

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
MIDLAND/ODESSA DIVISION**

ADVANCED CLUSTER SYSTEMS, INC.,

Plaintiff,

v.

INTEL CORPORATION,

Defendant.

Civil Action No. 7:24-cv-00245-ADA

**PLAINTIFF ADVANCED CLUSTER SYSTEMS, INC.’S
PRELIMINARY INFRINGEMENT CONTENTIONS**

Pursuant to the Court’s Standing Order Governing Proceedings 4.4 – Patent Cases, dated January 23, 2024 (“OGP”), at p. 2 (Section II.2), Plaintiff Advanced Cluster Systems, Inc. (“ACS”) hereby serves on Defendant Intel Corporation (“Intel”) the following Preliminary Infringement Contentions.¹

The following disclosures are based on information currently available to ACS. Fact and expert discovery in this action has not begun, and ACS reserves the right to modify, amend, or otherwise supplement these disclosures as discovery reveals new documents and information. ACS further reserves the right to modify, amend, or otherwise supplement these disclosures should Defendant alter, supplement, or otherwise clarify documents and information currently available

¹ ACS anticipates that its contentions will require supplementation because Intel: (1) has not produced any versions, subroutines or APIs for the source code for the Accused Products; (2) has not provided any relevant and responsive technical documents for the Accused Products; and (3) has not produced documents or otherwise provided information or evidence responsive to ACS’s forthcoming Requests for Production or other discovery requests. *See* OGP at p. 11, n. 8.

to ACS. ACS notes that Defendant has not yet provided any versions, subroutines, or APIs for the source code for the Accused Products. Defendant has not yet produced any relevant and responsive technical documents that show the operation of the Accused Products and have not produced any documents responsive to ACS’s forthcoming Requests for Production. Such responsive documents are relevant to several of the required disclosures below.

I. DISCLOSURES

A. ASSERTED CLAIMS

Pursuant to the Court’s OGP, at p. 2 (Section II.2), ACS identifies the following asserted claims (the “Asserted Claims”):

U.S. Patent No.	Asserted Claims
10,333,768	1, 4, 20, 21, 26, 27, 29, 30, 31, 33, 34, 35, 36, 37, 39
11,563,621	1, 3, 4, 6, 7, 8, 9, 10, 11, 13, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30
11,570,034	1, 2, 3, 8, 10, 24, 25, 27, 28, 30
11,811,582	1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
12,021,679	1, 2, 3, 8, 10, 17, 18

B. ACCUSED PRODUCTS AND METHODS

Based on presently available information, ACS contends that Intel infringes the Asserted Claims based on the following instrumentalities (the “Accused Products”). Specifically, ACS asserts that the Accused Products meet the limitations of the asserted claims identified below.

Asserted Patent	Asserted Claims	Accused Product(s)
10,333,768	1, 4, 20, 21, 26, 27, 29, 30, 31, 33, 35, 37, 39	<p>All server and workstation products that utilize Intel’s Habana AI accelerator architectures, including at least the Gaudi HLS-1 AI Training System, HLS-Gaudi2 server, and Gaudi 3 AI Accelerator HLB-325 Baseboard products, all variants thereof, and all products including to related to those products, as well as any products incorporating those items and any products that are substantially similar to those products (the “Accused Habana Server Products”).</p> <p>All AI accelerator products that utilize Intel’s Habana AI accelerator architectures and are especially adapted for use in server and workstation products, including at least the Gaudi, Gaudi 2, Gaudi 3, and Goya AI accelerator products, all variants thereof, and any products that are substantially similar to those products (the “Accused Habana AI Accelerator Products”).</p> <p><i>See Exhibit A1 hereto. Each of the Accused Habana AI Accelerator Products and each of the Accused Habana Server Products is configured and operates in a substantially similar manner in relevant part.</i></p>
10,333,768	1, 4, 20, 21, 26, 27, 29, 30, 31, 33, 34, 35, 36, 37, 39	All Xeon Scalable Processors (Skylake-SP architecture), 2nd Gen Intel Xeon Scalable Processors, 3rd Gen Intel Xeon Scalable Processors, 4th Gen Intel Xeon Scalable Processors, 5th Gen Intel Xeon Scalable Processors, and Intel Xeon 6 Processors, all variants thereof, all products incorporating those products, and all products that are substantially similar to those products (the “Accused Xeon Products”). ²

² Collectively, the Accused Habana Server Products, Accused Habana AI Accelerator Products, and the Accused Xeon Products are referred to herein as the Accused Products.

Asserted Patent	Asserted Claims	Accused Product(s)
		<p><i>See Exhibit B1 hereto. Each of the Accused Xeon Products is configured and operates in a substantially similar manner in relevant part.</i></p>
<p>11,563,621</p>	<p>1, 3, 4, 6, 7, 8, 9, 11, 13, 15, 16, 17, 21, 23, 24, 26, 27, 28, 29, 30</p>	<p>All server and workstation products that utilize Intel’s Habana AI accelerator architectures, including at least the Gaudi HLS-1 AI Training System, HLS-Gaudi2 server, and Gaudi 3 AI Accelerator HLB-325 Baseboard products, all variants thereof, and all products including to related to those products, as well as any products incorporating those items and any products that are substantially similar to those products (the “Accused Habana Server Products”).</p> <p>All AI accelerator products that utilize Intel’s Habana AI accelerator architectures and are especially adapted for use in server and workstation products, including at least the Gaudi, Gaudi 2, Gaudi 3, and Goya AI accelerator products, all variants thereof, and any products that are substantially similar to those products (the “Accused Habana AI Accelerator Products”).</p> <p><i>See Exhibit A2 hereto. Each of the Accused Habana AI Accelerator Products and each of the Accused Habana Server Products is configured and operates in a substantially similar manner in relevant part.</i></p>
<p>11,563,621</p>	<p>1, 3, 4, 6, 7, 8, 9, 10 11, 13, 15, 16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30</p>	<p>All Xeon Scalable Processors (Skylake-SP architecture), 2nd Gen Intel Xeon Scalable Processors, 3rd Gen Intel Xeon Scalable Processors, 4th Gen Intel Xeon Scalable Processors, 5th Gen Intel Xeon Scalable Processors, and Intel Xeon 6 Processors, all</p>

Asserted Patent	Asserted Claims	Accused Product(s)
		<p>variants thereof, all products incorporating those products, and all products that are substantially similar to those products (the “Accused Xeon Products”).</p> <p><i>See Exhibit B2 hereto. Each of the Accused Xeon Products is configured and operates in a substantially similar manner in relevant part.</i></p>
11,570,034	1, 2, 3, 8, 10, 24, 25, 27, 28, 30	<p>All server and workstation products that utilize Intel’s Habana AI accelerator architectures, including at least the Gaudi HLS-1 AI Training System, HLS-Gaudi2 server, and Gaudi 3 AI Accelerator HLB-325 Baseboard products, all variants thereof, and all products including to related to those products, as well as any products incorporating those items and any products that are substantially similar to those products (the “Accused Habana Server Products”).</p> <p>All AI accelerator products that utilize Intel’s Habana AI accelerator architectures and are especially adapted for use in server and workstation products, including at least the Gaudi, Gaudi 2, Gaudi 3, and Goya AI accelerator products, all variants thereof, and any products that are substantially similar to those products (the “Accused Habana AI Accelerator Products”).</p> <p><i>See Exhibit A3 hereto. Each of the Accused Habana AI Accelerator Products and each of the Accused Habana Server Products is configured and operates in a substantially similar manner in relevant part.</i></p> <p>All Xeon Scalable Processors (Skylake-SP architecture), 2nd Gen Intel Xeon Scalable Processors, 3rd Gen Intel Xeon Scalable</p>

Asserted Patent	Asserted Claims	Accused Product(s)
		<p>Processors, 4th Gen Intel Xeon Scalable Processors, 5th Gen Intel Xeon Scalable Processors, and Intel Xeon 6 Processors, all variants thereof, all products incorporating those products, and all products that are substantially similar to those products (the “Accused Xeon Products”).</p> <p><i>See Exhibit B3 hereto. Each of the Accused Xeon Products is configured and operates in a substantially similar manner in relevant part.</i></p>
11,811,582	1, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20	<p>All server and workstation products that utilize Intel’s Habana AI accelerator architectures, including at least the Gaudi HLS-1 AI Training System, HLS-Gaudi2 server, and Gaudi 3 AI Accelerator HLB-325 Baseboard products, all variants thereof, and all products including to related to those products, as well as any products incorporating those items and any products that are substantially similar to those products (the “Accused Habana Server Products”).</p> <p>All AI accelerator products that utilize Intel’s Habana AI accelerator architectures and are especially adapted for use in server and workstation products, including at least the Gaudi, Gaudi 2, Gaudi 3, and Goya AI accelerator products, all variants thereof, and any products that are substantially similar to those products (the “Accused Habana AI Accelerator Products”).</p> <p><i>See Exhibit A4 hereto. Each of the Accused Habana AI Accelerator Products and each of the Accused Habana Server Products is configured and operates in a substantially similar manner in relevant part.</i></p>

Asserted Patent	Asserted Claims	Accused Product(s)
11,811,582	1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20	<p>All Xeon Scalable Processors (Skylake-SP architecture), 2nd Gen Intel Xeon Scalable Processors, 3rd Gen Intel Xeon Scalable Processors, 4th Gen Intel Xeon Scalable Processors, 5th Gen Intel Xeon Scalable Processors, and Intel Xeon 6 Processors, all variants thereof, all products incorporating those products, and all products that are substantially similar to those products (the “Accused Xeon Products”).</p> <p><i>See Exhibit B4 hereto. Each of the Accused Xeon Products is configured and operates in a substantially similar manner in relevant part.</i></p>
12,021,679	1, 2, 3, 8, 10, 17, 18	<p>All server and workstation products that utilize Intel’s Habana AI accelerator architectures, including at least the Gaudi HLS-1 AI Training System, HLS-Gaudi2 server, and Gaudi 3 AI Accelerator HLB-325 Baseboard products, all variants thereof, and all products including to related to those products, as well as any products incorporating those items and any products that are substantially similar to those products (the “Accused Habana Server Products”).</p> <p>All AI accelerator products that utilize Intel’s Habana AI accelerator architectures and are especially adapted for use in server and workstation products, including at least the Gaudi, Gaudi 2, Gaudi 3, and Goya AI accelerator products, all variants thereof, and any products that are substantially similar to those products (the “Accused Habana AI Accelerator Products”).</p> <p><i>See Exhibit A5 hereto. Each of the Accused Habana AI Accelerator Products and each of the Accused Habana Server Products is</i></p>

Asserted Patent	Asserted Claims	Accused Product(s)
		<p>configured and operates in a substantially similar manner in relevant part.</p> <p>All Xeon Scalable Processors (Skylake-SP architecture), 2nd Gen Intel Xeon Scalable Processors, 3rd Gen Intel Xeon Scalable Processors, 4th Gen Intel Xeon Scalable Processors, 5th Gen Intel Xeon Scalable Processors, and Intel Xeon 6 Processors, all variants thereof, all products incorporating those products, and all products that are substantially similar to those products (the “Accused Xeon Products”).</p> <p><i>See Exhibit B5 hereto. Each of the Accused Xeon Products is configured and operates in a substantially similar manner in relevant part.</i></p>

ACS does not intend for the product names or model numbers in this chart of Accused Products, or in the appended infringement charts in Exhibits A1-A5 and B1-B5, to be a conclusive list. By identifying product names and model numbers in this chart of Accused Products, and in the appended infringement charts in Exhibits A1-A5 and B1-B5, ACS intends to include all model numbers and configurations for each identified Accused Product. ACS has identified additional model numbers where available based on currently available information. ACS reserves the right to supplement this table of Accused Products as additional information becomes available through discovery in this action.

C. INFRINGEMENT CHARTS

For the Asserted Patent, ACS attaches claim charts (*see* Exhibits A1-A5 and B1-B5) to these Preliminary Infringement Contentions identifying specifically where and how each

limitation of each Asserted Claim is found within each Accused Product. The attached claim charts are exemplary.

ACS has not completed its investigation of the facts relating to this case, has not completed discovery in this action, and has not completed preparation for trial. Therefore, ACS expressly reserves the right to amend, supplement, or otherwise modify the attached charts as additional information becomes available through discovery and/or as necessitated by any claim construction ruling from the Court. *See* OGP at p. 11, n. 8.

D. INDIRECT INFRINGEMENT

In addition to directly infringing the Asserted Patents by, for example, making, using, selling, offering for sale, and/or importing into the United States the Accused Products, Intel has indirectly infringed the Asserted Patents by actively inducing direct infringement of the Asserted Patents by others.

Defendant actively induces others to infringe the Asserted Patents by marketing and selling the Accused Products, knowing, no later than the date ACS served the original complaint in this case on September 26, 2024, and intending that the Accused Products would be used by customers, users, system builders, Intel Partners, and retailers/distributors in a manner that infringes the Asserted Patent.

ACS is informed and believes, and thereon alleges that Intel makes, uses, sells, offers for sale, and/or imports one or more of the Accused Products for or on behalf of third parties such as customers, users, system builders, partners, and retailers/distributors, knowing and intending that the Accused Products will be used by the third parties in a manner that practices the Asserted Claims as shown in Exhibits A1-A5 and B1-B5, respectively. For example, Intel published and provided marketing materials, technical specifications, whitepapers, datasheets, user manuals, and

development and testing information, and other resources on its website (<http://www.intel.com/>) that instructed and encouraged third parties to integrate the Accused Products into products using Intel's technology that were then made, used, sold, offered for sale and/or imported into the United States.³ Intel has also established the "Intel AI Partner ecosystem" to assist customers with training, professional services, and service and support.⁴ These activities were designed to bring infringing products that incorporate Intel's Accused Products to market in the United States. As a result of Intel's activities, the Accused Products have been used in a manner that directly infringes the Asserted Patents. Intel continues to engage in acts of inducement of infringement of the Asserted Patents.

Defendant induces others to directly infringe in numerous ways. As a non-exhaustive exemplary list, a direct infringer may make server products by combining Accused Habana AI Accelerator Products and/or Accused Xeon Products into an infringing system; use a server product owned by the direct infringer, use a server product owned by a third party, use a server product owned by Defendant; sell and/or offer to sell server products made by the direct infringer, sell and/or offer to sell server products acquired from another party, including possibly from Defendant; and/or import server products into the United States.

Based on currently available information, Defendant provides marketing materials, technical specifications, whitepapers, datasheets, user manuals, development and testing

³ *E.g.*, <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi-overview.html>;
<https://www.intel.com/content/www/us/en/products/docs/processors/xeon/xeon-scalable-platform-where-to-buy.html>.

⁴ *E.g.*,
<https://www.intel.com/content/www/us/en/partner-alliance/membership/benefits/partners-ai.html>.

information, and other resources that instruct and encourage customers, users, system builders, and/or retailers/distributors to make, use, sell, offer for sale, and import the server products in an infringing manner in the United States. Defendant instructs and encourages this infringement by, for example, establishing the Intel AI Partner ecosystem to assist Intel partners with marketing, training, professional services, sales and distribution, and service and support for the server products. The server products embody or practice the Asserted Claims as set forth in Exhibits A1-A5 and B1-B5.

Additionally, Defendant provides marketing materials, technical specifications, whitepapers, datasheets, user manuals, development and testing information, reference architectures, and other resources that instruct and encourage customers, users, system builders, Intel AI Partners, and retailers/distributors to use the Accused Products in an infringing manner in the United States. For example, Intel provides instructions for integrating the Accused Products using into other products that infringe the Asserted Patents. These third parties then use the materials and support provided by Defendant to make, use, sell, offer for sale, and/or import into the United States infringing products that use the Accused Products therein. Defendant instructs and encourages this by, for example, establishing the Intel AI Partner ecosystem to assist Intel Partners with marketing, training, professional services, sales and distributions, and service and support.

Intel also encourages others to provide, or provides itself, use of Accused Products in a compute as a service model. Their compute as a service model encourages use of Accused Products by a party different from the owner of the Accused Products. This model encourages many infringing users to share access to and use of the Accused Products. Intel also offers support for Accused Products allowing, and inducing, infringing use to continue.

In view of at least Defendant's knowledge of the Asserted Patents, Defendant knew or should have known that such activities would cause direct infringement of the Asserted Patents by Defendant's customers, end users, system builders, Intel AI Partners, and/or retailers/distributors of the Accused Products. Defendant's inducement constitutes infringement of the Asserted Patents in violation of 35 U.S.C. § 271(b).

ACS has not completed its investigation of the facts relating to this case, has not completed discovery in this action, and has not completed preparation for trial. Therefore, ACS expressly reserves the right to amend, supplement, or otherwise modify the attached charts as additional information becomes available through discovery. *See* OGP at p. 11, n. 8.

E. LITERAL INFRINGEMENT AND INFRINGEMENT UNDER THE DOCTRINE OF EQUIVALENTS

Based on currently available information, ACS alleges that each of the Accused Products literally infringes each of the Asserted Claims. *See Supra* §§ I.B, I.C & I.D, and Exhibits A1-A5 and B1-B5 hereto.

To the extent any claim limitations are not literally infringed by any Accused Product, ACS contends the limitations are infringed under the doctrine of equivalents because the identified features of the Accused Products are insubstantially different from the literal claim elements, for example because they perform substantially the same function in substantially the same way to achieve substantially the same result. ACS has not completed its investigation of the facts relating to this case and has not yet completed discovery in this action. Furthermore, the Asserted Claims have not yet been construed by the Court, nor has Defendant addressed claim construction. Defendant also has not yet set forth its bases, if any, for non-infringement of the Asserted Claims. Accordingly, ACS reserves the right to amend, supplement, or otherwise modify these

infringement contentions as additional information becomes available through the claim construction process and discovery, including the right to further supplement its contentions with regard to infringement pursuant to the doctrine of equivalents. *See* OGP at p. 11, n. 8.

F. PRIORITY DATE OF ASSERTED CLAIMS

ACS contends that the Asserted Claims of the Asserted Patents are entitled to a priority date no later than the filing of provisional application No. 60/813,738, which was filed on June 13, 2006, and of provisional application No. 60/850,908, which was filed on October 11, 2006. The Asserted Patents claim priority to both provisional applications on their faces as related U.S. Applications.

II. DOCUMENT PRODUCTION ACCOMPANYING DISCLOSURE

Pursuant to the Court's OGP, at p. 2 (Section II.2), concurrently with the service of its foregoing disclosures and contentions, ACS will be producing via an FTP link today a copy of the file history for each Asserted Patent and documents evidencing conception and reduction to practice for each claimed invention (ACS_0000167 – ACS_0076737).

These disclosures, contentions, and document production are based on information currently available to ACS. Fact and expert discovery in this action has not begun, and ACS reserves the right to modify, amend, or otherwise supplement its disclosures, contentions, and document production as new documents and information are revealed through discovery and/or to the extent Defendant alters, supplants, or otherwise clarifies documents and information currently available to ACS. *See* OGP at p. 11, n. 8.

Respectfully submitted,

By: /s/ Reynaldo C. Barceló

BARCELÓ, HARRISON & WALKER, LLP

Reynaldo C. Barceló (admitted *pro hac vice*)
rey@bhiplaw.com
2901 West Coast Hwy, Suite 200
Newport Beach, CA 92663
Telephone: (949) 340-9736

CROWELL & MORING LLP

Jon Gurka (admitted *pro hac vice*)
jgurka@crowell.com
Kainoa Asuega (admitted *pro hac vice*)
kasuega@crowell.com
3 Park Plaza, 20th Floor
Irvine, CA 92614
Telephone: (949) 263-8400

David Lindner (admitted *pro hac vice*)
dlindner@crowell.com
Andrew McElligott (admitted *pro hac vice*)
amcelligott@crowell.com
455 N Cityfront Plaza Drive, Ste 3600
Chicago, IL 60611
Telephone: (312) 321-4200

Michelle Wang
michellewang@crowell.com
Two Manhattan West
375 Ninth Avenue
New York, NY 10001
Telephone: (212) 223-4000

CHERRY JOHNSON SIEGMUND JAMES, PLLC

Mark D. Siegmund (TX Bar No. 24117055)

msiegmund@cjsjlaw.com

William D. Ellerman (TX Bar No. 24007151)

wellerman@cjsjlaw.com

7901 Fish Pond Rd., 2nd Floor

Waco, TX 76710

Telephone: (254) 732-2242

Attorneys for Plaintiff

Advanced Cluster Systems, Inc.

CERTIFICATE OF SERVICE

I hereby certify that on January 22, 2025, I served the foregoing document (including the accompanying claim chart exhibits A1-A5 and B1-B5 identified therein) via e-mail on Defendant's counsel of record at Intel-ACS@mto.com and BNash@mofo.com.

Dated: January 22, 2025

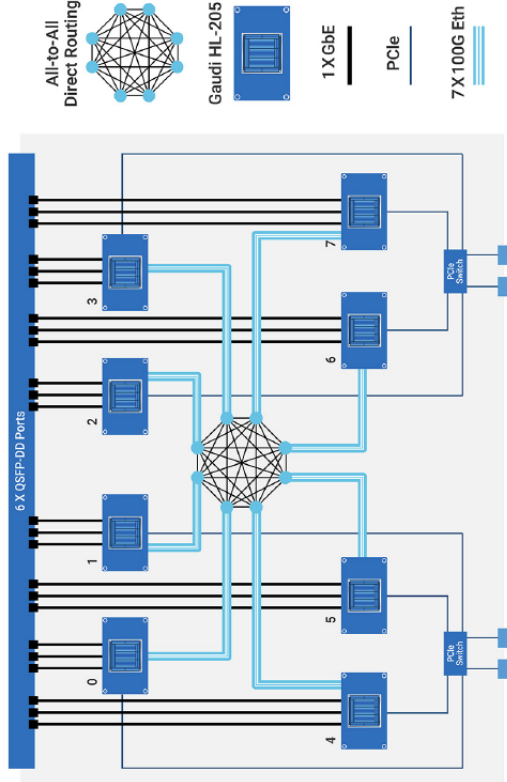
By: /s/ Reynaldo C. Barceló
Reynaldo C. Barceló

Exhibit A1
U.S. Pat. No. 10,333,768 in view of
Accused Habana Server Products and Accused Habana AI Accelerator Products

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p>
<p>1. A computer cluster comprising:</p>	<p>Each of the Accused Habana Server Products¹ comprises a computer cluster. For example, as depicted below and as described further herein, Intel’s Gaudi® HLS-1 AI Training System (“HLS-1”) includes a computer cluster.</p> <div style="text-align: center;">  <p>The image shows a server rack with multiple circuit boards. On the left, there is a logo for Intel Gaudi, with the text 'An Intel Company' and 'GAUDI'. On the right, there is text that reads 'GAUDI® HLS-1 AI Training System'.</p> </div> <p>Each of the Accused Habana Server Products includes a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product.² For example, as depicted below, the Intel HLS-1 system</p>

¹ The Accused Habana Server Products include, but are not limited to, all products including or related to Intel’s Gaudi® HLS-1 AI Training System (“HLS-1”) (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>), HLS-Gaudi2 server (https://habana.ai/wp-content/uploads/2023/10/HLS-Gaudi2_Datasheet_10_23.pdf), Gaudi 3 AI Accelerator HL.B-325 Baseboard (<https://www.intel.com/content/www/us/en/content-details/817489/intel-gaudi-3-ai-accelerator-hlb-325-baseboard-product-brief.html>), as well as any products incorporating those items.

² Accused Habana AI Accelerator Products include, but are not limited to all products including or related to intel’s Gaudi (https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html); <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi.html>), Gaudi 2 (<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html>), Gaudi 3 (<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html>), and Goya (https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf) AI accelerator products, as well as any products incorporating those items.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.³</p>  <p style="text-align: center;">HLS-1 Block Diagram</p>
<p>a plurality of nodes, wherein each of the plurality of nodes comprises a hardware processor, wherein one or more of the nodes are</p>	<p>Each of the Accused Habana Server Products comprises a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.⁴</p>

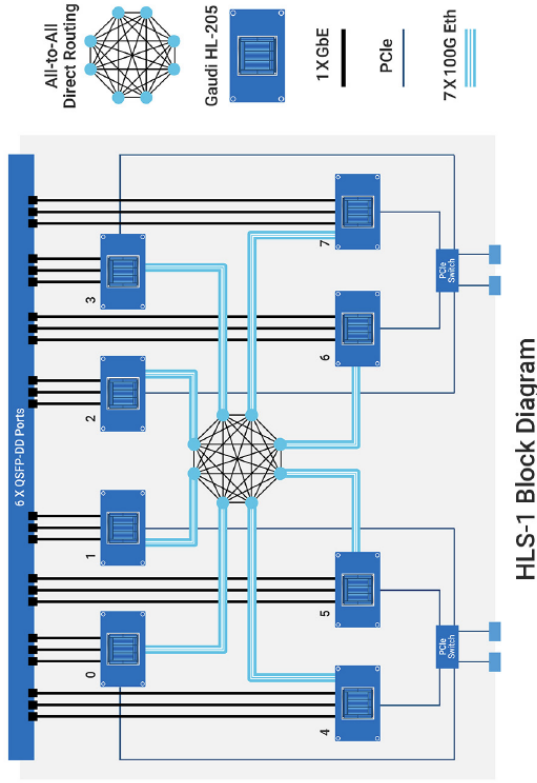
³ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

⁴ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

Accused Habana Server Products and Accused Habana AI Accelerator Products

U.S. Pat. No. 10,333,768

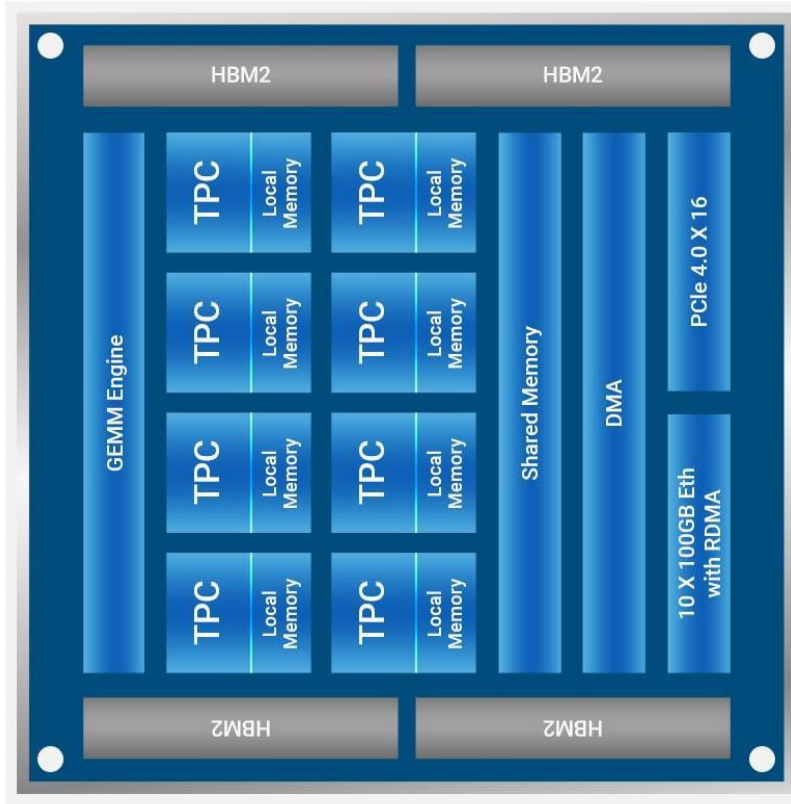
configured to receive a command to start a cluster initialization process for the computer cluster, and wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions; and



Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises a hardware processor, wherein one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster, and wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions.

Specifically, each Accused Habana AI Accelerator Product comprises a hardware processor that comprises multiple processor cores. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM). The architecture of the Gaudi HL-2000 processor is depicted in the following figure:⁵

⁵ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi.html> (Gaudi), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html> (Gaudi 2),



Each Accused Habana AI Accelerator Product comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Habana Server Product. For example, each Gaudi HL-205 Mezzanine card in each HLS-1 system comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the eight Gaudi HL-205 Mezzanine cards in the HLS-1

<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>This is done, for example, by using the Habana Collective Communication Library (HCCL).⁶ HCCL is included in Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”⁷</p> <p>Each Accused Habana AI Accelerator Product is configured to access a non-transitory computer-readable medium (e.g., memory accessible by a Gaudi HL-2000 processor, including local memory, shared memory, and HBM2 memory, as well as memory external to the Gaudi HL-2000 processor).⁸</p> <p>Each Accused Habana AI Accelerator Product in each Accused Habana Server Product comprises program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions (e.g., executable SynapseAI® Software Suite program code stored in memory accessible by the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card). For example, when executed, the code stored in the memory accessible by a Gaudi HL-2000 processor causes the processor to evaluate mathematical expressions (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library). TPCs are configured to evaluate multiple mathematical expressions, such as square root, sine, cosine, tangent, and reciprocal, as well as matrix operations such as Vector Reduction Intrinsics.⁹</p> <p>Each Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card of the HLS-1 system is configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the processor to produce results of mathematical expression evaluation. A first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other Gaudi HL-205 accelerators for execution. The other Gaudi HL-205</p>
--	--

⁶ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

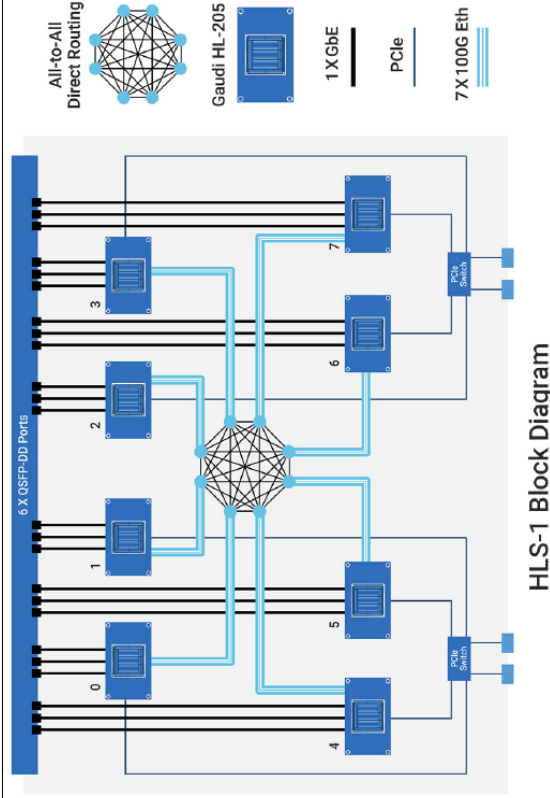
⁷ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

⁸ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

⁹ See, e.g., Built-in Functions (“https://docs.habana.ai/en/latest/TPC_User_Guide/Built_in_Functions.html”).

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>accelerators in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user. Hence, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card is configured to access executable program code stored in memory accessible by the Gaudi HL-2000 processor, where the executable program code is program code for a single-node kernel that, when executed, causes the processor to interpret user instructions.</p>
<p>a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture;</p>	<p>Each of the Accused Habana AI Accelerator Products in each Accused Habana Server Product comprises a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture (e.g., the all-to-all direct routing connectivity between the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system shown in the following block diagram,¹⁰ along with related collective HCCL commands included in Intel's SynapseAI® Software Suite).</p>

¹⁰ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video ("Habana Labs: How to Scale AI Training with Gaudi Processors"), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.



For example, Intel also provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”¹¹ The source code for HCCL Demo¹² includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, and for the Gaudi HL-205 Mezzanine card nodes in the HLS-1 system to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture.

wherein the plurality of nodes comprises: a first node comprising a first hardware processor configured to access a first

Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution.

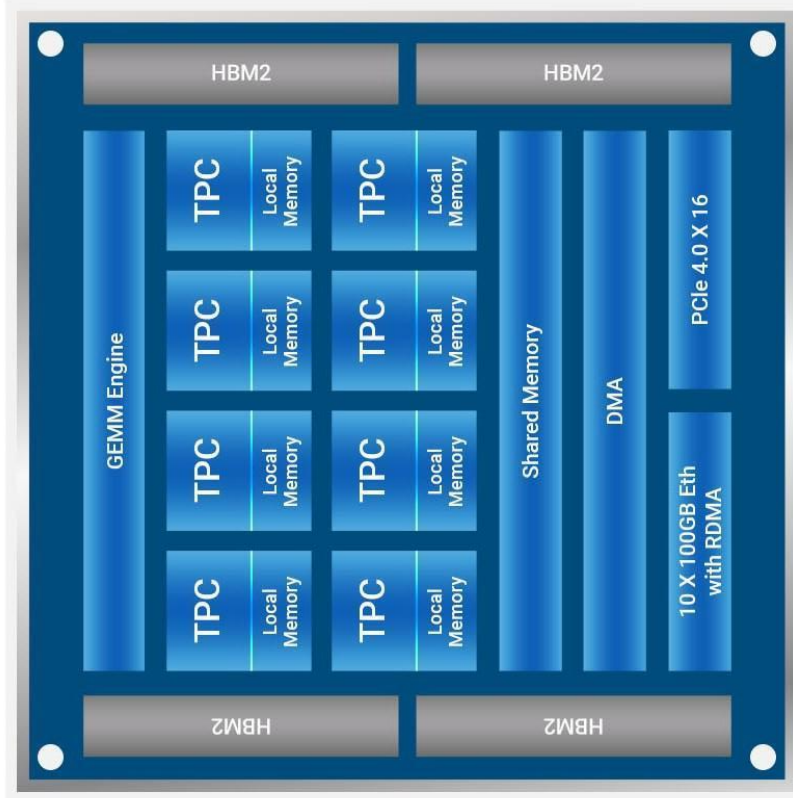
¹¹ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).
¹² See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p> <p>memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure:¹³</p>
---	--

¹³ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi.html> (Gaudi), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).

U.S. Pat. No. 10,333,768

Accused Habana Server Products and Accused Habana AI Accelerator Products



For example, the Habana Collective Communication Library (HCCL) includes commands that cause a Gaudi HL-2000 processor of a Gaudi HL-205 mezzanine card in a HLS-1 system to distribute calls to at least one other Gaudi HL-205 mezzanine card in the HLS-1 system for execution.¹⁴ HCCL is included in Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”¹⁵

¹⁴ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

¹⁵ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

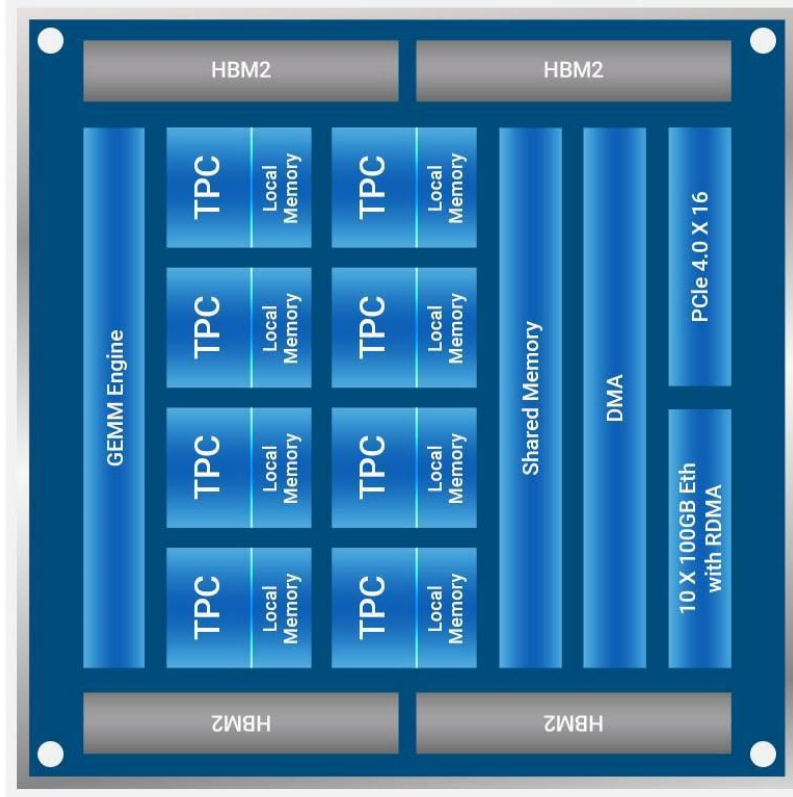
U.S. Pat. No. 10,333,768 in view of Accused Habana Server / AI Accelerator Products
ACS’s Prelim. Infringement Contentions

Exh. A1
January 22, 2025

U.S. Pat. No. 10,333,768	Accused Habana Server Products and Accused Habana AI Accelerator Products
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node;</p>	<p>The source code for Intel's HCCL Demo program¹⁶ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for a user interface and for a first single-node kernel, where the single-node kernel is configured to interpret user instructions and to distribute calls to at least one of a plurality of other Gaudi HL-205 Mezzanine card nodes for execution.</p>
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third Accused Habana AI Accelerator Product;</p>	<p>Each of the Accused Habana Server Products comprises a second Accused Habana AI Accelerator Product comprising a second hardware processor with a plurality of processing cores, wherein the second Accused Habana AI Accelerator Product is configured to receive calls from the first Accused Habana AI Accelerator Product, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third Accused Habana AI Accelerator Product.</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure:¹⁷</p>

¹⁶ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

¹⁷ GaudiTM Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/audi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/audi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



For example, Intel provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”¹⁸ The source code for HCCL Demo¹⁹ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for a user interface and for a first single-node kernel, for interpreting user instructions, for distributing calls to at least one of a plurality of other Gaudi HL-205 Mezzanine card nodes for execution, for receiving

¹⁸ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

¹⁹ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also [https://github.com/HabanaAI/hccl_demo/blob/main/README.md](https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py).

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>the result of mathematical expression from other nodes, for executing mathematical expression evaluations, and for communicating a result of mathematical expression evaluation to other nodes and to the user interface.</p>
<p>wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of the first mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;</p>	<p>A third Accused Habana AI Accelerator Product in an Accused Habana Server Product comprises a hardware processor with multiple processing cores, and is configured to receive the result of a first mathematical expression evaluation from the second Accused Habana AI Accelerator Product in the Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the received result (e.g., via executable SynapseAI® Software Suite program code stored in memory in communication with a Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card), and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product. For example, when executed, the code stored in the memory accessible by the Gaudi HL-2000 processor of the third Gaudi HL-205 Mezzanine card in a HLS-1 system causes the processor to receive a result of a first mathematical expression evaluation from a second Gaudi HL-205 Mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library), and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-205 Mezzanine card in the HLS-1 system.</p> <p>Also for example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a third Gaudi HL-205 mezzanine card in an HLS-1 system is configured to receive the result of a first mathematical expression evaluation from the second Gaudi HL-205 mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result, and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-205 Mezzanine card in the HLS-1 system. The source code for Intel’s HCCL Demo program,²⁰ for example, includes the details for using HCCL commands to receive the result of a first mathematical</p>

²⁰ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>expression evaluation from a second Accused Habana AI Accelerator Product in an Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the received result, and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product.</p>
<p>wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface;</p>	<p>A first Accused Habana AI Accelerator Product in each Accused Habana Server Product is configured to return the result of a second mathematical expression evaluation to the user interface.</p> <p>For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card node in a HLS-1 system is configured to communicate a result of mathematical expression evaluation to other nodes and to a user interface. Intel provides ‘HCCL Demo,’ which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”²¹ The source code for Intel’s HCCL Demo program²² includes the details for using HCCL commands to return the result of mathematical expression evaluations from a first Gaudi HL-205 mezzanine card node in a HLS-1 system to the user interface.</p>
<p>wherein one or more of the nodes are configured to: accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least</p>	<p>One or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product are configured to: (1) accept user instructions; (2) after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and (3) after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p> <p>For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI</p>

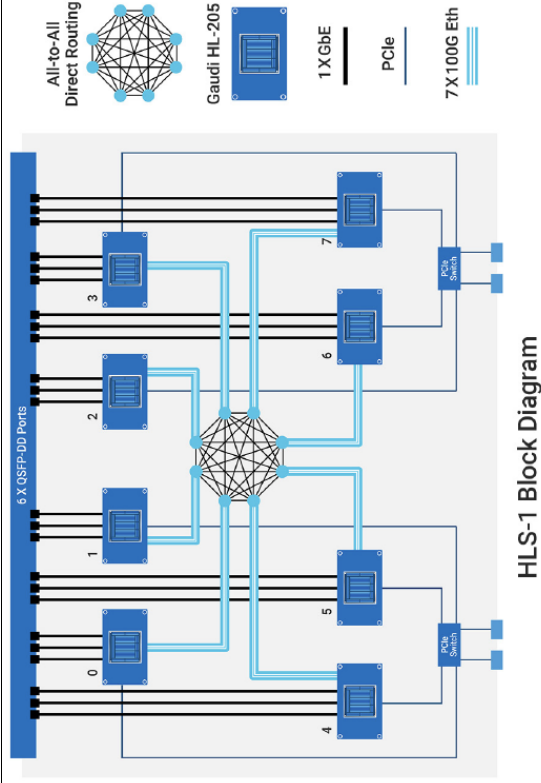
²¹ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

²² See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p> <p>some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p> <p>Also for example, the source code for Intel’s HCCL Demo program²³ includes the details for using HCCL commands to cause an Accused Habana AI Accelerator Products in an Accused Habana Server Product to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p>
<p>4. The computer cluster of claim 1, wherein each of the nodes comprises one or more cluster node modules.</p>	<p>Each of the Accused Habana Server Products comprises a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.²⁴</p>

²³ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py; https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

²⁴ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

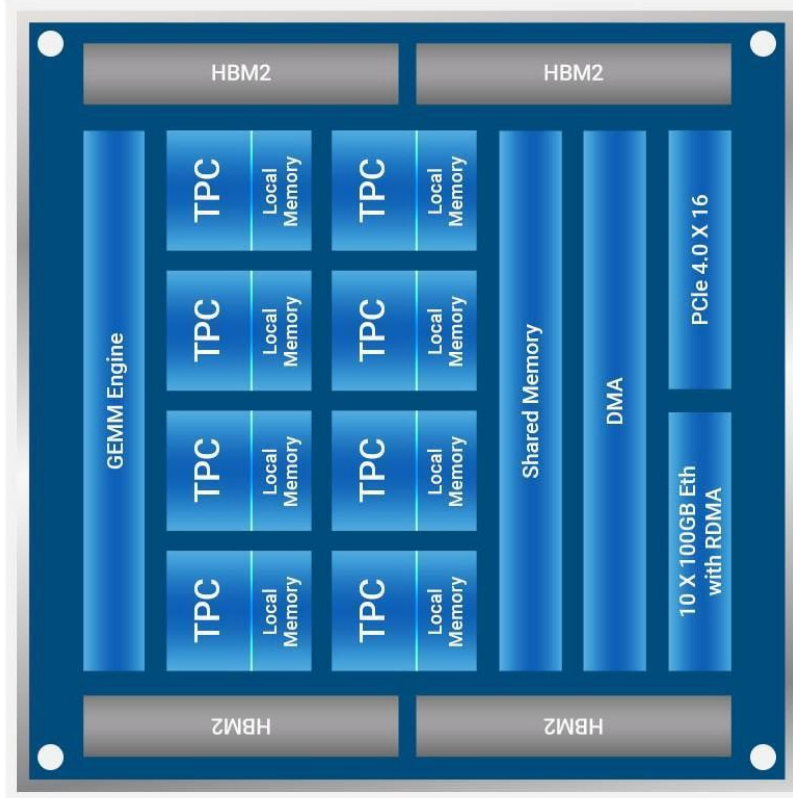


Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises one or more cluster node modules. Specifically, each Accused Habana AI Accelerator Product comprises a hardware processor that comprises multiple processor cores. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM). The architecture of the Gaudi HL-2000 processor is depicted in the following figure.²⁵

²⁵ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

U.S. Pat. No. 10,333,768

Accused Habana Server Products and Accused Habana AI Accelerator Products



Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises one or more cluster node modules, for example, by supporting and/or executing the Habana Collective Communication Library (HCCL).²⁶ HCCL is included in Intel's SynapseAI® Software Suite. SynapseAI® is "Habana's complete software stack custom designed to support Habana's Gaudi implementation."²⁷

²⁶ "Habana Collective Communications Library (HCCL) is Habana's emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library." (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

²⁷ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

U.S. Pat. No. 10,333,768 in view of Accused Habana Server / AI Accelerator Products
ACS's Prelim. Infringement Contentions

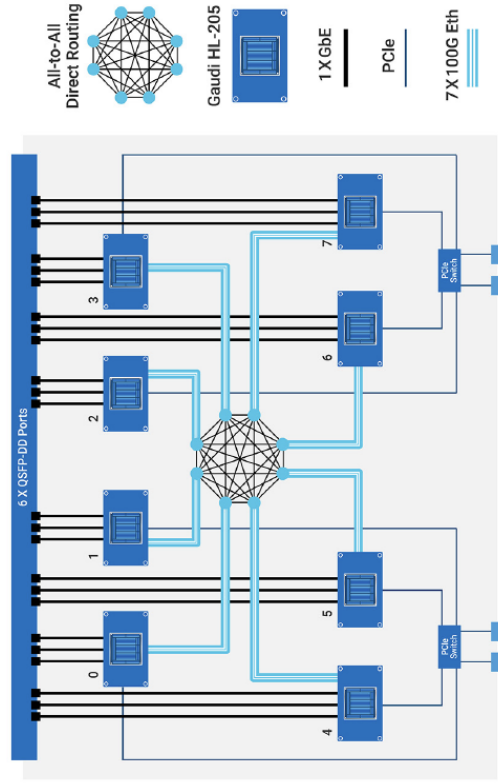
Exh. A1
January 22, 2025

U.S. Pat. No. 10,333,768

20. The computer cluster of claim 1, wherein one or more of the nodes are configured to accept user instructions via one or more of the nodes.

Accused Habana Server Products and Accused Habana AI Accelerator Products

One or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product are configured to accept user instructions via one or more of the Accused Habana AI Accelerator Products in the Accused Habana Server Product. For example, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” . . . “7” in the figure below) in which the nodes are configured to communicate commands including user instructions with one another via network connections that connect the nodes in an all-to-all local bus routing by eight 7x 100G Ethernet connections.²⁸



At least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, each of the cluster of eight Gaudi HL-205 Mezzanine cards in an HLS-1 system corresponds to a node that can access and execute kernel software residing in memory, and that is configured to enable cluster node modules to accept user instructions and communicate at least some of the user instructions to each other using the communications network

²⁸ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

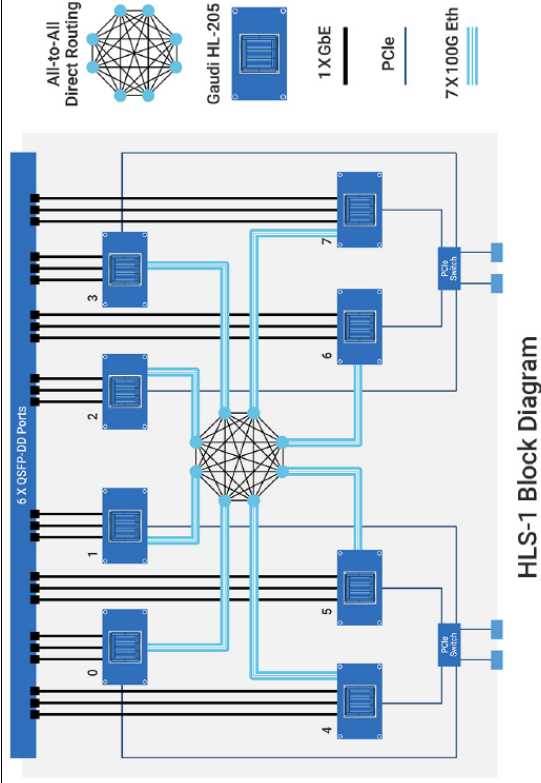
<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>interface shown above, and further communicate at least some of the user instructions to one or more of the kernels. The source code for Intel’s HCCL Demo program²⁹ includes the details for using HCCL commands to cause one or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product to accept user instructions via one or more of the nodes.^{30,31}</p> <p>One or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product are configured to communicate some of the user instructions using the mechanism for the nodes to communicate with each other. For example, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1”, “...7” in the figure below) in which the nodes are configured to communicate commands including user instructions with one another via network connections that connect the nodes in an all-to-all local bus routing by eight 7x 100G Ethernet connections.³²</p>
<p>21. The computer cluster of claim 20, wherein one or more of the nodes are configured to communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other.</p>	

²⁹ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py; https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

³⁰ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

³¹ https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/Using_HCCL.html.

³² Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.



At least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card is configured to communicate user instructions with other nodes using the network connections shown above. The source code for Intel’s HCCL Demo program³³ includes the details for using HCCL commands to cause one or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product to communicate with each other at least some of the user instructions using the network connections between the nodes.^{34,35}

³³ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py; https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

³⁴ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

³⁵ https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/Using_HCCL.html.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p>
<p>26. A computer cluster comprising: "</p>	<p>Each of the Accused Habana Server Products³⁶ comprises a computer cluster. For example, as depicted below and as described further herein, Intel's Gaudi® HLS-1 AI Training System ("HLS-1") includes a computer cluster.</p>  <p>Each of the Accused Habana Server Products includes a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product.³⁷ For example, as depicted below, the Intel HLS-1</p>

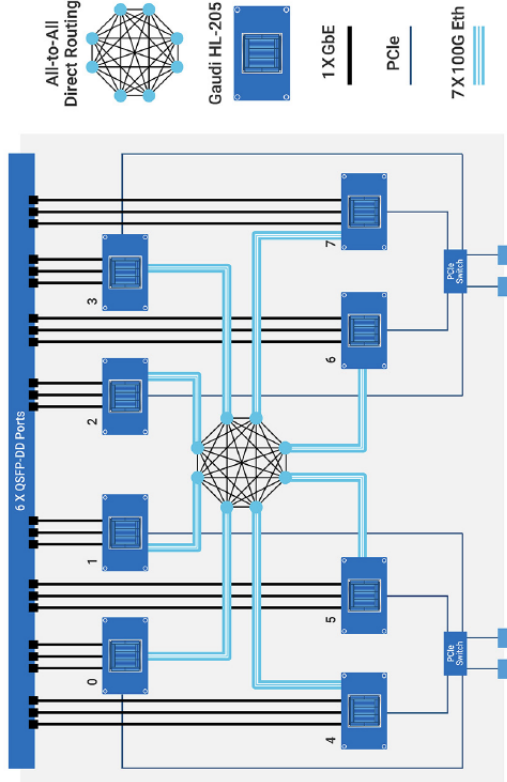
³⁶ The Accused Habana Server Products include, but are not limited to, all products including or related to Intel's Gaudi® HLS-1 AI Training System ("HLS-1") (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>), HLS-Gaudi2 server ([https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html](https://habana.ai/wp-content/uploads/2023/10/HLS-Gaudi2_Datasheet_10_23.pdf)), Gaudi 3 AI Accelerator HLB-325 Baseboard (<https://www.intel.com/content/www/us/en/content-details/817489/intel-gaudi-3-ai-accelerator-hlb-325-baseboard-product-brief.html>), as well as any products incorporating those items.

³⁷ Accused Habana AI Accelerator Products include, but are not limited to all products including or related to intel's Gaudi (https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi.html>), Gaudi 3 (<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html>), and Goya (https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf) AI accelerator products, as well as any products incorporating those items.

U.S. Pat. No. 10,333,768 in view of Accused Habana Server / AI Accelerator Products
 ACS's Prelim. Infringement Contentions

20

Exh. A1
 January 22, 2025

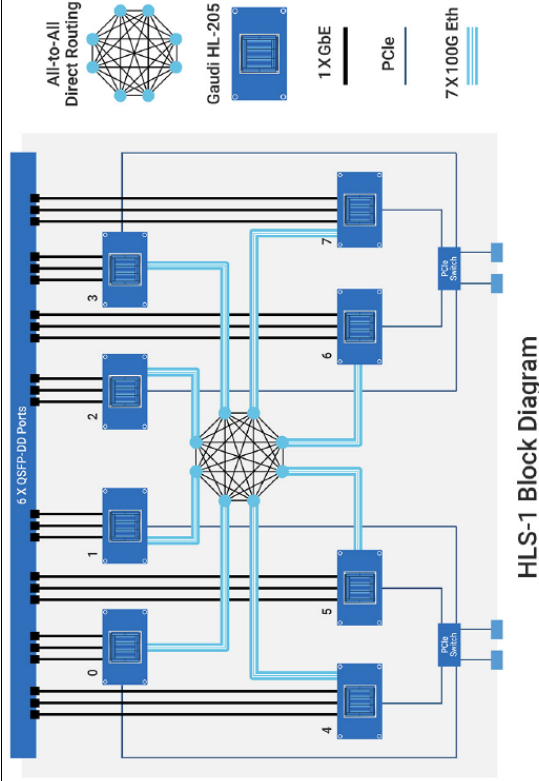
<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.³⁸</p>  <p style="text-align: center;">HLS-1 Block Diagram</p>	<p>Each of the Accused Habana Server Products comprises a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.³⁹</p>
<p>a plurality of nodes, wherein one or more of the nodes are configured to receive: a command to start a cluster initialization process for the computer</p>	<p>Each of the Accused Habana Server Products comprises a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.³⁹</p>	<p>Each of the Accused Habana Server Products comprises a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.³⁹</p>

³⁸ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/audi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

³⁹ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/audi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

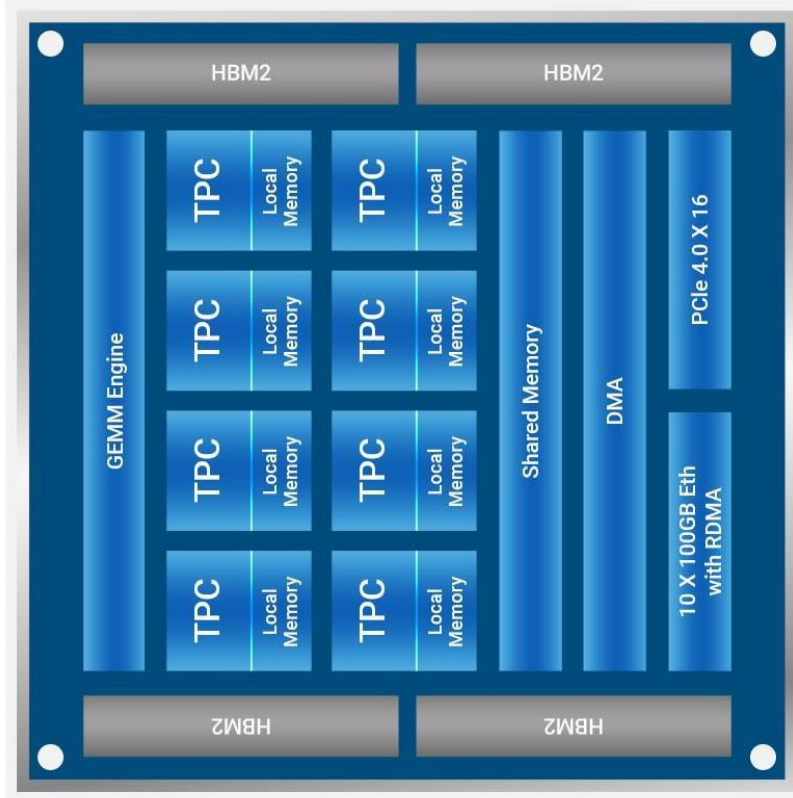
Accused Habana Server Products and Accused Habana AI Accelerator Products

U.S. Pat. No. 10,333,768 cluster, wherein the cluster initialization process comprises establishing communication among two or more of the nodes; and an instruction from a user interface or a script; and



One or more of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product are configured to receive a command to start a cluster initialization process for the computer cluster that establishes communication among two or more of the nodes as depicted above. Specifically, each Accused Habana AI Accelerator Product comprises a hardware processor that comprises multiple processor cores. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM). The architecture of the Gaudi HL-2000 processor is depicted in the following figure:⁴⁰

⁴⁰ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



Each Accused Habana AI Accelerator Product is configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Habana Server Product. For example, each Gaudi HL-205 Mezzanine card in each HLS-1 system is configured to receive a command from a user interface or a script to start a cluster initialization process for the computer cluster that includes the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system. This is done, for example, by using the Habana Collective Communication Library (HCCL).⁴¹ HCCL is included in Intel's SynapseAI®

⁴¹ "Habana Collective Communications Library (HCCL) is Habana's emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library." (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”⁴²</p>
<p>a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls;</p>	<p>Each of the Accused Habana AI Accelerator Products in each Accused Habana Server Product comprises a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls over a peer-to-peer network architecture (e.g., the all-to-all direct routing connectivity between the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system shown in the following block diagram,⁴³ along with related collective HCCL commands included in Intel’s SynapseAI® Software Suite).</p> <div data-bbox="600 478 1144 1270" data-label="Diagram"> <p>The diagram, titled 'HLS-1 Block Diagram', illustrates a network architecture. At the top, a horizontal bar represents '6 X QSFP-DD Ports'. Below this, eight Gaudi HL-205 mezzanine cards are arranged in two rows of four, numbered 0 through 7. Each card is connected to the top bar. A central 'All-to-All Direct Routing' network is shown as a mesh of nodes. Light blue lines represent connections between the cards and the central network. A legend at the top right identifies connection types: a single line for '1 X GbE', a double line for 'PCIe', and a thick blue line for '7 X 100G Eth'. Two 'PCI Switch' components are shown at the bottom, connected to the network.</p> </div>

⁴² Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

⁴³ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>For example, Intel also provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁴⁴ The source code for HCCL Demo⁴⁵ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, and for the Gaudi HL-205 Mezzanine card nodes in the HLS-1 system to communicate results of mathematical expression evaluation with each other using asynchronous calls over a peer-to-peer network architecture. The Intel Gaudi software uses stream architecture to manage concurrent execution of asynchronous tasks.⁴⁶ For example, HCCL supports asynchronicity of issued operations and “all collective operations are asynchronous and implemented as non-blocking calls. After an asynchronous call, another collective operation may be called immediately after as long as it uses the same stream.”⁴⁷</p>
<p>wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions; wherein the plurality of nodes comprises: a first node comprising a first hardware</p>	<p>Each Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card of the HLS-1 system is configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the processor to evaluate mathematical expressions.</p> <p>For example, each Accused Habana AI Accelerator Product is configured to access a non-transitory computer-readable medium (e.g., memory accessible by a Gaudi HL-2000 processor, including local memory, shared memory, and HBM2 memory, as well as memory external to the Gaudi HL-2000 processor).⁴⁸ Each Accused Habana AI Accelerator Product in each Accused Habana Server Product comprises program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions (e.g., executable SynapseAI® Software Suite program code stored in memory accessible by the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card). For example, when executed, the code stored in the memory accessible by a Gaudi HL-2000 processor causes the processor to evaluate mathematical expressions (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating</p>

⁴⁴ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

⁴⁵ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

⁴⁶ https://docs.habana.ai/en/latest/Gaudi_Overview/Intel_Gaudi_Software_Suite.html.

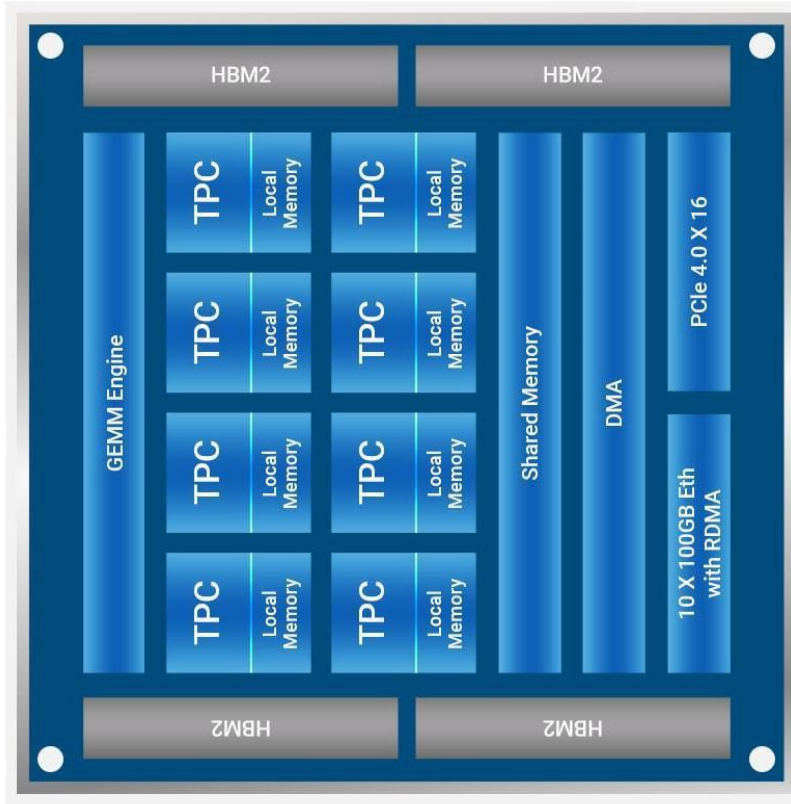
⁴⁷ See https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Using_HCCL.html.

⁴⁸ GaudiTM Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

<p>U.S. Pat. No. 10,333,768</p> <p>processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library). TPCs are configured to evaluate multiple mathematical expressions, such as square root, sine, cosine, tangent, and reciprocal, as well as matrix operations such as Vector Reduction Intrinsic.⁴⁹</p> <p>A first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other Gaudi HL-205 accelerators for execution. The other Gaudi HL-205 accelerators in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user. Hence, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card is configured to access executable program code stored in memory accessible by the Gaudi HL-2000 processor, where the executable program code is program code for a single-node kernel that, when executed, causes the processor to interpret user instructions.</p> <p>Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution.</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure.⁵⁰</p>
--	---

⁴⁹ See, e.g., Built-in Functions (“https://docs.habana.ai/en/latest/TPC/TPC_User_Guide/Built_in_Functions.html”).

⁵⁰ GaudiTM Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/processors/ai-accelerators/tpc.html> (Gaudi), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/tpc.html> (Gaudi 2),



For example, the Habana Collective Communication Library (HCCL) includes commands that cause a Gaudi HL-2000 processor of a Gaudi HL-205 mezzanine card in a HLS-1 system to distribute calls to at least one other Gaudi HL-205 mezzanine card in the HLS-1 system for execution.⁵¹ HCCL is

<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).

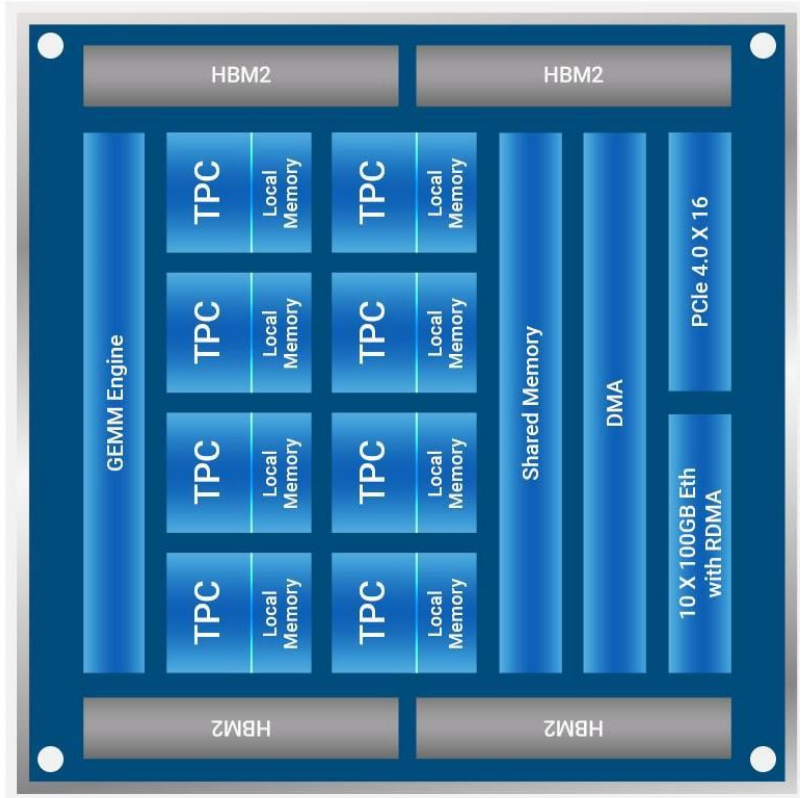
⁵¹ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>included in Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”⁵²</p> <p>The source code for Intel’s HCCL Demo program⁵³ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for a user interface and for a first single-node kernel, where the single-node kernel is configured to interpret user instructions and to distribute calls to at least one of a plurality of other Gaudi HL-205 Mezzanine card nodes for execution.</p> <p>Each of the Accused Habana Server Products comprises a second Accused Habana AI Accelerator Product comprising a second hardware processor with a plurality of processing cores, wherein the second Accused Habana AI Accelerator Product is configured to receive calls from the first Accused Habana AI Accelerator Product, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third Accused Habana AI Accelerator Product.</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure:⁵⁴</p>
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node; wherein</p>	

⁵² Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

⁵³ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

⁵⁴ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



For example, Intel provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁵⁵ The source code for HCCL Demo⁵⁶ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for a user interface and for a first single-node kernel, for interpreting user instructions, for distributing calls to at least one of a plurality of other Gaudi HL-205 Mezzanine card nodes for execution, for receiving

⁵⁵ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).
⁵⁶ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p>
<p>the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;</p>	<p>the result of mathematical expression from other nodes, for executing mathematical expression evaluations, and for communicating a result of mathematical expression evaluation to other nodes and to the user interface.</p>
<p>A third Accused Habana AI Accelerator Product in an Accused Habana Server Product comprises a hardware processor with multiple processing cores, and is configured to receive the result of mathematical expression evaluation from the second Accused Habana AI Accelerator Product in the Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the received result (e.g., via executable SynapseAI® Software Suite program code stored in memory in communication with a Gaudi HL-2000 processor in a Gaudi HL-2005 mezzanine card), and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product. For example, when executed, the code stored in the memory accessible by the Gaudi HL-2000 processor of the third Gaudi HL-2005 Mezzanine card in a HLS-1 system causes the processor to receive a result of mathematical expression evaluation from a second Gaudi HL-2005 Mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library), and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-2005 Mezzanine card in the HLS-1 system.</p> <p>Also for example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a third Gaudi HL-2005 mezzanine card in an HLS-1 system is configured to receive the result of mathematical expression evaluation from the second Gaudi HL-2005 mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result, and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-2005 Mezzanine card in the HLS-1 system. The source code for Intel’s HCCL Demo program,⁵⁷ for example, includes the details for using HCCL commands to receive the result of mathematical</p>	<p>A third Accused Habana AI Accelerator Product in an Accused Habana Server Product comprises a hardware processor with multiple processing cores, and is configured to receive the result of mathematical expression evaluation from the second Accused Habana AI Accelerator Product in the Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the received result (e.g., via executable SynapseAI® Software Suite program code stored in memory in communication with a Gaudi HL-2000 processor in a Gaudi HL-2005 mezzanine card), and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product. For example, when executed, the code stored in the memory accessible by the Gaudi HL-2000 processor of the third Gaudi HL-2005 Mezzanine card in a HLS-1 system causes the processor to receive a result of mathematical expression evaluation from a second Gaudi HL-2005 Mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library), and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-2005 Mezzanine card in the HLS-1 system.</p> <p>Also for example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a third Gaudi HL-2005 mezzanine card in an HLS-1 system is configured to receive the result of mathematical expression evaluation from the second Gaudi HL-2005 mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result, and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-2005 Mezzanine card in the HLS-1 system. The source code for Intel’s HCCL Demo program,⁵⁷ for example, includes the details for using HCCL commands to receive the result of mathematical</p>

⁵⁷ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>expression evaluation from a second Accused Habana AI Accelerator Product in an Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the received result, and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product.</p>
<p>wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface or the script;</p>	<p>A first Accused Habana AI Accelerator Product in each Accused Habana Server Product is configured to return the result of a second mathematical expression evaluation to the user interface or a script.</p> <p>For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card node in a HLS-1 system is configured to communicate a result of mathematical expression evaluation to other nodes and to a user interface. Intel provides ‘HCCL Demo,’ which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁵⁸ The source code for Intel’s HCCL Demo program⁵⁹ includes the details for using HCCL commands to return the result of mathematical expression evaluations from a first Gaudi HL-205 mezzanine card node in a HLS-1 system to the user interface or a script.</p>
<p>wherein one or more of the nodes are configured to: accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least</p>	<p>One or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product are configured to: (1) accept user instructions; (2) after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and (3) after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p> <p>For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI</p>

⁵⁸ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

⁵⁹ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p> <p>some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.</p>	<p style="text-align: center;">Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p> <p>Also for example, the source code for Intel’s HCCL Demo program⁶⁰ includes the details for using HCCL commands to cause an Accused Habana AI Accelerator Products in an Accused Habana Server Product to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p>
<p>27. The computer cluster of claim 26, wherein the asynchronous calls comprise a first command to create a first packet to be sent as payload; and an expression should be sent; wherein the first command is configured to send the first packet to a local cluster node module, and wherein the local cluster node module is configured to forward the expression to the target node. A single-node kernel is, for example, a software program that runs on the first node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other nodes for execution by cluster node modules on those nodes. For example, the HLS-1 system comprises a plurality of Gaudi HL-205 Mezzanine cards that implement asynchronous calls that enable a single-node kernel to perform computation tasks while the Gaudi HL-205 Mezzanine cards are simultaneously communicating with one another via the all-to-all direct non-blocking routing connectivity between the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system, as shown in the following block diagram.⁶¹</p>	<p>The asynchronous calls (see claim 26, above) comprise a first command to create a first packet containing an expression to be sent as payload, and a target node where the expression should be sent, wherein the first command is configured to be called from within a single-node kernel, wherein the single-node kernel is configured to send the first packet to a local cluster node module, and wherein the local cluster node module is configured to forward the expression to the target node. A single-node kernel is, for example, a software program that runs on the first node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other nodes for execution by cluster node modules on those nodes. For example, the HLS-1 system comprises a plurality of Gaudi HL-205 Mezzanine cards that implement asynchronous calls that enable a single-node kernel to perform computation tasks while the Gaudi HL-205 Mezzanine cards are simultaneously communicating with one another via the all-to-all direct non-blocking routing connectivity between the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system, as shown in the following block diagram.⁶¹</p>

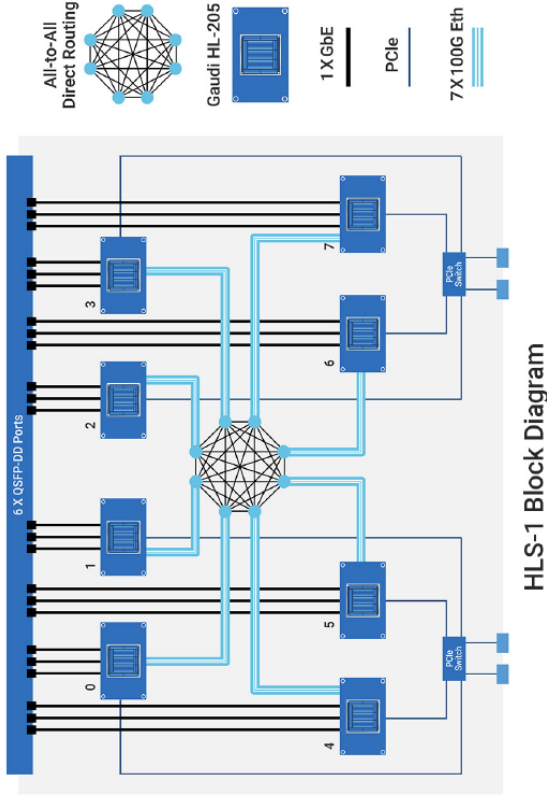
⁶⁰ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py; https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

⁶¹ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

Accused Habana Server Products and Accused Habana AI Accelerator Products

U.S. Pat. No. 10,333,768

to send the first packet to a local cluster node module; and wherein the local cluster node module is configured to forward the expression to the target node.



HLS-1 Block Diagram

For example, Intel provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁶² The source code for HCCL Demo⁶³ includes the details for using HCCL commands included in Intel’s SynapseAI® Software Suite to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, and for the Gaudi HL-205 Mezzanine card nodes in the HLS-1 system to implement asynchronous calls that enable single-node kernel software residing in memory to perform computation tasks while cluster node modules are simultaneously communicating with one another. The Intel Gaudi software uses stream architecture to manage concurrent execution of asynchronous tasks.⁶⁴ For example, HCCL supports asynchronicity of issued operations and “all collective

⁶² See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

⁶³ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

⁶⁴ https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Software_Suite.html.

U.S. Pat. No. 10,333,768	<p data-bbox="186 193 228 1528">Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p data-bbox="228 193 305 1528">operations are asynchronous and implemented as non-blocking calls. After an asynchronous call, another collective operation may be called immediately after as long as it uses the same stream.”⁶⁵</p> <p data-bbox="305 193 776 1528">Also, for example, HCCL supports asynchronous progress threads that allow for managing communication in parallel with application computation and, as a result, achieve better communication/computation overlapping. For example, the asynchronous calls <code>hcclSend()</code> and <code>hcclRecv()</code> are used for asynchronous communications.⁶⁶ For example, a first single-node kernel that generates a <code>hcclSend()</code> call, therefore, creates a first packet containing an expression to be sent as payload (see, for example, the call <code>hcclSend()</code>), and a target node (see, for example the destination / <code>sendToRank</code> field of <code>hcclSend()</code>) where the expression should be sent, where the first command is configured to be called from within a single-node kernel. Because of the asynchronous nature of <code>hcclSend()</code> and <code>hcclRecv()</code> calls, the single-node kernel is configured to send the first packet to a local cluster node module and the local cluster node module is configured to forward the expression to the target node (see, for example the destination / <code>sendToRank</code> field of the call <code>hcclSend()</code>).</p>
29. A computer cluster comprising:	<p data-bbox="776 193 922 1528">Each of the Accused Habana Server Products⁶⁷ comprises a computer cluster. For example, as depicted below and as described further herein, Intel’s Gaudi® HLS-1 AI Training System (“HLS-1”) includes a computer cluster.</p>

⁶⁵ https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/Using_HCCL.html.

⁶⁶ https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/Overview.html; https://github.com/HabanaAI/hccl_demo/blob/main/send_recv.cpp;

⁶⁷ The Accused Habana Server Products include, but are not limited to, all products including or related to Intel’s Gaudi® HLS-1 AI Training System (“HLS-1”) (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>), HLS-Gaudi2 server (https://habana.ai/wp-content/uploads/2023/10/HLS-Gaudi2_Datasheet_10_23.pdf), Gaudi 3 AI Accelerator HLB-325 Baseboard (<https://www.intel.com/content/www/us/en/content-details/817489/intel-gaudi-3-ai-accelerator-hlb-325-baseboard-product-brief.html>), as well as any products incorporating those items.

U.S. Pat. No. 10,333,768

Accused Habana Server Products and Accused Habana AI Accelerator Products



Each of the Accused Habana Server Products includes a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product.⁶⁸ For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.⁶⁹

⁶⁸ Accused Habana AI Accelerator Products include, but are not limited to all products including or related to intel’s Gaudi (https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/audi.html>), Gaudi 2 (<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html>), Gaudi 3 (<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html>), and Goya (https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf) AI accelerator products, as well as any products incorporating those items.

⁶⁹ Gaudi@ HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

U.S. Pat. No. 10,333,768 in view of Accused Habana Server / AI Accelerator Products

ACS’s Prelim. Infringement Contentions

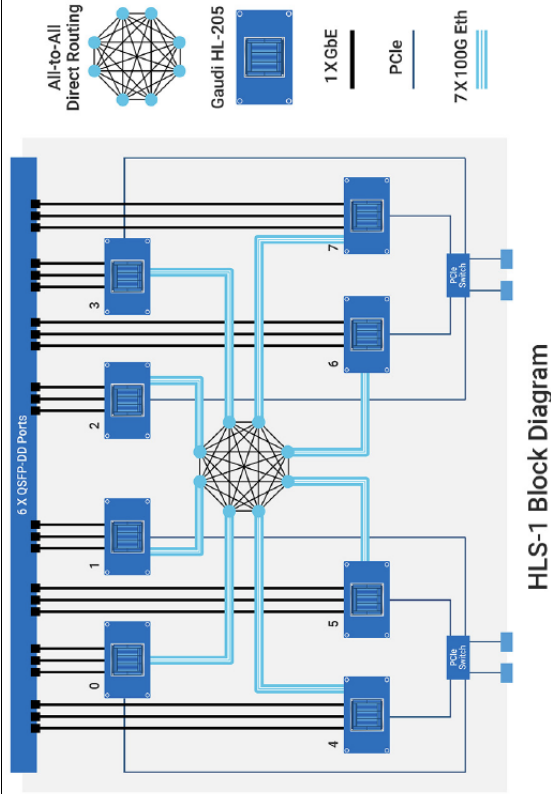
35

Exh. A1

January 22, 2025

U.S. Pat. No. 10,333,768

Accused Habana Server Products and Accused Habana AI Accelerator Products



a plurality of nodes, wherein one or more of the nodes are configured to receive: a command to start a cluster initialization process for the computer cluster, wherein the cluster initialization process comprises establishing communication among two or more of the nodes; and

Each of the Accused Habana Server Products comprises a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1” ... “7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.⁷⁰

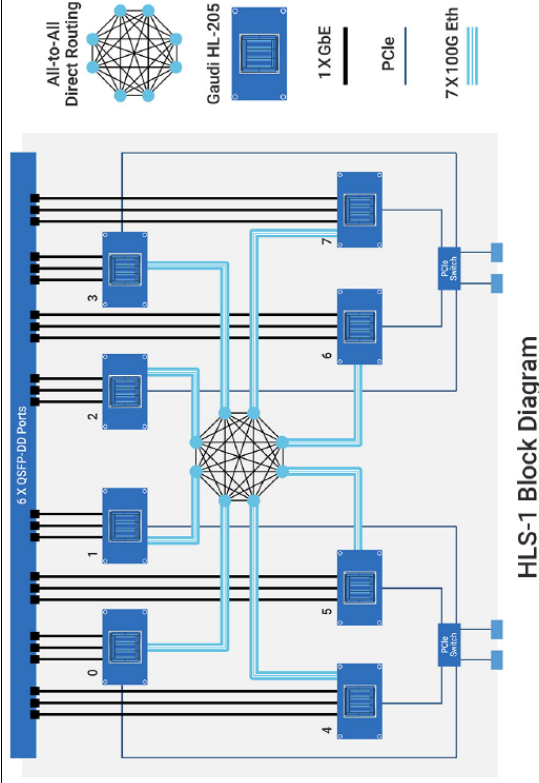
⁷⁰ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

U.S. Pat. No. 10,333,768 in view of Accused Habana Server / AI Accelerator Products
ACS’s Prelim. Infringement Contentions

Exh. A1
January 22, 2025

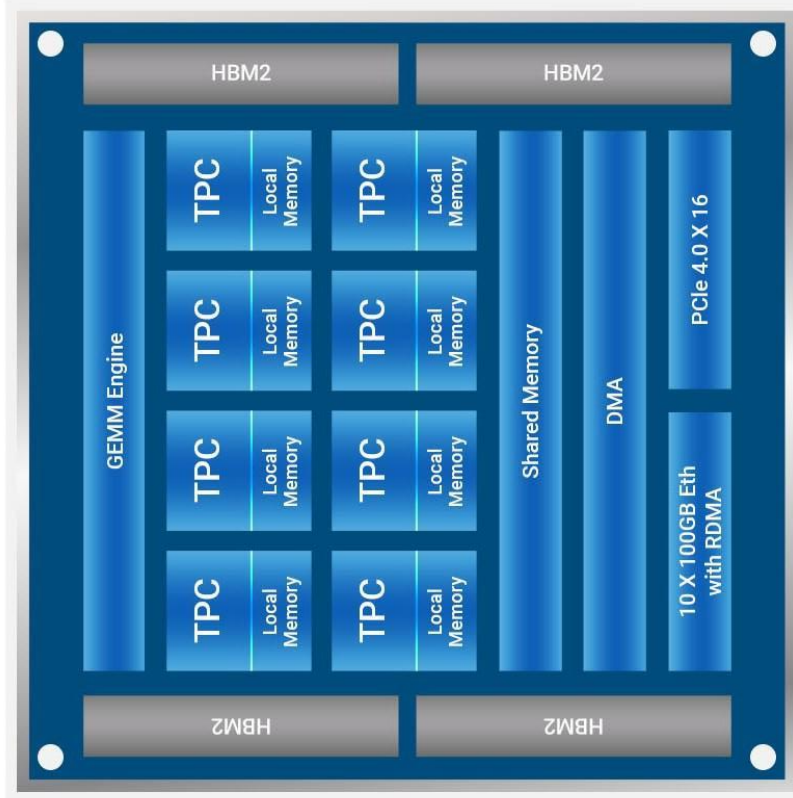
Accused Habana Server Products and Accused Habana AI Accelerator Products

U.S. Pat. No. 10,333,768
 an instruction from a user interface or a script; and



One or more of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product are configured to receive a command to start a cluster initialization process for the computer cluster that establishes communication among two or more of the nodes as depicted above. Specifically, each Accused Habana AI Accelerator Product comprises a hardware processor that comprises multiple processor cores. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM). The architecture of the Gaudi HL-2000 processor is depicted in the following figure:⁷¹

⁷¹ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



Each Accused Habana AI Accelerator Product is configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Habana Server Product. For example, each Gaudi HL-205 Mezzanine card in each HLS-1 system is configured to receive a command from a user interface or a script to start a cluster initialization process for the computer cluster that includes the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system. This is done, for example, by using the Habana Collective Communication Library (HCCL).⁷² HCCL is included in Intel's SynapseAI®

⁷² "Habana Collective Communications Library (HCCL) is Habana's emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library." (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”⁷³</p>
<p>a mechanism for the nodes to communicate results of mathematical expression evaluation with each other;</p>	<p>Each of the Accused Habana AI Accelerator Products in each Accused Habana Server Product comprises a mechanism for the nodes to communicate results of mathematical expression evaluation with each other (e.g., via the all-to-all direct routing connectivity between the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system shown in the following block diagram,⁷⁴ along with related collective HCCL commands included in Intel’s SynapseAI® Software Suite).</p> <div data-bbox="560 472 1112 1270" data-label="Diagram"> <p>The diagram, titled 'HLS-1 Block Diagram', illustrates the connectivity of eight Gaudi HL-205 Mezzanine cards. At the top, a horizontal bar represents '6 X QSFP-DD Ports'. Below this, eight Gaudi HL-205 cards are arranged in two rows of four, numbered 0 through 7. Each card is connected to the QSFP-DD ports above it. The cards are also interconnected via a central mesh network, labeled 'All-to-All Direct Routing'. Two PCIe switches are shown at the bottom, connected to the cards via PCIe lines. A legend on the right side of the diagram identifies the components: 'All-to-All Direct Routing' (represented by a mesh icon), 'Gaudi HL-205' (represented by a blue card icon), '1 X GbE' (represented by a single line), 'PCIe' (represented by a double line), and '7 X 100G Eth' (represented by a thick blue line).</p> </div>

⁷³ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

⁷⁴ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>For example, Intel also provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁷⁵ The source code for HCCL Demo⁷⁶ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, and for the Gaudi HL-205 Mezzanine card nodes in the HLS-1 system to communicate results of mathematical expression evaluation with each other.</p> <p>Each Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card of the HLS-1 system is configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the processor to evaluate mathematical expressions.</p> <p>For example, each Accused Habana AI Accelerator Product is configured to access a non-transitory computer-readable medium (e.g., memory accessible by a Gaudi HL-2000 processor, including local memory, shared memory, and HBM2 memory, as well as memory external to the Gaudi HL-2000 processor).⁷⁷ Each Accused Habana AI Accelerator Product in each Accused Habana Server Product comprises program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions (e.g., executable SynapseAI® Software Suite program code stored in memory accessible by the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card). For example, when executed, the code stored in the memory accessible by a Gaudi HL-2000 processor causes the processor to evaluate mathematical expressions (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library). TPCs are configured to evaluate multiple mathematical expressions, such as square root, sine, cosine, tangent, and reciprocal, as well as matrix operations such as Vector Reduction Intrinsic.⁷⁸</p>
<p>wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions; wherein the plurality of nodes comprises: a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first</p>	

⁷⁵ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

⁷⁶ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

⁷⁷ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

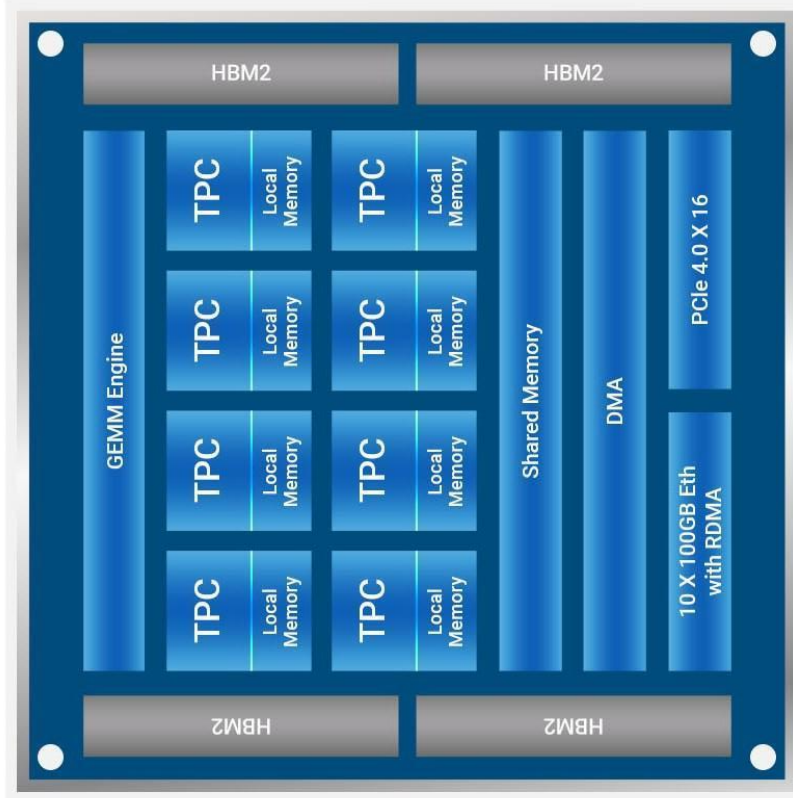
⁷⁸ See, e.g., Built-in Functions (“https://docs.habana.ai/en/latest/TPC/TPC_User_Guide/Built_in_Functions.html”).

<p>U.S. Pat. No. 10,333,768 single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and</p>	<p style="text-align: center;">Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>A first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other Gaudi HL-205 accelerators for execution. The other Gaudi HL-205 accelerators in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user. Hence, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card is configured to access executable program code stored in memory accessible by the Gaudi HL-2000 processor, where the executable program code is program code for a single-node kernel that, when executed, causes the processor to interpret user instructions.</p> <p>Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution.</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure:⁷⁹</p>
---	--

⁷⁹ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), [https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html](https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf) (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).

U.S. Pat. No. 10,333,768

Accused Habana Server Products and Accused Habana AI Accelerator Products



For example, the Habana Collective Communication Library (HCCL) includes commands that cause a Gaudi HL-2000 processor of a Gaudi HL-205 mezzanine card in a HLS-1 system to distribute calls to at least one other Gaudi HL-205 mezzanine card in the HLS-1 system for execution.⁸⁰ HCCL is included in Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”⁸¹

⁸⁰ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

⁸¹ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

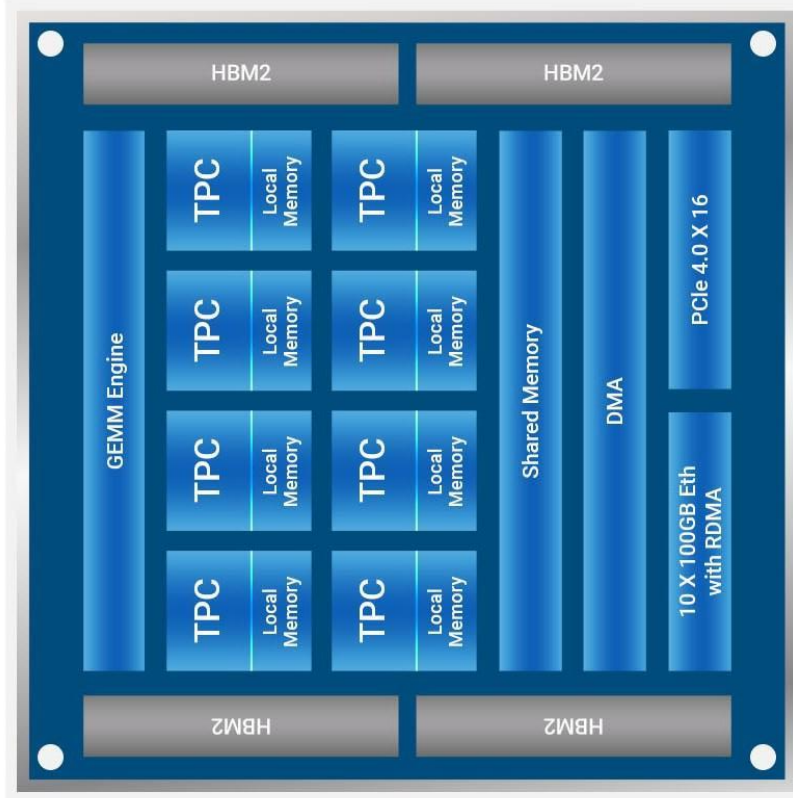
U.S. Pat. No. 10,333,768 in view of Accused Habana Server / AI Accelerator Products
ACS’s Prelim. Infringement Contentions

Exh. A1
January 22, 2025

U.S. Pat. No. 10,333,768	Accused Habana Server Products and Accused Habana AI Accelerator Products
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node; wherein</p>	<p>The source code for Intel's HCCL Demo program⁸² includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for a user interface and for a first single-node kernel, where the single-node kernel is configured to interpret user instructions and to distribute calls to at least one of a plurality of other Gaudi HL-205 Mezzanine card nodes for execution.</p> <p>Each of the Accused Habana Server Products comprises a second Accused Habana AI Accelerator Product comprising a second hardware processor with a plurality of processing cores, wherein the second Accused Habana AI Accelerator Product is configured to receive calls from the first Accused Habana AI Accelerator Product, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third Accused Habana AI Accelerator Product.</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure:⁸³</p>

⁸² See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

⁸³ GaudiTM Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/audi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/audi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



For example, Intel provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁸⁴ The source code for HCCL Demo⁸⁵ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for a user interface and for a first single-node kernel, for interpreting user instructions, for distributing calls to at least one of a plurality of other Gaudi HL-205 Mezzanine card nodes for execution, for receiving

⁸⁴ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

⁸⁵ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>the result of mathematical expression from other nodes, for executing mathematical expression evaluations, and for communicating a result of mathematical expression evaluation to other nodes and to the user interface.</p>
<p>the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node; and</p>	<p>A third Accused Habana AI Accelerator Product in an Accused Habana Server Product comprises a hardware processor with multiple processing cores, and is configured to receive the result of mathematical expression evaluation from the second Accused Habana AI Accelerator Product in the Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the received result (e.g., via executable SynapseAI® Software Suite program code stored in memory in communication with a Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card), and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product. For example, when executed, the code stored in the memory accessible by the Gaudi HL-2000 processor of the third Gaudi HL-205 Mezzanine card in a HLS-1 system causes the processor to receive a result of mathematical expression evaluation from a second Gaudi HL-205 Mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library), and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-205 Mezzanine card in the HLS-1 system.</p> <p>Also for example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a third Gaudi HL-205 mezzanine card in an HLS-1 system is configured to receive the result of mathematical expression evaluation from the second Gaudi HL-205 mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the received result, and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-205 Mezzanine card in the HLS-1 system. The source code for Intel’s HCCL Demo program,⁸⁶ for example, includes the details for using HCCL commands to receive the result of mathematical</p>

⁸⁶ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>expression evaluation from a second Accused Habana AI Accelerator Product in an Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the received result, and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product.</p>
<p>wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface or the script;</p>	<p>A first Accused Habana AI Accelerator Product in each Accused Habana Server Product is configured to return the result of a second mathematical expression evaluation to the user interface or a script.</p> <p>For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card node in a HLS-1 system is configured to communicate a result of mathematical expression evaluation to other nodes and to a user interface. Intel provides ‘HCCL Demo,’ which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁸⁷ The source code for Intel’s HCCL Demo program⁸⁸ includes the details for using HCCL commands to return the result of mathematical expression evaluations from a first Gaudi HL-205 mezzanine card node in a HLS-1 system to the user interface or a script.</p>
<p>wherein one or more of the nodes are configured to: accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least</p>	<p>One or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product are configured to: (1) accept user instructions; (2) after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and (3) after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p> <p>For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI</p>

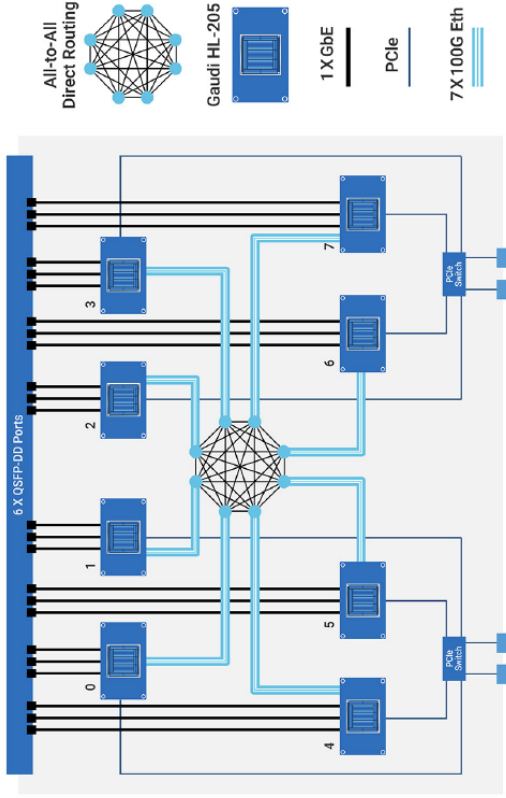
⁸⁷ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

⁸⁸ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p> <p>some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p> <p>Also for example, the source code for Intel’s HCCL Demo program⁸⁹ includes the details for using HCCL commands to cause an Accused Habana AI Accelerator Products in an Accused Habana Server Product to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p>
<p>30. The computer cluster of claim 4, wherein each of the plurality of nodes implements asynchronous calls that enable the single-node kernel to perform computation tasks while the cluster node modules are simultaneously communicating with one another.</p>	<p>Each of the Accused Habana AI Accelerator Products in each Accused Habana Server Product comprises a plurality of nodes that implement asynchronous calls that enable a single-node kernel to perform computation tasks while cluster node modules are simultaneously communicating with one another via the all-to-all direct routing connectivity between the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system shown in the following block diagram.⁹⁰</p>

⁸⁹ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py; https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

⁹⁰ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.



HLS-1 Block Diagram

For example, Intel provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”⁹¹ The source code for HCCL Demo⁹² includes the details for using HCCL commands included in Intel’s SynapseAI® Software Suite to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, and for the Gaudi HL-205 Mezzanine card nodes in the HLS-1 system to implement asynchronous calls that enable single-node kernel software residing in memory to perform computation tasks while cluster node modules are simultaneously communicating with one another. The Intel Gaudi software uses stream architecture to manage concurrent execution of asynchronous tasks.⁹³ For example, HCCL supports asynchronicity of issued operations and “all collective

⁹¹ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).
⁹² See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.
⁹³ https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Software_Suite.html.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>operations are asynchronous and implemented as non-blocking calls. After an asynchronous call, another collective operation may be called immediately after as long as it uses the same stream.”⁹⁴</p> <p>Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises one or more cluster node modules, for example, by supporting and/or executing the Habana Collective Communication Library (HCCL).⁹⁵ HCCL is included in Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”⁹⁶</p> <p>In each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product, intercommunication among the plurality of single-node kernels during thread execution is enabled by the plurality of cluster node modules, and the computer cluster is configured to permit exchange of information between nodes during the course of a parallel computation. For example, the Intel Gaudi Software Suite, which comprises Intel’s SynapseAI® Software Suite and the Intel Gaudi graph compiler, “generates optimized binary code that implements the given model topology on Gaudi. It performs operator fusion, data layout management, parallelization, pipelining and memory management, as well as graph-level optimizations. The graph compiler uses the rich TPC kernel library⁹⁷ which contains a wide variety of operations (for example, elementwise, non-linear, non-GEMM operators).”⁹⁸ The source code for Intel software such as Intel’s HCCL Demo program⁹⁹ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, enable intercommunication among a plurality of single-node kernels during thread execution by a plurality of cluster node modules, and permit exchange of information between the nodes during the course of a parallel computation.</p>
--	--

⁹⁴ https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/Using_HCCL.html.

⁹⁵ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

⁹⁶ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

⁹⁷ “Intel® Gaudi® processors TPC kernel library with supporting firmware, drivers and tools” (<https://developer.habana.ai/get-started/kernel-libraries/>).

⁹⁸ “Intel Gaudi Software Suite” (https://docs.habana.ai/en/latest/Gaudi_Overview/Intel_Gaudi_Software_Suite.html).

⁹⁹ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py.

U.S. Pat. No. 10,333,768	Accused Habana Server Products and Accused Habana AI Accelerator Products
<p>33. The computer cluster of claim 1, wherein the plurality of nodes are configured to permit exchange of information between nodes during the course of parallel computation.</p>	<p>The Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprise one or more cluster node modules, for example, by supporting and/or executing the Habana Collective Communication Library (HCCL).¹⁰⁰ HCCL is included in Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”¹⁰¹</p> <p>In each of the Accused Habana Server Products, the plurality of nodes of the computer cluster is configured to permit exchange of information between nodes during the course of parallel computation.</p> <p>For example, the Intel Gaudi Software Suite, which comprises Intel’s SynapseAI® Software Suite and the Intel Gaudi graph compiler, “generates optimized binary code that implements the given model topology on Gaudi. It performs operator fusion, data layout management, parallelization, pipelining and memory management, as well as graph-level optimizations. The graph compiler uses the rich TPC kernel library¹⁰² which contains a wide variety of operations (for example, elementwise, non-linear, non-GEMM operators).”¹⁰³ The source code for Intel software such as Intel’s HCCL Demo program¹⁰⁴ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, enable intercommunication among a plurality of single-node kernels during thread execution by a plurality of cluster node modules, and permit exchange of information between the nodes during the course of parallel computation.</p>

¹⁰⁰ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

¹⁰¹ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

¹⁰² “Intel® Gaudi® processors TPC kernel library with supporting firmware, drivers and tools” (<https://developer.habana.ai/get-started/kernel-libraries/>).

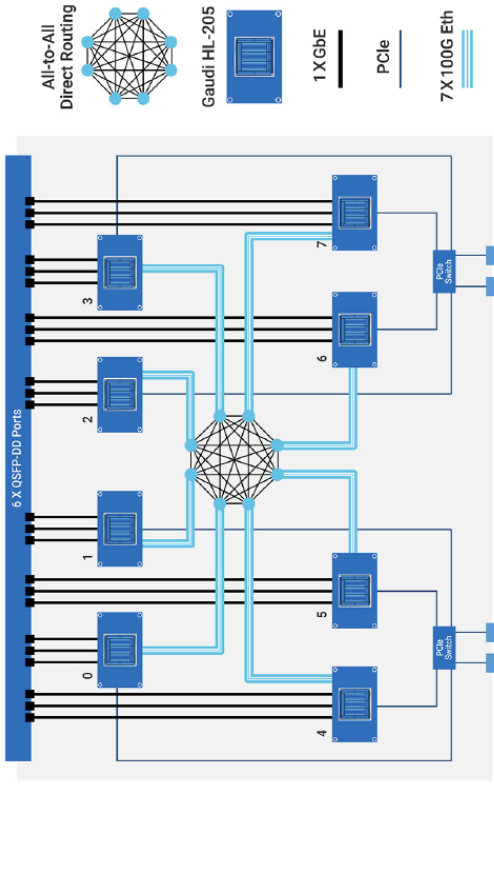
¹⁰³ “Intel Gaudi Software Suite” (https://docs.habana.ai/en/latest/Gaudi_Overview/Intel_Gaudi_Software_Suite.html).

¹⁰⁴ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p>
<p>35. A computer cluster node for evaluating expressions in parallel with other computer cluster nodes, the computer cluster node comprising:</p>	<p>Each of the Habana AI Accelerator Products¹⁰⁵ (for example, in an Accused Habana Server Product)¹⁰⁶ comprises a computer cluster node for evaluating expressions in parallel with other computer cluster nodes. For example, as depicted below and as described further herein, Intel’s Gaudi® HLS-1 AI Training System (“HLS-1”) includes a computer cluster comprising a computer cluster node for evaluating expressions in parallel with other computer cluster nodes.</p>
	<div data-bbox="451 382 906 1339" data-label="Image"> </div> <p>Each of the Accused Habana Server Products includes a plurality of computer cluster nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”,</p>

¹⁰⁵ Accused Habana AI Accelerator Products include, but are not limited to all products including or related to intel’s Gaudi (https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi.html>), Gaudi 2 (<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html>), Gaudi 3 (<https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html>), and Goya (https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf) AI accelerator products, as well as any products incorporating those items.

¹⁰⁶ The Accused Habana Server Products include, but are not limited to, all products including or related to Intel’s Gaudi® HLS-1 AI Training System (“HLS-1”) (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>), HLS-Gaudi2 server ([https://www.intel.com/content/www/us/en/content/uploads/2023/10/HLS-Gaudi2_Datasheet_10_23.pdf](https://habana.ai/wp-content/uploads/2023/10/HLS-Gaudi2_Datasheet_10_23.pdf)), Gaudi 3 AI Accelerator HLB-325 Baseboard (<https://www.intel.com/content/www/us/en/content-details/817489/intel-gaudi-3-ai-accelerator-hlb-325-baseboard-product-brief.html>), as well as any products incorporating those items.

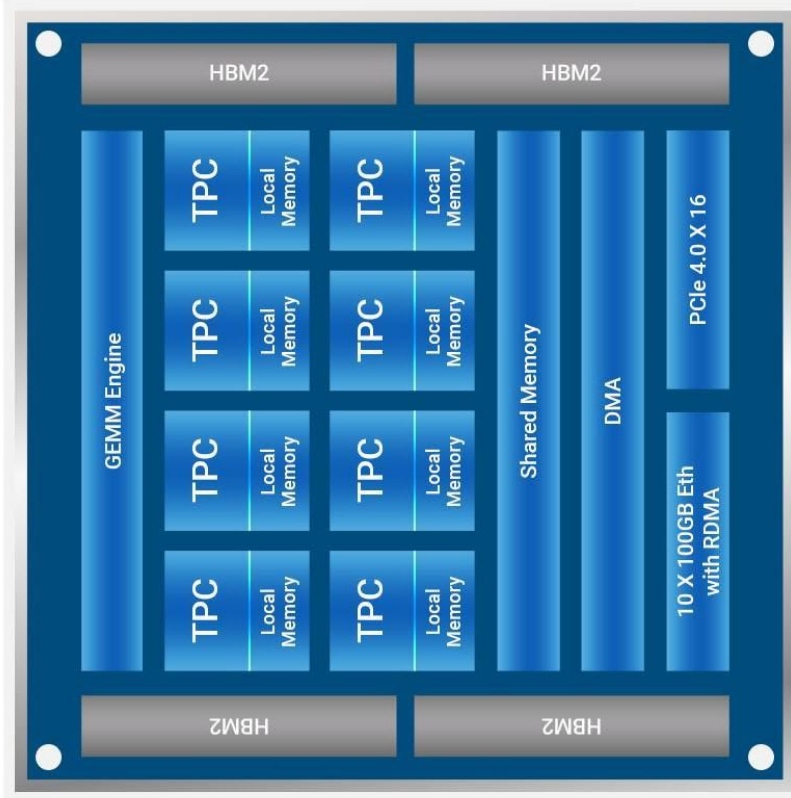
<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>“1”...“7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.¹⁰⁷</p>
<p>a hardware processor configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the hardware processor to interpret user instructions, to evaluate mathematical expressions, and to produce results of mathematical expression evaluation, wherein the hardware processor comprises multiple processor cores; a user connection interface configured to receive a command to start a cluster initialization process for a computer cluster;</p>	<p>Each of the Accused Habana Server Products comprises a plurality of nodes, each of which comprises an Accused Habana AI Accelerator Product. For example, as depicted below, the Intel HLS-1 system comprises eight Gaudi HL-205 Mezzanine cards (designated as nodes “0”, “1”...“7” in the figure below) connected in an all-to-all local bus routing by eight 7x 100G Ethernet connections.¹⁰⁸</p>  <p style="text-align: center;">HLS-1 Block Diagram</p>

¹⁰⁷ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

¹⁰⁸ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises a hardware processor configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the hardware processor to interpret user instructions, to evaluate mathematical expressions, and to produce results of mathematical expression evaluation, wherein the hardware processor comprises multiple processor cores.</p> <p>Specifically, each Accused Habana AI Accelerator Product comprises a hardware processor that comprises multiple processor cores. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM). The architecture of the Gaudi HL-2000 processor is depicted in the following figure:¹⁰⁹</p>
--	--

¹⁰⁹ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20Gaudi%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/processors/ai-accelerators/audi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/audi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



Each Accused Habana AI Accelerator Product comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Habana Server Product. For example, each Gaudi HL-205 Mezzanine card in each HLS-1 system comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system. This is done, for example, by using the Habana Collective Communication Library

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Habana Server Products and Accused Habana AI Accelerator Products (HCCL).¹¹⁰ HCCL is included in Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”¹¹¹</p> <p>Each Accused Habana AI Accelerator Product is configured to access a non-transitory computer-readable medium (<i>e.g.</i>, memory accessible by a Gaudi HL-2000 processor, including local memory, shared memory, and HBM2 memory, as well as memory external to the Gaudi HL-2000 processor).¹¹²</p> <p>Each Accused Habana AI Accelerator Product in each Accused Habana Server Product comprises program code for a single-node kernel that, when executed, causes the hardware processor in the Accused Habana AI Accelerator Product to interpret user instructions, to evaluate mathematical expressions, and to produce results of mathematical expression evaluation (<i>e.g.</i>, executable SynapseAI® Software Suite program code stored in memory accessible by the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card). For example, when executed, the code stored in the memory accessible by a Gaudi HL-2000 processor causes the hardware processor to interpret user instructions, to evaluate mathematical expressions (<i>e.g.</i>, by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library), and to produce results of mathematical expression evaluation. TPCs are configured to evaluate multiple mathematical expressions, such as square root, sine, cosine, tangent, and reciprocal, as well as matrix operations such as Vector Reduction Intrinsic.¹¹³</p> <p>Each Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card of the HLS-1 system is configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the processor to produce results of mathematical expression evaluation. A first single-node kernel is responsible for interpreting user instructions and distributing</p>
--	---

¹¹⁰ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

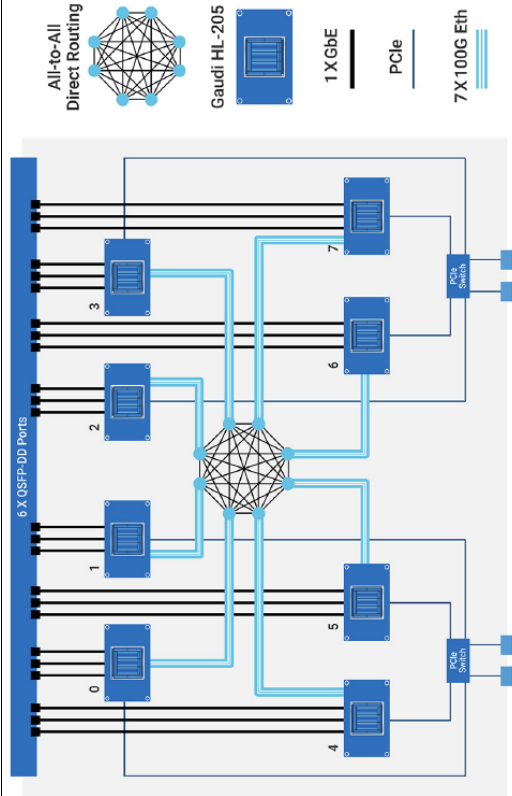
¹¹¹ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

¹¹² Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

¹¹³ See, *e.g.*, Built-in Functions (“https://docs.habana.ai/en/latest/TPC/TPC_User_Guide/Built_in_Functions.html”).

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>calls to at least one of the other Gaudi HL-205 accelerators for execution. The other Gaudi HL-205 accelerators in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user. Hence, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card is configured to access executable program code stored in memory accessible by the Gaudi HL-2000 processor, where the executable program code is program code for a single-node kernel that, when executed, causes the processor to interpret user instructions.</p>
<p>a mechanism to communicate results of evaluation with other computer cluster nodes using a peer-to-peer architecture; and</p>	<p>Each of the Accused Habana AI Accelerator Products in each Accused Habana Server Product comprises a mechanism for the Accused Habana AI Accelerator Products to communicate results of evaluation with other Accused Habana AI Accelerator Products using a peer-to-peer architecture (e.g., the all-to-all direct routing connectivity between the eight Gaudi HL-205 Mezzanine cards in the HLS-1 system shown in the following block diagram,¹¹⁴ along with related collective HCCL commands included in Intel’s SynapseAI® Software Suite).</p>

¹¹⁴ Gaudi® HLS-1 AI Training System (<https://www.intel.com/content/www/us/en/content-details/784787/gaudi-hls-1-ai-training-system.html>); see also HLS-1 introductory video (“Habana Labs: How to Scale AI Training with Gaudi Processors”), at <https://www.youtube.com/watch?v=u0siCfmCNfg>; see also <https://habana.ai/products/gaudi/>.



HLS-1 Block Diagram

For example, Intel also provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”¹¹⁵ The source code for HCCL Demo¹¹⁶ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, and for the Gaudi HL-205 Mezzanine card nodes in the HLS-1 system to communicate results of evaluation with other Gaudi HL-205 Mezzanine card nodes using a peer-to-peer architecture.

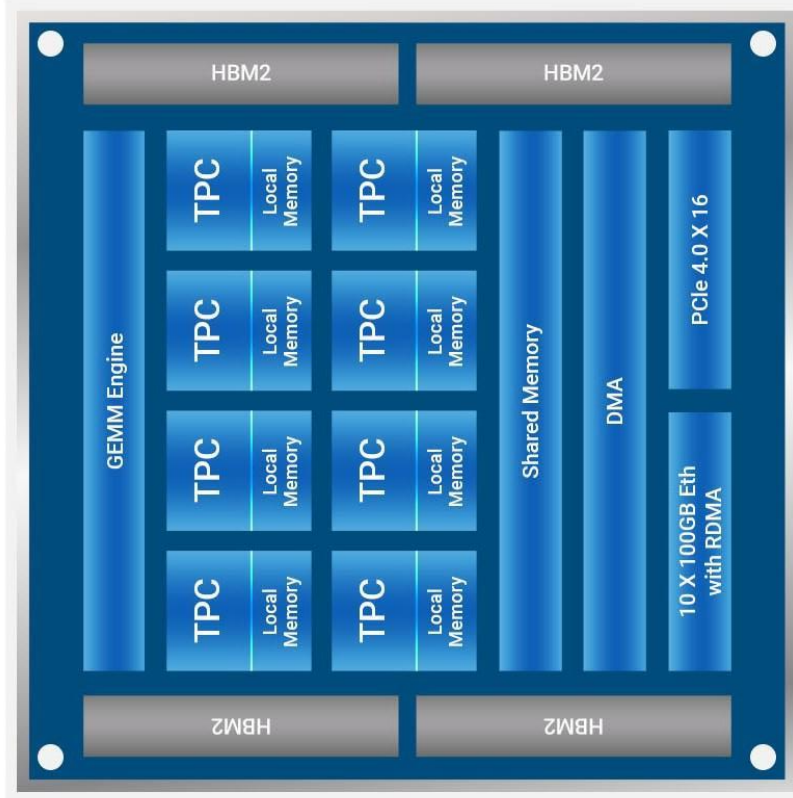
Each of the plurality of Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprises program code that, when executed, is capable of causing the hardware processor to: receive calls from a second node comprising a second hardware processor configured to access a second memory comprising program code for a user interface and program code for a second

program code that, when executed, is capable of causing the hardware processor to: receive calls from a second node

¹¹⁵ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/Testing_and_Benchmarking.html).
¹¹⁶ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

<p>U.S. Pat. No. 10,333,768</p> <p>comprising a second hardware processor configured to access a second memory comprising program code for a user interface and program code for a second single-node kernel, the second single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution;</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>single-node kernel, the second single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution.</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure:¹¹⁷</p>
--	--

¹¹⁷ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also (https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html); <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi.html> (Gaudi), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi2.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



For example, the Habana Collective Communication Library (HCCL) includes commands that cause a Gaudi HL-2000 processor of a Gaudi HL-205 mezzanine card in a HLS-1 system to receive calls from a second Gaudi HL-205 mezzanine card in the HLS-1 system comprising a second Gaudi HL-2000 processor configured to access a second memory comprising program code for a user interface and program code configured to interpret user instructions and distribute calls to at least one of the other Gaudi HL-205 mezzanine card nodes in the HLS-1 system for execution.¹¹⁸ HCCL is included in

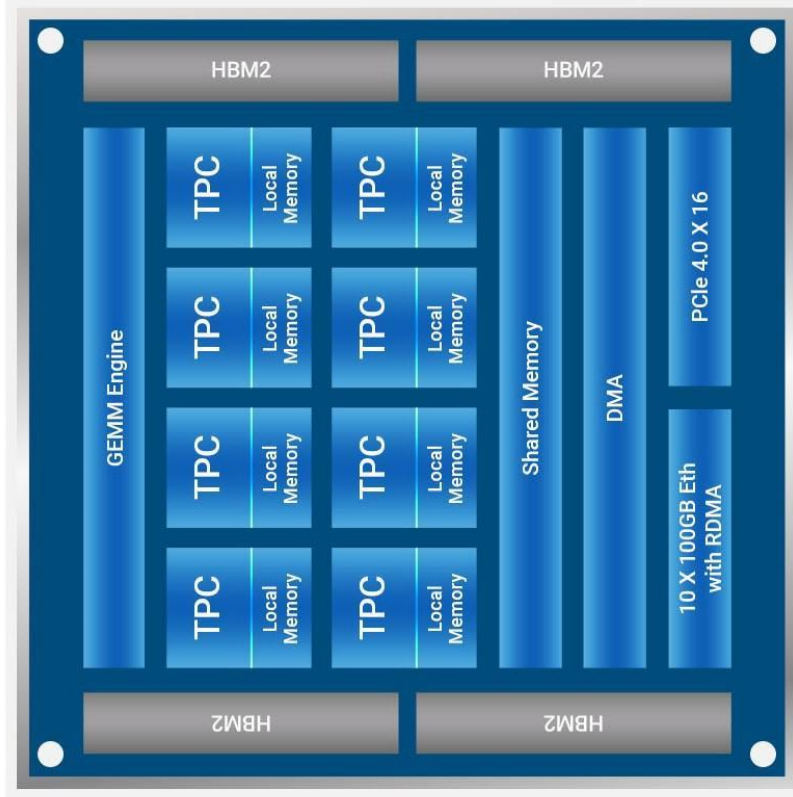
¹¹⁸ “Habana Collective Communications Library (HCCL) is Habana’s emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library.” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p>
<p>execute, using the hardware processor, at least a first mathematical expression evaluation; and communicate a result of the first mathematical expression evaluation to a third node comprising</p>	<p>Intel’s SynapseAI® Software Suite. SynapseAI® is “Habana’s complete software stack custom designed to support Habana’s Gaudi implementation.”¹¹⁹</p> <p>The source code for Intel’s HCCL Demo program¹²⁰ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for a user interface and program code configured to interpret user instructions and distribute calls to at least one of the other Gaudi HL-205 Mezzanine card nodes in the HLS-1 system for execution.</p> <p>A first Accused Habana AI Accelerator Product in an Accused Habana Server Product comprises program code that, when executed, is capable of causing a hardware processor in the first Accused Habana AI Accelerator Product to execute, using the hardware processor, at least a first mathematical expression evaluation and communicate a result of the first mathematical expression evaluation to a third Accused Habana AI Accelerator Product in the Accused Habana Server Product.</p> <p>Each Accused Habana AI Accelerator Product comprises a hardware processor. For example, each Gaudi HL-205 Mezzanine card is a node that includes a Gaudi HL-2000 processor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as depicted in the following figure:¹²¹</p>

¹¹⁹ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

¹²⁰ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py.

¹²¹ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); see also https://docs.habana.ai/en/latest/Gaudi_Overview/Gaudi_Architecture.html; <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi.html> (Gaudi 2), <https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html> (Gaudi 3), https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf (Goya).



For example, Intel provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”¹²² The source code for HCCL Demo¹²³ includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, accessing program code for executing mathematical expression evaluations, and for communicating a result of mathematical expression evaluations to other nodes.

¹²² See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/Testing_and_Benchmarking.html).
¹²³ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

U.S. Pat. No. 10,333,768	Accused Habana Server Products and Accused Habana AI Accelerator Products
<p>a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the computer cluster node, execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation, and communicate a result of the second mathematical expression evaluation to the first node;</p>	<p>A third Accused Habana AI Accelerator Product in an Accused Habana Server Product to which a first Accused Habana AI Accelerator Product in an Accused Habana Server Product communicates a result of a first mathematical expression evaluation comprises a hardware processor with multiple processing cores, and is configured to receive the result of mathematical expression evaluation from the first Accused Habana AI Accelerator Product in the Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation (e.g., via executable SynapseAI® Software Suite program code stored in memory in communication with a Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card), and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product.</p> <p>For example, when executed, the code stored in the memory accessible by the Gaudi HL-2000 processor of the third Gaudi HL-205 Mezzanine card in a HLS-1 system causes the processor to receive a result of a first mathematical expression evaluation from a first Gaudi HL-205 Mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation (e.g., by performing matrix multiplications using the General Matrix Multiplication (“GEMM”) engine, evaluating mathematical expressions using the Tensor Processing Cores in the Gaudi HL-2000 processor, and/or executing commands in the HCCL library), and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-205 Mezzanine card in the HLS-1 system.</p> <p>Also for example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a third Gaudi HL-205 mezzanine card in an HLS-1 system is configured to receive the result of a first mathematical expression evaluation from the first Gaudi HL-205 mezzanine card in the HLS-1 system, to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation, and to communicate the result of the second mathematical expression evaluation to the first Gaudi HL-205 Mezzanine card in the HLS-1 system. The source</p>

U.S. Pat. No. 10,333,768	<p data-bbox="185 182 483 1528">Accused Habana Server Products and Accused Habana AI Accelerator Products code for Intel’s HCCL Demo program,¹²⁴ for example, includes the details for using HCCL commands to receive the result of a first mathematical expression evaluation from a first Accused Habana AI Accelerator Product in an Accused Habana Server Product, to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation, and to communicate the result of the second mathematical expression evaluation to the first Accused Habana AI Accelerator Product in the Accused Habana Server Product.</p> <p data-bbox="483 182 602 1528">A user connection interface in a first Accused Habana AI Accelerator Product in each Accused Habana Server Product is configured to return at least one result of mathematical expression evaluation to a user interface or a script.</p> <p data-bbox="602 182 971 1528">For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel’s SynapseAI® Software Suite, the Gaudi HL-2000 processor in a first Gaudi HL-205 mezzanine card node in a HLS-1 system comprises a user connection interface that is configured to return at least one result of mathematical expression evaluation to a user interface or a script. Intel provides “HCCL Demo,” which is “a program utilizing HCCL APIs, demonstrating both collective and point to point communication.”¹²⁵ The source code for Intel’s HCCL Demo program¹²⁶ includes the details for using HCCL commands to implement a user connection interface configured to return at least one result of mathematical expression evaluation from a Gaudi HL-205 mezzanine card node in a HLS-1 system to the user interface or a script.</p> <p data-bbox="971 182 1218 1528">One or more of the Accused Habana AI Accelerator Products in an Accused Habana Server Product are configured to: (1) accept user instructions; (2) after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and (3) after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernel on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p>
<p data-bbox="228 1528 483 1904">wherein the user connection interface is configured to return at least one result of mathematical expression evaluation to a user interface or a script; and</p> <p data-bbox="483 1528 997 1904">wherein the computer cluster node is configured to: accept user instructions; after accepting user instructions, communicate at least some of the user</p>	

¹²⁴ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

¹²⁵ See “Testing and Benchmarking – HCCL Demo” (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIS/Testing_and_Benchmarking.html).

¹²⁶ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

U.S. Pat. No. 10,333,768	Accused Habana Server Products and Accused Habana AI Accelerator Products
<p>instructions using the mechanism for the nodes to communicate with each other; and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernel.</p>	<p>For example, at least because of the architecture of the Gaudi HL-2000 processor along with related resources included in Intel's SynapseAI® Software Suite, the Gaudi HL-2000 processor in a Gaudi HL-205 mezzanine card is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p> <p>Also for example, the source code for Intel's HCCL Demo program¹²⁷ includes the details for using HCCL commands to cause an Accused Habana AI Accelerator Products in an Accused Habana Server Product to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the Accused Habana AI Accelerator Products to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more Accused Habana AI Accelerator Products in an Accused Habana Server Product.</p>
<p>37. The computer cluster node of claim 35, wherein the computer cluster node is configured to permit exchange of information with other computer cluster nodes during the course of parallel computation.</p>	<p>The Accused Habana AI Accelerator Product nodes in each Accused Habana Server Product comprise one or more cluster nodes, for example, by supporting and/or executing the Habana Collective Communication Library (HCCL).¹²⁸ HCCL is included in Intel's SynapseAI® Software Suite. SynapseAI® is "Habana's complete software stack custom designed to support Habana's Gaudi implementation."¹²⁹</p> <p>Each of the cluster nodes of the Accused Habana Server Products is configured to permit exchange of information between nodes during the course of parallel computation.</p>

¹²⁷ See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py; https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

¹²⁸ "Habana Collective Communications Library (HCCL) is Habana's emulation layer of the NVIDIA Collective Communication Library (NCCL) and is included in the SynapseAI® Software library." (https://docs.habana.ai/en/latest/API_Reference_Guides/HCCL_APIs/index.html).

¹²⁹ Gaudi® Training Platform White Paper, Nov 2020, p. 7 (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>).

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Habana Server Products and Accused Habana AI Accelerator Products</p> <p>For example, the Intel Gaudi Software Suite, which comprises Intel’s SynapseAI® Software Suite and the Intel Gaudi graph compiler, “generates optimized binary code that implements the given model topology on Gaudi. It performs operator fusion, data layout management, parallelization, pipelining and memory management, as well as graph-level optimizations. The graph compiler uses the rich TPC kernel library¹³⁰ which contains a wide variety of operations (for example, elementwise, non-linear, non-GEMM operators).”¹³¹ The source code for Intel software such as Intel’s HCCL Demo program¹³² includes the details for using HCCL commands to initialize a HCCL communicator on a HLS-1 system comprising multiple Gaudi HL-205 Mezzanine card nodes, enable intercommunication among a plurality of single-node kernels during thread execution by a plurality of cluster node modules, and permit exchange of information between the nodes during the course of parallel computation.</p>
<p>39. The computer cluster node of claim 35, wherein the hardware processor comprises a special purpose microprocessor.</p>	<p>Each Habana AI Accelerator Product is a hardware processor that comprises a special purpose microprocessor. For example, each Gaudi HL-205 Mezzanine card includes special purpose Gaudi HL-2000 microprocessor that contains a cluster of eight programmable Tensor Processing Cores (TPC) and a General Matrix Multiplication Engine (GEMM), as shown below.¹³³</p>

¹³⁰ “Intel® Gaudi® processors TPC kernel library with supporting firmware, drivers and tools” (<https://developer.habana.ai/get-started/kernel-libraries/>).

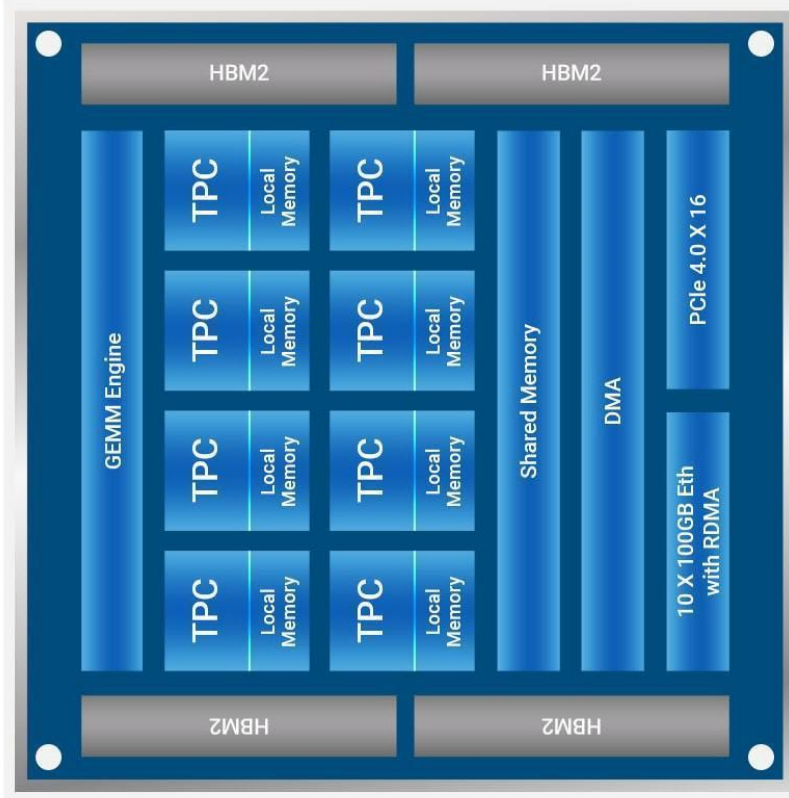
¹³¹ “Intel Gaudi Software Suite” (https://docs.habana.ai/en/latest/Gaudi_Overview/Intel_Gaudi_Software_Suite.html).

¹³² See https://github.com/HabanaAI/hccl_demo/blob/main/hccl_demo.cpp; see also https://github.com/HabanaAI/hccl_demo/blob/main/run_hccl_demo.py, https://github.com/HabanaAI/hccl_demo/blob/main/README.md.

¹³³ Gaudi™ Training Platform White Paper, Nov 2020, p. 6, (<https://habana.ai/wp-content/uploads/pdf/2020/Habana%20GAUDI%20Training%20Whitepaper%20v1.2.pdf>); Gaudi HL-2000 Data Sheet (<https://habana.ai/wp-content/uploads/2019/06/Gaudi-Datasheet.pdf>) (“The Gaudi™ HL-2000 is an advanced AI and Deep Learning Training Processor, leveraging purpose-built architecture and delivering superior performance, scalability, power efficiency and cost savings.... The Gaudi is designed to accelerate various AI Training workloads such as image recognition, neural machine translation, sentiment analysis, recommender systems and many others.”).

U.S. Pat. No. 10,333,768

Accused Habana Server Products and Accused Habana AI Accelerator Products



U.S. Pat. No. 10,333,768

a plurality of nodes, wherein each of the plurality of nodes comprises a hardware processor, wherein one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster, and wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions; and

Accused Xeon Products

Each of the Accused Xeon Products comprises a plurality of nodes. For example, as depicted below, the 4th Gen Xeon Scalable Processor comprises four nodes (described as domains in the figure below), which are configured to be connected in an all-to-all local bus interconnection configuration.³

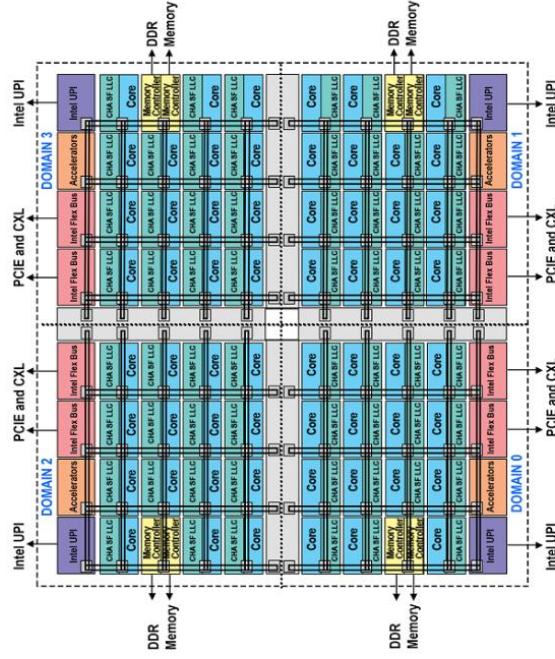


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Each of the 4th Gen Xeon Scalable Processors is configured to operate in a Sub-NUMA clustering 4 (SNC-4) affinity mode, comprising four domains of sub-NUMA clusters / nodes.

Each Accused Xeon Product comprises a plurality of nodes, wherein each of the plurality of nodes comprises a hardware processor, wherein one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster, and wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code

³ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions.

Each Accused Xeon Product comprises a hardware processor that comprises multiple processor cores. For example, the 4th Gen Xeon Scalable Processor is configured to operate in an SNC-4 mode where each of the four domains is a node that includes a plurality of processing cores (fifteen) as shown in the above figure.

Each Accused Xeon Product comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Xeon Product. For example, each 4th Gen Xeon Scalable Processor comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the four domains (node). This is done, for example, by using Intel's oneCCL software, which is part of Intel's oneAPI Collective Communications Library.⁴ More particularly, Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...."⁵ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.⁶ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.⁷

Each Accused Xeon Product is configured to access a non-transitory computer-readable medium as shown in the figure above, including Last Level Cache (LLC), DDR memory, and High-bandwidth Memory (HBM).

⁴ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.
⁵ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87Ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/communicator.html>.
⁶ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
⁷ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

Each Accused Xeon Product is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions. For example, the 4th Gen Xeon Scalable Processor is configured to access and execute code stored in the memory, and is further configured to cause the processor to evaluate mathematical expressions.⁸ ⁹ For example, Intel's 4th Gen Xeon Scalable Processors are configured such that each domain / node in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes at least two types of compute engines for evaluating mathematical expressions including matrix operations using multiple processors and cores:

- Intel Advanced Matrix Extensions Accelerator (AMX): AMX is a specialized hardware accelerator for evaluating matrix multiplication expressions.¹⁰
- Processing Cores (PCs): The PCs are general-purpose hardware accelerators that can be used to evaluate a variety of mathematical expressions.¹¹

Each 4th Gen Xeon Scalable Processor is, therefore, configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the processor to produce results of mathematical expression evaluation. More particularly, a first single-node kernel is responsible for interpreting user instructions and distributing calls among the domains of the 4th Gen Xeon Scalable Processor for execution. Other domains in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. When the user submits a job to the cluster, for example, the first single-node kernel parses the job and distributes the tasks to the other domains (nodes) for execution. The other domains (nodes) then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user. Hence, the 4th Gen Xeon Scalable Processor is configured to access

⁸ "Technical Overview Of The 4th Gen Intel Xeon Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

⁹ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., "Intel® AVX-512 - FP16 Instruction Set for Intel® Xeon® Processor Based Products Technology Guide," <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

¹⁰ <https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/advanced-matrix-extensions/overview.html>.

¹¹ <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

U.S. Pat. No. 10,333,768	Accused Xeon Products
<p>wherein the plurality of nodes comprises: a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution;</p> <p>and</p>	<p>The 4th Gen Xeon Scalable Processor is configured to implement Intel's oneCCL library,¹⁴ which includes commands that provide a mechanism for the domains / nodes in a 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture, either standing alone or, for example, in conjunction with the processor core built-in mathematical functions.^{15 16 17}</p> <p>Each of the Accused Xeon Products comprises a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.¹⁸</p>

¹⁴ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.
¹⁵ "Technical Overview Of The 4th Gen Intel Xeon Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

¹⁶ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. *See, e.g.,* "Intel® AVX-512 - FPU Instruction Set for Intel® Xeon® Processor Based Products Technology Guide," <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fpu-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

¹⁷ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.
¹⁸ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.²² The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:²³</p>
--	---

²² “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

²³ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

Accused Xeon Products

The CCL library provides multiple commands for creating, managing, and using CCL communicators.²⁴ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.²⁵ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

The first single-node kernel is, for example, a software program that runs on the first domain / node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other domains / nodes for execution. The other domains / nodes in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. This is known as the master-slave model, where a first domain / node executes in a supervisory role, and the

²⁴ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
²⁵ <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768	Accused Xeon Products
	<p>other domains / nodes serve as compute nodes. For example, initially, the first node is configured to interpret user instructions and distribute calls to at least one of the other nodes for execution.</p> <p>Unlike a master-slave model, the nodes communicate results of mathematical expression evaluation with each other during execution of the tasks that are distributed to them by the first node using a peer-to-peer architecture without being required to go through the first node.</p> <p>When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel. The first single-node kernel then collects the results from the other nodes and returns them to the user.</p>
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node;</p>	<p>Each of the Accused Xeon Products comprises a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.²⁶</p>

²⁶ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

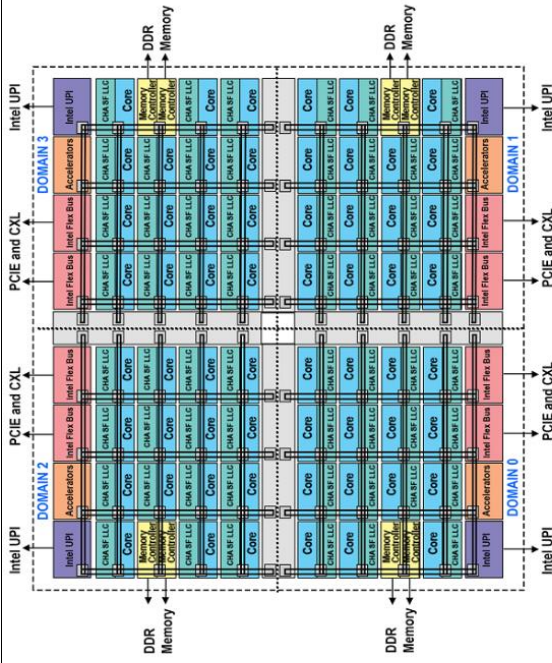


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors....”²⁷ The oneCCL library is one of the 10 core elements of Intel’s oneAPI software suite.²⁸ Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.²⁹

“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL

²⁷ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87c1qd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
²⁸ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
²⁹ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.³⁰ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors.³¹</p>
--	---

³⁰ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

³¹ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

Accused Xeon Products

The diagram illustrates the Intel oneAPI ecosystem, organized into several layers:

- Platforms and Kits:** Includes Intel® Developer Catalog Pretrained Models, Intel® Developer Cloud, cnvrg.io* MLOps, AI Tools/AI Acceleration, SIGOPT Optimization (oneAPI XGBoost Integration), BigDL, Spark, and OpenVINO Deployment.
- Frameworks:** Includes pandas, MODIN, Spark, XGBoost, Leaven, TensorFlow, and PyTorch.
- Libraries:** Includes Intel® oneAPI Data Analytics Library (oneDAL), Intel® oneAPI Deep Neural Network Library (oneDNN), Intel® oneAPI Collective Communications Library (oneCCL), and Intel® oneAPI Math Kernel Library (oneMKL).
- Languages:** Fortran, SYCL*, DPC++/C++.
- Analyzers:** Intel® Profiler, Intel® Trace Collector and Analyzer.
- Debuggers:** GDB*, Intel® SoC Watch, Intel® System Debugger, Intel® Cluster Checker, Model Zoo.
- Migration:** SYCLomatic, Intel® DPC++ Compatibility Tool.
- Hypervisors / Orchestration:** KVM 5.17, Hyper-V 1H*22, ESXi*/vSphere* 8.0, Kubernetes*, RedHat OpenShift*.
- OS & Kernel:** Microsoft Windows Server* 2022, Linux* 5.16, Red Hat 8.6, Ubuntu* 22.04, SUSE Linux Enterprise Server* 15 SP4.
- Hardware:** 4th & 5th Gen Intel® Xeon® Scalable Processor, Intel® Xeon® CPU Max Series, Intel® Data Center GPU Max Series.

The CCL library provides multiple commands for creating, managing, and using CCL communicators.³² The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.³³ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Each of the Accused Xeon Products comprises a hardware processor with multiple processing cores, and is configured to receive the result of a first mathematical expression evaluation from the second node and to execute at least a second mathematical expression evaluation using the received result.

wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is

³² Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
³³ <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768

configured to receive the result of the first mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;

Accused Xeon Products

For example, when executed, the code stored in the memory accessible by the 4th Gen Xeon Scalable Processor causes the processor to receive a result of a first mathematical expression evaluation from a second node (domain) in the 4th Gen Xeon Scalable Processor and to execute at least a second mathematical expression evaluation using the received result (e.g., by performing matrix multiplications).

For example, as depicted below, the Intel 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.

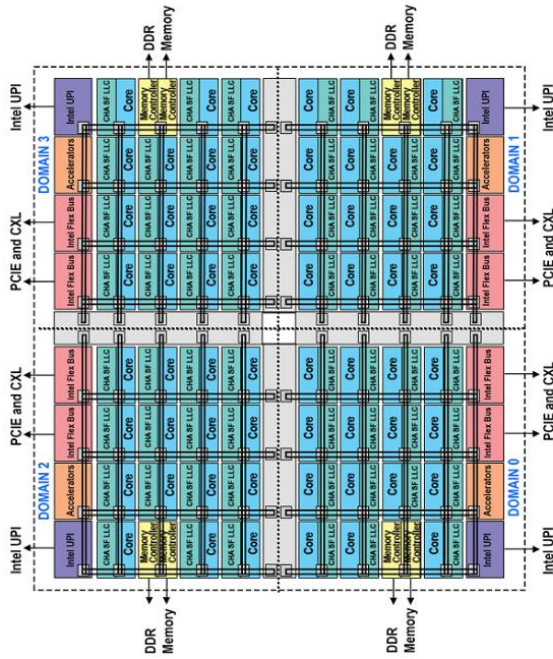


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of

U.S. Pat. No. 10,333,768	<p style="text-align: center;">Accused Xeon Products</p> <p>4th and 5th gen Intel® Xeon® processors...³⁴ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.³⁵ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.³⁶</p> <p>“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”³⁷ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel's oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel's 4th Gen Xeon Scalable Processors:³⁸</p>
--------------------------	---

³⁴ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/communicator.html>.

³⁵ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

³⁶ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide/2021-12/overview.html>.

³⁷ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

³⁸ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

Accused Xeon Products

The diagram illustrates the Intel oneAPI ecosystem, organized into several layers:

- Platforms and Kits:** Includes Intel Developer Catalog, Intel Developer Cloud, cnvrg.io* MLOps, AI Tools/AI Acceleration, SIGOPT Optimization (oneAPI XGBoost Integration), BigDL, Spark, and OpenVINO Deployment.
- Frameworks:** Includes pandas, MODIN, Spark, XGBoost, Leaven, TensorFlow, and PyTorch.
- Libraries:** Includes Intel oneAPI Data Analytics Library (oneDAL), Intel oneAPI Deep Neural Network Library (oneDNN), Intel oneAPI Collective Communications Library (oneCCL), and Intel oneAPI Math Kernel Library (oneMKL).
- Languages:** Includes Fortran, SYCL*, and DPC++/C++.
- Analyzers:** Includes Intel Inspector, Intel Profiler, Intel Trace Collector and Analyzer, and Intel Neural Compressor.
- Debuggers:** Includes GDB*, Intel SoC Watch, Intel System Debugger, Intel ClusterChecker, and Model Zoo.
- Migration:** Includes SYCLomatic and Intel DPC++ Compatibility Tool.
- Hypervisors / Orchestration:** Includes KVM 5.17, Hyper-V 1H*22, ESXi*/vSphere* 8.0, Kubernetes*, and RedHat OpenShift*.
- OS & Kernel:** Includes Microsoft Windows Server* 2022, Linux* 5.16, Red Hat 8.6, Ubuntu* 22.04, and SUSE Linux Enterprise Server* 15 SP4.
- Hardware:** Includes 4th & 5th Gen Intel Xeon Scalable Processor, Intel Xeon CPU Max Series, and Intel Data Center GPU Max Series.

The CCL library provides multiple commands for creating, managing, and using CCL communicators.³⁹ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.⁴⁰ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Also, for example, the architecture of the 4th Gen Xeon Scalable Processor is configured to receive the result of a first mathematical expression evaluation from the second node (domain) and to execute at least a second mathematical expression evaluation using the received result. The source code for Intel's oneCCL software, for example, includes the details for using CCL commands to receive the

³⁹ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
⁴⁰ <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>result of a first mathematical expression evaluation from a second node and to execute at least a second mathematical expression evaluation using the received result.</p>
<p>wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface;</p>	<p>Each of the Accused Xeon Products includes a first node configured to return the result of a second mathematical expression evaluation to the user interface.</p> <p>Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors..."⁴¹ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.⁴² Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.⁴³</p> <p>The CCL library provides multiple commands for creating, managing, and using CCL communicators.⁴⁴ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.⁴⁵ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.</p>
<p>wherein one or more of the nodes are configured to: accept user instructions; after accepting user instructions, communicate</p>	<p>Each of the Accused Xeon Products includes one or more nodes configured to accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.</p>

⁴¹ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁴² <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁴³ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

⁴⁴ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> ("Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.")

⁴⁵ <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
<p>U.S. Pat. No. 10,333,768</p> <p>at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.</p>	<p>For example, the architecture of the 4th Gen Xeon Scalable Processor is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the domains to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p> <p>Also for example, the source code for Intel's oneCCL software includes the details for using CCL commands to cause 4th Gen Xeon Scalable Processor to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the 4th Gen Xeon Scalable Processor to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p>
<p>4. The computer cluster of claim 1, wherein each of the nodes comprises one or more cluster node modules.</p>	<p>Each of the Accused Xeon Products comprises a plurality of nodes that comprises one or more cluster node modules. For example, as depicted below, the 4th Gen Xeon Scalable Processor comprises four nodes (described as domains in the figure below), each of which is configured to access memory including, for example, LLC and DDR memory.⁴⁶</p>

⁴⁶ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

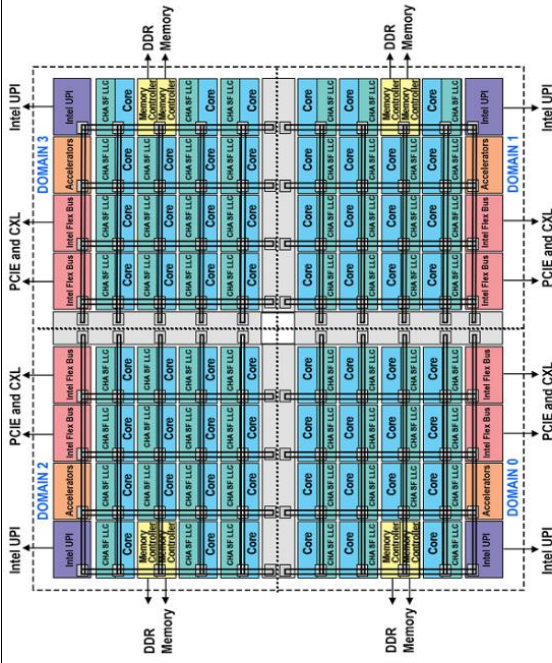


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Each Accused Xeon Product comprises a plurality of nodes, wherein each of the plurality of nodes comprises at least one computer-readable medium in communication with at least one of the first processor, the second processor, or the third processor. For example, the 4th Gen Xeon Scalable Processor, as shown above, is configured to access and be in communication with a non-transitory computer-readable medium that includes, for example, local memory (e.g., LLC), and external memory (e.g., DDR memory). The 4th Gen Xeon Scalable Processor is, therefore, configured (see Memory Controller) to have memory accessible by and in communication with the processor.

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, including Intel’s oneCCL software.⁴⁷ For example, Intel’s 4th Gen Xeon Scalable Processor are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon®

⁴⁷ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

processors....⁴⁸ The oneCCL library is one of the 10 core elements of Intel’s oneAPI software suite.⁴⁹ Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.⁵⁰

“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL.* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”⁵¹ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.

The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the exemplary context of the software suite for Intel’s 4th Gen Xeon Scalable Processors.⁵²

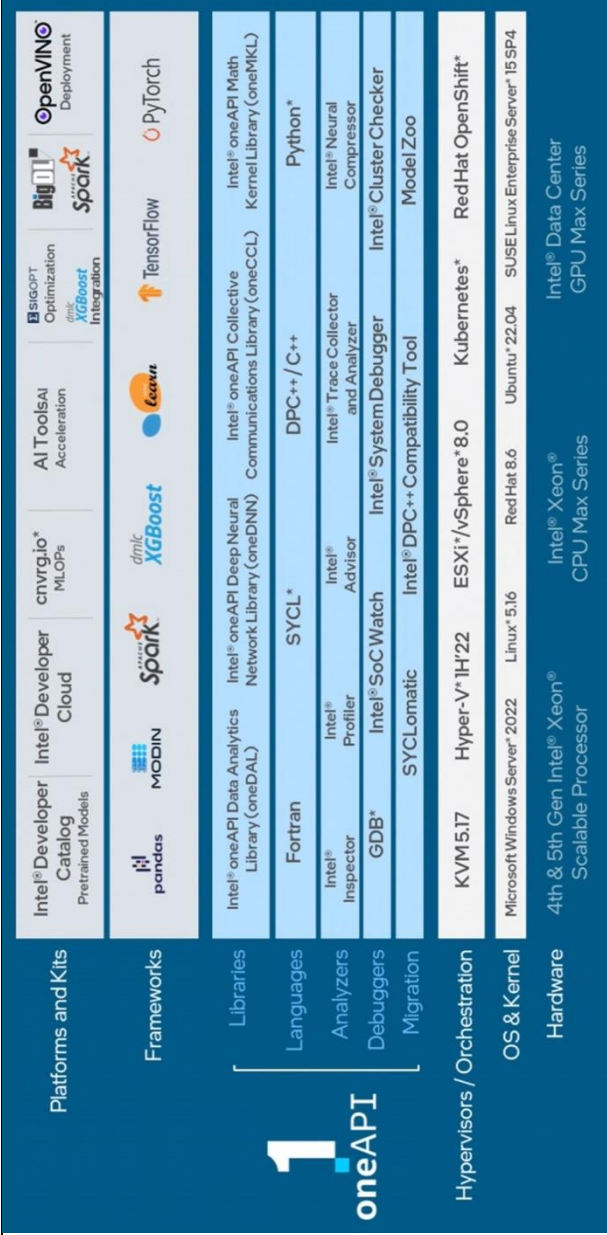
⁴⁸ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁴⁹ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁵⁰ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

⁵¹ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

⁵² “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.



The CCL library provides multiple commands for creating, managing, and using CCL communicators.⁵³ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.⁵⁴ The CCL library is responsible for receiving first commands from a user interface without the first commands first passing through the first kernel, and after receiving the first commands from the user interface, send second commands to the first kernel.

The source code for the CCL library includes the details for using oneCCL commands to initialize communications on a 4th Gen Xeon Scalable Processor, receive first commands from a user interface without the first commands first passing through the first kernel, and after receiving the first commands from the user interface, send second commands to the first kernel.

⁵³ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
⁵⁴ <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768

20. The computer cluster of claim 1, wherein one or more of the nodes are configured to accept user instructions via one or more of the nodes.

Accused Xeon Products

Each of the Accused Xeon Products comprises a computer cluster wherein one or more of the nodes are configured to accept user instructions via one or more of the nodes.

For example, as depicted below, the 4th Gen Xeon Scalable Processor is configured to provide all-to-all communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.⁵⁵

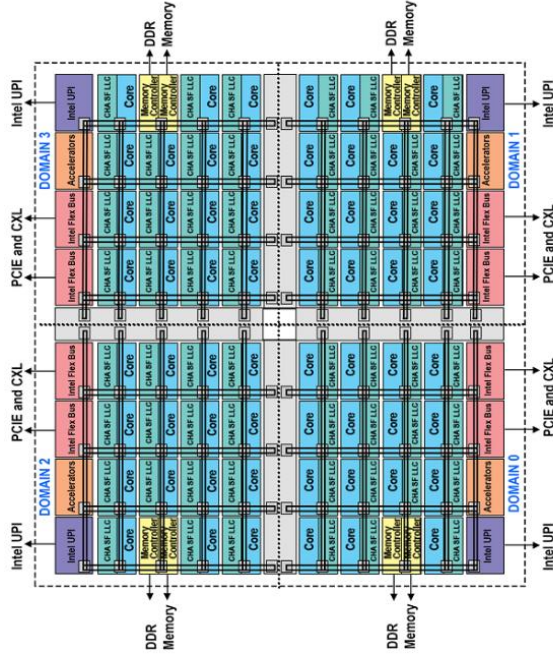


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Each of the 4th Gen Xeon Scalable Processors is configured to operate in a Sub-NUMA clustering 4 (SNC-4) affinity mode, comprising four domains of sub-NUMA clusters / nodes.

Each Accused Xeon Product comprises a hardware processor that comprises multiple processor cores. For example, the 4th Gen Xeon Scalable Processor is configured to operate in an SNC-4 mode where

⁵⁵ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

each of the four domains is a node that includes a plurality of processing cores (fifteen) as shown in the above figure.

Each Accused Xeon Product comprises one or more of the nodes configured to accept user instructions via one or more of the nodes a user connection interface. For example, each Accused Xeon Product comprises a node configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Xeon Product. For example, each 4th Gen Xeon Scalable Processor comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the four domains (node). For example, Intel's 4th Gen Xeon Scalable Processors are configured such that each domain / node in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes at least two types of compute engines for evaluating mathematical expressions including matrix operations using multiple processors and cores:

- Intel Advanced Matrix Extensions Accelerator (AMX): AMX is a specialized hardware accelerator for evaluating matrix multiplication expressions.⁵⁶
- Processing Cores (PCs): The PCs are general-purpose hardware accelerators that can be used to evaluate a variety of mathematical expressions.⁵⁷

This is done, for example, by using Intel's oneCCL software, which is part of Intel's oneAPI Collective Communications Library.⁵⁸ More particularly, Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...."⁵⁹ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.⁶⁰

⁵⁶ <https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/advanced-matrix-extensions/overview.html>.

⁵⁷ <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

⁵⁸ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

⁵⁹ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87cigd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/communicator.html>.

⁶⁰ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

U.S. Pat. No. 10,333,768	<p style="text-align: center;">Accused Xeon Products</p> <p>Each Accused Xeon Product comprises a hardware processor that comprises multiple processor cores. For example, the 4th Gen Xeon Scalable Processor is configured to operate in an SNC-4 mode where each of the four domains is a node that includes a plurality of processing cores (fifteen) as shown in the above figure.</p> <p>Each Accused Xeon Product comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Xeon Product. For example, each 4th Gen Xeon Scalable Processor comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the four domains (node). This is done, for example, by using Intel's oneCCL software, which is part of Intel's oneAPI Collective Communications Library.⁶³ More particularly, Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...."⁶⁴ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.⁶⁵ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.⁶⁶</p> <p>Each Accused Xeon Product is configured to access program code for a single-node kernel that, when executed, causes the hardware processor to interpret user instructions to evaluate mathematical expressions and to produce results of mathematical expression evaluation. For example, the 4th Gen Xeon Scalable Processor is configured to access and execute code stored in the memory, and is further</p>
--------------------------	--

⁶³ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

⁶⁴ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87Ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/communicator.html>.

⁶⁵ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁶⁶ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

U.S. Pat. No. 10,333,768	<p style="text-align: center;">Accused Xeon Products</p> <p>configured to cause the processor to evaluate mathematical expressions.⁶⁷ For example, Intel’s 4th Gen Xeon Scalable Processors are configured such that each domain / node in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes at least two types of compute engines for evaluating mathematical expressions including matrix operations using multiple processors and cores:</p> <ul style="list-style-type: none"> • Intel Advanced Matrix Extensions Accelerator (AMX): AMX is a specialized hardware accelerator for evaluating matrix multiplication expressions.⁶⁹ • Processing Cores (PCs): The PCs are general-purpose hardware accelerators that can be used to evaluate a variety of mathematical expressions.⁷⁰ <p>Each 4th Gen Xeon Scalable Processor is configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the processor to produce results of mathematical expression evaluation. A first single-node kernel is responsible for interpreting user instructions and distributing calls among the domains of the 4th Gen Xeon Scalable Processor for execution. Other domains in the cluster are responsible for executing the tasks that are distributed to them. When the user submits a job to the cluster, for example through a user interface, the first single-node kernel parses the job and distributes the tasks to the other domains (nodes) for execution.</p>
---------------------------------	--

⁶⁷ “Technical Overview Of The 4th Gen Intel Xeon Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

⁶⁸ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., “Intel® AVX-512 - FP16 Instruction Set for Intel® Xeon® Processor Based Products Technology Guide,” <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

⁶⁹ <https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/advanced-matrix-extensions/overview.html>.

⁷⁰ <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

U.S. Pat. No. 10,333,768

26. A computer cluster comprising: "

Accused Xeon Products

Each of the Accused Xeon Products⁷¹ comprises a computer cluster. For example, as depicted below and as described further herein, Intel's 4th Gen Xeon Scalable Processor⁷² includes a computer cluster.

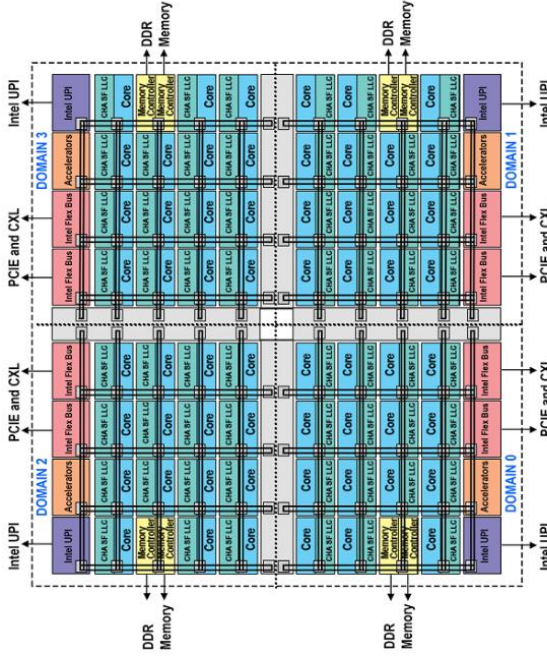


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

⁷¹ The Accused Xeon Products include, but are not limited to, all products including or related to Intel's Xeon Scalable Processors (Skylake-SP architecture) (<https://www.intel.com/content/www/us/en/developer/articles/technical/xeon-processor-scalable-family-technical-overview.html>), 2nd Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/developer/articles/news/second-generation-intel-xeon-processor-scalable-family-technical-overview.html>), 3rd Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/developer/articles/technical/intel-xeon-processor-scalable-family-overview.html>), 4th Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/products/docs/processors/xeon/5th-gen-xeon-scalable-family-overview.html>), 5th Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/products/details/processors/xeon.html>), as well as any products incorporating those items.

⁷² "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

Each Accused Xeon Product comprises a hardware processor that comprises multiple processor cores. For example, the 4th Gen Xeon Scalable Processor is configured to operate in an SNC-4 mode where each of the four domains is a node that includes a plurality of processing cores (fifteen) as shown in the above figure.

Each Accused Xeon Product comprises a user connection interface configured to receive a command to *start a cluster initialization process* for the computer cluster of each Accused Xeon Product. For example, each 4th Gen Xeon Scalable Processor comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the four domains (node) (e.g., communication among two or more nodes. This is done, for example, by using Intel's oneCCL software, which is part of Intel's oneAPI Collective Communications Library.⁷⁴ More particularly, Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...."⁷⁵ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.⁷⁶ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.⁷⁷

More particularly, a first single-node kernel is responsible for interpreting instructions from a user interface or script and distributing calls among the domains of the 4th Gen Xeon Scalable Processor for execution. Other domains in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. When the user submits a job to the cluster, for example, the first single-node kernel parses the job and distributes the tasks to the other domains (nodes) for execution. The other domains (nodes) then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user.

⁷⁴ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

⁷⁵ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/communicator.html>.

⁷⁶ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁷⁷ See <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

Hence, the 4th Gen Xeon Scalable Processor is configured to access executable program code stored in memory accessible by the 4th Gen Xeon Scalable Processor, where the executable program code is program code for a single-node kernel that, when executed, causes the processor, as it is configured, to interpret user instructions.

a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls;

Each of the Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls. For example, as depicted below, the 4th Gen Xeon Scalable Processor is configured to provide all-to-all communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.⁷⁸

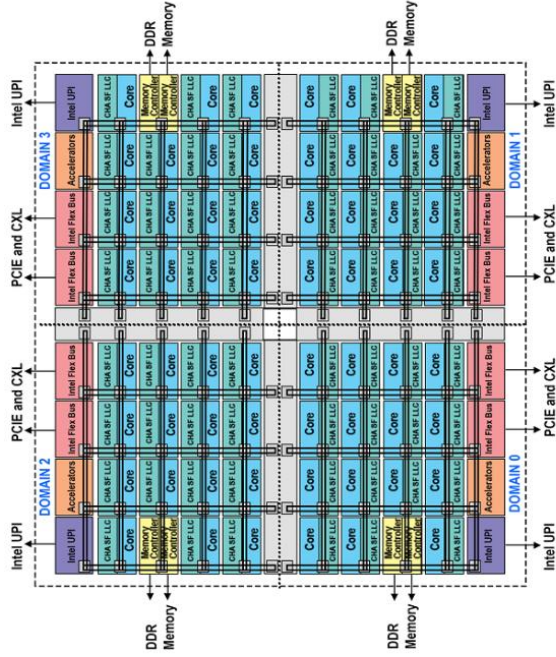


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

⁷⁸ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

The 4th Gen Xeon Scalable Processor is configured to implement Intel's oneCCL library,⁷⁹ which includes commands that provide a mechanism for the domains / nodes in a 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, to communicate results of mathematical expression evaluation with each other using an asynchronous calls, either standing alone or, for example, in conjunction with the processor core built-in mathematical functions.^{80 81 82 83}

Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors..."⁸⁴ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.⁸⁵ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.⁸⁶

"The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for

⁷⁹ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

⁸⁰ "Technical Overview Of The 4th Gen Intel Xeon Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

⁸¹ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., "Intel® AVX-512 - FP16 Instruction Set for Intel® Xeon® Processor Based Products Technology Guide," <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

⁸² See, e.g., "Developer Guide for Linux* OS," <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.

⁸³ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>; see also <https://www.intel.com/content/dam/develop/external/us/en/documents/mpi-devref-oneapi-linux-beta10.pdf> (e.g., I MPI_ADJUST_ALLTOALL, I MPI_ADJUST_ALLTOALLY, I MPI_ADJUST_ALLTOALLW, MPI_Isend, MPI_Irecv, and MPI_test).

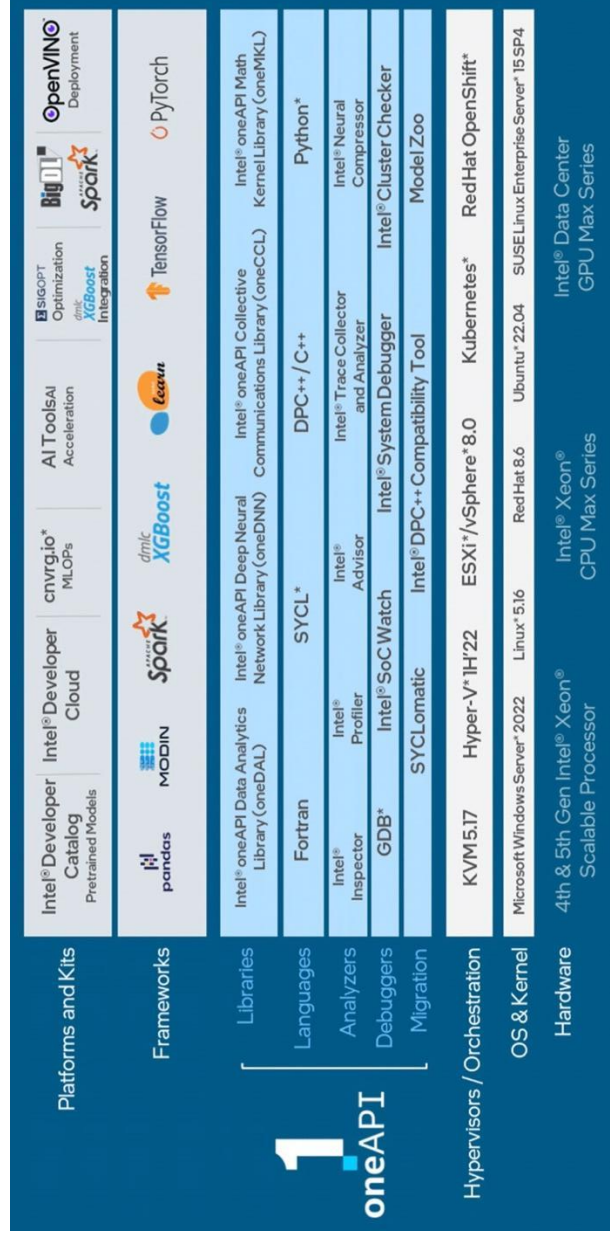
⁸⁴ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/communicator.html>.

⁸⁵ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁸⁶ See <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”⁸⁷ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.

The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors.⁸⁸



⁸⁷ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.
⁸⁸ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>The CCL library provides multiple commands for creating, managing, and using CCL communicators.⁸⁹ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.⁹⁰ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.</p> <p>Intel® MPI Library supports asynchronous progress threads that allow you to manage communication in parallel with application computation and, as a result, achieve better communication/computation overlapping.⁹¹</p> <p>Each of the Accused Xeon Products comprises a plurality of nodes wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions, wherein the plurality of nodes comprises a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution..</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.⁹²</p>
<p>wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions; wherein the plurality of nodes comprises: a first node comprising a first hardware processor configured to</p>	

⁸⁹ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

⁹⁰ <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-9/device-communication.html>.

⁹¹ Intel® MPI Library Developer Guide for Linux* OS <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.

⁹² “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

U.S. Pat. No. 10,333,768

access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and

Accused Xeon Products

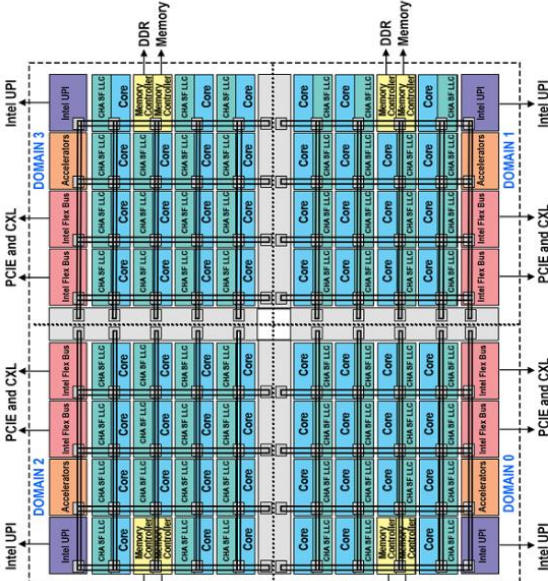


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

As shown above, each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel (see, e.g., DDR Memory, LLC, and Memory Controller).

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors....”⁹³ The oneCCL library is one of the 10 core elements of

⁹³ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/communicator.html>.

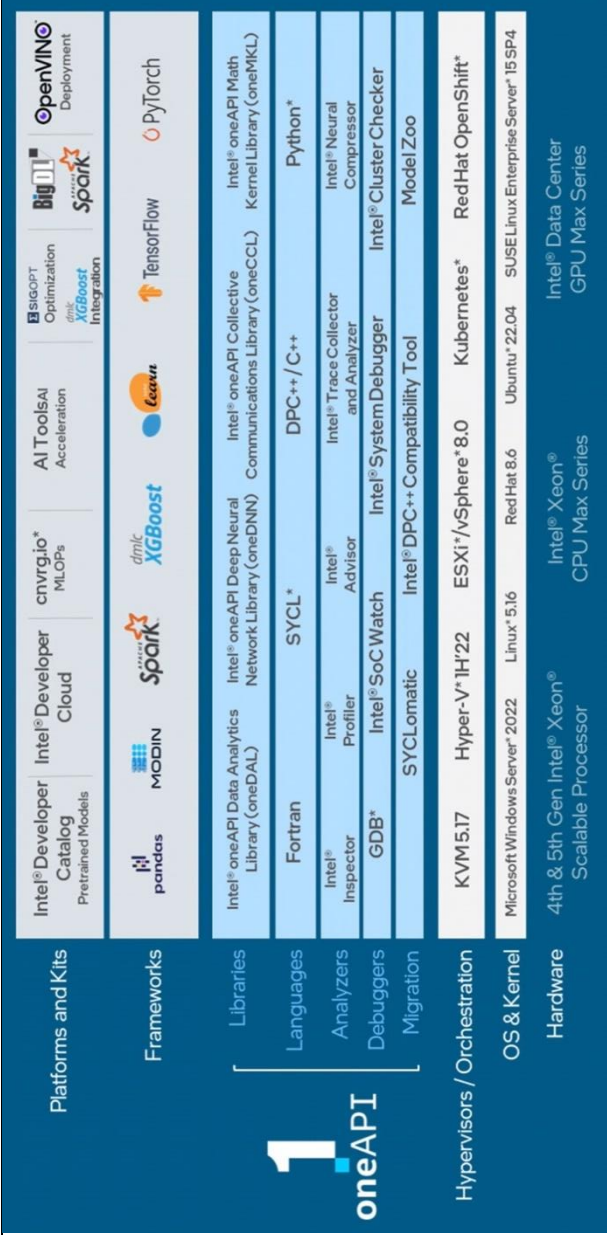
<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>Intel’s oneAPI software suite.⁹⁴ Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.⁹⁵</p> <p>“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”⁹⁶ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:⁹⁷</p>
--	--

⁹⁴ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

⁹⁵ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/onecc/developerguide-reference/2021-12/overview.html>.

⁹⁶ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

⁹⁷ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqgd>.



The CCL library provides multiple commands for creating, managing, and using CCL communicators.⁹⁸ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.⁹⁹ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

The first single-node kernel is, for example, a software program that runs on the first domain / node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other domains / nodes for execution. The other domains / nodes in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. This is known as the master-slave model, where a first domain / node executes in a supervisory role, and the

⁹⁸ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
⁹⁹ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768	Accused Xeon Products
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node; wherein</p>	<p>other domains / nodes serve as compute nodes. For example, initially, the first node is configured to interpret user instructions and distribute calls to at least one of the other nodes for execution.</p> <p>Unlike a master-slave model, the nodes communicate results of mathematical expression evaluation with each other during execution of the tasks that are distributed to them by the first node using a peer-to-peer architecture without being required to go through the first node.</p> <p>When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel. The first single-node kernel then collects the results from the other nodes and returns them to the user.</p>
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.¹⁰⁰</p>	<p>Each of the Accused Xeon Products comprises a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.¹⁰⁰</p>

¹⁰⁰ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

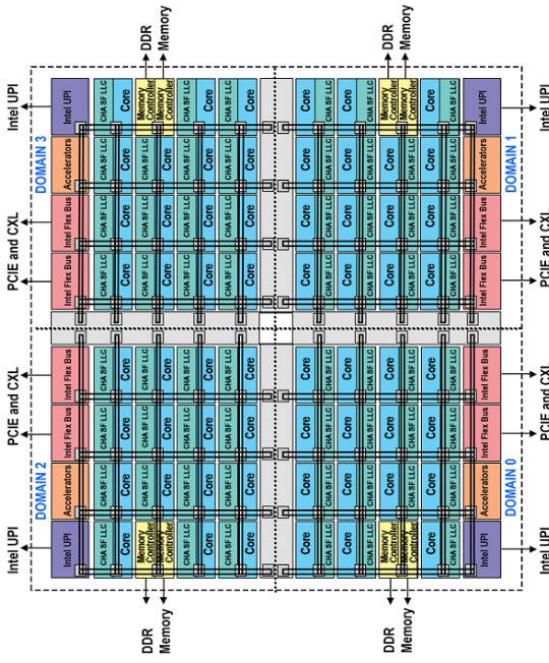


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...”¹⁰¹ The oneCCL library is one of the 10 core elements of Intel’s oneAPI software suite.¹⁰² Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹⁰³

“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL

¹⁰¹ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87Ciqd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

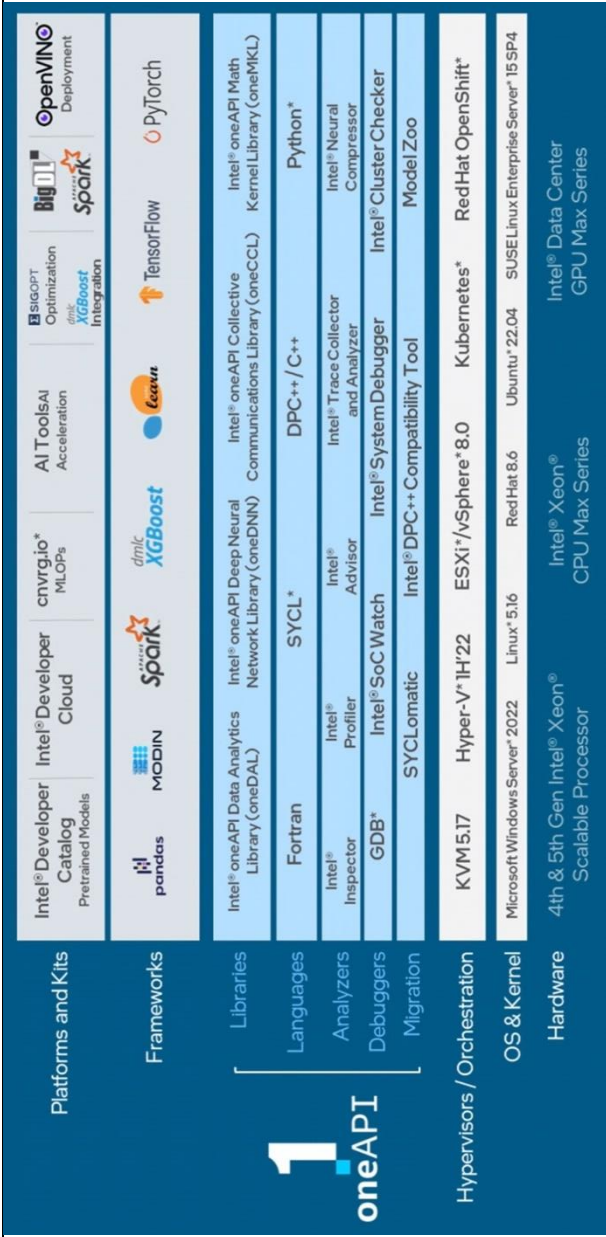
¹⁰² <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

¹⁰³ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”¹⁰⁴ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:¹⁰⁵</p>
--	--

¹⁰⁴ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

¹⁰⁵ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.



The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁰⁶ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁰⁷ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Each of the Accused Xeon Products comprises a hardware processor with multiple processing cores, and is configured to receive the result of mathematical expression evaluation from the second node and to execute at least a second mathematical expression evaluation using the received result.

the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured

¹⁰⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
¹⁰⁷ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768

to receive the result of mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;

Accused Xeon Products

For example, when executed, the code stored in the memory accessible by the 4th Gen Xeon Scalable Processor causes the processor to receive a result of a first mathematical expression evaluation from a second node (domain) in the 4th Gen Xeon Scalable Processor and to execute at least a second mathematical expression evaluation using the received result (e.g., by performing matrix multiplications).

For example, as depicted below, the Intel 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.

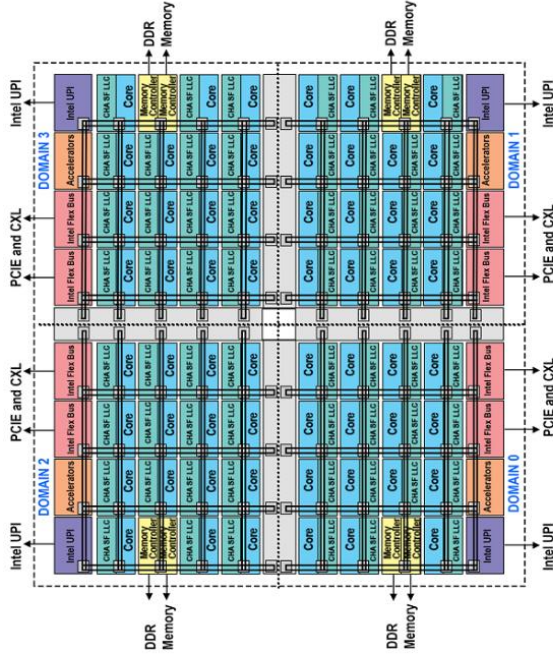


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of

U.S. Pat. No. 10,333,768	<p style="text-align: center;">Accused Xeon Products</p> <p>4th and 5th gen Intel® Xeon® processors...⁹¹⁰⁸ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.¹⁰⁹ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹¹⁰</p> <p>“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”¹¹¹ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel's oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel's 4th Gen Xeon Scalable Processors:¹¹²</p>
--------------------------	---

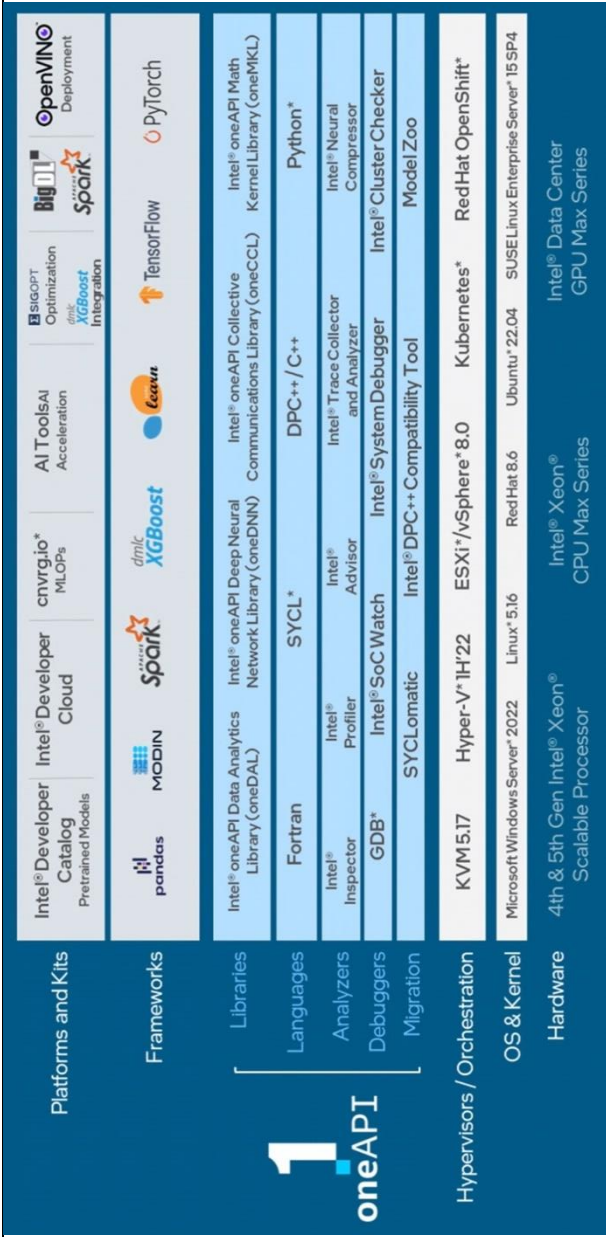
¹⁰⁸ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,”

<https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/communicator.html>.

¹⁰⁹ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
¹¹⁰ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

¹¹¹ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-oneMKL.html>.

¹¹² “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqtd>.



The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹¹³ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹¹⁴ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Also, for example, the architecture of the 4th Gen Xeon Scalable Processor is configured to receive the result of a first mathematical expression evaluation from the second node (domain) and to execute at least a second mathematical expression evaluation using the received result. The source code for Intel's oneCCL software, for example, includes the details for using CCL commands to receive the

¹¹³ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
¹¹⁴ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>result of a first mathematical expression evaluation from a second node and to execute at least a second mathematical expression evaluation using the received result.</p>
<p>wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface or the script;</p>	<p>Each of the Accused Xeon Products includes a first node configured to return the result of a second mathematical expression evaluation to the user interface or the script.</p> <p>Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors..."¹¹⁵ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.¹¹⁶ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹¹⁷</p> <p>The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹¹⁸ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹¹⁹ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.</p>
<p>wherein one or more of the nodes are configured to: accept user instructions; after accepting user instructions, communicate</p>	<p>Each of the Accused Xeon Products includes one or more nodes configured to accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.</p>

¹¹⁵ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

¹¹⁶ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

¹¹⁷ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

¹¹⁸ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onecl.html#gs.ejo7gf> ("Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.")

¹¹⁹ <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
<p>U.S. Pat. No. 10,333,768</p> <p>at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p>	<p>For example, the architecture of the 4th Gen Xeon Scalable Processor is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the domains to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p> <p>Also for example, the source code for Intel's oneCCL software includes the details for using CCL commands to cause 4th Gen Xeon Scalable Processor to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the 4th Gen Xeon Scalable Processor to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p>
<p>27. The computer cluster of claim 26, wherein the asynchronous calls comprise a first command to create a first packet containing: an expression to be sent as payload; and a target node where the expression should be sent; wherein the first command is configured to be called from within a single-node kernel; and wherein the local cluster node module is configured to forward the expression to the target node.</p>	<p>Each of the Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls wherein the asynchronous calls comprise a first command to create a first packet containing: an expression to be sent as payload; and a target node where the expression should be sent; wherein the first command is configured to be called from within a single-node kernel; wherein the single-node kernel is configured to send the first packet to a local cluster node module; and wherein the local cluster node module is configured to forward the expression to the target node.</p> <p>Each of Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls. For example, as depicted below, the 4th Gen Xeon Scalable Processor is configured to provide communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.¹²⁰</p>

¹²⁰ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>mathematical expression evaluation with each other using a asynchronous calls, either standing alone or, for example, in conjunction with the processor core built-in mathematical functions.^{122 123 124 125}</p> <p>Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...”¹²⁶ The oneCCL library is one of the 10 core elements of Intel’s oneAPI software suite.¹²⁷ Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹²⁸</p> <p>“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other</p>
--	---

¹²² “Technical Overview Of The 4th Gen Intel Xeon Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

¹²³ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., “Intel® AVX-512 - FPU Instruction Set for Intel® Xeon® Processor Based Products Technology Guide,” <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

¹²⁴ See, e.g., “Developer Guide for Linux* OS,” <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.

¹²⁵ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onecccl.html#gs.ejo7gf>; see also <https://www.intel.com/content/dam/develop/external/us/en/documents/mpi-devref-oneapi-linux-beta10.pdf> (e.g., MPI_Isend, MPI_Irecv, MPI_Test, LMPI_ADJUST_ALLTOALL, I_MPI_ADJUST_ALLTOALLV, I_MPI_ADJUST_ALLTOALLW).

¹²⁶ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87Ciqd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/onecccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

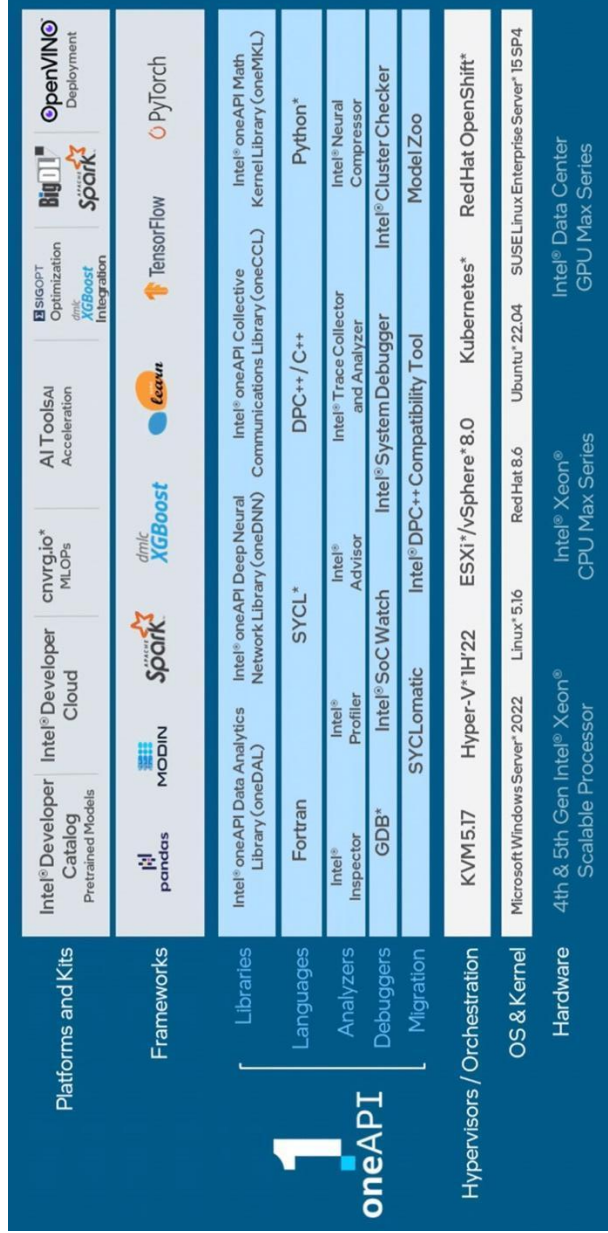
¹²⁷ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

¹²⁸ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/onecccl/developer-guide-reference/2021-12/overview.html>.

¹²⁹ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/onecccl/developer-guide-reference/2021-12/overview.html>.

functionality.¹²⁹ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.

The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors.¹³⁰



The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹³¹ The CCL library includes software objects that are used to manage the

¹²⁹ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

¹³⁰ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

¹³¹ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onecl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹³² The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.</p> <p>Intel® MPI Library supports asynchronous progress threads that allow for managing communication in parallel with application computation and, as a result, achieve better communication/computation overlapping.¹³³</p> <p>A single-node kernel is, for example, a software program that runs on the first domain / node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other domains / nodes for execution by cluster node modules on those nodes. For example, the asynchronous calls MPI_Isend and MPI_Irecv as well as the MPI_Test are used for asynchronous communications.¹³⁴ For example, a first single-node kernel that generates an MPI_Isend call, therefore, creates a first packet containing an expression to be sent as payload (see, for example, the call MPI_Isend) and a target node (see, for example the dest field of MPI_Isend) where the expression should be sent where the first command is configured to be called from within a single-node kernel. Because of the asynchronous nature of MPI_Isend and MPI_Irecv calls, the single-node kernel is configured to send the first packet to a local cluster node module and the local cluster node module is configured to forward the expression to the target node (see, for example the dest field of the call MPI_Isend).</p>
--	---

¹³² <https://www.intel.com/content/www/us/en/docs/onecl/developer-guide-reference/2021-9/device-communication.html>.

¹³³ Intel® MPI Library Developer Guide for Linux* OS <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.

¹³⁴ See “Intel MPI Library for Linux* OS,” <https://www.intel.com/content/dam/develop/external/us/en/documents/mpi-devref-oneapi-linux-beta10.pdf>.

U.S. Pat. No. 10,333,768

Accused Xeon Products

29. A computer cluster comprising:

Each of the Accused Xeon Products¹³⁵ comprises a computer cluster. For example, as depicted below and as described further herein, Intel's 4th Gen Xeon Scalable Processor¹³⁶ includes a computer cluster.

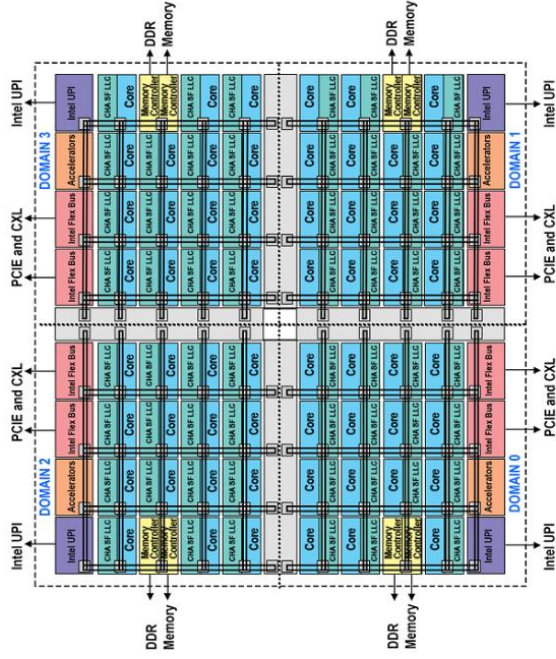


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

a plurality of nodes, wherein one or more of the

Each of the Accused Xeon Products comprises a plurality of nodes. For example, as depicted below, the 4th Gen Xeon Scalable Processor comprises four nodes (described as domains in the figure

¹³⁵ The Accused Xeon Products include, but are not limited to, all products including or related to Intel's Xeon Scalable Processors (Skylake-SP architecture) (<https://www.intel.com/content/www/us/en/developer/articles/technical/xeon-processor-scalable-family-technical-overview.html>), 2nd Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/developer/articles/news/second-generation-intel-xeon-processor-scalable-family-technical-overview.html>), 3rd Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-processor-scalable-family-overview.html>), 4th Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/products/docs/processors/xeon/5th-gen-xeon-scalable-family-overview.html>), 5th Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/products/details/processors/xeon.html>), as well as any products incorporating those items.

¹³⁶ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

U.S. Pat. No. 10,333,768

nodes are configured to receive: a command to start a cluster initialization process for the computer cluster, wherein the cluster initialization process comprises establishing communication among two or more of the nodes; and an instruction from a user interface or a script; and

Accused Xeon Products

below), which are configured to be connected in an all-to-all local bus interconnection configuration.¹³⁷

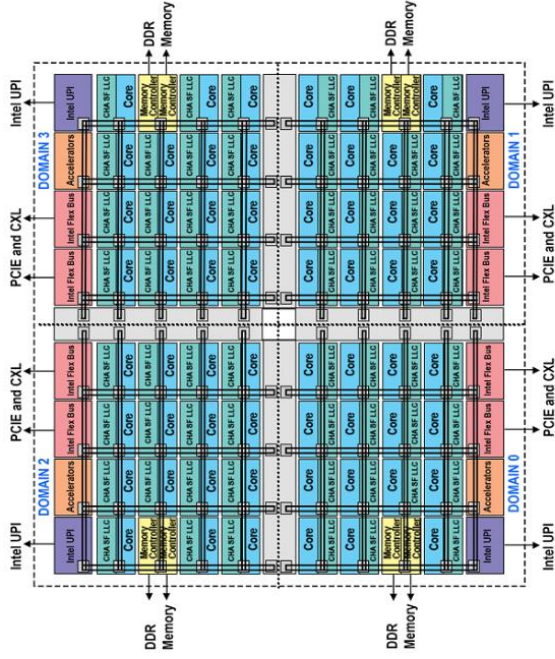


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Each of the 4th Gen Xeon Scalable Processors is configured to operate in a Sub-NUMA clustering 4 (SNC-4) affinity mode, comprising four domains of sub-NUMA clusters / nodes.

Each Accused Xeon Product comprises a plurality of nodes, wherein one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster, wherein the cluster initialization process comprises establishing communication among two or more of the nodes; and an instruction from a user interface or a script.

¹³⁷ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

Each Accused Xeon Product comprises a hardware processor that comprises multiple processor cores. For example, the 4th Gen Xeon Scalable Processor is configured to operate in an SNC-4 mode where each of the four domains is a node that includes a plurality of processing cores (fifteen) as shown in the above figure.

Each Accused Xeon Product comprises a user connection interface configured to receive a command to *start a cluster initialization process* for the computer cluster of each Accused Xeon Product. For example, each 4th Gen Xeon Scalable Processor comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the four domains (node) (e.g., communication among two or more nodes. This is done, for example, by using Intel's oneCCL software, which is part of Intel's oneAPI Collective Communications Library.¹³⁸ More particularly, Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...."¹³⁹ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.¹⁴⁰ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹⁴¹

More particularly, a first single-node kernel is responsible for interpreting instructions from a user interface or script and distributing calls among the domains of the 4th Gen Xeon Scalable Processor for execution. Other domains in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. When the user submits a job to the cluster, for example, the first single-node kernel parses the job and distributes the tasks to the other domains (nodes) for execution. The other domains (nodes) then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user.

¹³⁸ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

¹³⁹ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneapi/technical/oneapi-what-is-it.html>.

¹⁴⁰ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
¹⁴¹ See <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

Hence, the 4th Gen Xeon Scalable Processor is configured to access executable program code stored in memory accessible by the 4th Gen Xeon Scalable Processor, where the executable program code is program code for a single-node kernel that, when executed, causes the processor, as it is configured, to interpret user instructions.

a mechanism for the nodes to communicate results of mathematical expression evaluation with each other;

Each of the Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of mathematical expression evaluation with each other. For example, as depicted below, the 4th Gen Xeon Scalable Processor is configured to provide all-to-all communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.¹⁴²

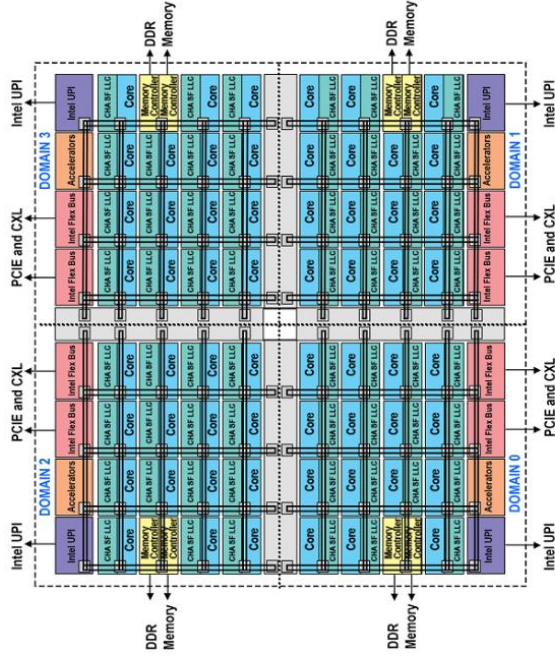


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

¹⁴² “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>The 4th Gen Xeon Scalable Processor is configured to implement Intel's oneCCL library,¹⁴³ which includes commands that provide a mechanism for the domains / nodes in a 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, to communicate results of mathematical expression evaluation with each other, either standing alone or, for example, in conjunction with the processor core built-in mathematical functions.^{144 145 146}</p> <p>Each of the Accused Xeon Products comprises a plurality of nodes wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions, wherein the plurality of nodes comprises a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution..</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.¹⁴⁷</p>
<p>wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions; wherein the plurality of nodes comprises: a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and</p>	

¹⁴³ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

¹⁴⁴ "Technical Overview Of The 4th Gen Intel Xeon Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

¹⁴⁵ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., "Intel® AVX-512 - FPU Instruction Set for Intel® Xeon® Processor Based Products Technology Guide," <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fpu-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

¹⁴⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

¹⁴⁷ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

U.S. Pat. No. 10,333,768
 program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and

Accused Xeon Products

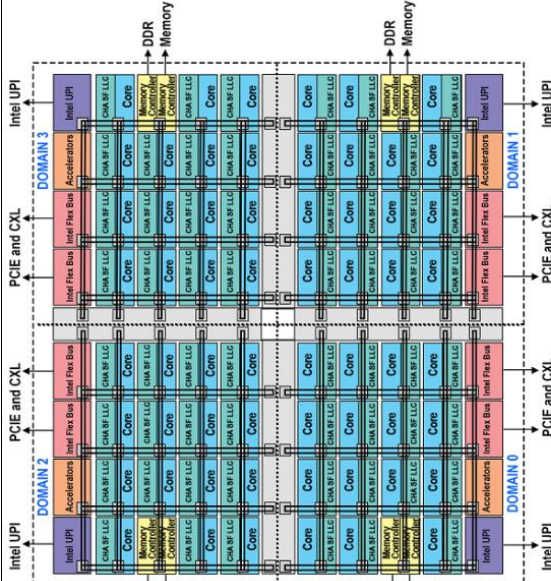


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

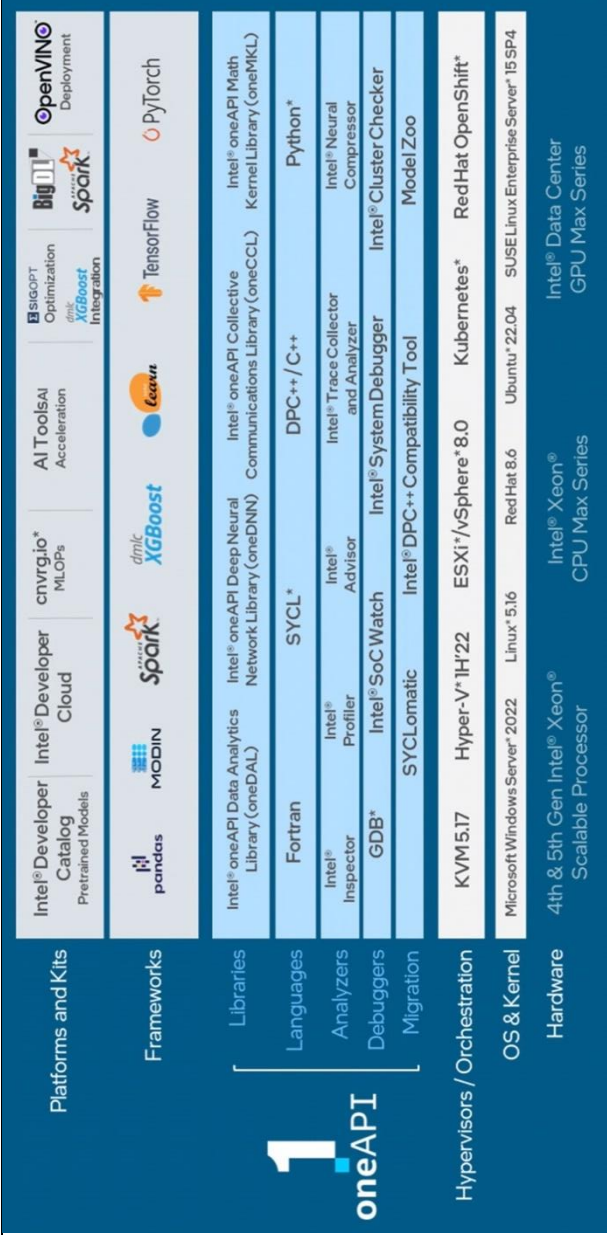
As shown above, each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel (see, e.g., DDR Memory, LLC, and Memory Controller).

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors....”¹⁴⁸ The oneCCL library is one of the 10 core elements of

¹⁴⁸ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/communicator.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>Intel’s oneAPI software suite.¹⁴⁹ Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹⁵⁰</p> <p>“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”¹⁵¹ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:¹⁵²</p>
--	--

¹⁴⁹ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
¹⁵⁰ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.
¹⁵¹ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.
¹⁵² “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.



The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁵³ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁵⁴ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

The first single-node kernel is, for example, a software program that runs on the first domain / node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other domains / nodes for execution. The other domains / nodes in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. This is known as the master-slave model, where a first domain / node executes in a supervisory role, and the

¹⁵³ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
¹⁵⁴ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768	Accused Xeon Products
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node; wherein</p>	<p>other domains / nodes serve as compute nodes. For example, initially, the first node is configured to interpret user instructions and distribute calls to at least one of the other nodes for execution.</p> <p>Unlike a master-slave model, the nodes communicate results of mathematical expression evaluation with each other during execution of the tasks that are distributed to them by the first node using a peer-to-peer architecture without being required to go through the first node.</p> <p>When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel. The first single-node kernel then collects the results from the other nodes and returns them to the user.</p>
<p>a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.¹⁵⁵</p>	<p>Each of the Accused Xeon Products comprises a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.¹⁵⁵</p>

¹⁵⁵ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”¹⁵⁹ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:¹⁶⁰</p>
--	--

¹⁵⁹ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

¹⁶⁰ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁶¹ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁶² The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Each of the Accused Xeon Products comprises a hardware processor with multiple processing cores, and is configured to receive the result of mathematical expression evaluation from the second node and to execute at least a second mathematical expression evaluation using the received result.

the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured

¹⁶¹ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
¹⁶² <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768

to receive the result of mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node; and

Accused Xeon Products

For example, when executed, the code stored in the memory accessible by the 4th Gen Xeon Scalable Processor causes the processor to receive a result of a first mathematical expression evaluation from a second node (domain) in the 4th Gen Xeon Scalable Processor and to execute at least a second mathematical expression evaluation using the received result (e.g., by performing matrix multiplications).

For example, as depicted below, the Intel 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.

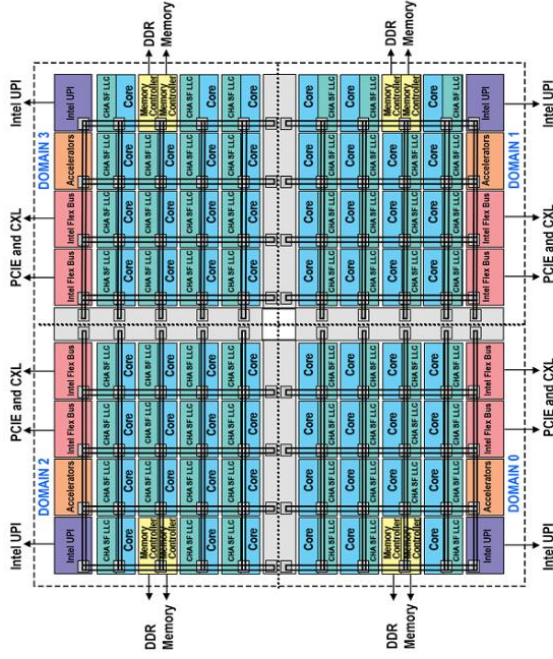


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>4th and 5th gen Intel® Xeon® processors...¹⁶³ The oneCCL library is one of the 10 core elements of Intel’s oneAPI software suite.¹⁶⁴ Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹⁶⁵</p> <p>“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”¹⁶⁶ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:¹⁶⁷</p>
--	--

¹⁶³ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,”

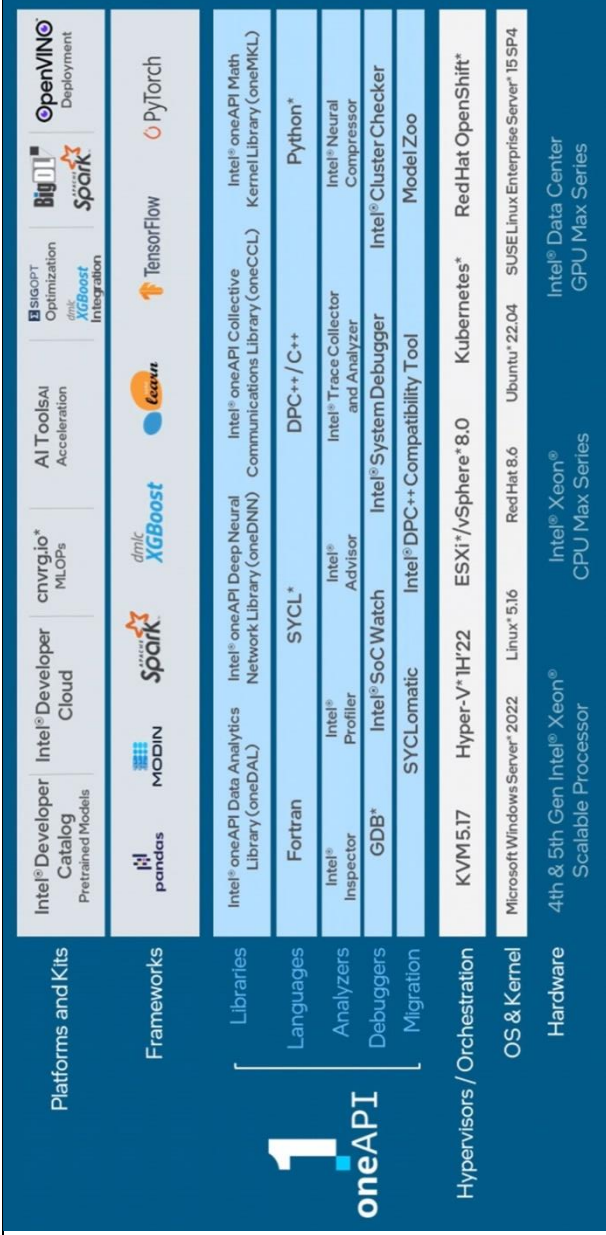
<https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/communicator.html>.

¹⁶⁴ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
¹⁶⁵ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

¹⁶⁶ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-oneMKL.html>.

¹⁶⁷ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

Accused Xeon Products



The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁶⁸ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁶⁹ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Also, for example, the architecture of the 4th Gen Xeon Scalable Processor is configured to receive the result of a first mathematical expression evaluation from the second node (domain) and to execute at least a second mathematical expression evaluation using the received result. The source code for Intel's oneCCL software, for example, includes the details for using CCL commands to receive the

¹⁶⁸ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
¹⁶⁹ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>result of a first mathematical expression evaluation from a second node and to execute at least a second mathematical expression evaluation using the received result.</p>
<p>wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface or the script;</p>	<p>Each of the Accused Xeon Products includes a first node configured to return the result of a second mathematical expression evaluation to the user interface or the script.</p> <p>Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors..."¹⁷⁰ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.¹⁷¹ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹⁷²</p> <p>The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁷³ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁷⁴ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.</p>
<p>wherein one or more of the nodes are configured to: accept user instructions; after accepting user instructions, communicate</p>	<p>Each of the Accused Xeon Products includes one or more nodes configured to accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.</p>

¹⁷⁰ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

¹⁷¹ <https://www.intel.com/content/www/us/en/developer/communications/library-developer-guide-and-reference/>.

¹⁷² See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

¹⁷³ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneapi.html#gs.ejo7gf> ("Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.")

¹⁷⁴ <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
<p>U.S. Pat. No. 10,333,768</p> <p>at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p>	<p>For example, the architecture of the 4th Gen Xeon Scalable Processor is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the domains to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p> <p>Also for example, the source code for Intel's oneCCL software includes the details for using CCL commands to cause 4th Gen Xeon Scalable Processor to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the 4th Gen Xeon Scalable Processor to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p>
<p>30. The computer cluster of claim 4, wherein each of the plurality of nodes implements asynchronous calls that enable the single-node kernel to perform computation tasks while the cluster node modules are simultaneously communicating with one another.</p>	<p>Each of the Accused Xeon Products comprises a plurality of nodes, wherein each of the plurality of nodes implements asynchronous calls that enable the single-node kernel to perform computation tasks while the cluster node modules are simultaneously communicating with one another.</p> <p>Each of Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls. For example, as depicted below, the 4th Gen Xeon Scalable Processor is configured to provide communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.¹⁷⁵</p>

¹⁷⁵ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

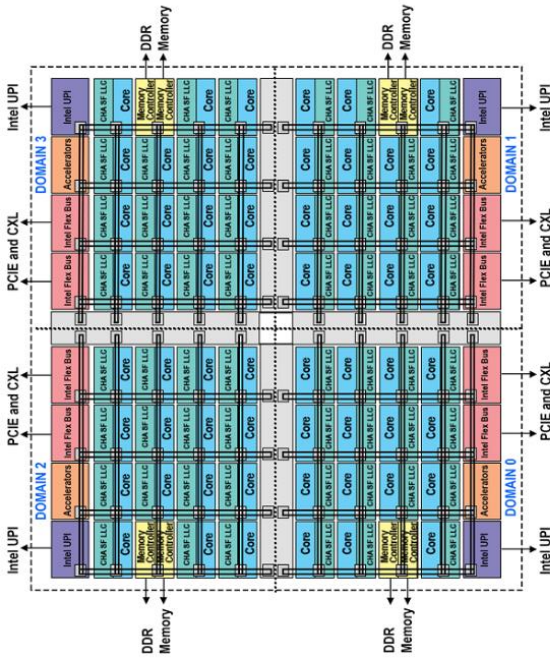


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

The 4th Gen Xeon Scalable Processor is configured to implement Intel's oneCCL library,¹⁷⁶ which includes commands that provide a mechanism for the domains / nodes in a 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering_4 (SNC-4) affinity mode, to communicate results of

¹⁷⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>mathematical expression evaluation with each other using asynchronous calls, either standing alone or, for example, in conjunction with the processor core built-in mathematical functions.¹⁷⁷ 178 179 180</p> <p>Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors..."¹⁸¹ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.¹⁸² Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.¹⁸³</p> <p>"The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other</p>
--	---

¹⁷⁷ "Technical Overview Of The 4th Gen Intel Xeon Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

¹⁷⁸ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., "Intel® AVX-512 - FPU Instruction Set for Intel® Xeon® Processor Based Products Technology Guide," <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fpu-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

¹⁷⁹ See, e.g., "Developer Guide for Linux* OS," <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.

¹⁸⁰ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>; see also <https://www.intel.com/content/dam/develop/external/us/en/documents/mpi-devref-oneapi-linux-beta10.pdf> (e.g., I_MPI_ADJUST_ALLTOALL, I_MPI_ADJUST_ALLTOALLY, I_MPI_ADJUST_ALLTOALLW, MPI_Isend, MPI_Irecv, and MPI_test).

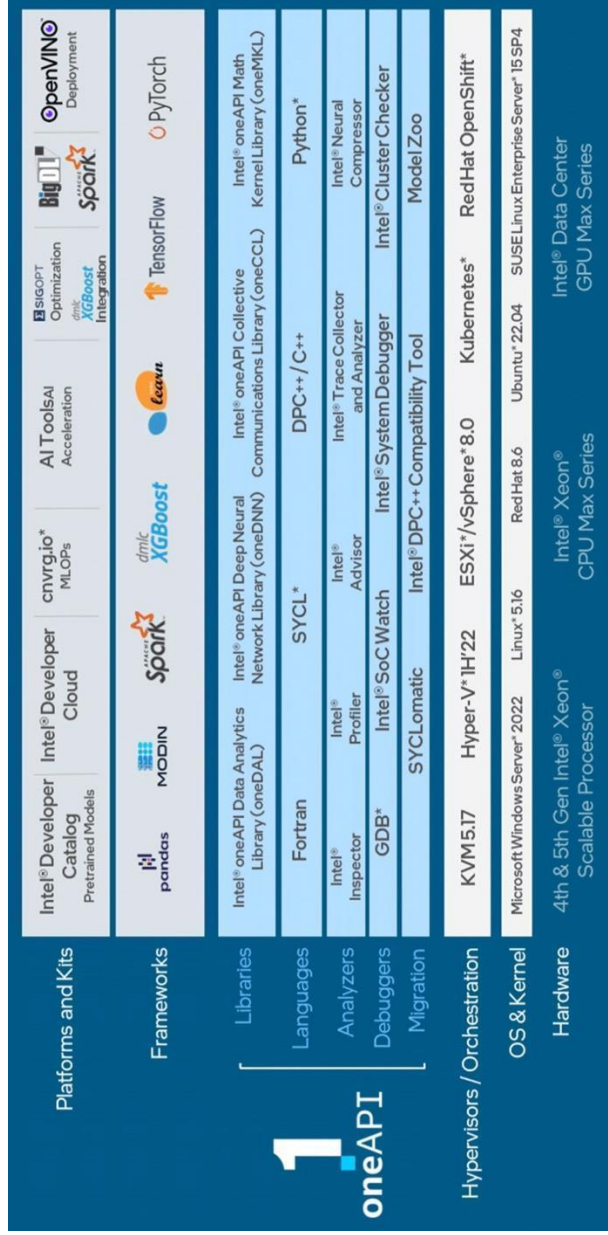
¹⁸¹ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87Ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

¹⁸² <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

¹⁸³ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

functionality.¹⁸⁴ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.

The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:¹⁸⁵



The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁸⁶ The CCL library includes software objects that are used to manage the

¹⁸⁴ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

¹⁸⁵ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

¹⁸⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onecl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

U.S. Pat. No. 10,333,768	Accused Xeon Products
	<p>communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁸⁷ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.</p> <p>Intel® MPI Library supports asynchronous progress threads that allow for managing communication in parallel with application computation and, as a result, achieve better communication/computation overlapping.¹⁸⁸</p> <p>A single-node kernel is, for example, a software program that runs on the first domain / node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other domains / nodes for execution by cluster node modules on those nodes. The first single-node kernel thus is configured to send the first packet to a local cluster node module. The cluster domain modules on those other domains / nodes in the cluster are responsible for executing the tasks that are distributed to them by the first single-node kernel. The local cluster node module is configured to forward the expression to the target node.</p> <p>When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel. The first single-node kernel then collects the results from the other nodes and returns them to the user.</p> <p>The cluster node modules thus enable the single-node kernel to perform computation tasks while the cluster node modules are simultaneously communicating with one another.</p>
31. The computer cluster of claim 4, wherein the plurality of single-node kernels during thread	<p>Each of the Accused Xeon Products comprises a plurality of single-node kernels and a plurality of nodes, each of which comprises one or more cluster node modules, wherein intercommunication among the plurality of single-node kernels during thread execution is enabled by the plurality of cluster node modules, and wherein the computer cluster is configured to permit exchange of information between nodes during the course of a parallel computation.</p>

¹⁸⁷ <https://www.intel.com/content/www/us/en/docs/onecl/developer-guide-reference/2021-9/device-communication.html>.

¹⁸⁸ Intel® MPI Library Developer Guide for Linux* OS <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.

U.S. Pat. No. 10,333,768

execution is enabled by the plurality of cluster node modules, and wherein the computer cluster is configured to permit exchange of information between nodes during the course of parallel computation.

Accused Xeon Products

For example, as depicted below and as described further herein, Intel's 4th Gen Xeon Scalable Processor¹⁸⁹ includes a computer cluster comprising a computer cluster node (e.g., domain) for evaluating expressions in parallel with other computer cluster nodes.

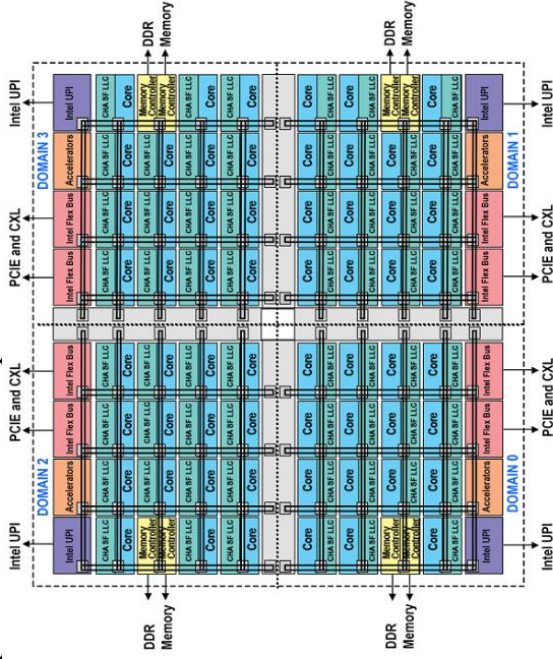


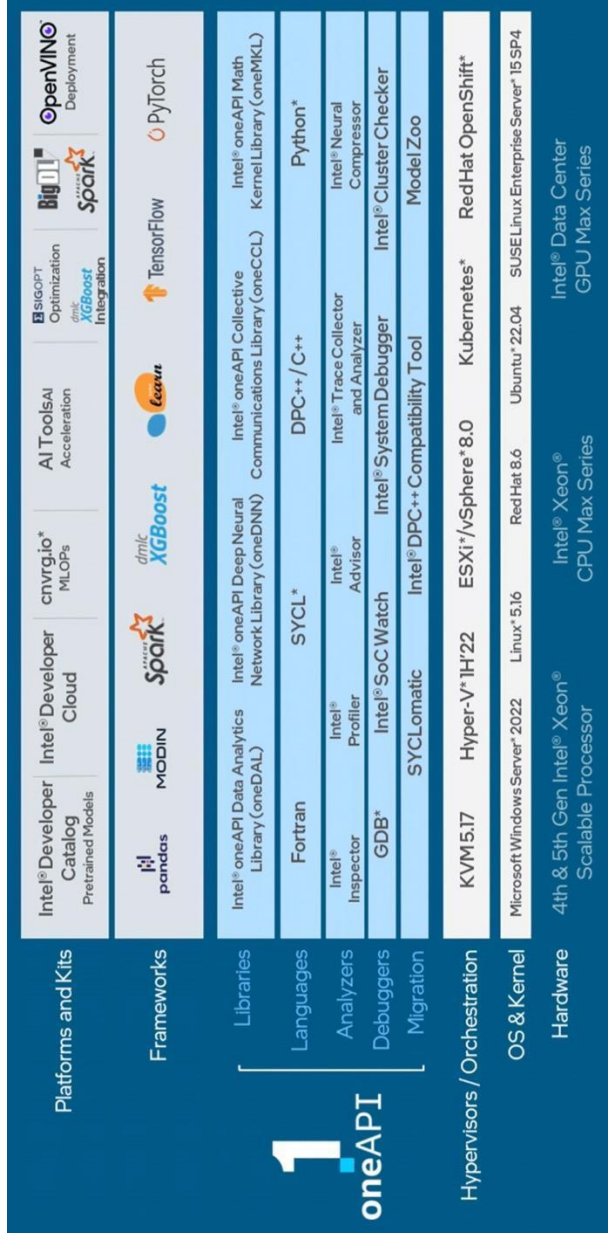
Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Each of Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of computations, including results of thread execution among various single-node kernels, with each other. For example, as depicted above, the 4th Gen Xeon Scalable Processor is configured to provide all-to-all communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.¹⁹⁰

¹⁸⁹ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

¹⁹⁰ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

The figure below depicts Intel's oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the exemplary context of the software suite for Intel's 4th Gen Xeon Scalable Processors:¹⁹¹



The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁹² The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁹³ The CCL library is responsible for receiving first commands from a user interface without the first commands first passing through the first kernel, and after receiving the first commands from the user interface, send second commands to the first kernel.

¹⁹¹ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87c1qd>.

¹⁹² Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

¹⁹³ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

The source code for the CCL library includes the details for using oneCCL commands to permit exchange of information among nodes during the course of parallel computation.

33. The computer cluster of claim 1, wherein the plurality of nodes are configured to permit exchange of information between nodes during the course of parallel computation.

Each of the Accused Xeon Products comprises a plurality of nodes, wherein the computer cluster is configured to permit exchange of information between nodes during the course of parallel computation.

For example, as depicted below and as described further herein, Intel's 4th Gen Xeon Scalable Processor¹⁹⁴ includes a computer cluster comprising a computer cluster node (e.g., domain) for evaluating expressions in parallel with other computer cluster nodes.

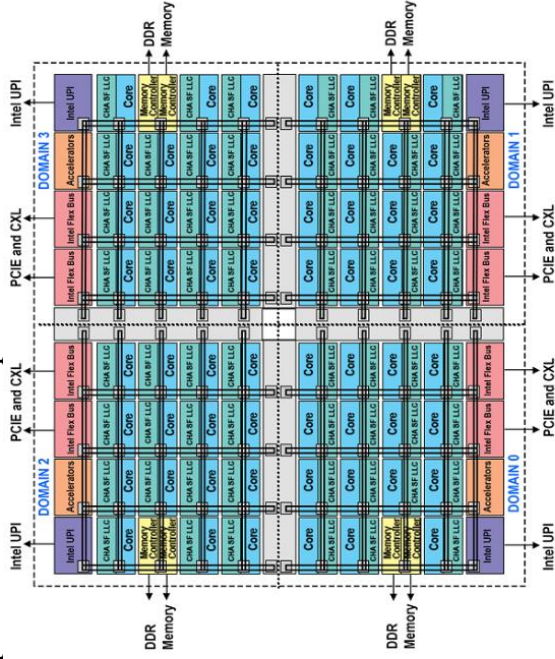
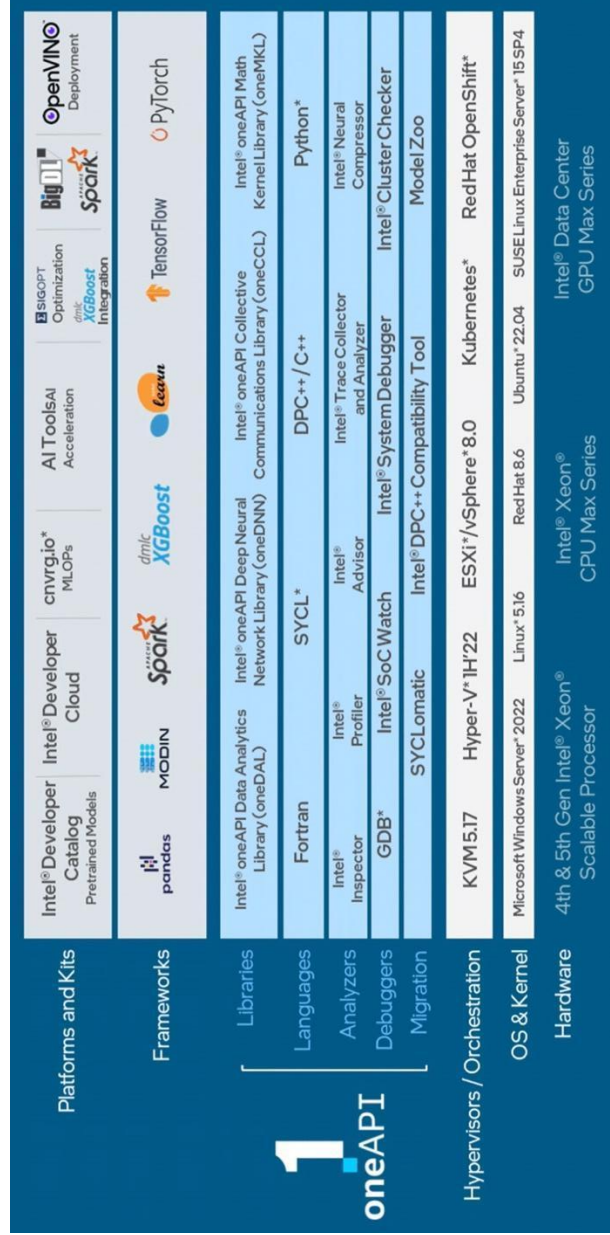


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

¹⁹⁴ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

Each of Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of computations with each other. For example, as depicted above, the 4th Gen Xeon Scalable Processor is configured to provide all-to-all communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.¹⁹⁵

The figure below depicts Intel's oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the exemplary context of the software suite for Intel's 4th Gen Xeon Scalable Processors.¹⁹⁶



¹⁹⁵ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

¹⁹⁶ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

U.S. Pat. No. 10,333,768	Accused Xeon Products
	<p>The CCL library provides multiple commands for creating, managing, and using CCL communicators.¹⁹⁷ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.¹⁹⁸ The CCL library is responsible for receiving first commands from a user interface without the first commands first passing through the first kernel, and after receiving the first commands from the user interface, send second commands to the first kernel.</p> <p>The source code for the CCL library includes the details for using oneCCL commands to permit exchange of information among nodes during the course of parallel computation.</p>
<p>34. The computer cluster of claim 1, wherein each of the plurality of nodes comprises instructions executable by the hardware processor and configured to implement asynchronous behavior, wherein the instructions comprise: a first instruction to asynchronously send a payload to another node; a second instruction to asynchronously receive a payload from another node; and a third instruction to search for a payload matching a message specifier.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor comprises four nodes (described as domains in the figure below), which are configured to be connected in an all-to-all local bus interconnection configuration.¹⁹⁹</p>	<p>Each of the Accused Xeon Products comprises a plurality of nodes, wherein each of the plurality of nodes comprises instructions executable by the hardware processor and configured to implement asynchronous behavior, wherein the instructions comprise: a first instruction to asynchronously send a payload to another node; a second instruction to asynchronously receive a payload from another node; and a third instruction to search for a payload matching a message specifier.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor comprises four nodes (described as domains in the figure below), which are configured to be connected in an all-to-all local bus interconnection configuration.¹⁹⁹</p>

¹⁹⁷ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

¹⁹⁸ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

¹⁹⁹ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

U.S. Pat. No. 10,333,768
 payload matching a
 message specifier.

Accused Xeon Products

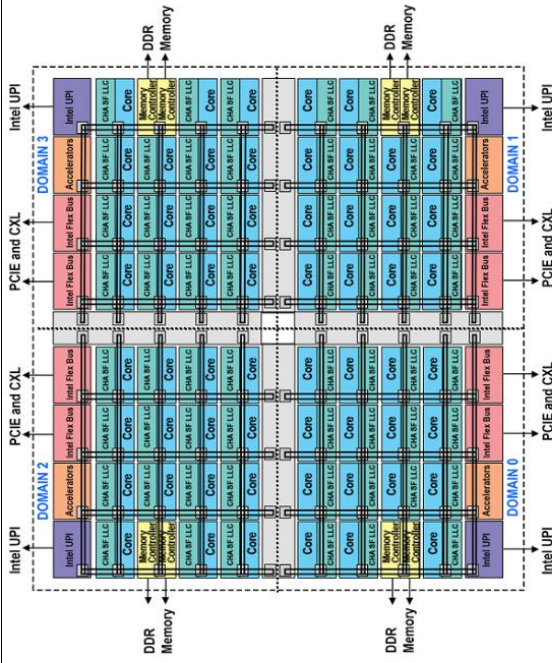


Figure 3 - Block Diagram Representing Domains Of sub-NUMA With Four Clusters

The 4th Gen Xeon Scalable Processor is configured to implement Intel's oneCCL library,²⁰⁰ which includes commands that provide a mechanism for the domains / nodes in a 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, to communicate results of mathematical expression evaluation with each other using an asynchronous calls, either standing alone or, for example, in conjunction with the processor core built-in mathematical functions.^{201 202 203 204}

²⁰⁰ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.
²⁰¹ "Technical Overview Of The 4th Gen Intel Xeon Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.
²⁰² Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., "Intel® AVX-512 - FPU Instruction Set for Intel® Xeon® Processor Based Products Technology Guide," <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.
²⁰³ See, e.g., "Developer Guide for Linux* OS," <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.
²⁰⁴ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>; see also <https://www.intel.com/content/dam/develop/external/us/en/documents/mpi-devref-oneapi-linux-beta10.pdf> (e.g., MPI_Isend, MPI_Irecv, and MPI_Test).

Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...."²⁰⁵ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.²⁰⁶ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.²⁰⁷

"The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality."²⁰⁸ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.

The figure below depicts Intel's oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel's 4th Gen Xeon Scalable Processors:²⁰⁹

²⁰⁵ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

²⁰⁶ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.
²⁰⁷ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

²⁰⁸ "Intel® oneAPI Programming Guide," <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

²⁰⁹ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

The diagram illustrates the Intel oneAPI ecosystem. At the top, it lists various development and optimization tools: Intel® Developer Catalog Pretrained Models, Intel® Developer Cloud, cnvrg.io* MLOPs, AI ToolsAI Acceleration, SIGOPT Optimization, Intel® XGB Boost Integration, BigDL, Spark, and OpenVINO Deployment. Below this, it shows frameworks like pandas, MODIN, Spark, XGB Boost, Leven, TensorFlow, and PyTorch. The core oneAPI ecosystem is divided into several categories:

- Platforms and Kits:** Intel® Developer Catalog Pretrained Models, Intel® Developer Cloud, cnvrg.io* MLOPs, AI ToolsAI Acceleration, SIGOPT Optimization, Intel® XGB Boost Integration, BigDL, Spark, and OpenVINO Deployment.
- Frameworks:** pandas, MODIN, Spark, XGB Boost, Leven, TensorFlow, and PyTorch.
- Libraries:** Intel® oneAPI Data Analytics Library (oneDAL), Intel® oneAPI Deep Neural Network Library (oneDNN), Intel® oneAPI Collective Communications Library (oneCCL), Intel® oneAPI Math Kernel Library (oneMKL).
- Languages:** Fortran, SYCL*, DPC++/C++.
- Analyzers:** Intel® Profiler, Intel® Trace Collector and Analyzer, Intel® Neural Compressor.
- Debuggers:** GDB*, Intel® SoC Watch, Intel® System Debugger, Intel® Cluster Checker.
- Migration:** SYCLomatic, Intel® DPC++ Compatibility Tool, Model Zoo.
- Hypervisors / Orchestration:** KVM 5.17, Hyper-V* 1H'22, ESXi*/vSphere* 8.0, Kubernetes*, Red Hat OpenShift*.
- OS & Kernel:** Microsoft Windows Server* 2022, Linux* 5.16, Red Hat 8.6, Ubuntu* 22.04, SUSE Linux Enterprise Server* 15 SP4.
- Hardware:** 4th & 5th Gen Intel® Xeon® Scalable Processor, Intel® Xeon® CPU Max Series, Intel® Data Center GPU Max Series.

The CCL library provides multiple commands for creating, managing, and using CCL communicators.²¹⁰ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.²¹¹ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Intel® MPI Library supports asynchronous progress threads that allow you to manage communication in parallel with application computation and, as a result, achieve better communication/computation overlapping.²¹²

²¹⁰ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
²¹¹ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.
²¹² Intel® MPI Library Developer Guide for Linux* OS <https://www.intel.com/content/www/us/en/docs/mpi-library/developer-guide-linux/2021-14/asynchronous-progress-control.html>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>A single-node kernel is, for example, a software program that runs on the first domain / node. The first single-node kernel is responsible for interpreting user instructions and distributing calls to at least one of the other domains / nodes for execution by cluster node modules on those nodes. For example, the calls MPI_Isend and MPI_Irecv as well as the MPI_Test algorithm correspond to a first instruction to asynchronously send a payload to another node (see, for example the dest field of the MPI_Isend call), a second instruction to asynchronously receive a payload from another node (see, for example the source field of the MPI_Irecv call), and a third instruction to search for a payload matching a message specifier (see, for example, the MPI_Test call)²¹³, respectively. In particular, the MPI_Isend call includes a tag field that acts as a message specifier that must be verified by the designated node dest to complete an asynchronous receive using the MPI_Irecv call. Dependencies are released when the MPI_Test call indicates the payload has been sent and received.</p>
35. A computer cluster node for evaluating expressions in parallel with other computer cluster nodes, the computer cluster node comprising:	<p>Each of the Accused Xeon Products²¹⁴ comprises a computer cluster node for evaluating expressions in parallel with other computer cluster nodes. For example, as depicted below and as described further herein, Intel's 4th Gen Xeon Scalable Processor²¹⁵ includes a computer cluster comprising a computer cluster node (e.g., domain) for evaluating expressions in parallel with other computer cluster nodes.</p>

²¹³ See also MPI_Test in "Intel MPI Library for Linux* OS," <https://www.intel.com/content/dam/develop/external/us/en/documents/mmpi-devref-oneapi-linux-beta10.pdf>.

²¹⁴ The Accused Xeon Products include, but are not limited to, all products including or related to Intel's Xeon Scalable Processors (Skylake-SP architecture) (<https://www.intel.com/content/www/us/en/developer/articles/technical/xeon-processor-scalable-family-technical-overview.html>), 2nd Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/developer/articles/news/second-generation-intel-xeon-processor-scalable-family-technical-overview.html>), 3rd Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/developer/articles/technical/intel-xeon-processor-scalable-family-overview.html>), 4th Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/products/docs/processors/xeon/5th-gen-xeon-scalable-family-overview.html>), 5th Gen Xeon Scalable Processors (<https://www.intel.com/content/www/us/en/products/details/processors/xeon.html>), as well as any products incorporating those items.

²¹⁵ "Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family," <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p>Accused Xeon Products</p>
<p>Each Accused Xeon Product comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster of each Accused Xeon Product. For example, each 4th Gen Xeon Scalable Processor comprises a user connection interface configured to receive a command to start a cluster initialization process for the computer cluster that includes the four domains (node). This is done, for example, by using Intel's oneCCL software, which is part of Intel's oneAPI Collective Communications Library.²¹⁷ More particularly, Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors...."²¹⁸ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.²¹⁹ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.²²⁰</p> <p>Each Accused Xeon Product is configured to access a non-transitory computer-readable medium as shown in the figure above, including Last Level Cache (LLC), DDR memory, and High-bandwidth Memory (HBM).</p> <p>Each Accused Xeon Product is configured to access program code for a single-node kernel that, when executed, causes the hardware processor to interpret user instructions to evaluate mathematical expressions and to produce results of mathematical expression evaluation. For example, the 4th Gen Xeon Scalable Processor is configured to access and execute code stored in the memory, and is further</p>	

²¹⁷ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

²¹⁸ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87Ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

²¹⁹ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

²²⁰ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>configured to cause the processor to evaluate mathematical expressions.²²¹ ²²² For example, Intel's 4th Gen Xeon Scalable Processors are configured such that each domain / node in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes at least two types of compute engines for evaluating mathematical expressions including matrix operations using multiple processors and cores:</p> <ul style="list-style-type: none"> • Intel Advanced Matrix Extensions Accelerator (AMX): AMX is a specialized hardware accelerator for evaluating matrix multiplication expressions.²²³ • Processing Cores (PCs): The PCs are general-purpose hardware accelerators that can be used to evaluate a variety of mathematical expressions.²²⁴ <p>Each 4th Gen Xeon Scalable Processor is configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the processor to produce results of mathematical expression evaluation. A first single-node kernel is responsible for interpreting user instructions and distributing calls among the domains of the 4th Gen Xeon Scalable Processor for execution. Other domains in the cluster are responsible for executing the tasks that are distributed to them. When the user submits a job to the cluster, for example, the first single-node kernel parses the job and distributes the tasks to the other domains (nodes) for execution. The other domains (nodes) then execute the tasks and return the results to the first single-node kernel or communicate the results of mathematical expression evaluation to other nodes. The first single-node kernel then collects the results from the other nodes and returns them to the user. Hence, the 4th Gen Xeon Scalable Processor configured to access executable program code stored in memory accessible by the 4th Gen Xeon Scalable Processor, where the executable program code is program code for a single-node kernel that, when executed, causes the processor, as it is configured, to interpret user instructions.</p>
--	---

²²¹ “Technical Overview Of The 4th Gen Intel Xeon Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

²²² Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., “Intel® AVX-512 - FP16 Instruction Set for Intel® Xeon® Processor Based Products Technology Guide,” <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

²²³ <https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/advanced-matrix-extensions/overview.html>.

²²⁴ <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

U.S. Pat. No. 10,333,768

a mechanism to communicate results of evaluation with other computer cluster nodes using a peer-to-peer architecture; and

Accused Xeon Products

Each of the Accused Xeon Products comprises a mechanism such as a communication network interface for one node to communicate results of evaluation with other nodes using a peer-to-peer architecture. For example, as depicted below, the 4th Gen Xeon Scalable Processor is configured to provide all-to-all communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.²²⁵

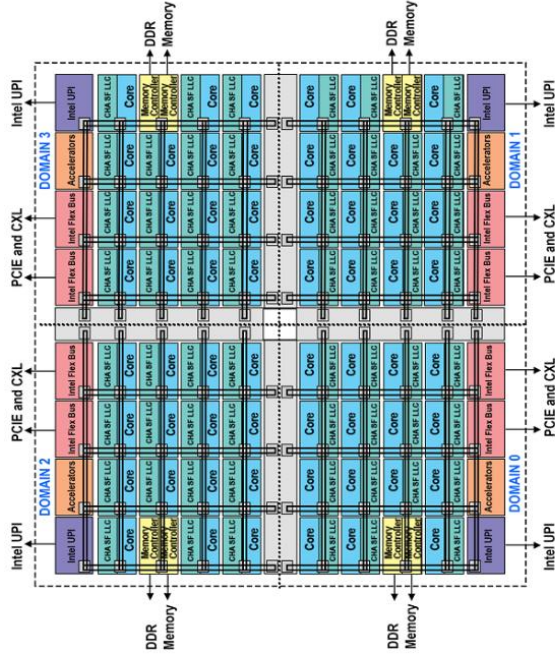


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

The 4th Gen Xeon Scalable Processor is configured to implement Intel’s oneCCL library,²²⁶ which includes commands that provide a mechanism for the domains / nodes in a 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, to communicate results of

²²⁵ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

²²⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>evaluation with each other nodes using a peer-to-peer architecture, either standing alone or, for example, in conjunction with the processor core built-in mathematical functions.^{227 228 229}</p> <p>Each of the Accused Xeon Products comprises program code that, when executed, is capable of causing the hardware processor to: receive calls from a second node comprising a second hardware processor configured to access a second memory comprising program code for a user interface and program code for a second single-node kernel, the second single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.²³⁰</p>
<p>program code that, when executed, is capable of causing the hardware processor to: receive calls from a second node comprising a second hardware processor configured to access a second memory comprising program code for a user interface and program code for a second single-node kernel, the second single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution;</p>	

²²⁷ “Technical Overview Of The 4th Gen Intel Xeon Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

²²⁸ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. *See, e.g.*, “Intel® AVX-512 - FPU Instruction Set for Intel® Xeon® Processor Based Products Technology Guide,” <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

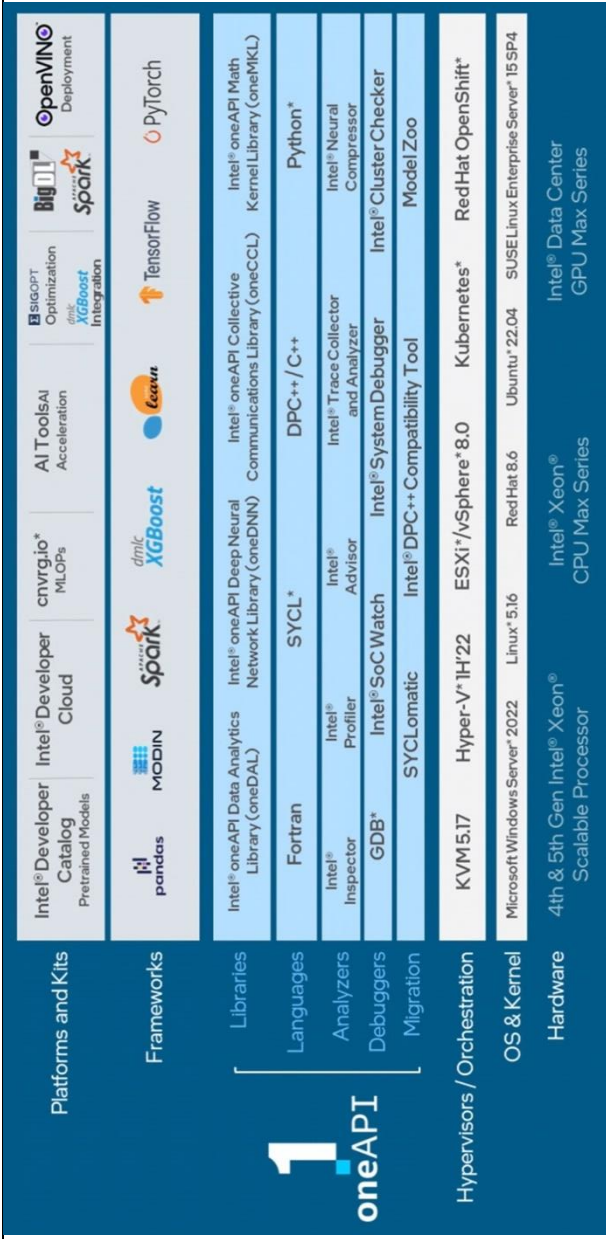
²²⁹ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onecl.html#gs.ejo7gf>.

²³⁰ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.²³⁴ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors.²³⁵</p>
--	---

²³⁴ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

²³⁵ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.



The CCL library provides multiple commands for creating, managing, and using CCL communicators.²³⁶ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.²³⁷ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

The first single-node kernel is, for example, a software program that runs on the first domain / node. The first single-node kernel is responsible for interpreting user instructions and receiving calls from a second domain / node configured to access a second memory comprising program code for a user interface and program code configured to interpret user instructions and distributing calls to at least one of the other domains / nodes for execution. The other domains / nodes in the cluster are

²³⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
²³⁷ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768	Accused Xeon Products
	<p>responsible for executing the tasks that are distributed to them by the first single-node kernel. This is known as the master-slave model, where a first domain / node executes in a supervisory role, and the other domains / nodes serve as compute nodes. For example, initially, the first node is configured to interpret user instructions and distribute calls to at least one of the other nodes for execution.</p> <p>Unlike a master-slave model, the nodes communicate results of mathematical expression evaluation with each other during execution of the tasks that are distributed to them by the first node using a peer-to-peer architecture without being required to go through the first node.</p> <p>When the user submits a job to the cluster, the first single-node kernel parses the job and distributes the tasks to the other nodes for execution. The other nodes then execute the tasks and return the results to the first single-node kernel. The first single-node kernel then collects the results from the other nodes and returns them to the user.</p>
<p>execute, using the hardware processor, at least a first mathematical expression evaluation; and communicate a result of the first mathematical expression evaluation to a third node comprising</p>	<p>Each of the Accused Xeon Products comprises program code that, when executed, is capable of causing a hardware processor in the first node to execute, using the hardware processor, at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node.</p> <p>For example, as depicted below, the 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.²³⁸</p>

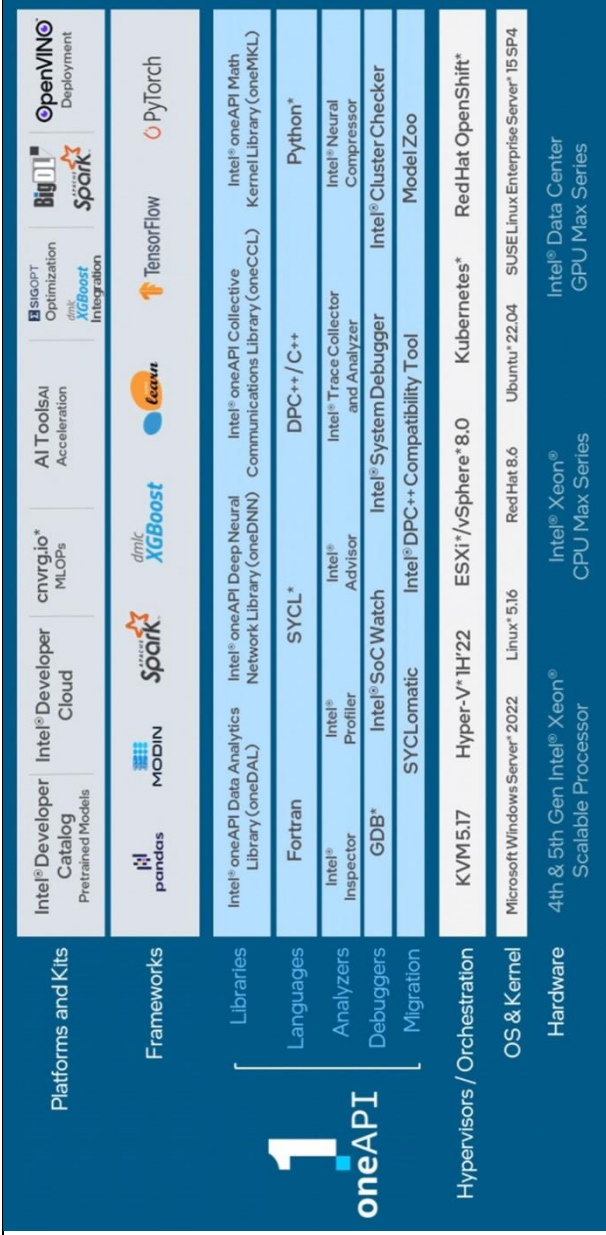
²³⁸ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.²⁴² The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors.²⁴³</p>
--	---

²⁴² “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

²⁴³ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

Accused Xeon Products



The CCL library provides multiple commands for creating, managing, and using CCL communicators.²⁴⁴ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.²⁴⁵ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Each of the Accused Xeon Products comprises a hardware processor with multiple processing cores, and is configured to receive the result of mathematical expression evaluation from the first node and to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation.

a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of

²⁴⁴ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

²⁴⁵ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

U.S. Pat. No. 10,333,768

mathematical expression evaluation from the computer cluster node, execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation, and communicate a result of the second mathematical expression evaluation to the first node;

Accused Xeon Products

For example, when executed, the code stored in the memory accessible by the 4th Gen Xeon Scalable Processor causes the processor to receive a result of a first mathematical expression evaluation from a first node (domain) in the 4th Gen Xeon Scalable Processor and to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation (e.g., by performing matrix multiplications).

For example, as depicted below, the Intel 4th Gen Xeon Scalable Processor, configured in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes four domains where each domain includes fifteen processing cores, each of which is configured to access memory including, for example, LLC, DDR, and HBM memory.

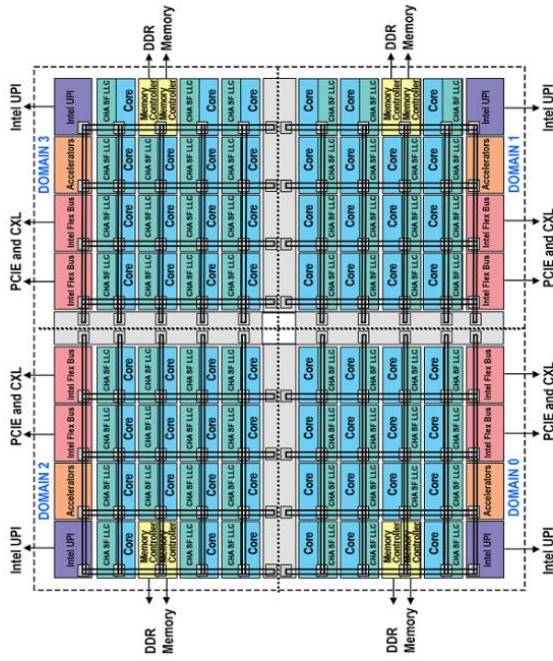


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Intel’s Accused Xeon Products are configured to be supported by Intel’s oneAPI software suite, which, for example, “maximize[s] application performance by activating the advanced capabilities of

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>4th and 5th gen Intel® Xeon® processors...²⁴⁶ The oneCCL library is one of the 10 core elements of Intel’s oneAPI software suite.²⁴⁷ Intel’s oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.²⁴⁸</p> <p>“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”²⁴⁹ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements.</p> <p>The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the context of the software suite for Intel’s 4th Gen Xeon Scalable Processors:²⁵⁰</p>
--	--

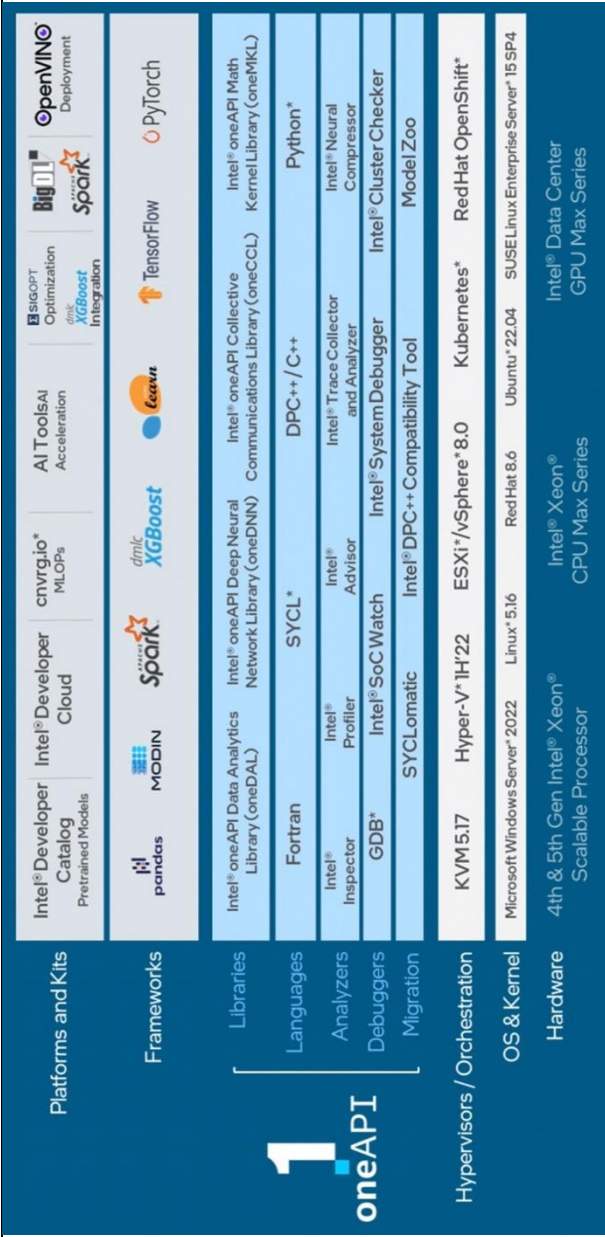
²⁴⁶ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

²⁴⁷ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

²⁴⁸ See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/2021-12/overview.html>.

²⁴⁹ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

²⁵⁰ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.



The CCL library provides multiple commands for creating, managing, and using CCL communicators.²⁵¹ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.²⁵² The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.

Also, for example, the architecture of the 4th Gen Xeon Scalable Processor is configured to receive the result of a first mathematical expression evaluation from the first node (domain) and to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation. The source code for Intel’s oneCCL software, for example, includes the details for using CCL commands to receive the result of a first mathematical expression evaluation from a first node

²⁵¹ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).

²⁵² <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<p>and to execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation.</p>
<p>wherein the user connection interface is configured to return at least one result of mathematical expression evaluation to a user interface or a script;</p> <p>and</p>	<p>A user connection interface in each of the Accused Xeon Products includes a first node configured to return at least one result of mathematical expression evaluation to a user interface or a script.</p> <p>Intel's Accused Xeon Products are configured to be supported by Intel's oneAPI software suite, which, for example, "maximize[s] application performance by activating the advanced capabilities of 4th and 5th gen Intel® Xeon® processors..."²⁵³ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.²⁵⁴ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.²⁵⁵</p> <p>The CCL library provides multiple commands for creating, managing, and using CCL communicators.²⁵⁶ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.²⁵⁷ The CCL library is responsible for distributing the workload across the domains / nodes and for collecting the results from the domains / nodes.</p>
<p>wherein the computer cluster node is configured to: accept user instructions; after accepting user instructions, communicate</p>	<p>Each of the Accused Xeon Products includes a node configured to accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernels.</p>

²⁵³ "Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors," <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; "oneAPI Specification, Release 1.3-rev-1," pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

²⁵⁴ <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/communication.html>.

²⁵⁵ See "Intel® oneAPI Collective Communications Library Developer Guide and Reference," <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

²⁵⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneapi.html#gs.ejo7gf> ("Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.")

²⁵⁷ <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-9/device-communication.html>.

Accused Xeon Products	
<p>U.S. Pat. No. 10,333,768</p> <p>at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p>	<p>For example, the architecture of the 4th Gen Xeon Scalable Processor is configured to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the domains to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p> <p>Also for example, the source code for Intel's oneCCL software includes the details for using CCL commands to cause 4th Gen Xeon Scalable Processor to accept user instructions, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the 4th Gen Xeon Scalable Processor to communicate with each other, and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernels on one or more domains in the 4th Gen Xeon Scalable Processor.</p>
<p>36. The computer cluster node of claim 35, wherein the one or more non-transitory memory devices comprise program code for performing a parallel fast Fourier transform, wherein the program code causes the hardware processor to evaluate a command to perform a Fourier transform on an array comprising a first data portion that is stored on the computer cluster node and a second data portion that is not stored on the computer cluster node.</p>	<p>Each of the Accused Xeon Products comprises one or more non-transitory memory devices which comprise program code for performing a parallel fast Fourier transform, wherein the program code causes the hardware processor to evaluate a command to perform a Fourier transform on an array comprising a first data portion that is stored on the computer cluster node and a second data portion that is not stored on the computer cluster node.</p> <p>Each of the Accused Xeon Products comprises a plurality of nodes that comprises one or more cluster node modules. For example, as depicted below, the 4th Gen Xeon Scalable Processor comprises four nodes (described as domains in the figure below), each of which is configured to access memory including, for example, LLC and DDR memory.²⁵⁸</p>

²⁵⁸ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

processors....²⁶⁰ The oneCCL library is one of the 10 core elements of Intel's oneAPI software suite.²⁶¹ Intel's oneCCL package comprises the oneCCL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.²⁶²

“The Intel® oneAPI Math Kernel Library (oneMKL) is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance. oneMKL contains the high-performance optimizations from the full Intel® Math Kernel Library for CPU architectures (with C/Fortran programming language interfaces) and adds to them a set of SYCL.* interfaces for achieving performance on various CPU architectures and Intel Graphics Technology for certain key functionalities. oneMKL provides BLAS and LAPACK linear algebra routines, fast Fourier transforms, vectorized math functions, random number generation functions, and other functionality.”²⁶³ The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements. Utilizing the highly optimized FFT implementations within the Intel oneMKL library, oneMKL provides optimized routines for various FFT algorithms, leveraging advanced instruction sets (like AVX-512) and multi-core parallelism available on Xeon Scalable processors. The highly optimized Basic Linear Algebra Subprograms (BLAS) routines within oneMKL are used to perform the matrix multiplications involved in the butterfly operations of the FFT algorithm. BLAS provides highly optimized implementations for matrix-vector and matrix-matrix multiplications. The full capabilities of the oneMKL library, including its optimized FFT routines, BLAS functions, and threading support are used to calculate a Fourier transform across the computer cluster in parallel.

²⁶⁰ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>; “oneAPI Specification, Release 1.3-rev-1,” pp. 388-389 <https://spec.oneapi.io/versions/latest/oneAPI-spec.pdf>; “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>, <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/communicator.html>.

²⁶¹ <https://www.intel.com/content/www/us/en/developer/articles/technical/oneapi-what-is-it.html>.

²⁶² See “Intel® oneAPI Collective Communications Library Developer Guide and Reference,” <https://www.intel.com/content/www/us/en/docs/oneapi/developer-guide-reference/2021-12/overview.html>.

²⁶³ “Intel® oneAPI Programming Guide,” <https://www.intel.com/content/www/us/en/docs/oneapi/programming-guide/2024-1/intel-oneapi-math-kernel-library-onemkl.html>.

The source code for the CCL library includes the details for using oneCCL commands to initialize communications on a 4th Gen Xeon Scalable Processor, receive first commands from a user interface without the first commands first passing through the first kernel, and after receiving the first commands from the user interface, send second commands to the first kernel.

As described above, oneMKL is a computing math library of highly optimized and extensively parallelized routines for applications that require maximum performance that is an advanced functions module. Among its functionality is Fast Fourier Transforms²⁶⁷ that are used within the parallelized routines of oneMKL to calculate a Fourier transform across the computer cluster in parallel.²⁶⁸

The mathematical routines addressed in oneMKL include matrix operations using, for example, arrays of data elements. The program code thus directs calculation of a fast Fourier transform in parallel by directing the hardware processor to evaluate a command to perform a Fourier transform on an array comprising a first data portion that is stored on the computer cluster node and a second data portion that is not stored on the computer cluster node.

The processors are configured to evaluate multiple mathematical expressions and matrix operations.²⁶⁹ For example, Intel's 4th Gen Xeon Scalable Processors are configured such that each domain / node in a Sub-NUMA clustering 4 (SNC-4) affinity mode, includes at least two types of compute engines for evaluating mathematical expressions including matrix operations using multiple processors and cores:

²⁶⁷ See, e.g., <https://www.intel.com/content/www/us/en/developer/articles/technical/performance-benchmarks-onemkl-on-xeon-processors.html>; <https://www.intel.com/content/www/us/en/docs/onemkl/developer-reference-c/2023-1/cluster-fft-functions.html>.

²⁶⁸ “Developer Reference for Intel® oneAPI Math Kernel Library for C,” <https://www.intel.com/content/www/us/en/docs/onemkl/developer-reference-c/2023-1/cluster-fft-functions.html> (“One or more [Cluster FFT] processes may be running in parallel on each cluster node.”).

²⁶⁹ “Technical Overview Of The 4th Gen Intel Xeon Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

²⁷⁰ Processor cores are configured to evaluate multiple mathematical expressions, such as addition, subtraction, multiplication, division, and square root. See, e.g., “Intel® AVX-512 - FP16 Instruction Set for Intel® Xeon® Processor Based Products Technology Guide,” <https://www.intel.com/content/www/us/en/content-details/669773/intel-avx-512-fp16-instruction-set-for-intel-xeon-processor-based-products-technology-guide.html>.

Accused Xeon Products	
U.S. Pat. No. 10,333,768	<ul style="list-style-type: none"> ● Intel Advanced Matrix Extensions Accelerator (AMX): AMX is a specialized hardware accelerator for evaluating matrix multiplication expressions.²⁷¹ ● Processing Cores (PCs): The PCs are general-purpose hardware accelerators used to evaluate a variety of mathematical expressions.²⁷²
37. The computer cluster node of claim 35, wherein the computer cluster node is configured to permit exchange of information with other computer cluster nodes during the course of parallel computation.	<p>Each of the Accused Xeon Products comprises a plurality of nodes, wherein the computer cluster node is configured to permit exchange of information with other computer cluster nodes during the course of parallel computation.</p> <p>For example, as depicted below and as described further herein, Intel’s 4th Gen Xeon Scalable Processor²⁷³ includes a computer cluster comprising a computer cluster node (e.g., domain) for evaluating expressions in parallel with other computer cluster nodes.</p>

²⁷¹ <https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/advanced-matrix-extensions/overview.html>.

²⁷² <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

²⁷³ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

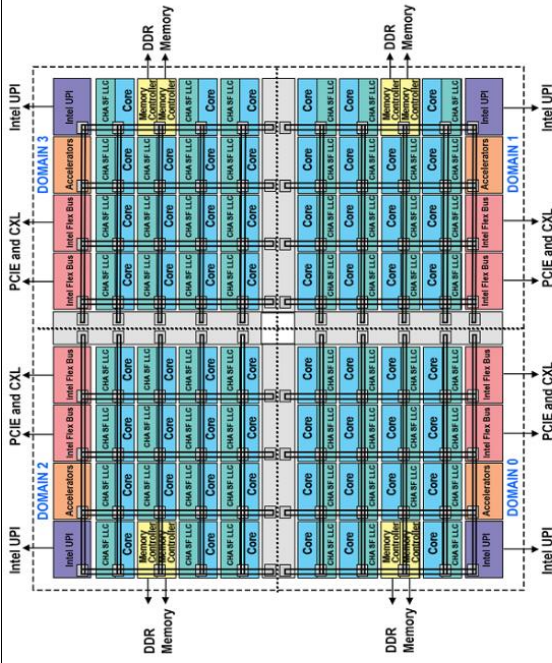


Figure 3 – Block Diagram Representing Domains Of sub-NUMA With Four Clusters

Each of Accused Xeon Products comprises a mechanism such as a communication network interface for the nodes to communicate results of computations with each other. For example, as depicted above, the 4th Gen Xeon Scalable Processor is configured to provide all-to-all communication using Sub-NUMA clustering 4 (SNC-4) affinity mode using four domains of sub-NUMA clusters / nodes.²⁷⁴

The figure below depicts Intel’s oneAPI software suite and its Collective Communications Library (oneCCL) and Math Kernel Library (oneMKL) in the exemplary context of the software suite for Intel’s 4th Gen Xeon Scalable Processors.²⁷⁵

²⁷⁴ “Technical Overview Of The 4th Gen Intel® Xeon® Scalable processor family,” <https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html>.

²⁷⁵ “Software for 4th and 5th Gen Intel® Xeon® and Intel® Max Series Processors,” <https://www.intel.com/content/www/us/en/developer/platform/4gen-xeon-max-series-cpu.html#gs.87ciqd>.

The diagram illustrates the Intel oneAPI ecosystem. At the top, it lists 'Platforms and Kits' including Intel® Developer Catalog Pretrained Models, Intel® Developer Cloud, cnvrg.io* MLOps, AI Tools/AI Acceleration, SIGOPT Optimization, Intel® XGB Boost Integration, BigDL, Spark, and OpenVINO Deployment. Below this are 'Frameworks' such as pandas, MODIN, Spark, XGB Boost, Leaven, TensorFlow, and PyTorch. The 'Libraries' section includes Intel® oneAPI Data Analytics Library (oneDAL), Intel® oneAPI Deep Neural Network Library (oneDNN), Intel® oneAPI Collective Communications Library (oneCCL), and Intel® oneAPI Math Kernel Library (oneMKL). 'Languages' listed are Fortran, SYCL*, and DPC++/C++. 'Analyzers' include Intel® Profiler, Intel® Trace Collector and Analyzer, and Intel® Neural Compressor. 'Debuggers' include GDB*, Intel® SoC Watch, Intel® System Debugger, Intel® Cluster Checker, and Model Zoo. 'Migration' is supported by SYCLomatic and Intel® DPC++ Compatibility Tool. 'Hypervisors / Orchestration' includes KVM 5.17, Hyper-V* IH*22, ESXi*/vSphere* 8.0, Kubernetes*, and Red Hat OpenShift*. 'OS & Kernel' options are Microsoft Windows Server* 2022, Linux* 5.16, Red Hat 8.6, Ubuntu* 22.04, and SUSE Linux Enterprise Server* 15 SP4. 'Hardware' includes 4th & 5th Gen Intel® Xeon® Scalable Processor, Intel® Xeon® CPU Max Series, and Intel® Data Center GPU Max Series.

The CCL library provides multiple commands for creating, managing, and using CCL communicators.²⁷⁶ The CCL library includes software objects that are used to manage the communication between the domains / nodes accelerators in the 4th Gen Xeon Scalable Processor.²⁷⁷ The CCL library is responsible for receiving first commands from a user interface without the first commands first passing through the first kernel, and after receiving the first commands from the user interface, send second commands to the first kernel.

The source code for the CCL library includes the details for using oneCCL commands to permit exchange of information among nodes during the course of parallel computation.

²⁷⁶ Intel oneAPI Collective Communications Library, <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneccl.html#gs.ejo7gf> (“Enables efficient implementations of collectives that are heavily used for neural network training, including all-gather, all-reduce, and reduce-scatter.”).
²⁷⁷ <https://www.intel.com/content/www/us/en/docs/oneapi/oneccl/developer-guide-reference/2021-9/device-communication.html>.

<p>U.S. Pat. No. 10,333,768</p>	<p style="text-align: center;">Accused Xeon Products</p> <p>for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions.</p> <p>Each Accused Xeon Product comprises a hardware processor that comprises multiple processor cores. For example, the 4th Gen Xeon Scalable Processor is configured to operate in an SNC-4 mode where each of the four domains is a node that includes a plurality of processing cores (fifteen) as shown in the above figure. Each 4th Gen Xeon Scalable Processor is, therefore, a special purpose microprocessor.</p>
--	--