IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAP AMERICA, INC.
Petitioner,

v.

VALTRUS INNOVATIONS LTD. AND KEY PATENT INNOVATIONS LIMITED
Patent Owner.

Case No. IPR2025-00415
U.S. Patent No. 6,691,139

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 6,691,139

TABLE OF CONTENTS

Page

TABLE OF ABBREVIATIONS AND CONVENTIONS

Petitioner	SAP America, Inc.
Patent Owners	Valtrus Innovations Ltd. ("Valtrus") and Key Patent innovations Ltd. ("KPI")
'139 Patent	U.S. Patent No. 6,691,139 (the patent being challenged herein)
Challenged Claims	Claims 1-18
POSITA	Person of Ordinary Skill in the Art

CLAIM LISTING

Claim 1	
1[preable]	A method for creating a standby data processing system located at a standby site, which is remotely located from a primary site, comprising the steps of:
1[a]	(a) computing a difference between an active transaction redo log and an archive transaction redo log to produce a delta image at the primary site;
1[b]	(b) transmitting the active transaction redo log and the delta image, separately in time sequence, from the primary site to the standby site; and
1[c]	(c) combining the delta image with the active transaction redo log at the standby site, to produce a standby archive transaction redo log.
Claim 2	
2	A method as in claim 1, further comprising the step of storing the standby archive transaction redo log on the standby site.
Claim 3	
3	A method as in claim 1, further comprising the step of storing transactions in the standby active transaction redo log as they are trapped at the primary site.
Claim 4	
4	A method as in claim 1, further comprising the step of entering transactions from the archive transaction redo log into the standby transaction processing system.
Claim 5	
5[preamble]	A method for creating archive redo logs for a standby database at a remote location, comprising the steps of:

5[a]	(a) trapping active log writes using a remote mirroring device that is capable of mirroring the log writes to a standby site, to create an online redo log at the standby site;
5[b]	(b) computing a difference between an online redo log and archive redo log to create a delta image at a primary database, wherein the delta image includes an archive header;
5[c]	(c) transmitting the delta image to a standby archive log replicator contained in the standby database; and
5[d]	(d) combining the delta image and archive header with the online redo log to create a standby archive redo log on the standby database.
Claim 6	
6	A method as in claim 5, further comprising the step of applying the standby archive redo log by creating links to the standby archive redo log.
Claim 7	
7	A method as in claim 5, further comprising the step of transmitting the delta image to a standby archive log replicator using a first remote mirroring device in the primary database and a second remote mirroring device in the standby database.
Claim8	
8	A method as in claim 7, further comprising the step of using replication handles to track the progress of mirror writes on the standby database as compared to the primary database.
Claim 9	
9	A method as in claim 8, further comprising the step of checking the primary database queries using the archive log replicator to determine whether archiving is complete.

Claim 10	
10	A method as in claim 5, further comprising the step of shipping an entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously after archiving becomes slow enough that there is a risk of primary database blocking.
Claim 11	
11	A method as in claim 10, further comprising the step of shipping an entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously when a spool buffer in a remote mirroring system has been filled.
Claim 12	
12[preamble]	A method for creating archive redo logs from a primary database, for a standby database at a remote location, comprising the steps of:
12[a]	(a) trapping file system transactions to create an online redo log at the standby database;
12[b]	(b) computing a difference between an online redo log and archive redo log in the primary database to find the delta image of the archive redo log;
12[c]	(c) transmitting the delta image to a standby database; and
12[d]	(d) combining the delta image with the online redo log in the standby database to create standby archive redo log on the standby database.
Claim 13	
13	A method as in claim 12, further comprising the step of applying the standby archive redo log by creating a link between the standby database and the standby archive redo log.

Claim 14	
14	A method as in claim 12, further comprising the step of transmitting the delta image to a standby archive log replicator using a first remote mirroring device in the primary database and a second remote mirroring device in the standby database.
Claim 15	
15	A method as in claim 14, further comprising the step of using replication handles to track the progress of mirror writes on the standby database as compared to the primary database.
Claim 16	
16	A method as in claim 12, further comprising the step of checking the primary database with database queries using the archive log replicator to determine whether archiving is complete.
Claim 17	
17	A method as in claim 12, further comprising the step of shipping an entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously after archiving becomes slow enough that there is a risk of the primary database blocking.
Claim 18	
18	A method as in claim 12, further comprising the step of shipping the entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously when a spool buffer in a remote mirroring system has been filled.

PETITIONER'S EXHIBIT LIST

No.	Description
1001	U.S. Patent No. 6,691,139
1002	File History of U.S. Patent No. 6,691,139
1003	Declaration of Kevin Almeroth, Ph.D.
1004	*Curriculum Vitae* of Kevin Almeroth, Ph.D.
1005	U.S. Pat. No. 6,823,336 ("Srinivasan")
1006	U.S. Pat. No. 6,658,589 ("Taylor")
1007	Ramez Elmasri and Shamkant B. Navathe, FUNDAMENTALS OF DATABASE SYSTEMS (2d ed.1994)
1008	Original Complaint for Patent Infringement in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1009	Original Complaint for Patent Infringement in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00533-JRG (E.D. Tex.)
1010	Director Vidal's Memorandum Regarding Discretionary Denial Procedure, Dated June 21, 2022
1011	First Amended Docket Control Order in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1012	Motion to Dismiss for Lack of Standing (Public Version) in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1013	Motion to Reconsider Ruling on Motion to Dismiss for Lack of Standing (Public Version) in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)

No.	Description
1014	Plaintiffs' Notice Regarding *Valtrus II* in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)
1015	SAP's Invalidity Contentions (Cover Pleading) in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.)

I. INTRODUCTION

Petitioner requests *inter partes* review ("IPR") of claims 1-18 ("Challenged Claims") of U.S. Patent No 6,691,139 ("'139 Patent"). The claims of the '139 patent are directed to disaster recovery for online computers including methods (claims 1, 5, and 12). Disaster recovery for online computers using the concept of "deltas" was commonplace in the industry years before the '139 Patent's earliest claimed priority date.

The '139 Patent was allowed based on an amendment that the "delta image" is computed from a difference between an active transaction redo log and an archive transaction redo log. Ex-1002, 37-43.

II. COMPLIANCE WITH IPR REQUIREMENTS

A. Grounds for Standing

As required by 37 C.F.R. § 42.104(a), Petitioner certifies that the '139 Patent "is available for inter partes review and that the Petitioner is not barred or estopped from requesting an inter partes review challenging the patent claims on the grounds identified in the petition." Neither Petitioner nor any real party in interest has filed a civil action challenging the validity of a claim of the '139 Patent. This petition is being filed no more than 1 year after the date on which Petitioner or real parties in interest or privies of Petitioner were served with a complaint alleging infringement of the '139 Patent.

B. Mandatory Notices

Real Party-in-Interest: Petitioner identifies SAP America, Inc. and SAP SE (collectively, "SAP") as real parties-in-interest.

Related Matters: No previous IPR or post-grant proceeding has been filed challenging the '139 Patent. Valtrus Innovations LTD. ("Valtrus") has asserted the '139 Patent in litigation against SAP in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00021-JRG (E.D. Tex.), Valtrus and Key Patent Innovations LTD. ("KPI") (collectively, "Patent Owners") have asserted the '139 Patent in litigation against SAP in *Valtrus Innovations Ltd. v. SAP America, Inc. et al*, 2:24-cv-00533 (E.D. Tex.), and SAP filed a declaratory judgment action of non-infringement of the '139 Patent against Valtrus, KPI, and Patent Platform Services, LLC in *SAP America, Inc. v. Valtrus Innovations Limited et al*, 1:24-cv-00756-GBW (D. Del.).

Lead and Back-up Counsel:

Lead Counsel	Back-up Counsel
Louis L. Campbell (Reg. # 59,963) WINSTON & STRAWN LLP 255 Shoreline Drive, Suite 520 Redwood City, CA 94065 Tel: (650) 858-6500 LLcampbell@winston.com	**Scott M. Border (Reg. # 77,744)** WINSTON & STRAWN LLP 1901 L Street NW Washington, DC 20036 Tel: (202) 282-5000 SBorder@winston.com

Lead Counsel	Back-up Counsel
	Samuel W. Riebe (Reg. # 78,044) WINSTON & STRAWN LLP 2121 N. Pearl Street, Suite 900 Dallas, TX 75201 Tel: (214) 453-6500 SRiebe@winston.com
	Michael Bittner (*Pro hac vice* to be submitted) WINSTON & STRAWN LLP 2121 N. Pearl Street, Suite 900 Dallas, TX 75201 Tel: (214) 453-6500 MBittner@winston.com
	Matt R. McCullough (*Pro hac vice* to be submitted) WINSTON & STRAWN LLP *Pro hac vice* to be submitted Winston & Strawn LLP 255 Shoreline Drive, Suite 520 Redwood City, CA 94065 Tel: (650) 858-6500 MRMcCullough@winston.com
	William Logan (*Pro hac vice* to be submitted) *Pro hac vice* to be submitted WINSTON & STRAWN LLP 800 Capitol Street, Suite 2400 Houston, TX 77002 Tel: (713) 651-2600 WLogan@winston.com

Lead Counsel	Back-up Counsel
	Patrick Clark (*Pro hac vice* to be submitted) WINSTON & STRAWN LLP *Pro hac vice* to be submitted 2121 N. Pearl Street, Suite 900 Dallas, TX 75201 Tel: (214) 453-6500 PClark@winston.com

Service Information: Service information is provided in the designation of counsel above. Petitioner consents to service of all documents via electronic mail at IPR-SAP-Val@winston.com.

C. Fee Authorization

The Office is authorized to charge all fees due in connection with this proceeding to Deposit Account No. 50-1814.

III. THE CHALLENGED CLAIMS

Under 37 C.F.R. §§ 42.104(b) and 42.22, Petitioner requests that the Board institute this IPR on the Challenged Claims of the '139 Patent and cancel those claims as unpatentable on the following grounds:

Ground	Claims	Basis for Unpatentability
1	1-18	Obviousness under 35 U.S.C. § 103 over Srinivasan and Taylor

IV. THE '139 PATENT

A. Effective Filing Date

The '139 Patent issued from Application No. 09/774,839, filed January 31, 2001, which is a standalone non-provisional application with no priority claim. Consequently, the earliest possible effective filing date for the '139 Patent is January 31, 2001.

B. Level of Ordinary Skill in the Art

As of January 31, 2001, a POSITA in the '139 Patent's field would have a working knowledge of data storage, including data redundancy systems and methods used in a fault-tolerant manner. Such a POSITA would have had a bachelor's degree in electrical engineering, computer science, or equivalent training, and approximately two years of work experience in software development in the field of network data processing and storage. Additional education can compensate for work experience, and vice versa. Ex-1003, ¶ 33. Based on his education and experience, Dr. Almeroth has at least the knowledge of and qualifies as a POSITA. Ex-1003, ¶¶ 7-31.

C. Technology Overview

The '139 Patent describes techniques for disaster recovery for online computer systems, including the creation of a standby data processing site at a remote location using reduced communication bandwidth. Ex-1001 at 1:6-10. The '139 Patent explains that the Internet and increasing use of networked computers has resulted in

a significant increase in network traffic based on the number of transactions performed by remote servers. Ex-1001, 1:13-16. Since the information in many transactions is critically important (e.g., credit card transactions, email, etc.), the information needs to be accessible 24/7. Ex-1001, 1:18-25.

The ’139 Patent recognizes that systems fail, resulting in unacceptable downtime, and so database systems have been developed in which a redundant backup system allows for a fast failover, in which the backup (standby) system takes over transaction processing and storage if the primary system fails to avoid interruptions. Ex-1001, 1:26-37. The ’139 Patent recognizes that certain failures may be caused by localized events, such as power outages, fires, floods, etc., and so it recognizes the benefit of having a secondary system at a remote location relative to the primary system to avoid the possibility of the localized event bringing both systems down. Ex-1001, 1:38-47.

The ’139 Patent explains that multiple systems can also safeguard against data corruption because the data can be stored in both the primary and secondary systems. Ex-1001, 1:49-55. The ’139 Patent recognizes that systems keep transaction logs, such as an “active log” for tracking recent transactions, and an “archive log” to store information from the active log once the active log has been filled or a certain time period has passed. Ex-1001, 1:57-60. Once created, the logs can be transferred to

backup systems using a "mirroring" process for both the active logs and archive logs, which in turn keeps the backup system current. Ex-1001, 1:60-65.

The primary and backup (standby) systems are remotely located from one another to improve reliability in the event one system goes down. Ex-1001, 1:37-47; 1:67-2:1. Data and the transaction logs (e.g., active logs or archive logs) that are sent to the remote (standby) system are often sent across a wide area network (WAN), which is slow compared to a local area network (LAN) or high-speed cluster connection. Ex-1001, 2:1-5. Therefore, the transmission rates across the WAN can cause significant network congestion, especially when the files transmitted are large (e.g., contain hundreds of megabytes of data per file). Ex-1001, 2:5-9.

The '139 Patent recognizes that "[n]ot only is it important to have a standby site, but both sites should be constantly synchronized. Ex-1001, 3:26-29. And so, to avoid "mirroring of the entire database [which] consumes a significant amount of bandwidth on a WAN which is already slow," the '139 Patent states, "it is valuable to compute a difference between an active data file and an archive data file to produce a delta image." Ex-1001 3:55-62. The '139 Patent shows the process of creating archive data files from online data files (shown in Fig. 1 below):

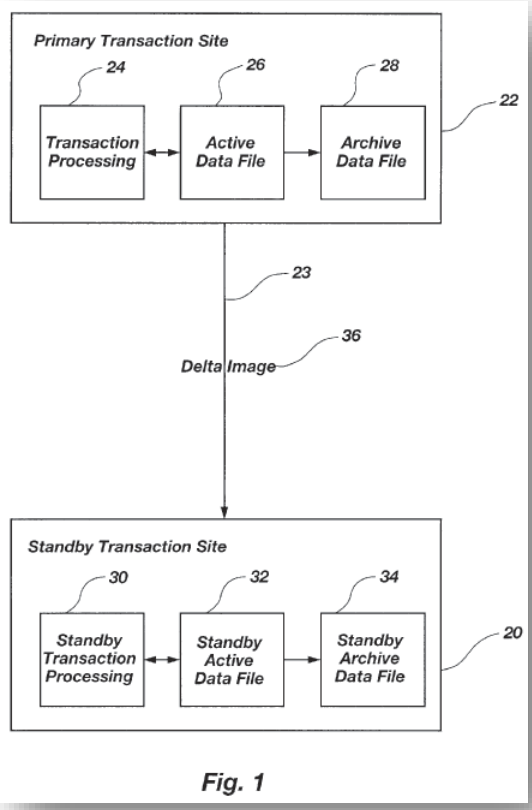


Fig. 1

Ex-1001, Fig. 1.

The ’139 Patent explains that the “delta image is useful because the archive data file is an archive of the active data file and there are large similarities between both,” but instead of the large amount of bandwidth used to send the whole archive data file, it “is more economical to send a smaller delta image.” Ex-1001, 3:62-66.

The system of the ’139 Patent includes a first primary cluster node 50 and a second primary cluster node 52. Each node contains a separate instance of a parallel database server 54, 56, and each node has its own copy of the online redo logs 58, 60 (shown in Fig. 3 below):

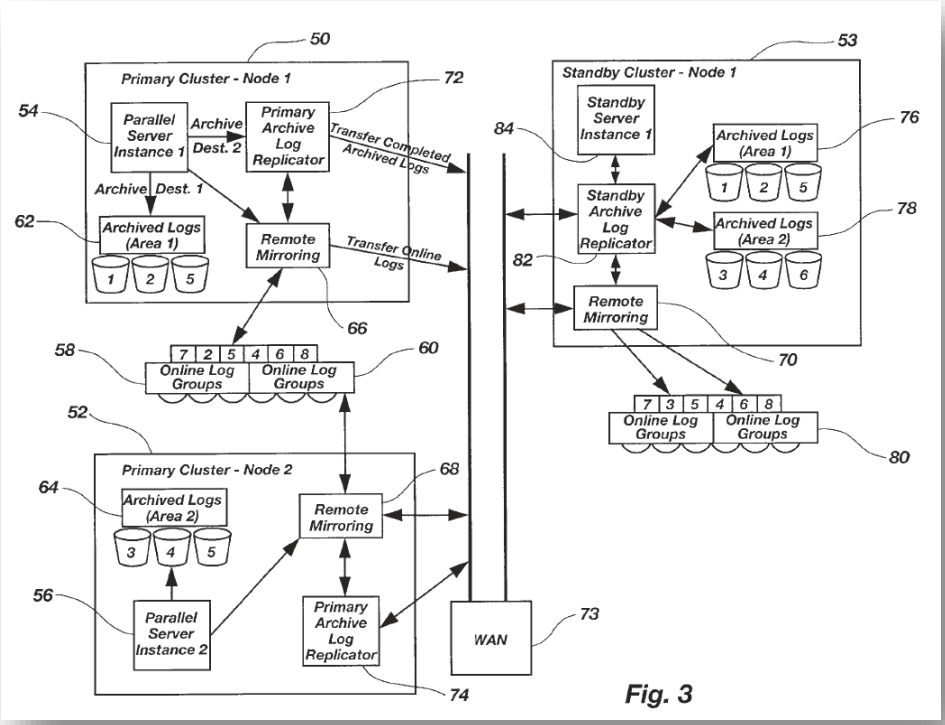


Fig. 3

Ex-1001, 4:43-49; Fig. 3.

Once the online redo logs are filled, they are then archived and transferred to the archive log areas 62, 64. Ex-1001, 4:52-55. The parallel database servers 54, 56 then fill up the online redo logs 58, 60 when transactions commit, and remote mirroring devices 66, 68 trap the disk subsystem writes to the online logs. Ex-1001, 4:56-60. Then the trapped information is sent (as a delta image) to the standby remote mirroring device 70 and a WAN 73 is used to transfer the information from the primary to the standby site. Ex-1001, 4:60-64; 5:9-15.

V. CLAIM CONSTRUCTION

Claims are given their "ordinary and customary meaning" as understood by a POSITA and the prosecution history pertaining to the patent. 37 C.F.R. § 42.100(b). Because a POSITA would find the challenged claims unpatentable under any interpretation consistent with their plain and ordinary meaning in the context of the '139 Patent, *see* Ex-1003, ¶ 53, the Board need not expressly construe the claim terms. *See Vivid Techs., Inc. v. Am. Sci. & Eng'g. Inc*., 200 F.3d 795, 803 (Fed. Cir. 1999).

VI. PRINCIPAL PRIOR ART

A. Srinivasan (Ex-1005)

U.S. Pat. No. 6,823,336 to Srinivasan et al. ("Srinivasan") was filed on September 26, 2000, and published upon issuance on November 23, 2004, and is therefore prior art under pre-AIA 35 U.S.C. § 102 (e).

Srinivasan describes a data storage system for transaction processing applications. Ex-1005, 11-15. Srinivasan teaches "a method of read-write access by a first host processor to a dataset in a first data storage system concurrent with uninterrupted read-only access by a second host processor to a consistent state of a copy of the dataset in a second data storage system." Ex-1005, 2:61-65. To perform the method, Srinivasan describes embodiments of a data processing system (one shown in Figure 1):

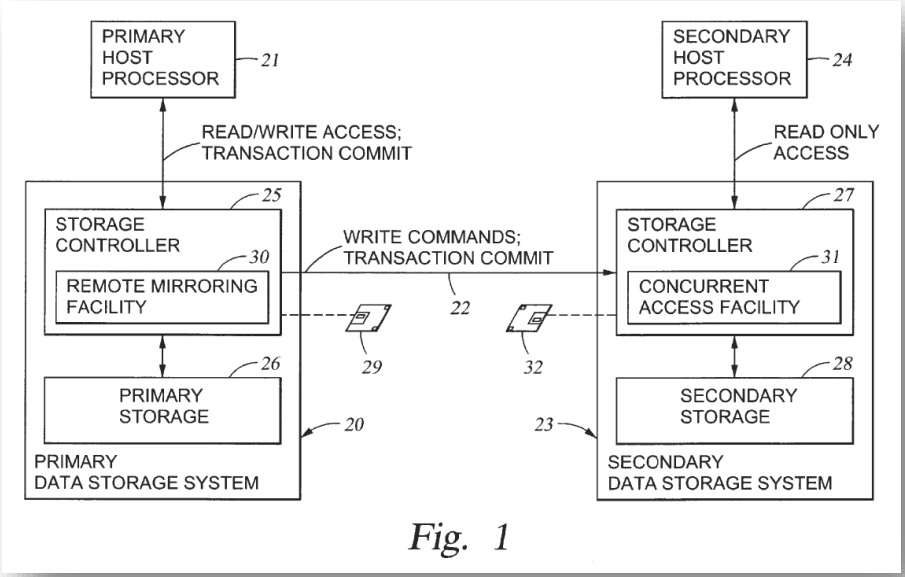


Fig. 1

Ex-1005, 5:16-20, Fig. 1. Srinivasan's Figure 8 shows a preferred construction of the data processing system of Figure 1:

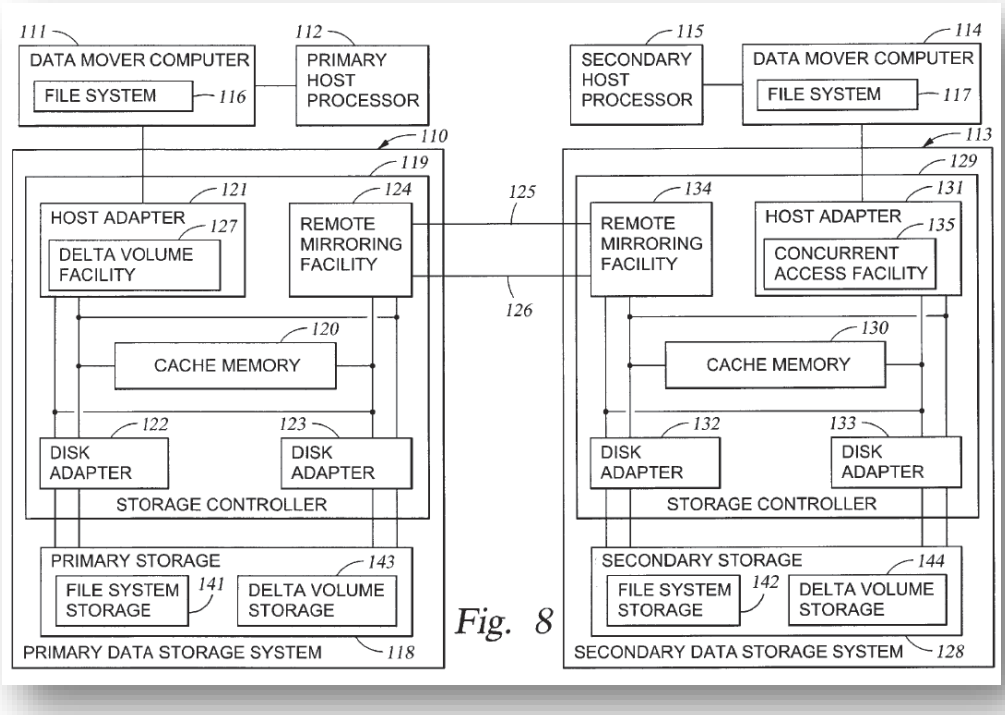


Fig. 8

Ex-1005, 4:52-53, 9:51-54; Fig. 8 (annotated). Srinivasan teaches that the "remote mirroring facility mirrors file system storage 141 in the primary storage 118" and does so "by mirroring delta volume storage 143 that is used to buffer the updates to file system storage 141 of the primary storage 118." Ex-1005, 10:17-22.

Srinivasan also teaches control flow through the secondary data storage system (shown in Fig. 2 below):

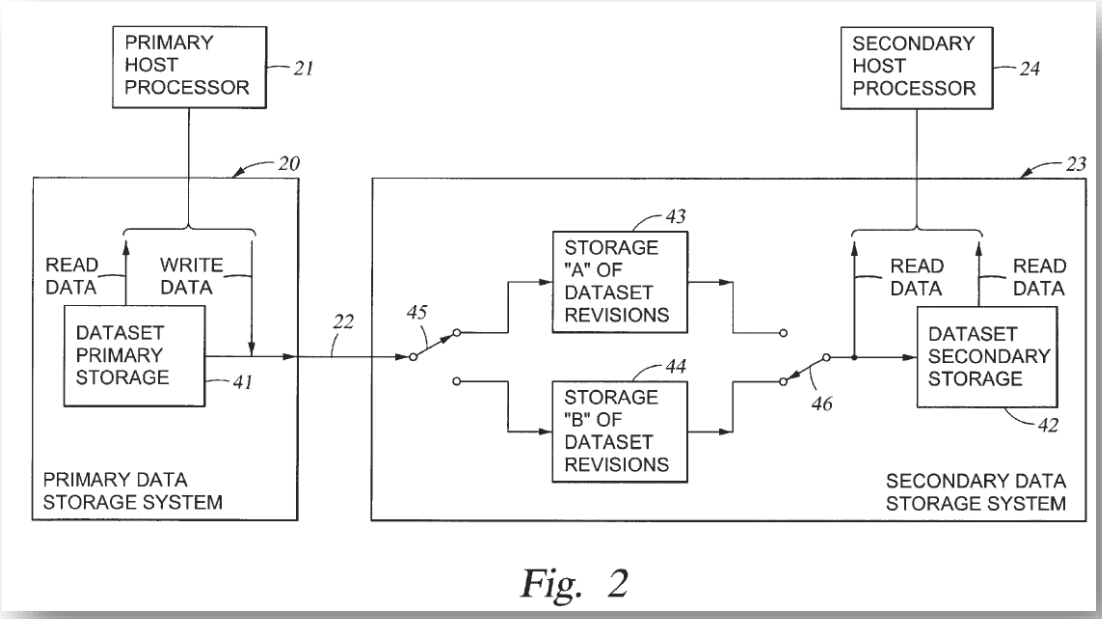


Fig. 2

Ex-1005, 5:56-57; Fig. 2. As shown in Figure 2, Srinivasan teaches two storages for separate dataset revisions (first storage "A" of dataset revisions 43, second storage "B" of dataset revisions 44), with switches 45, 46 directing the flow of data to and from each dataset. Ex-1005, 6:6-22. Srinivasan discloses that delta volumes are

broken down into delta chunks with each chunk including a header or trailer with information like a sequence number or delta set size (shown in Figure 10):

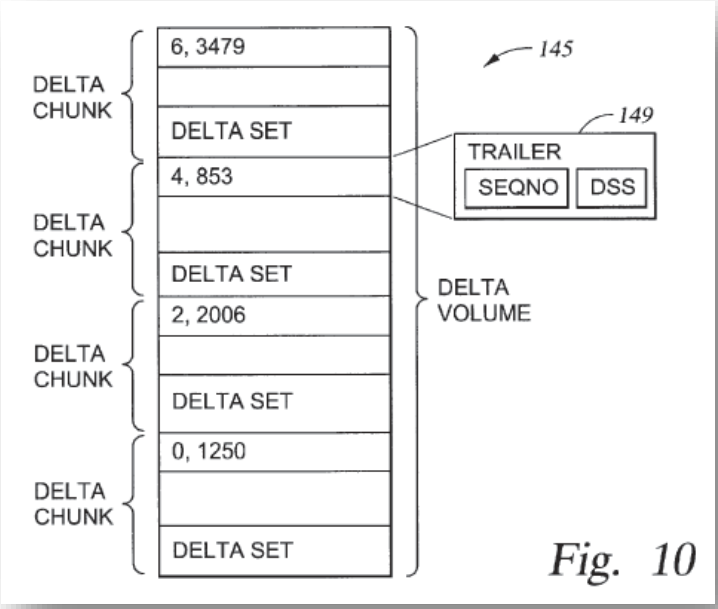


Fig. 10

Ex-1005, 11:22-33; Fig. 10. Figure 10 shows a block diagram of the delta volume in the data processing system depicted in Figure 8. Ex-1005, 4:54-55. Srinivasan explains that "[t]he delta volume therefore functions as a transaction log for updates to the file system, and also as a buffer for transmitting the updates between the primary data storage system and the secondary data storage system," Ex-1005, 11:33-36, and "the delta volume could use a conventional transaction log data structure," Ex-1005, 11:55-56. Further, Srinivasan teaches that "the storage dataset revisions could be implemented as a sequential transactional log of the file system modifications on the primary site, with sufficient information stored in the log, such

as inode numbers and old values and new values, to allow the secondary concurrent access facility to 'replay' the transactions into the 'live' file system in the file system secondary storage." Ex-1005, 16:64-17:3.

Srinivasan teaches that "[t]o reduce the computational overhead for accessing such a conventional transaction log, the delta volume could also include a delta set directory overlaid upon the conventional transaction data log structure" (Figure 11):

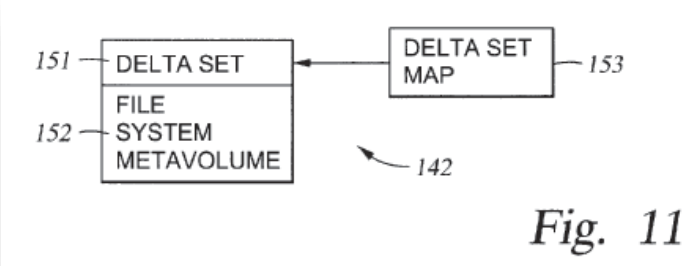


Fig. 11

Ex-1005, 11:57-61; Fig. 11, 4:52-58 (explaining Figure 11 is a block diagram of data structures in the secondary storage of the secondary data storage system in Figure 8).

B. Taylor (Ex-1006)

U.S. Pat. No. 6,658,589 to Taylor ("Taylor") was filed on December 20, 1999, and issued on December 2, 2003, and is therefore prior art under at least pre-AIA 35 U.S.C. § 102(e).

Taylor teaches "[a] system and method for safe and effective backup and restore of parallel server databases stored in data storage systems." Ex-1006, Abstract. As shown in Figure 3 below, a parallel server database architecture

generally includes instances (30a, 30b) that can independently access a database 40, archive logs 34 and online redo logs 36:

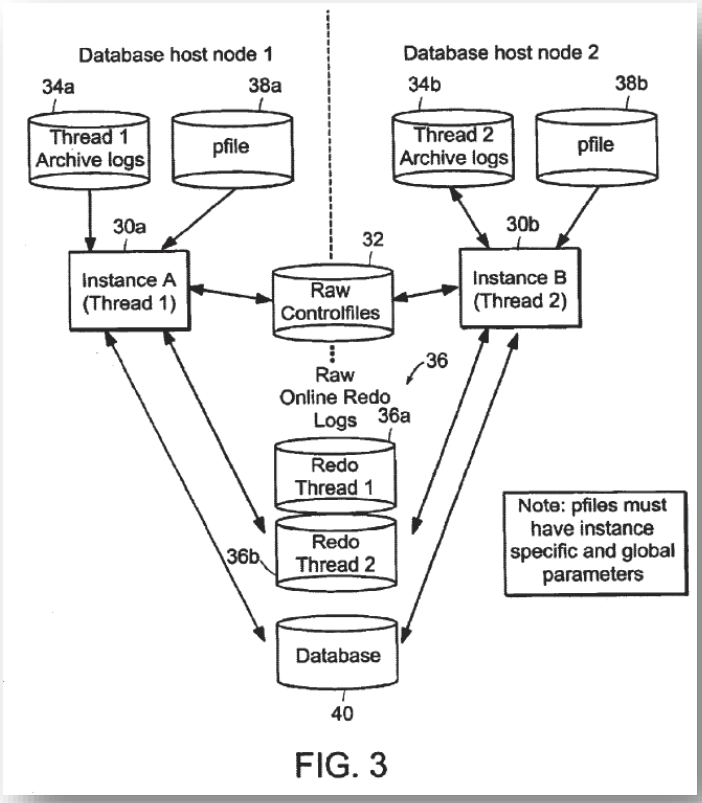


FIG. 3

Ex-1006, 5:27-44; FIG. 3.

The "log usually consists of at least two files; one is optionally being spooled to disk (archived) 34 while the other is being written (online) 36." Ex-1006, 3:3-5. The "[o]nline redo logs 36 are redo logs that have not been archived," and "may be available to the instance 30 for recording activity, or have previously been written but are awaiting archiving." Ex-1006, 3:5-8. Taylor explains all data files and online redo logs 36 reside in raw partitions and are visible to all parallel server database

nodes. Ex-1006, 3:27-32. In its background of the prior art, Taylor explains that online redo logs are generally not backed up, whereas all archive logs 34 and backup control file 32 information is backed up and exist as files in a file system. Ex-1006, 3:17-24. Overcoming the deficiencies of the prior art, Taylor teaches a parallel server database architecture that includes instances (30a, 30b) that can independently access a database, control files 32, archive logs (34a, 34b), directories to archive logs 34a', 34b', online redo logs 36, and parameter or configuration files (38a, 38b) (shown in FIG. 4):

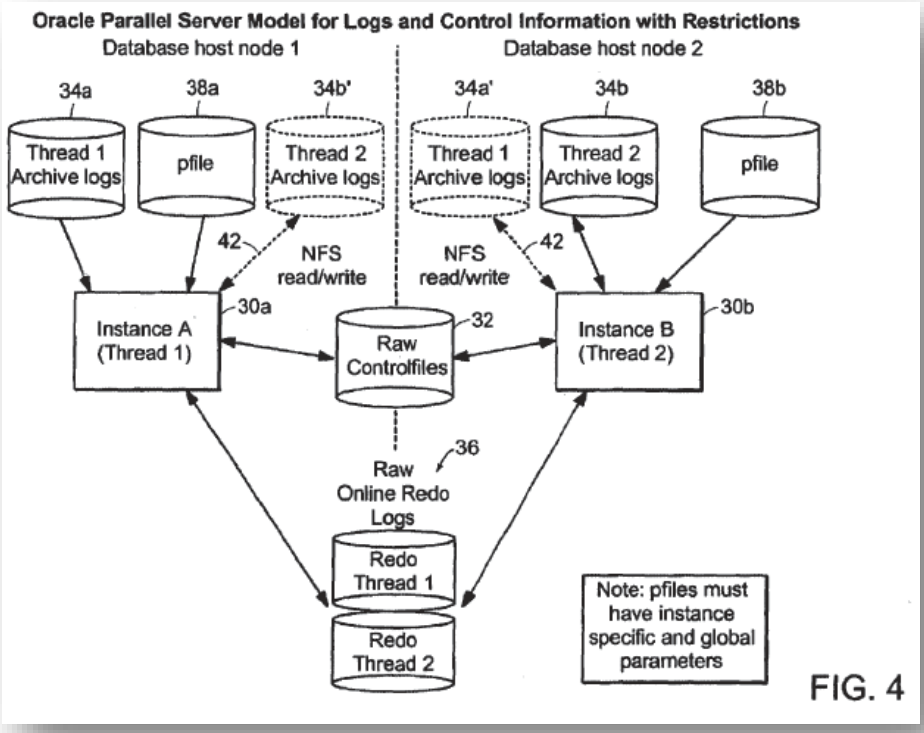


FIG. 4

Ex-1006, 3:3-11, 5:27-28, 44-46; FIG. 4 (annotated). Taylor also explains advantages of the invention over the prior art like allowing online or offline backup

and restore mechanisms for parallel server databases, which is effective for redo archive logs as well as database information and allows for single node discovery, preparation, and release of a multiple node database. Ex-1006, 4:31-36. Another advantage is allowing files to be backed up from any node in a parallel database environment where each node can see all the database and control files. Ex-1006, 4:37-42.

VII. ALL CLAIMS ARE UNPATENTABLE

All Challenged Claims are unpatentable as obvious under pre-AIA 35 U.S.C. § 103 over Srinivasan in view of Taylor.

As explained below, the scope and content of Srinivasan alone or in combination with Taylor includes every element of claims 1-18 without any meaningful difference between the teachings of those references and the elements of the asserted claims. After considering the teachings in the prior art from the perspective of a POSITA, it would have been obvious to combine the teachings of those known elements in the way they are recited in claims 1-18 as of January 31, 2001. Ex-1003, ¶¶ 54-57. Patent Owners have identified no secondary considerations purporting to show claims 1-18 would not have been obvious as of January 31, 2001.

A. Claims 1-18 are Obvious over Srinivasan in view of Taylor

For the reasons below, Srinivasan and Taylor would have rendered obvious claims 1-18. Ex-1003, ¶¶ 54-57, 74-210.

The "*online redo logs*" (also called "*active transaction redo logs*") of the '139 Patent are kept by each node of a parallel database server and contain information needed to perform database recovery by recording every change made to the database structure. Ex-1001, 4:45-52. Similarly, the "*archive redo logs*" of the '139 Patent are archived, or stored, versions of the online redo logs once the online redo logs are filled. Ex-1001, 4:52-55. Ex-1003, ¶ 75

It would have been obvious to organize the transaction logs of Srinivasan into "online" redo logs, and "archive" redo logs, as taught by Taylor. Ex-1003, ¶ 76. The Srinivasan system teaches that delta volumes are mirrored between a primary data storage system and a secondary data storage system. Ex-1005, 4:45-50. Srinivasan expressly teaches that the delta volume "could use a conventional transaction log data structure." Ex-1005, 11:54-57. Srinivasan explains that the delta volume functions as both a transaction log for updates and as a buffer for transmitting the updates to the secondary storage system. Ex-1005, 11:33-36. Srinivasan describes "live" or online (active) transactions being transmitted to the secondary storage system before those transactions are committed to storage. Ex-1005, 5:48-52 (describing transaction commit commands originating from the primary host

processor 21 and being sent over link 22 to the secondary storage controller 27 for execution), 16:64-17:3 (explaining "the storage of dataset revisions could be implemented as a sequential transactional log of the file system modifications on the primary side, with sufficient information stored in the log … to allow the secondary concurrent access facility to 'replay' the transactions into the 'live' file system in the file system secondary storage"). Ex-1003, ¶ 76.

A POSITA would have recognized that organizing Srinivasan's transaction logs into "*online redo logs*" and "*archive redo logs*," as taught by Taylor, would merely involve tracking which transactions have already been committed, or stored, and which transactions have not, as Srinivasan already teaches. Ex-1003, ¶ 77; Ex-1005, 6:57-7:11 (explaining sending transaction commit commands, which can be encoded in the write commands send over the link, from the primary data storage system 20 to the secondary data storage system 23), 13:49-53 (explaining the secondary data storage system should have received all updates in the delta set of the delta chunk before receiving the transaction commit command). And like Srinivasan, Taylor teaches that "*online redo logs*" are redo logs (sequential log of actions to be reapplied to database if not written to disk) that have not been archived and may be available to an instance for recording activity, whereas "*archive redo logs*" are redo logs that have been archived by being spooled to disk. Ex-1003, ¶ 77; Ex-1006, 2:67-3:8.

A POSITA would thus have been motivated to structure Srinivasan's transaction logs into "*online redo logs*" and "*archive redo logs*," as taught by Taylor. Srinivasan recognizes the need to balance between computational overhead and storage requirements, and that using transaction logs would minimize storage requirements. Ex-1003, ¶ 78. Ex-1005, 11:48-61. Using Taylor's "*archive redo logs*" in Srinivasan would allow certain critical transactions to be maintained in storage at the primary and/or backup sites for backup and restoration purposes; and using Taylor's "*online redo logs*" in Srinivasan would further reduce storage requirements because, as Taylor explains, the *online redo logs* are redo logs that have not yet been archived or written to disk. Ex-1003, ¶ 78; Ex-1006, 3-11.

Both Srinivasan and Taylor are analogous art to the '139 Patent for at least two reasons. First, both Srinivasan and Taylor are directed toward the same field of endeavor of the '139 Patent—data storage systems for transaction processing. Ex-1001, 1:6-10, Ex-1005, 1:11-15, Ex-1006, 1:8-10. Second, both Srinivasan and Taylor are reasonably pertinent to a problem described by the '139 Patent—archiving all transactions at a primary site and creating network congestion by transmitting the entire archive logs to the standby site. Ex-1003, ¶ 79; Ex-1001, 2:5-7, Ex-1005, 6:51-64, 10:62-11:2, Ex-1006, 4:38-42.

In the combined teachings of Srinivasan and Taylor, a primary storage controller as taught in Srinivasan receives read and write commands to change the

file system from a primary host processor, and those read and write commands are organized into "*online redo logs*" and stored in cache memory. Ex-1003, ¶ 80. Ex-1005, 3:43-57 (describing storage controller receiving read and write commands, commit commands for data storage volumes, and remote mirroring facilities for volumes), 3:51-64 (explaining updating process for secondary data storage involving storage controllers and a primary host processor) 10:6-29 (describing remote mirroring facility interfacing cache memory to links connecting the primary data storage system to the secondary data storage system) 11:18-36 (explaining initialization and process of capturing changes to the file system in delta sets). Taylor recognizes the differences between "*online redo logs*" and "*archive redo logs*." Ex-1006, 3:3-5 (explaining "*archive redo logs*" have been stored on disk, whereas "*online redo logs*" are currently being written to and have not been stored on disk), 3:27-32 (explaining how data files in parallel server database configurations reside on raw partitions for visibility to all parallel server database nodes). Recognizing the differences taught by Taylor, the combined teachings of Srinivasan and Taylor would compare changes (captured in *"online redo logs*") to the transaction logs already written to disk ("*archive logs*") in the file system primary storage ("*computing a difference between an active transaction redo log and an archive transaction redo log*") to produce a delta volume ("*delta image*") that is stored in a delta volume storage. Ex-1003, ¶ 80.

Further, online redo logs and archive redo logs were well-known and common types of transaction logs or log files as of January 31, 2001, the filing date of the '139 Patent. Ex-1003, ¶ 81; *see, e.g.,* Ex-1007, 534-538 (explaining database transactions, including undo and redo, usually recorded in a system log that is "periodically backed up to archival storage (tape) to guard against such catastrophic failures"), 577-592 (describing redo and undo recovery algorithms as applied to system logs for recovery operations). Thus, based on the teachings of Srinivasan and Taylor, and what was well known in the art at the time, a POSITA would readily understand that known technique of creating a "delta" representing a difference in data over time could be applied to transaction logs, such as redo logs, to achieve predictable results. Ex-1003, ¶ 81. Further, a POSITA would have been motivated to compute a "delta" of redo logs based on the known benefit of reduced latency when smaller quantities of data are transmitted over a network. Ex-1003, ¶ 81; Ex-1007, 71-76 (describing latency associated with the storage type), 466-469 (explaining criteria for physical database design, including "Response time" and "Transaction throughput").

Additionally, a POSITA would have found it obvious to apply the known technique of creating a "delta" to redo logs as one of a finite number of predictable solutions with a reasonable expectation of success. Ex-1003, ¶ 82. For example, database systems commonly used transaction logs (e.g., redo logs) and data for

database replication and storage for disaster recovery before the application underlying the '139 Patent was filed. Ex-1003, ¶ 82; *see, e.g.,* Ex-1007, at 534-538, 577-592. It was already well known to create "deltas" or "snapshots" of data for transmission of the updated data to reduce latency on data transmitted to the standby site, particularly if the data was outdated and needed to be overwritten. Ex-1003, ¶ 82; Ex-1007, 792. Further, it was already well-known that transaction logs could be periodically backed up, such as by sending the logs from a primary database to a standby database after each database transaction. Ex-1003, ¶ 82; *see, e.g.,* Ex-1007, 527-553, 577-592. Thus, a POSITA would have recognized that same technique could be applied to create "deltas" of transaction logs, such as redo logs, to similarly reduce the effects of latency during transmission from the primary database to the standby database. Ex-1003, ¶ 82.

In the combined teachings of Srinivasan and Taylor, the data processing system transmits transaction logs (such as "*online redo logs*" and optionally "*archive redo logs*") from the primary data storage system to the secondary data storage system upon initialization ("*transmitting the active transaction redo log*"). Ex-1003, ¶ 83. Ex-1005, 11:18-36 (explaining initialization and process of capturing changes to the file system in delta sets). After initialization is complete, the primary data processing system captures changes in delta sets or delta volumes ("*delta image*") in delta volume storage, as described above; and the delta volumes ("*delta images*")

are remotely mirrored from the primary data storage system to the secondary data storage system ("*transmitting the active transaction redo log and the delta image, separately in time sequence*"). Ex-1003, ¶ 83.

It would have been obvious for the combined teachings of Srinivasan and Taylor to transmit the "*online redo logs*" (also called "*active transaction redo logs*") to the secondary data storage system for several reasons. Ex-1003, ¶ 84.

First, Srinivasan teaches how to avoid issues with data coherence. Ex-1003, ¶ 85; Ex-1005, 1:44-48 (explaining that concurrent database access by multiple host processors typically involves only one host processor with read-write access and the other host processors with read-only access), 6:51-55 (describing advantage of concurrent access facility providing secondary host processor with substantially uninterrupted and concurrent read-only access to a consistent dataset regardless of the rate at which the dataset secondary storage is updated), 6:64-66 (explaining transaction commit commands for writing to disk (archiving) can be encoded in write commands sent over the link), 11:18-22 (explaining that during initialization of the data processing system, file system access by the primary host processor is temporarily suspended while files are copied to the secondary data storage system). Thus, the technique of Srinivasan directly addresses the cache coherency problems discussed in Taylor related to transmitting "*online redo logs*" in parallel databases. Ex-1003, ¶ 85; Ex-1005, 2:46-50 (describing data coherence as a problem with

parallel databases since several nodes are accessing and writing to the same database), 2:58-65 (explaining "typical parallel server database applications" like those from Oracle use control files (paired with the database not server instance) to maintain database coherence and availability), 6:10-13 (describing how "the present invention will work for any of various parallel server databases running on any of various data storage systems."). Thus, a POSITA would have been motivated to use technique of Srinivasan to transmit "*online redo logs*" as taught by Taylor to overcome the problem identified by Taylor. Ex-1003, ¶ 85. And a POSITA would have reasonably expected to succeed in using the "*online redo logs*" of Taylor within the system and technique of Srinivasan to maintain data coherency by limiting write access to a single controller during backup or mirroring operations. Ex-1003, ¶ 85.

Second, Taylor describes limitations with transmitting logs in its background section, but Taylor describes an advance over the conventional background art, emphasizing that (1) "*online redo logs*" should be visible to all server databases, (2) the importance of data availability of non-stop functioning of the database. Ex-1003, ¶ 86; Ex-1005, 1:17-21 (describing need for continuous availability and round-the-clock access to databases stored in the one or more data storage system), 4:37-42 (explaining advantage of a system where files can be backed up from any node in a parallel server database), 4:12-14 (explaining desirability of the recovery node to have the network file system (NFS) write ability to the other nodes), 5:34-36

(describing how "*online redo logs*" reside in raw partitions and are visible to all parallel server database nodes), 8:60-62 (explaining standard vendor-specific DBMS backup utility functionality to ensure that logs are backed up). Accordingly, a POSITA would have been motivated to backup the "*online redo logs*" based on their visibility to all nodes and the need for their availability for round-the-clock database functionality. Ex-1003, ¶ 86.

In the combined teachings of Srinivasan and Taylor, the delta volume ("*delta image*") discussed above, would be combined with the "*online redo log*" (also called "*active transaction redo log*") at the secondary data storage system ("*combining the delta image with the active transaction redo log*"). Ex-1003, ¶ 87. For example, Srinivasan teaches that the delta volume ("*delta image*") could be overlaid on a conventional transaction log structure. Ex-1003, ¶ 87; Ex-1005, 11:57-61 (describing delta volume including delta set directory overlaid on transaction log data structure). Additionally, Srinivasan teaches that the delta volume can be overlaid on top of a file system metavolume. Ex-1005, 12:8-34 (explaining that updates are copied into corresponding blocks of file system metavolume). Further, one of ordinary skill in the art would have been motivated to combine the delta volume ("*delta image*") with the "*online redo log*" of the combined teachings of Srinivasan and Taylor to reduce computational overhead, as taught by Srinivasan.

Ex-1003, ¶ 87; Ex-1005, 11:57-61 (explaining overlaying delta volume delta set directory on transaction log data structure reduces computational overhead).

The '139 Patent explains that if the archive log is complete, the standby archive file replicator applies the completed archive to the standby database server by creating a link to the file in the archive destination where the standby database is looking for files. Ex-1001, 6:17-21. The '139 Patent describes that a disk subsystem can be shared by nodes in a parallel database and the nodes can be connected by a high-speed local link. Ex-1001, 4:45-47. Ex-1003, ¶ 88

It would have been obvious to use the logical-to-physical mapping of Srinivasan to create links to "*standby archive redo logs*," as taught by Taylor. Ex-1003, ¶ 89. Srinivasan teaches that each data storage system can be directly linked to multiple host processors, and that the files can be organized by a logical-to-physical mapping for access by any processor. Ex-1003, ¶ 89; Ex-1005, 17:34-38 (explaining each data storage system can be directly linked to multiple host processors), 15:63-16:1 (explaining concurrent access facility that interfaces between the virtual file system and the physical file system), 16:1-13 (describing a UNIX-based file system having a hierarchical file system structure including directories and files, and each directory and file has an "inode" containing metadata of the directory or file).

Taylor similarly teaches that the archive logs can use logical-to-physical mapping and expressly teaches creating links to the archive logs, which can be accessed from any node of a parallel database. Ex-1003, ¶ 90; Ex-1006, 5:39-46 (describing archive logs are maintained in directories on the node where the instance is running and how archive log directories are set to be readable and writable by all nodes servicing the database), 5:46-50 (explaining log directories are accessible from different nodes via a link to a network file system (NFS) or equivalent, in read/write mode on any nodes from which backup or restore will be performed), 5:50-55 (describing how log directories are mounted using the same names everywhere they are mounted for consistent access via the directory).

A POSITA would have been motivated to modify the teachings of Srinivasan to include links to the standby archive redo logs, as taught by Taylor, because of the multiple advantages it would provide in the context of parallel databases. Ex-1003, ¶ 91; Ex-1006, 3:53-55 (explaining each node of a parallel database environment can see all database and control files, allowing the files to be backed up from any node), 4:31-37 (explaining advantage of online or offline backup and restore mechanisms for parallel server databases), 4:38-42 (describing advantage of allowing files to be backed up from any node).

Further, using links to apply the standby archive redo from the combined teachings of Srinivasan and Taylor would merely involve known logical-to-physical

mapping techniques for the archive redo logs and allowing access by multiple processors, as Srinivasan already teaches. Ex-1005, 15:63-16: 13, 17:34-38. Ex-1003, ¶ 92.

Finally, Srinivasan and Taylor are at the same level of generality as the '139 Patent such that a POSITA would have been able to apply that same level of skill in the art in understanding Srinivasan and Taylor and the ways in which their teachings could be implemented. Ex-1003, ¶ 93. Thus, a POSITA would have reasonably expected to succeed in combining the teachings of Srinivasan and the teachings of Taylor. Ex-1003, ¶ 93.

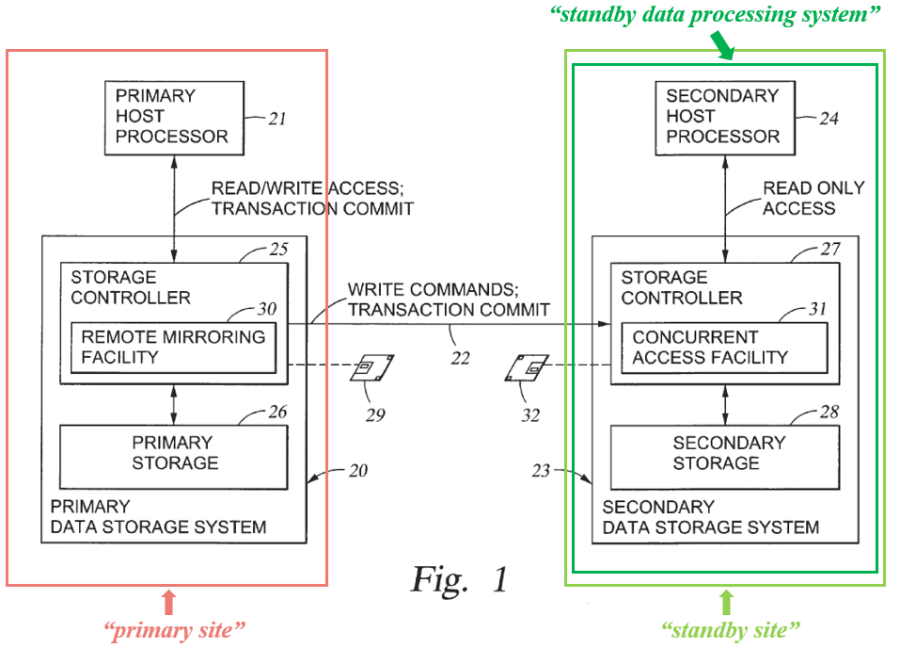
1. *Claim 1*

a. *[1preamble]: "A method for creating a standby data processing system located at a standby site, which is remotely located from a primary site, comprising the steps of:"*

The "standby data processing system" of the '139 Patent can include a standby database or a standby transaction processing system. Ex. 1001, 2:12-35 (using the terms "standby data processing system" and "standby database" interchangeably in the summary as performing the same functions), 4:25-35 (explaining a method used to create archive files on a standby system, which is also called a "standby transaction processing system"). Ex-1003, ¶ 95.

To the extent the preamble is limiting, Srinivasan teaches a data processing system that includes a "*standby data processing system*" including a secondary data

storage system 23 and secondary host processor 24 at a "*standby site*" remotely located from a "*primary site*" that has a primary data processing system, including a primary data storage system 20 and primary host processor 21 (shown in Fig. 1):



Ex-1005, Fig. 1 (annotated); Ex-1003, ¶ 96. A preferred construction for the data processing system of Figure 1 includes a "*standby data processing system*" with a secondary data storage system 113, a data mover computer 114, and a secondary host computer 115 at a "*standby site*" remotely located from a "*primary site*" that includes a primary data processing system with a primary data storage system 110, a data mover computer 111, a primary host processor 112 (shown in Figure 8):

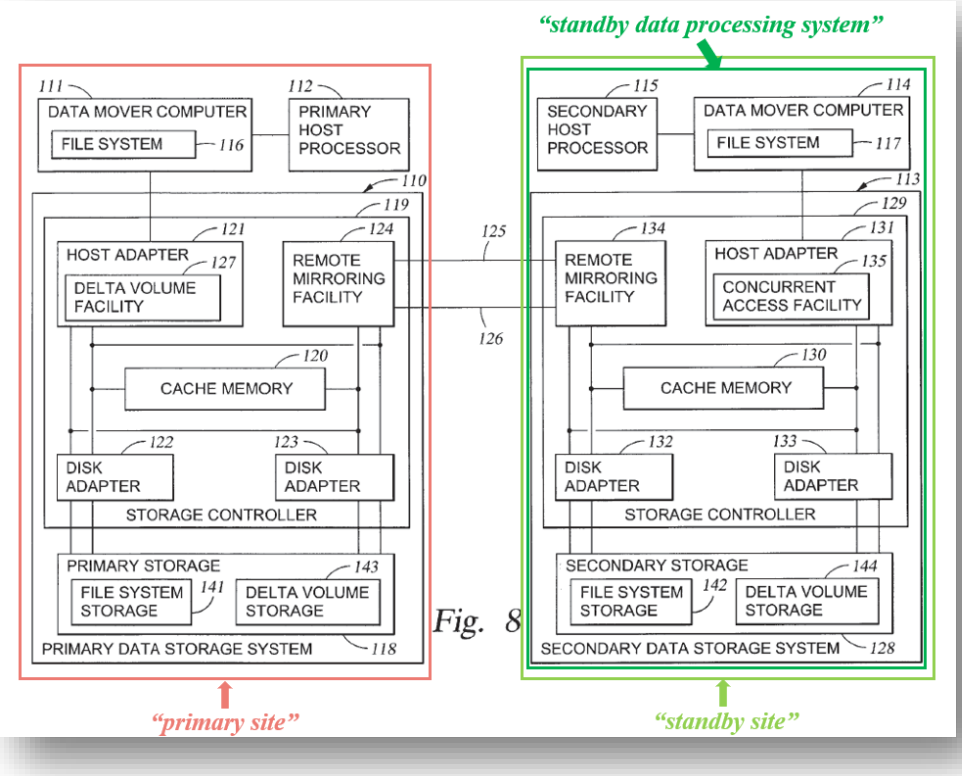


Fig. 8

Ex-1005, 9:55-58; Fig. 8 (annotated); Ex-1003, ¶ 97.

The secondary data storage system 113, the secondary data mover computer 114, and the secondary host processor 115 of Srinivasan are collectively a "*standby data processing system*," consistent with how that term is used in the '139 Patent, because they collectively store and process data at a "*standby site*." Ex-1003, ¶ 98; Ex-1005, 9:59-62 (explaining the data mover computer 111 includes a file system 116 that translates read and write commands from the primary host processor 112 to logical block read and write commands to the primary data storage system 110), 10:3-5 (explaining "the combination of the secondary data storage system 113 and the data mover computer 113 also functions as a file server").

The "*standby site*" of the '139 Patent can include a secondary or backup site that is remotely located from a primary site. Ex-1003, ¶ 99; Ex-1001, 1:39-45 (explaining standby site or remote site is desirable to protect against catastrophic failure if primary site is disabled due to power outage, fire, or flood), 1:60-65 (describing remotely mirroring of the active logs and archive logs from the primary site to the standby site), 2:5-7 (explaining data begin sent from the primary site across a wide area network (WAN) to the standby site can result in network congestion).

The "*standby data processing system*" described in Srinivasan is "*located at a standby site, which is remotely located from a primary site,*" consistent with how a "*standby site*" is used in the '139 Patent, because the write commands received at the "*primary site*" are remotely mirrored, via one or more transmission links to the "*standby data processing system*" at the "*standby site*." Ex-1003, ¶ 100; Ex-1005, 5:16-35 (explaining remote mirroring facilities for transmitting commands from the primary host processor 21 to the secondary storage controller 25), 10:6-29 (explaining remote mirroring of delta volume storage 143 of the primary storage 118 over dual redundant links 125, 126 at the "*primary site*" to the secondary storage 128 in the secondary data storage system 113 at the "*standby site*").

As Srinivasan explains, the disclosed system transmits and stores a backup copy of data at the "*standby data processing system*" ("*method for creating a*

standby data processing system"). Ex-1003, ¶ 101; Ex-1005, 5:56-6:22 (explaining a dataset at the "*secondary site*," like "a set of volumes, single volume, file system, set or files, or single file" is copied from the dataset at the "*primary site*" and transmitted from the primary host processor at the "*primary site*" over a link to the secondary data storage system), 10:58-11 (describing remotely mirroring an "A" delta volumes 145 and a "B" delta volume 146 from the primary data storage system 110 to the secondary data storage system 113).

The "*method for creating a standby data processing system*" described in Srinivasan is shown in annotated Figures 2 and Figure 9:

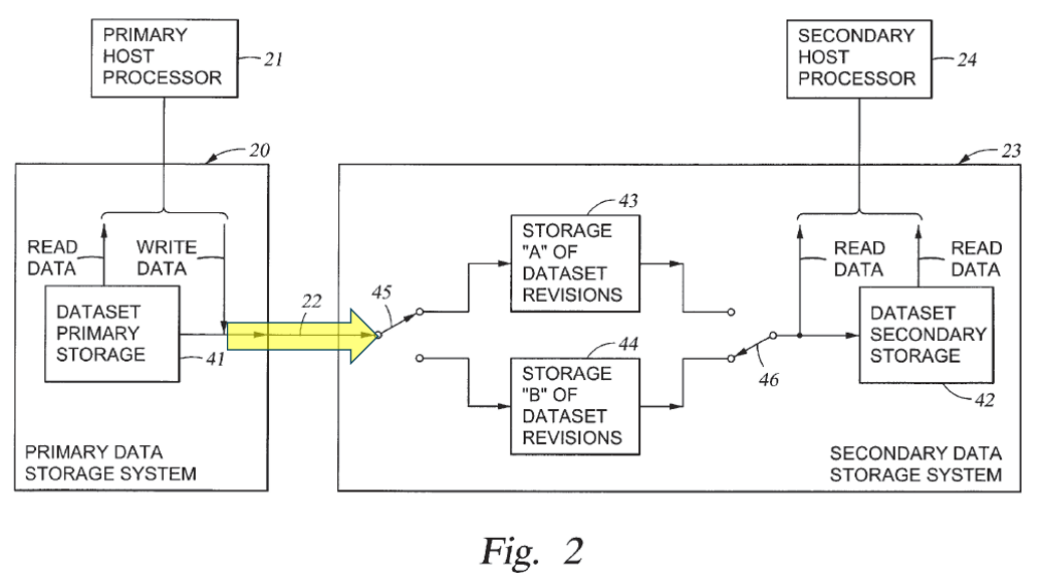


Fig. 2

Ex-1005, Fig. 2 (annotated).

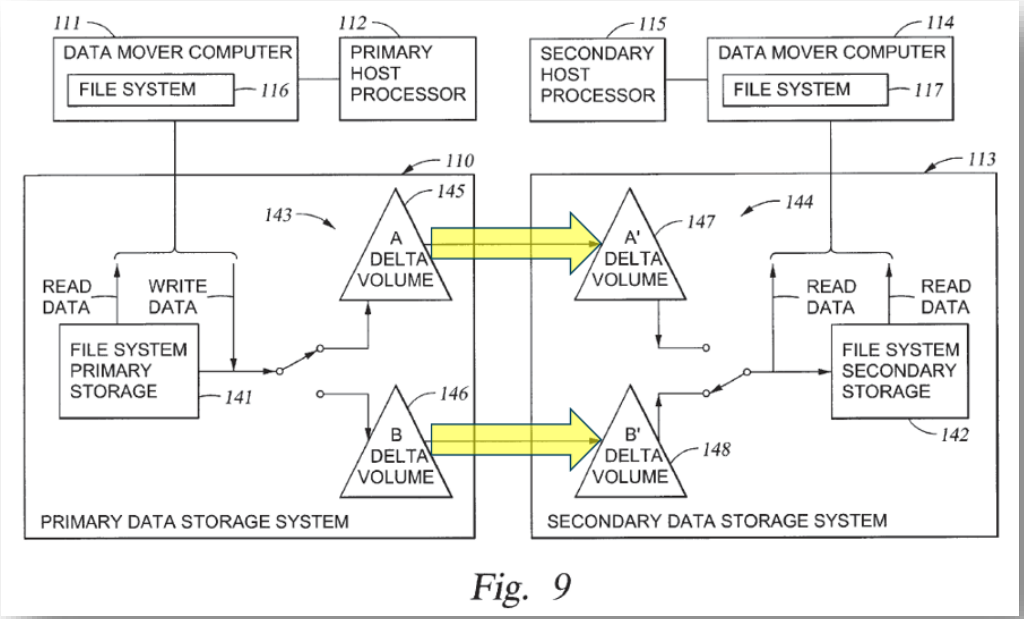


Fig. 9

Ex-1005, Fig. 9 (annotated), 4:45-53 (explaining Figure 9 shows data flow in the data processing system of Figure 8); Ex-1003, ¶ 102.

Accordingly, Srinivasan taught or suggested the preamble to the extent it is limiting. Ex-1003, ¶¶ 94-103.

b. *[1a]: "(a) computing a difference between an active transaction redo log and an archive transaction redo log to produce a delta image at the primary site;"*

This element would have been obvious based on the combined teachings of Srinivasan and Taylor. Ex-1003, ¶ 104. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

As previously discussed, it would have been obvious to organize the transaction logs of Srinivasan into online redo logs, and archive redo logs, as taught by Taylor. *See* §VII.A.

The '139 Patent distinguishes between an active transaction log that tracks recent transactions, and an archive log that is kept to store information from the active log after the active log has been filled or a certain time period has passed. Ex-1003, ¶ 106; Ex-1001, 1:56-60; *compare* Ex-1001, 4:47-49 (explaining "each node will have its own copy of the online redo logs 58, 60 (i.e., an active data file)"), *with* Ex-1001, 5:53-55 (describing when the "online redo logs are filled they are then archived and transferred into the archive log areas 62, 64 (i.e., archive data file)").

The online redo logs of the combined teachings of Srinivasan and Taylor are "*active transaction redo logs*," and the archive redo logs of the combined teachings of Srinivasan and Taylor are "*archive transaction redo logs*," consistent with how those terms are used in the '139 Patent. Taylor teaches that online redo logs ("*active transaction redo logs*") are redo logs (sequential logs of actions to be reapplied to database if not written to disk) that have not been archived and may be available to an instance for recording activity, whereas archive redo logs ("*archive transaction redo logs*") are redo logs that have been archived by being spooled to disk. Ex-1006, 2:67-3:8; Ex-1003, ¶ 107.

Srinivasan teaches that the primary storage controller 119, the primary host processor 112, identify and record file system changes (updates) in a delta set or delta volume ("*delta image*"), which is captured in the delta volume facility 127 and loaded to the delta volume storage 143 at the "*primary site*" where the primary data processing system is located. Ex-1003, ¶ 108; Ex-1005, 3:43-57 (explaining primary storage controller 119 receives read and write commands (for specified data in storage) from primary host processor 112), 10:22-24 (explaining "host adapter 121 is programmed with a 'delta volume facility' 127 that loads the updates into the delta volume storage 143 of the primary storage 118"), 11:18-26 (explaining initialization process and then using delta sets to capture changes to the file system).

As previously discussed, it would have been obvious to organize the read and write commands in online redo logs ("*active transaction redo logs*") stored in cache memory 120 of the storage controller 119. *See* §VII.A. Additionally, Taylor teaches that the online redo logs ("*active transaction redo logs*") contain real-time changes (updates) to the system that have not yet been written to disk, unlike the archive redo logs ("*archive transaction redo logs*"). Ex-1006, 2:67-3:8. Recognizing these differences taught by Taylor, the combined teachings of Srinivasan and Taylor system would include creating a delta volume ("*delta image*") containing changes (updates) to the system at the "*primary site*" by comparing the differences between online redo logs ("*active transaction redo logs*") and archive redo logs ("*archive*

transaction redo logs"). *See* §VII.A. And it would have been obvious to do so for the reasons previously discussed. Ex-1003, ¶ 109.

The primary data processing system at the "*primary site*," including the primary storage controller 119 and the primary host processor 112 for computing the differences, are shown in Figure 8 of Srinivasan, which is a preferred construction of Figure 1:

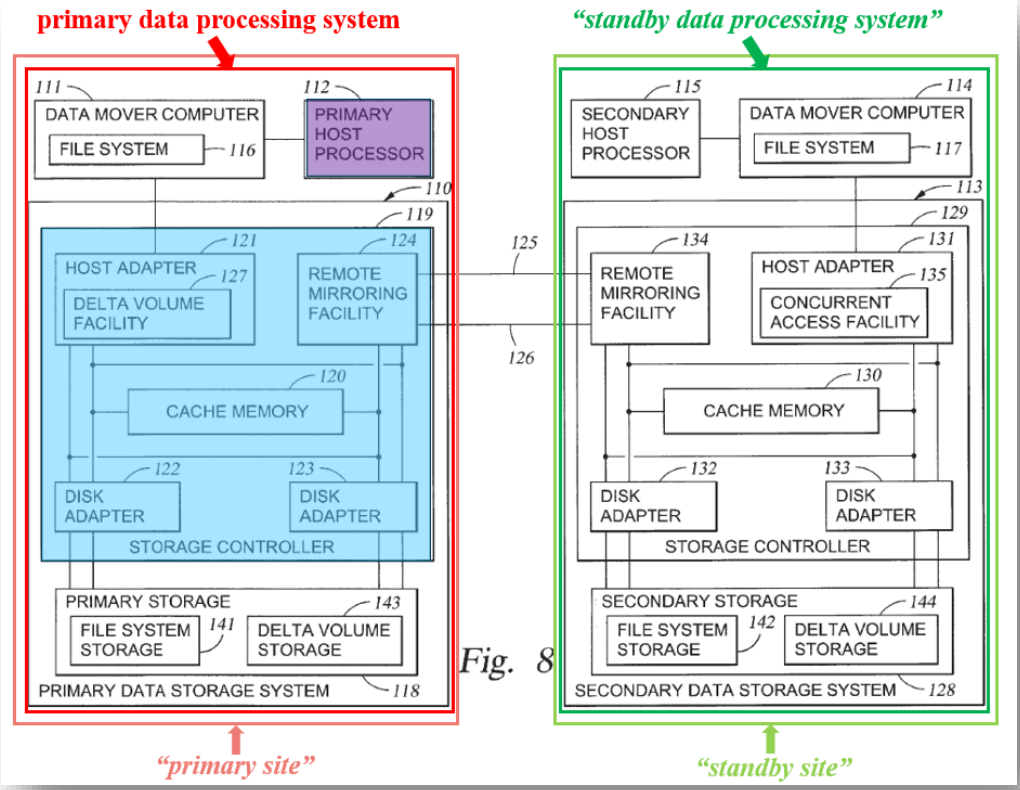
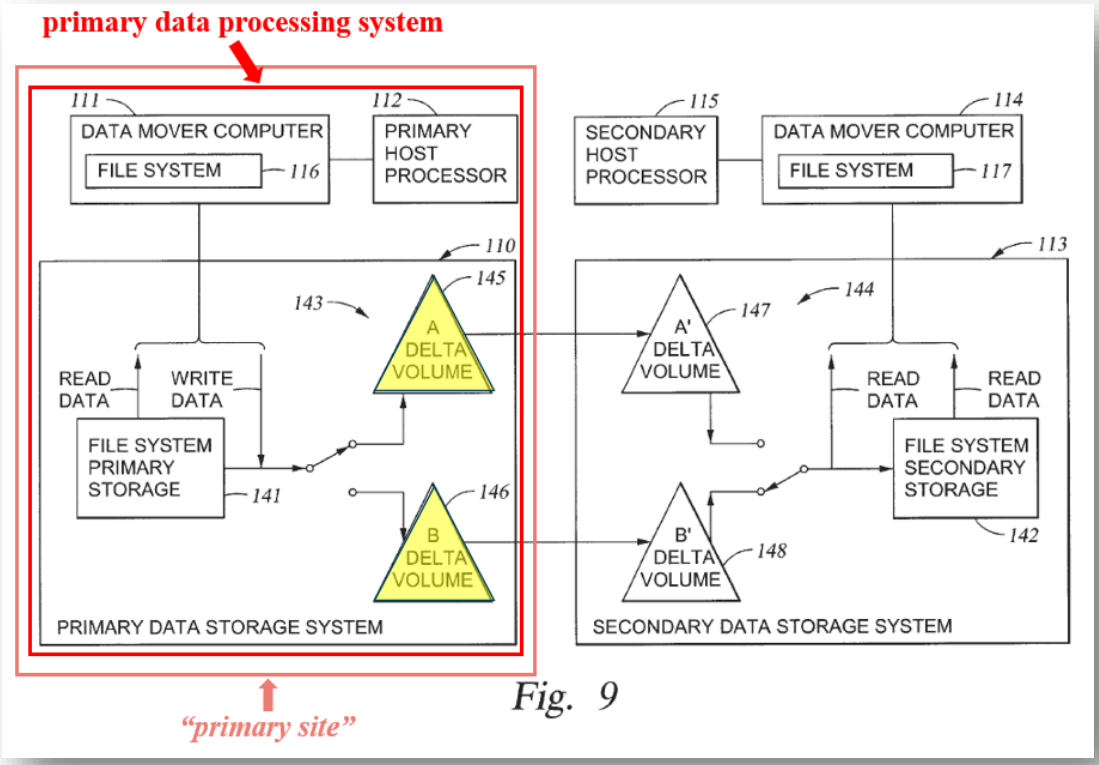


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶ 110. Additionally, Figure 9 of Srinivasan shows the flow of the delta volumes ("*delta images*") of the data processing system of Figure 8:



Ex-1005, Fig. 9 (annotated); Ex-1003, ¶ 111.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this claim element. Ex-1003, ¶¶ 104-112.

c. *[1b]: "(b) transmitting the active transaction redo log and the delta image, separately in time sequence, from the primary site to the standby site; and"*

This element would have been obvious based on the combined teachings of Srinivasan and Taylor. Ex-1003, ¶ 113. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

Srinivasan teaches that the datasets (e.g., set of volumes, file systems, file sets) of the primary data processing system and the "*standby data processing system*" are the same upon system initialization. Ex-1005, 5:57-66 (explaining each of the datasets are empty, or identical because they are loaded from the same external source, or the primary dataset is copied to the standby system), 11:18-26 (describing system initialization process and transmitting the file system primary storage 141 to the file system secondary storage 142). As previously discussed, during the initialization, all online redo logs "*active transaction redo logs*" and archive redo logs ("*archive transaction redo logs*") would be transmitted to the "*standby data processing system*" at the "*standby site*." *See* §VII.A; Ex-1003, ¶ 114.

Srinivasan explains that after initialization, write operations are written to the dataset, such as a delta volume ("*delta image*"), in the primary data storage system and then transmitted to the secondary data storage system. Ex-1005, 5:66-6:6 (explaining primary host processor 21 writes data to dataset 41, which is then forwarded over the link 22 to the secondary data storage system 23), 10:19-29 (describing that updates are loaded from delta volume facility 127 to delta volume storage 143 and then remotely mirrored to transmit the updates over redundant links 125, 126 to mirrored delta volume storage 144 in the secondary data storage system 113), 10:62-11:2 (explaining that "A" delta volume 145 and "B" delta volumes containing wrote data sets of sequential transactions are mirrored from the primary

data storage system 110 to the secondary data storage system 113 as "A'" and "B'" delta volumes 147, 148). For at least these reasons, and those discussed previously, the combined teachings of Srinivasan and Taylor would transmit "*active transaction redo logs*" and "*delta images, separately in time sequence, from the primary site to the standby site.*" Ex-1003, ¶ 115.

The transmission paths from the "*primary site*" to the "*standby site*" are shown below in Figures 8 and 9 of Srinivasan:

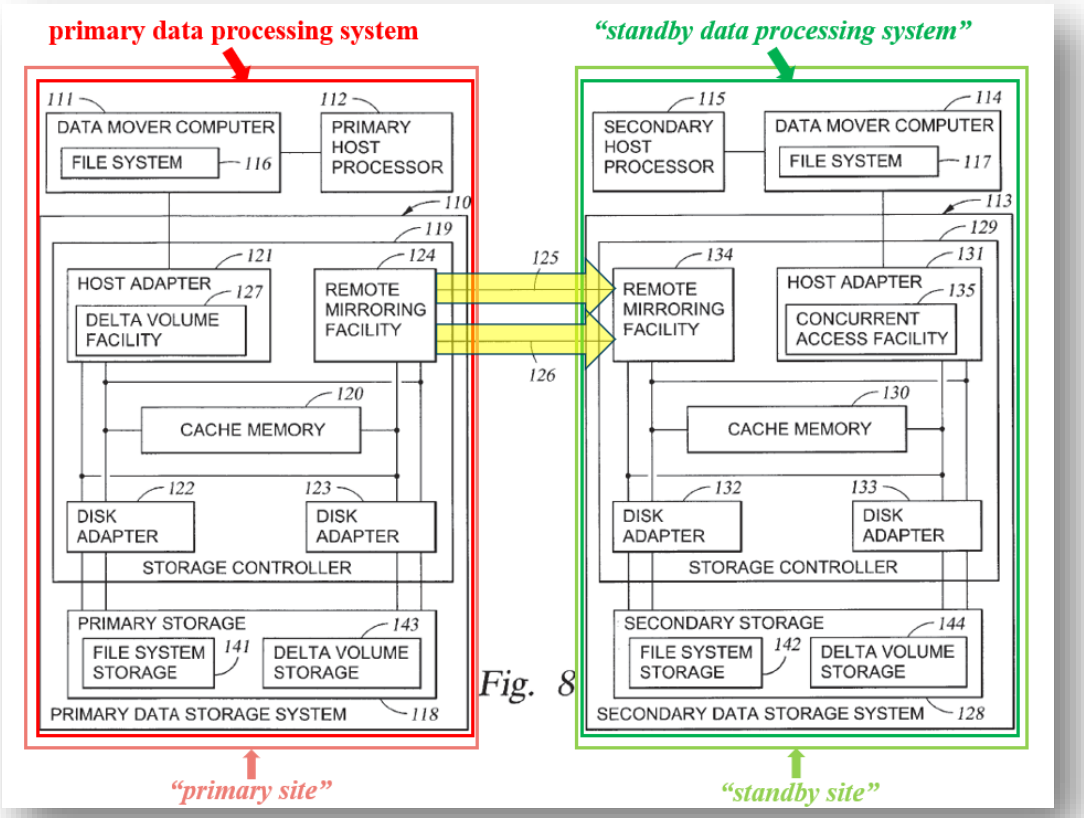


Fig. 8

Ex-1005, Fig. 8 (annotated).

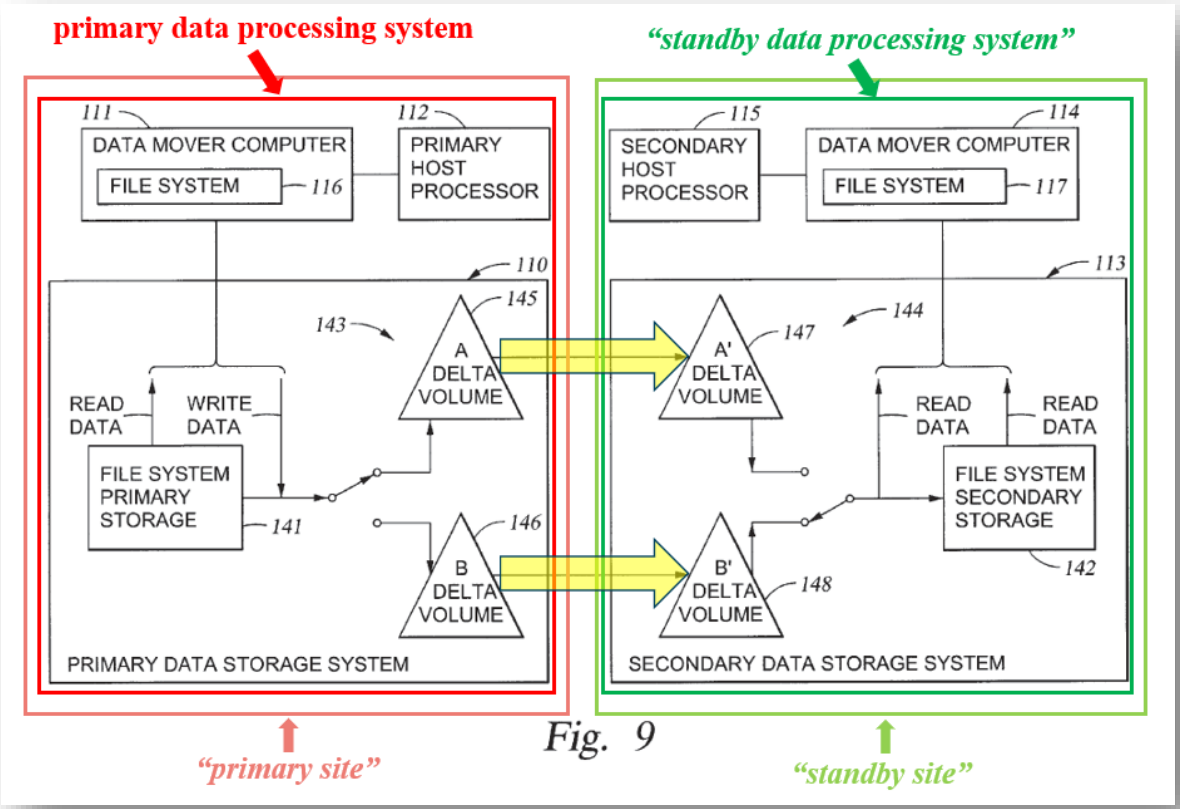


Fig. 9

Ex-1005, Fig. 9 (annotated); Ex-1003, ¶ 116.

Further, Srinivasan also teaches that delta volumes ("*delta images*") are broken down into delta sets or chunks identified by sequence number and delta set size that can be written to a delta chunk header or trailer 149. Ex-1005, 11:12-36 (explaining delta volumes are logically divided into delta chunks depending on various factors such as capacity and write activity of *primary site*, and explaining the size is large enough to contain all updates for any single transaction). Figure 10 of Srinivasan is a diagram of a delta volume in the data processing system of Figure 8:

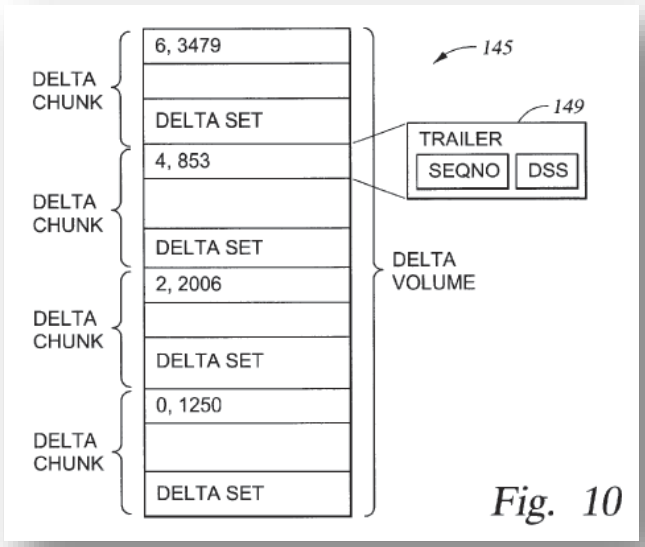


Fig. 10

Ex-1005, Fig. 10; Ex-1003, ¶ 117. Srinivasan also explains that the remote mirroring facilities can operate in asynchronous or semi-synchronous mode. Ex-1005, 13:49-60. And Srinivasan teaches that delta volumes containing updates can be sent on an ongoing basis to maintain consistency between the primary and standby systems. Ex-1003, ¶ 118; Ex-1005, 1:24-26 (explaining each transaction includes a set of read-write instructions that change the database from one consistent state to another), 2:26-32 (describing changes made to the local database could be recorded in the remote database on an ongoing basis using snapshots), 11:33-36 (explaining delta volumes function as buffers for transmitting updates between the primary data storage system and secondary data storage system). Thus, a POSITA would have reasonably expected to succeed in transmitting the online redo logs "*active transaction redo logs*" separately in time sequence from the delta volumes "*delta images*" of the

combined teachings of Srinivasan and Taylor because Srinivasan already contemplates asynchronous mirroring and sending delta volumes (e.g., "A" delta volume 145 and "B" delta volume 146) at separate times to the "*standby site*." Ex-1003, ¶ 119.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this claim element. Ex-1003, ¶¶ 113-120.

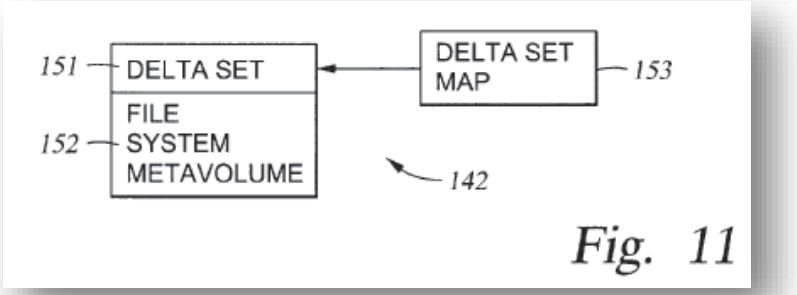
a. *[1c]: "(c) combining the delta image with the active transaction redo log at the standby site, to produce a standby archive transaction redo log."*

This element would have been obvious based on combining the teachings of Srinivasan and Taylor. Ex-1003, ¶ 121. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

Srinivasan teaches a delta volume ("*delta image*") could be overlaid on a conventional transaction log structure. Ex-1005, 11:57-61 (describing delta volume including delta set directory overlaid on transaction log data structure to reduce computational overhead). Additionally, Srinivasan teaches that the delta volume can be overlaid on top of a file system metavolume. Ex-1005, 12:8-34 (explaining that updates are copied into corresponding blocks of file system metavolume 152). For these reasons and those previously discussed, the combined teachings of Srinivasan and Taylor would have involved combining the delta volume ("*delta image*") with

the online redo logs ("*active transaction redo logs*") at the "*standby data processing system*" located at the "*standby site*" to produce archive logs ("*standby archive transaction redo logs*"). Ex-1003, ¶ 122.

Figure 11 of Srinivasan shows combined data structures at the "*standby site*":



Ex-1005, Fig. 11; Ex-1003, ¶ 123.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this claim element. Ex-1003, ¶¶ 121-124.

2. *Claim 2: "A method as in claim 1, further comprising the step of storing the standby archive transaction redo log on the standby site."*

As discussed above for claim 1, the combined teachings of Srinivasan and Taylor would have rendered obvious producing an archive log "*standby archive transaction redo log*" on the "*standby site*." Ex-1003, ¶ 125. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

Srinivasan teaches multiple techniques for storing transactions in the standby data storage system that are consistent with the primary data storage system. Ex-1005, 6:64-66 (explaining transaction commit commands can be encoded in write commands sent over the link), 5:48-55 (describing transaction commit commands originate from the primary host processor 21, and storage controller 25 forwards at least some transaction commit commands over the link 22 to storage controller 27), 8:59-9:7 (explaining switches 45 and 46 toggle between storage "A" dataset revisions 43 and storage "B" dataset revisions 44 where one is write-selected and one is read-selected), 10:66-11:2 (describing remote mirroring facility transferring write data to mirrored "A" and "B" delta volumes 147, 148), 5:29-33 (explaining storage controller 27 at the secondary storage system 28 receives write commands and executes them to maintain a copy of data (from the primary storage system 20) in the secondary storage system 28), 9:8-11 (describing that the secondary data storage system is programmed to integrate revisions into the dataset secondary storage as a background task). A POSITA would have recognized that storing the standby archive transaction redo log ("*storing the standby archive transaction redo log on the standby site*") would merely involve using the known storage techniques that Srinivasan already teaches. Ex-1003, ¶¶126-127.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 2. Ex-1003, ¶¶ 125-128.

3. *Claim 3: "A method as in claim 1, further comprising the step of storing transactions in the standby active transaction redo log as they are trapped at the primary site."*

As discussed above for claim 1, the combined teachings of Srinivasan and Taylor would have rendered obvious a standby online redo log "*standby active transaction redo log*." Ex-1003, ¶ 129. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The '139 Patent describes mirroring the online redo log using asynchronous remote mirroring technology that works by trapping all operating system writes for a storage device and mirroring those writes across disks on remote systems. Ex-1003, ¶ 130; Ex-1001, 5:30-35; *see* Ex-1001, 6:55-58 (explaining first step is trapping file system transactions or active log writes using a remote mirroring device capable of mirroring the log writes to a standby site).

Srinivasan teaches that the remote mirroring facility 124 ("*remote mirroring device*") traps and mirrors write commands to the secondary storage facility, consistent with how those terms are used in the '139 Patent. Ex-1005, 10:7-29 (describing remote mirroring facility 124 interfaces cache memory 120 to dual redundant links 125, 126 to mirror files and updates from primary storage 118 to secondary storage 128), 2:26-37 (explaining changes to the local database could be recorded in the remote database on an ongoing basis using a conventional remote

mirroring facility and conventional snapshot facility), 5:24-35 (describing that remote mirroring facility 30 transmits write commands from primary host processor 21 over the link 22 to the secondary storage controller 27 in the secondary storage system), 6:64-66 (explaining transaction commit commands can be encoded in the write commands send over the link). A POSITA would have recognized that storing transactions in a standby online redo log as they are trapped ("*storing transactions in the standby active transaction redo log as they are trapped at the primary site*") would merely involve using the known mirroring storage facility to perform its intended function, as Srinivasan already teaches. Ex-1003, ¶¶131-132.

Figure 8 of Srinivasan shows the remote mirroring facility 124 of the primary data storage system 110, and links 125, 126 connecting the remote mirroring facility 134 of the standby data storage system 113:

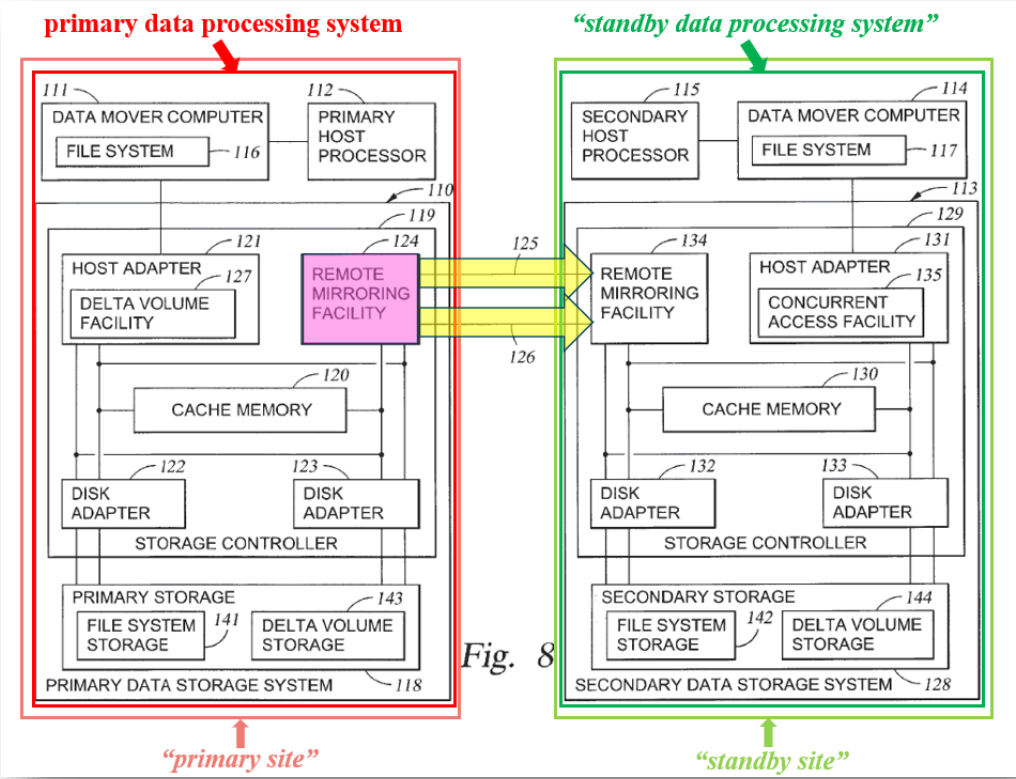


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶¶133.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 3. Ex-1003, ¶¶ 129-134.

4. *Claim 4: "A method as in claim 1, further comprising the step of entering transactions from the archive transaction redo log into the standby transaction processing system."*

As discussed above for claim 1, the combined teachings of Srinivasan and Taylor would have rendered obvious a standby archive redo log "*standby archive transaction redo log*." Ex-1003, ¶ 135. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

Further, Srinivasan teaches multiple techniques to enter transactions into the standby transaction processing system. Ex-1005, 7:13-31 (explaining secondary data storage system and director 47, 48 use a directory to confirm write command is accurate, and then write the dataset revisions and the update of the write command into the allocated storage), 5:29-33 (describing storage controller 27 at the secondary storage system 28 receives the write commands and executes them to maintain a copy of data (from the primary storage system 20) in the secondary storage system 28), 17:7-15 (explaining that each set of revisions is processed in a write-selected phrase in which revisions are written to a buffer and processed to create a directory, followed by a read-selected phase where revisions are read from the buffer and integrated into the dataset). A POSITA would have recognized that entering writes and write commands from the archive redo log ("*entering transactions from the archive transaction redo log into the standby transaction processing system*") would merely involve using the known write-execution techniques that Srinivasan already teaches. Ex-1003, ¶¶ 136-137.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 4. Ex-1003, ¶¶ 135-138.

5. *Claim 5*

a. *[5preamble]: "A method for creating archive redo logs for a standby database at a remote location, comprising the steps of:"*

As discussed above for claim 1, the combined teachings of Srinivasan and Taylor would have rendered obvious an "*archive redo log*." Ex-1003, ¶ 140. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

As discussed above in the claim [1preamble], to the extent the preamble is limiting, Srinivasan taught or suggested "*a method for creating a standby data processing system*" ("*method for creating … a standby database*") at a remote *standby site* ("*remote location*"). Ex-1003, ¶ 141.

Additionally, as discussed above in claim [1c], the combined teachings of Srinivasan and Taylor would have rendered obvious creating archive redo logs ("*archive redo logs*") as a result of the creation of "*the standby data processing system*." Ex-1003, ¶ 142.

The "*standby database*" at the "*remote location"* is shown in annotated Figure 8:

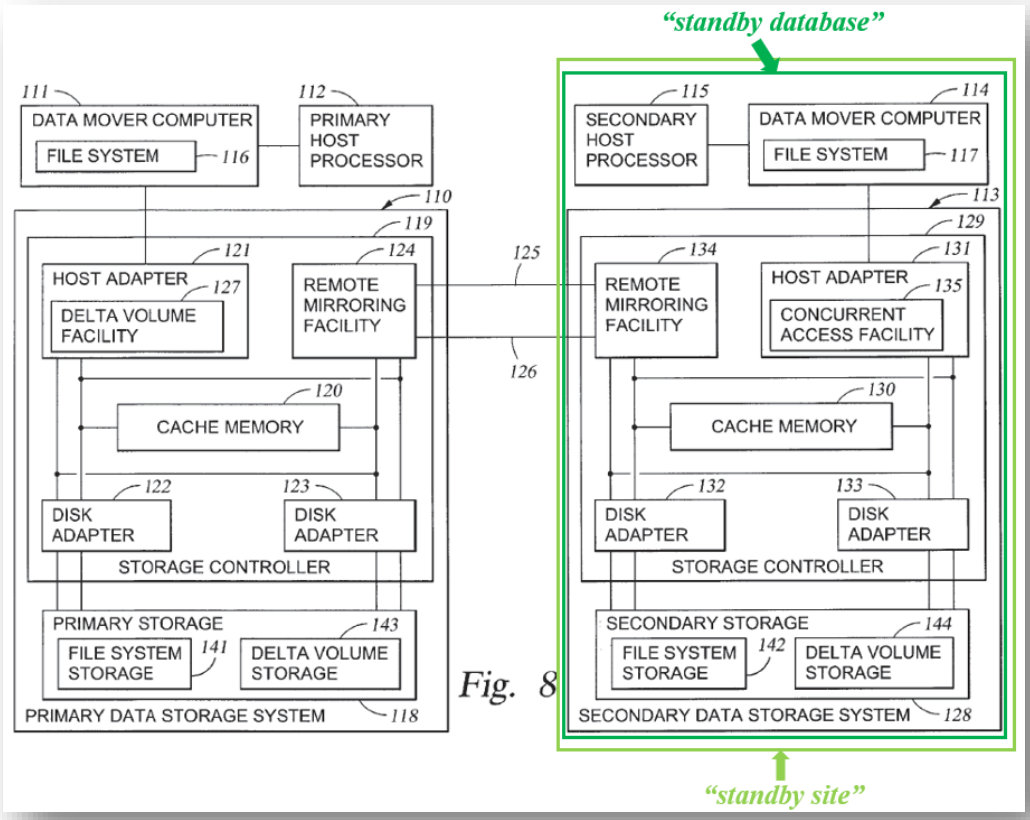


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶ 143.

Additionally, as discussed above in claim [1c], the combined teachings of Srinivasan and Taylor would have rendered obvious creating archive redo logs ("*archive redo logs*") as a result of the creation of "*the standby data processing system*." Ex-1003, ¶ 144.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this preamble to the extent it is limiting. Ex-1003, ¶¶ 139-145.

b. ***[5a]: "(a) trapping active log writes using a remote mirroring device that is capable of mirroring the log writes to a standby site, to create an online redo log at the standby site;"***

As discussed for claim 1, the combined teachings of Srinivasan and Taylor would have rendered obvious an "*online redo log*." Ex-1003, ¶ 146. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

Additionally, as discussed in claim 3, the combined teachings of Srinivasan and Taylor would have rendered obvious trapping write commands of transactions ("*active log writes*") using a remote mirroring facility 124 ("*remote mirroring device*") capable of mirroring the log writes to create a standby online redo log (shown below in Fig. 8):

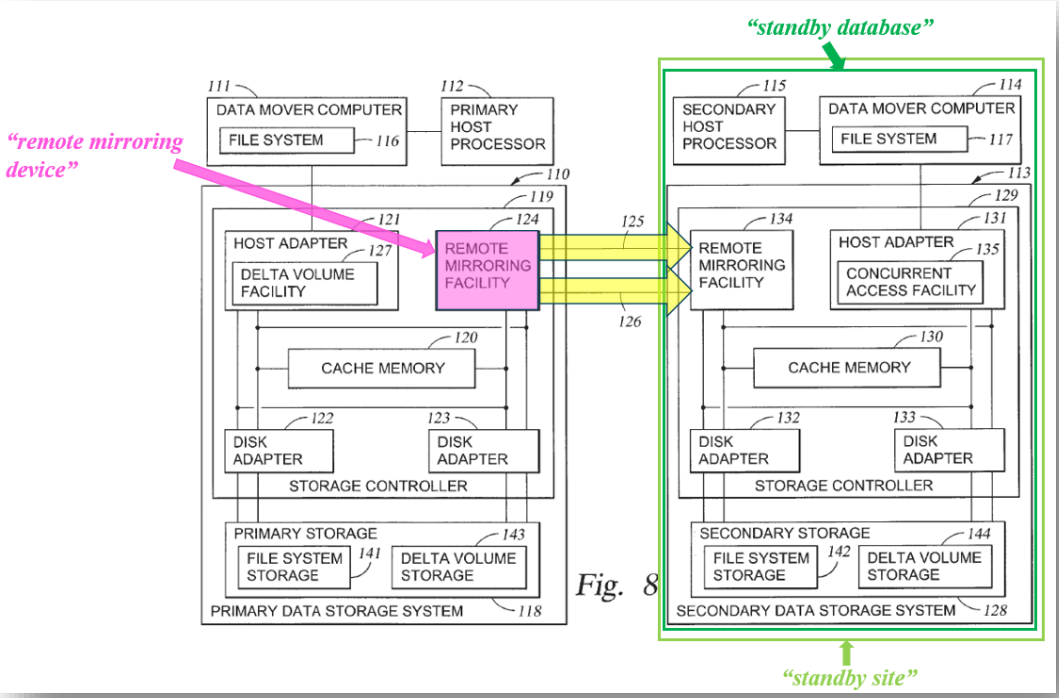


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶ 147; *see* claim [3].

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this element. Ex-1003, ¶¶ 146-148.

c. *[5b]: "(b) computing a difference between an online redo log and archive redo log to create a delta image at a primary database, wherein the delta image includes an archive header;"*

As discussed above for claim [1a], the combined teachings of Srinivasan and Taylor would have rendered obvious "*computing a difference between an online redo log and archive redo log to create a delta image at a primary database*." Ex-1003, ¶ 149. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The "*primary database*," including the primary storage controller 119 and the primary host processor 112 for computing the differences, are shown in Figure 8:

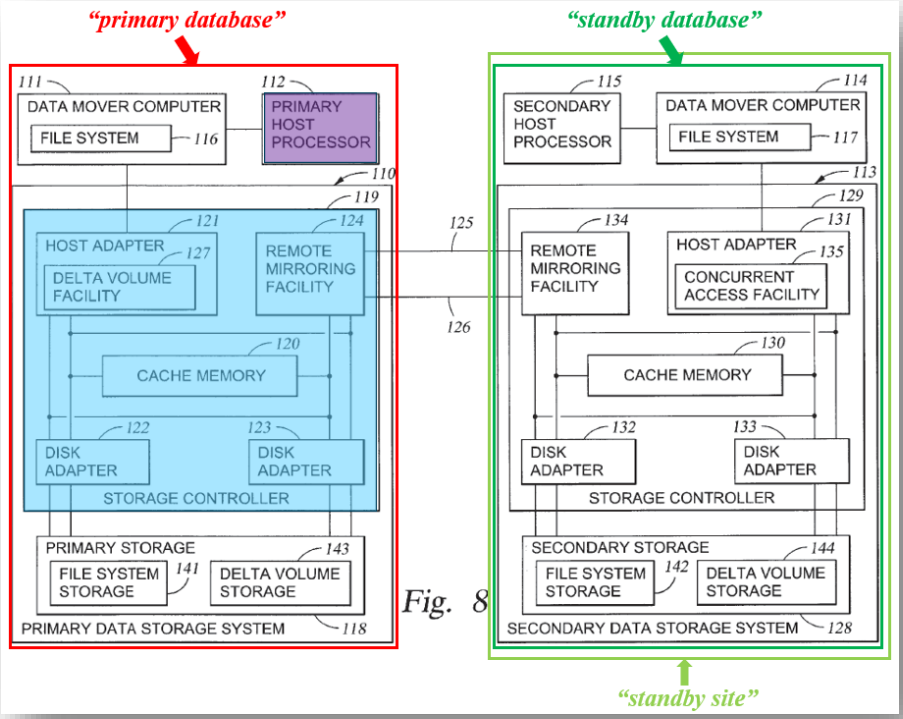


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶ 150.

Additionally, as discussed above for claim [1b], the combined teachings of Srinivasan and Taylor would have rendered obvious "*wherein the delta image includes an archive header*." Ex-1005, 11:12-36 (explaining that delta volumes ("*delta images*") are broken down into delta sets or chunks identified by sequence number and delta set size that can be written to a delta chunk header or trailer 149). The delta chunk header or trailer 149 of Srinivasan is an "*archive header*." Figure 10 of Srinivasan shows the header or trailer on the delta volume ("*delta image*"):

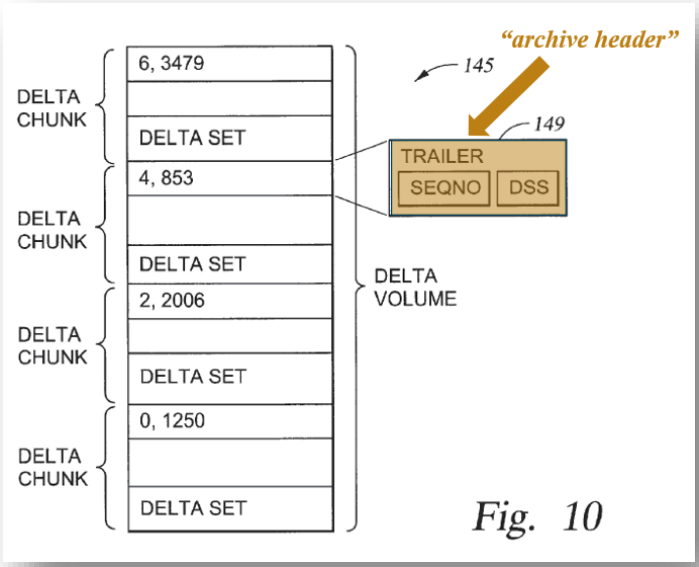


Fig. 10

Ex-1005, Fig. 10 (annotated); Ex-1003, ¶ 151.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this element. Ex-1003, ¶¶ 149-152.

d. *[5c]: "(c) transmitting the delta image to a standby archive log replicator contained in the standby database; and"*

As discussed above for claim [1b], the combined teachings of Srinivasan and Taylor would have rendered obvious "*transmitting the delta image to a standby database*." Ex-1003, ¶ 153. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The '139 Patent explains that the standby archive log replicator 82 communicates with the remote mirroring component 70 using replication handles (sequence numbers) to confirm whether the corresponding online redo log has been

completely mirrored, and then creates the archive redo log 76, 78 from the corresponding online redo log(s) 80. Ex-1001, 5:52-67. The standby archive log replicator also overlays the delta image on the standby online redo log to create the archive redo log. Ex-1001, 6:1-13. Ex-1003, ¶ 154.

The storage controller 129 of Srinivasan is a "*standby archive log replicator*," consistent with how that term is used in the '139 Patent. Ex-1005, 10:46-49 (explaining the storage controller 129 includes a semiconductor cache memory, host adapter 131, disk adapters 132, 133, and a remote mirroring facility 134), 10:24-29 (explaining, with reference to FIGS. 9-12, that the delta volume ("*delta image*") is remotely mirrored from the remote mirroring facility 124 at the primary site to the remote mirroring facility 134 at the standby site), 10:50-57 (describing host adapter 131 of the storage controller 129 is programmed with a concurrent access facility 135 that obtains updates from the mirrored delta volume storage 144 in the secondary storage 128), 12:10-15 (explaining the concurrent access facility 135 in the secondary storage system uses a volume manager utility that overlays the delta set 151 on the file system metavolume 152). A POSITA would have recognized that the storage controller 129 is a "*standby archive log replicator*" and merely performs its intended function taught by Srinivasan. Ex-1003, ¶¶ 155-156.

Figure 8 of Srinivasan shows the storage controller 129 ("*standby archive log replicator*"):

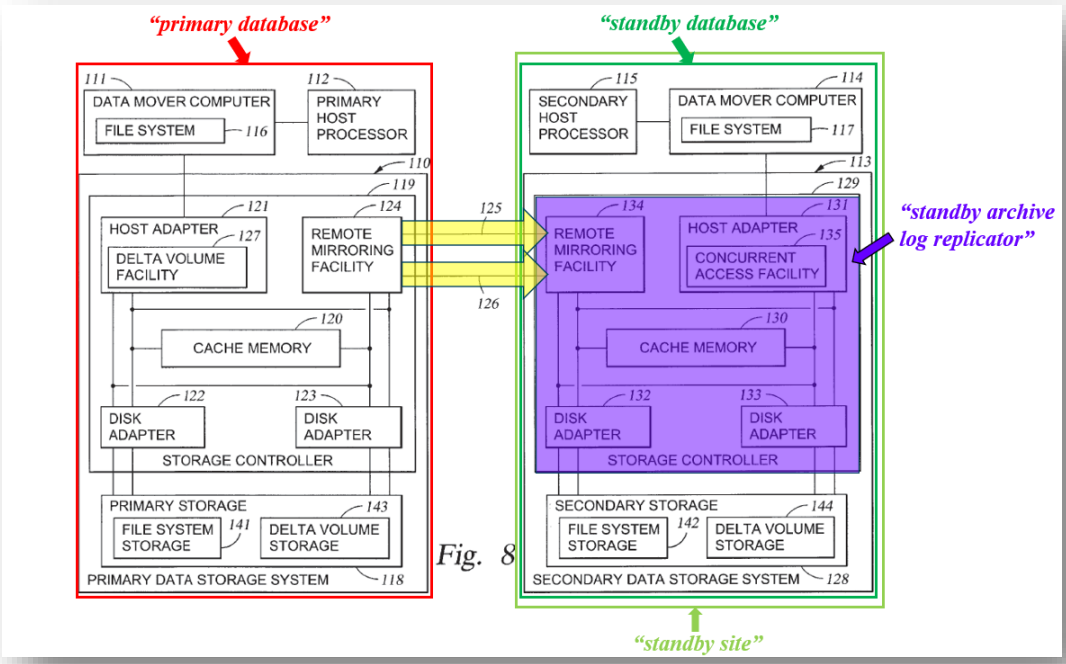


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶ 157.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this element. Ex-1003, ¶¶ 152-158.

e. *[5d]: "(d) combining the delta image and archive header with the online redo log to create a standby archive redo log on the standby database."*

As discussed above for claim [1c], the combined teachings of Srinivasan and Taylor would have rendered obvious an "*combining the delta image ... with the online redo log to create a standby archive redo log on the standby database*." Ex-1003, ¶ 159. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

Additionally, as discussed for claim [5b], the delta volume "*delta image*" includes an "*archive header*." Thus, the process of "*combining the delta image and archive header with the online redo log*" would be the same as claim [1c]. Ex-1003, ¶ 160.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 5. Ex-1003, ¶¶ 159-161.

6. *Claim 6: "A method as in claim 5, further comprising the step of applying the standby archive redo log by creating links to the standby archive redo log."*

As discussed above for claim 4, the combined teachings of Srinivasan and Taylor would have rendered obvious "*entering transactions from the archive transaction redo log into the standby transaction processing system*," which is equivalent to "*applying the standby archive redo log*." Ex-1003, ¶ 162. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The '139 Patent explains that the standby archive file replicator can apply the completed archive to the standby database server 84 by creating a link to the file in the archive destination where the standby database is looking for files. Ex-1001, 6:17-21. Taylor and Srinivasan would have rendered obvious "*creating links to the standby archive redo log*s" as that term is used in the '139 Patent. Ex-1003, ¶ 163.

As discussed above in Section VII.A, it would have been obvious to combine Taylor's teachings for "*creating links to the standby archive redo log*s" (e.g., creating links using directories to each archive redo log) and the teachings of Srinivasan. Srinivasan already teaches that each data storage system can be directly linked to multiple host processors, and that the files can be organized by a logical-to-physical mapping for access by any processor. Ex-1005, 17:34-38 (explaining each data storage system can be directly linked to multiple host processors), 15:63-16:1 (explaining concurrent access facility that interfaces between the virtual file system and the physical file system), 16:1-13 (describing a UNIX-based file system having a hierarchical file system structure including directories and files, and each directory and file has an "inode" containing metadata of the directory or file). Ex-1003, ¶ 164.

A POSITA would have recognized that creating links in this manner would merely involve combining the logical-to-physical mapping teachings of Srinivasan with the logical-to-physical mappings of Taylor such that the archive redo logs could be accessed by any processor. Like Srinivasan, Taylor also teaches that the archive logs can use logical-to-physical mapping by creating links to the archive logs, which are accessible from any node. Ex-1006, 5:39-46 (describing archive logs are maintained in directories on the node where the instance is running and how archive log directories are set to be readable and writable by all nodes servicing the database),

5:46-50 (explaining log directories are accessible from different nodes via a link to a network file system (NFS) or equivalent, in read/write mode on any nodes from which backup or restore will be performed), 5:50-55 (describing how directories are mounted using the same names everywhere they are mounted for consistent access via the directory). A POSITA would have been motivated to modify the teachings of Srinivasan and Taylor because creating links in this manner would improve the functionality of the Srinivasan system in parallel databases. Ex-1003, ¶¶ 165-166.

A POSITA would have had a reasonable expectation that creating links in this manner by combining the teachings of Srinivasan and Taylor because doing so would merely involve known logical-to-physical mapping techniques for the archive redo logs and allowing access by multiple processors, as Srinivasan already teaches. Ex-1005, 15:63-16: 13, 17:34-38; Ex-1003, ¶ 167.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 6. Ex-1003, ¶¶ 162-168.

7. *Claim 7: "A method as in claim 5, further comprising the step of transmitting the delta image to a standby archive log replicator using a first remote mirroring device in the primary database and a second remote mirroring device in the standby database."*

As discussed above for claim [5c], the combined teachings of Srinivasan and Taylor would have rendered obvious "*transmitting the delta image to a standby archive log replicator ... in the standby database*." Ex-1003, ¶ 169. A POSITA

would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

Srinivasan teaches a remote mirroring facility 124 ("*first remote mirroring device in the primary database*") that transmits delta volumes containing updates ("*delta images*") over dual redundant links 125, 126 to a remote mirroring facility 134 ("*second remote mirroring device in the standby database*") of the storage controller 129 ("*standby archive log replicator*"). Ex-1005, 10:46-49 (explaining the storage controller 129 includes a semiconductor cache memory, host adapter 131, disk adapters 132, 133, and a remote mirroring facility 134), 10:24-29 (explaining, with reference to FIGS. 9-12, that the delta volume ("*delta image*") is remotely mirrored from the remote mirroring facility 124 at the *primary site* to the remote mirroring facility 134 at the *standby site*), 10:50-57 (describing host adapter 131 of the storage controller 129 is programmed with a concurrent access facility 135 that obtains updates from the mirrored delta volume storage 144 in the secondary storage 128), 12:10-15 (explaining the concurrent access facility 135 in the secondary storage system uses a volume manager utility that overlays the delta set 151 on the file system metavolume 152). A POSITA would have recognized that using remote mirroring facility 124 ("*first remote mirroring device*") and remote mirroring facility 134 ("*second remote mirroring device*") to transmit the delta volume ("*delta image*") to the storage controller 129 ("*standby archive log replicator*") would

merely involve using the known mirroring storage facilities 124, 134 to perform their intended functions taught by Srinivasan. Ex-1003, ¶¶ 170-171.

Figure 8 of Srinivasan shows the transmission path of a delta volume ("*delta image*") using the remote mirroring facility 124 ("*first remote mirroring device*") and the remote mirroring facility 134 ("*second remote mirroring device*"):

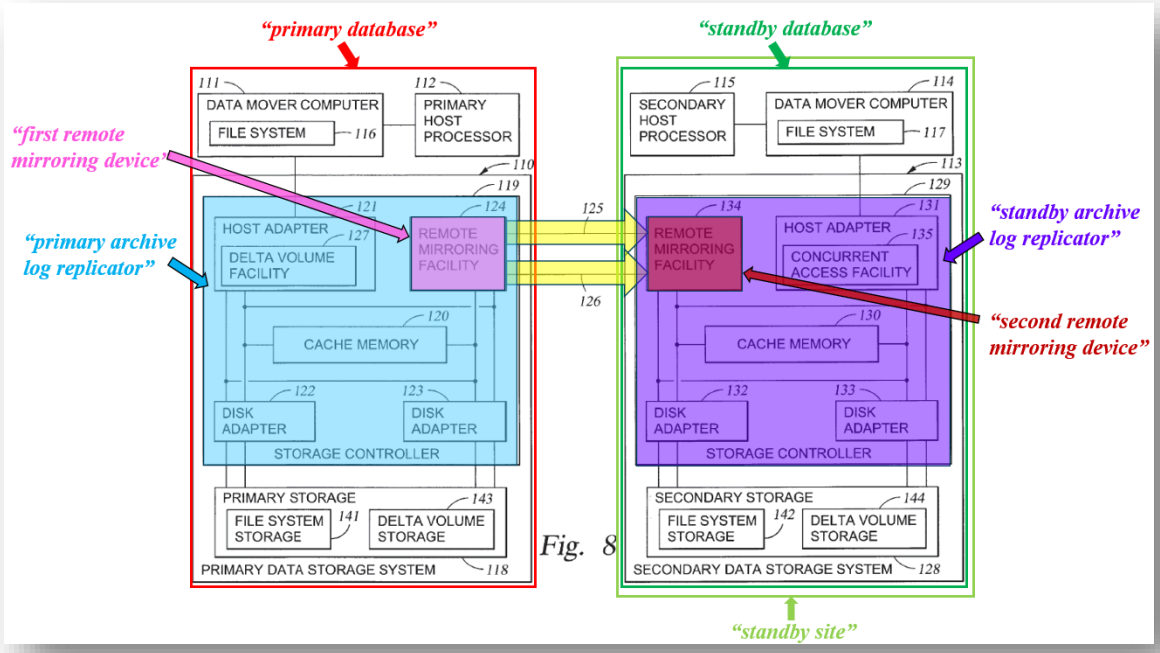


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶¶ 172.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 7. Ex-1003, ¶¶ 169-173.

8. *Claim 8: "A method as in claim 7, further comprising the step of using replication handles to track the progress of mirror writes on the standby database as compared to the primary database."*

As discussed above for claims [1b] and [5b], the combined teachings of Srinivasan and Taylor would have rendered obvious "*using replication handles to track the progress of mirror writes*" by including an "*archive header*" with sequence numbers as part of the delta volume ("*delta image*"). Ex-1003, ¶ 174. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The '139 Patent explains that replication handles are sequence numbers used to help track the progress of the mirror writes on the standby site as compared to the writes on the primary site. Ex-1001, 5:55-62. Ex-1003, ¶ 175.

Srinivasan teaches that each delta chunk includes a header or tailer 149 ("*archive header*") with a sequence number (SEQNO) and delta set size (DSS). Ex-1005, 11:12-36. The sequence number (SEQNO) is a "*replication handle*," consistent with how that term is used in the '139 Patent, because it is a sequence number that helps track the sequence of transactions, including the progress of mirror writes on the *standby site*. Ex-1005, 13:42-44 (explaining updating sequence number in delta chunk validates delta set), 49-52 (describing how the secondary data storage system should have received all updates in the delta set of the current delta chunk before receiving the transaction commit command), 52-60 (explaining the

remote mirroring facility can be operated in synchronous mode to synchronize the delta volume between the primary and secondary data storage systems, and then the transaction commit command can be sent upon synchronization of the delta volume). A POSITA would have recognized that using sequence numbers (SEQNO) ("*replication handles*") to track the progress of mirror writes on the "*standby database*" would merely involve using the known techniques to perform their intended functions, as Srinivasan already teaches. Ex-1003, ¶¶ 176-177.

Figure 10 of Srinivasan shows the sequence number (SEQNO) ("*replication handle*"):

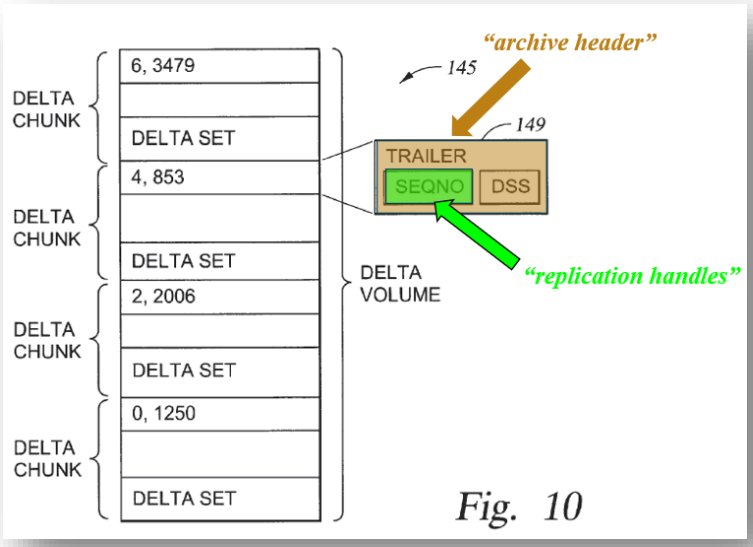


Fig. 10

Ex-1005, Fig. 10 (annotated); Ex-1003, ¶¶ 178.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 8. Ex-1003, ¶¶ 174-179.

9. *Claim 9: "A method as in claim 8, further comprising the step of checking the primary database queries using the archive log replicator to determine whether archiving is complete."*

As discussed above for claim 8, the combined teachings of Srinivasan and Taylor would have rendered obvious using sequence numbers (SEQNO) ("*replication handles*") "*to track the progress of mirror writes on the standby database*," including whether archiving is complete. Ex-1003, ¶ 180. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The '139 Patent explains that the primary archive log replicator signals completion of a new standby archive redo log by checking with the database system using database queries for archive completion. Ex-1001, 6:10-13. Conversely, the standby archive log replicator 82 checks with the corresponding database (through queries) to determine whether the archiving of a newly created standby archive redo log 76 is complete. Ex-1001, 6:14-17. Ex-1003, ¶ 181.

Srinivasan teaches that the storage controller 129 ("*archive log replicator*") checks "*primary database queries*," consistent with how that term is used in the '139 Patent, to determine whether archiving is complete. 13:40-44 (explaining the storage controller writes the delta set size (DSS) and sequence number (SEQNO) into an attributes block of the delta chunk, which is used to validate the delta set in the delta chunk), 49-52 (describing that secondary data storage system should have received

all updates in the delta set of the delta chunk before receipt of the transaction commit command), 14:4-17 (explaining updates based on sequence numbers and delta volume size for sequential loading and unloading), 35-41 (describing overflow detection if sequence number of delta chunk exceeds next number in sequence, which trigger controller error recovery procedure), 46-53 (explaining advantage of using two delta volumes per file size to ensure all updates in current delta chunk are flushed and received by secondary storage system before the transaction commit command at the secondary storage to minimize impact on host processing). A POSITA would have recognized that using the storage controller 129 ("*archive log replicator*") to check "*primary database queries*" using sequence numbers (SEQNO) ("*replication handles*") and delta set size (DSS) to determine whether archiving is complete would merely involve using the known techniques and components to perform their intended functions taught by Srinivasan. Ex-1003, ¶¶ 182-183.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 9. Ex-1003, ¶¶ 180-184.

10. *Claim 10: "A method as in claim 5, further comprising the step of shipping an entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously after archiving becomes slow enough that there is a risk of primary database blocking."*

As discussed above for claim [1a], the combined teachings of Srinivasan and Taylor would have rendered obvious "*shipping an entire archive redo log*," from the primary system to the standby system upon initialization. Ex-1003, ¶ 185. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The '139 Patent explains that the primary archive log replicator 72, 74 traps opens/writes, then ships the newly generated online redo log files to the standby site, and then asks the standby database to apply it in a fashion that 2-safeness and data integrity are always preserved. Ex-1001, 5:3-8. Additionally, the '139 Patent describes a "catch-up" mode in which the spool buffer in the remote mirroring device is filled or when archiving is slow enough (due to WAN delay or outages) that there is a risk of the primary database blocking because it cannot archive the active log files. Ex-1001, 6:29-40; Ex-1003, ¶ 186.

The storage controller 119 is a "*primary archive log replicator*," consistent with how that term is used in the '139 Patent. Ex-1005, 10: 6-15 (explaining the storage controller 119 includes a cache memory 120, host adapter 121, disk adapters 122, 123, and remote mirroring facility 124), 17-19 (describing remote mirroring

facility as mirroring file system storage), 3:54-57 (explaining the storage controller is programmed with a remote mirroring facility for mirroring volumes to a remote data storage system), 2:18-21 (explaining remotely mirroring a local copy of the database to a remote data storage system), 10:24-29 (explaining, with reference to FIGS. 9-12, that the remote mirroring facility 124 can also mirror the delta volume ("*delta image*") to the remote mirroring facility 134 at the *standby site*). A POSITA would have recognized that the storage controller 119 is a "*primary archive log replicator*" and merely involves using the known storage controller 119 to perform its intended function taught by Srinivasan. Ex-1003, ¶¶ 187-188.

As discussed for claim [5c], the storage controller 129 is a "*standby archive log replicator*." *See* claim [5c]; Ex-1003, ¶ 189.

Srinivasan also teaches that the system can transmit transaction logs asynchronously when the primary and standby systems are desynchronized, such as during a network delay or outage. Ex-1005, 14:39-45 (explaining detection of an overflow condition and control error recovery procedure that suspends writes, re-synchronizes primary and secondary storage file systems, and restarts the delta volume facility), 11:38-42 (describing asynchronous transmission of updates over the data link between the primary and secondary data storage systems), 13:52-60 (describing that remote data mirroring facility can be operated in asynchronous or semi-synchronous mode), 12:53-59 (explaining timer variable can be used to limit or

increase transmission frequency), 17:47-53 (explaining the processors and data storage systems can communicate with each other using various protocols, including Asynchronous Transfer Mode (ATM) protocols). A POSITA would have recognized that "*shipping an entire archive redo log … asynchronously after archiving becomes slow enough that there is a risk of primary database blocking*" using the storage controller 119 ("*primary archive log replicator*") and the storage controller 129 ("*standby archive log replicator*") would merely involve using the known techniques and components to perform their intended functions taught by Srinivasan. Ex-1003, ¶¶ 190-191.

Figure 8 of Srinivasan shows a primary storage controller 119 ("*primary archive log replicator*") to a secondary storage controller 129 ("*standby archive log replicator*"):

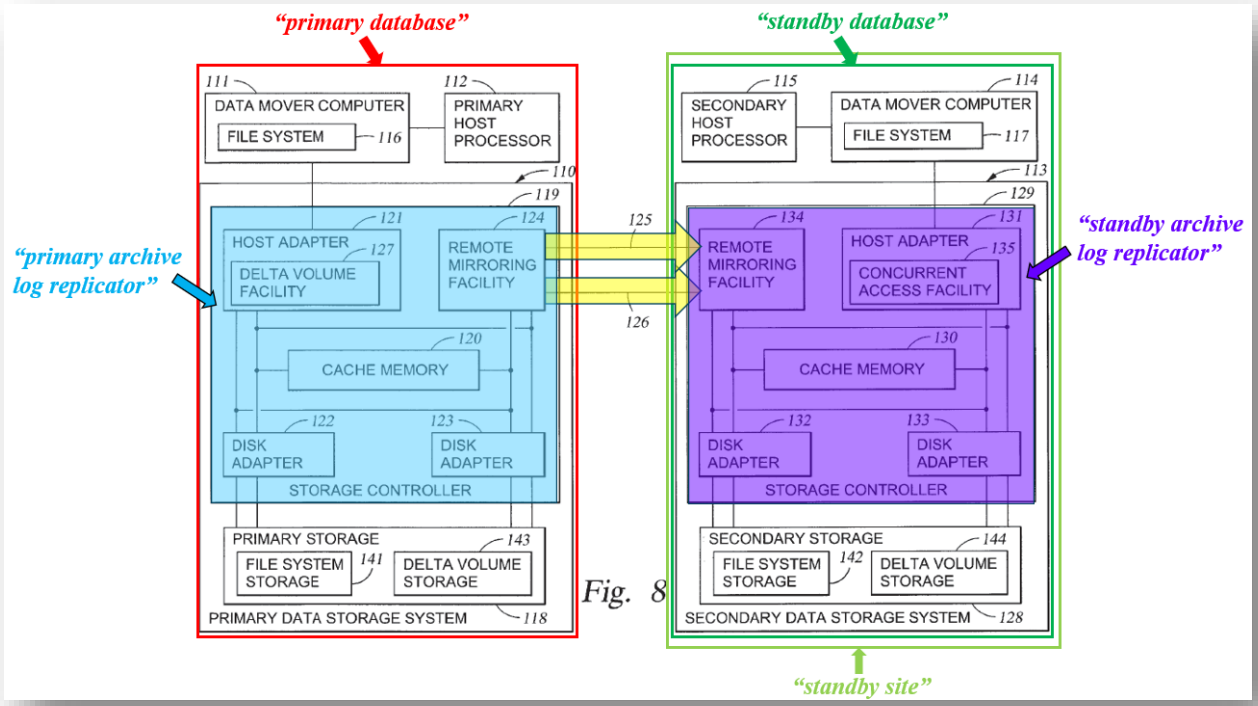


Fig. 8

Ex-1005, Fig. 8 (annotated); Ex-1003, ¶¶ 192.

Accordingly, Srinivasan and Taylor would have rendered obvious claim 10. Ex-1003, ¶¶ 185-193.

11. ***Claim 11: "A method as in claim 10, further comprising the step of shipping an entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously when a spool buffer in a remote mirroring system has been filled."***

As discussed above for claim 10, the combined teachings of Srinivasan and Taylor would have rendered obvious "*shipping an entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously*." Ex-1003, ¶ 194. A POSITA would have been motivated to

combine the teachings of Srinivasan and Taylor for the reasons described above. *See* §VII.A.

The '139 Patent does not describe what a spool buffer is, but rather that it is a subcomponent of a remote mirroring device. Ex-1001, 6:33-37. Consistent with how that term is used in the '139 Patent, Taylor describes the archive log as a file that is being "spooled" to disk, while the online log is being written. Ex-1006, 3:3-5. And Taylor describes buffers servicing the database in connection with discussing archive logs. Ex-1003, ¶ 195; Ex-1006, 3:8-11, 5:29-31.

Srinivasan teaches a buffer overflow can occur when a primary system tries to process new data before the secondary has processed old data. Ex-1005, 14:39-45 (explaining detection of an overflow condition and control error recovery procedure that suspends writes, re-synchronizes primary and secondary storage file systems, and restarts the delta volume facility), 18:51-56 (explaining, in claim 8, that the frequency of updating the dataset is permitted to exceed a limited frequency when required to avoid buffer overflow), 19:41-45 (explaining same for claim 14). A POSITA would have recognized that "*shipping an entire archive redo log ... asynchronously*" using the storage controller 119 ("*primary archive log replicator*") and the storage controller 129 ("*standby archive log replicator*") would merely involve applying the known technique to another condition already

contemplated by Srinivasan and using the components to perform their intended functions. Ex-1003, ¶¶ 196-197.

Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 11. Ex-1003, ¶¶ 194-198.

12. *Claim 12*

a. *[12preamble]: "A method for creating archive redo logs from a primary database, for a standby database at a remote location, comprising the steps of:"*

The combined teachings of Srinivasan and Taylor would have rendered obvious this preamble to the extent it is limiting for the reasons discussed for claims [1preamble] and [5preamble]. Ex-1003, ¶ 200.

b. *[12a]: "(a) trapping file system transactions to create an online redo log at the standby database;"*

The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5a]. Ex-1003, ¶ 201.

c. *[12b]: "(b) computing a difference between an online redo log and archive redo log in the primary database to find the delta image of the archive redo log;"*

The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5b]. Ex-1003, ¶ 202.

d. *[12c]: "(c) transmitting the delta image to a standby database; and"*

The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5c]. Ex-1003, ¶ 203.

e. ***[12d]: "(d) combining the delta image with the online redo log in the standby database to create standby archive redo log on the standby database."***

The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5d]. Ex-1003, ¶ 204.

13. ***Claim 13: "A method as in claim 12, further comprising the step of applying the standby archive redo log by creating a link between the standby database and the standby archive redo log."***

The combined teachings of Srinivasan and Taylor would have rendered obvious claim 13 for the reasons discussed in claim 6. Ex-1003, ¶ 205.

14. ***Claim 14: "A method as in claim 12, further comprising the step of transmitting the delta image to a standby archive log replicator using a first remote mirroring device in the primary database and a second remote mirroring device in the standby database."***

The combined teachings of Srinivasan and Taylor would have rendered obvious claim 14 for the reasons discussed in claim 7. Ex-1003, ¶ 206.

15. ***Claim 15: "A method as in claim 14, further comprising the step of using replication handles to track the progress of mirror writes on the standby database as compared to the primary database."***

The combined teachings of Srinivasan and Taylor would have rendered obvious claim 15 for the reasons discussed in claim 8. Ex-1003, ¶ 207.

16. ***Claim 16: "A method as in claim 12, further comprising the step of checking the primary database with database queries using the archive log replicator to determine whether archiving is complete."***

The combined teachings of Srinivasan and Taylor would have rendered obvious claim 16 for the reasons discussed in claim 9. Ex-1003, ¶ 208.

17. ***Claim 17: "A method as in claim 12, further comprising the step of shipping an entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously after archiving becomes slow enough that there is a risk of the primary database blocking."***

The combined teachings of Srinivasan and Taylor would have rendered obvious claim 17 for the reasons discussed in claim 10. Ex-1003, ¶ 209.

18. ***Claim 18: "A method as in claim 12, further comprising the step of shipping the entire archive redo log from the primary archive log replicator to the standby archive log replicator asynchronously when a spool buffer in a remote mirroring system has been filled."***

The combined teachings of Srinivasan and Taylor would have rendered obvious claim 18 for the reasons discussed in claim 11. Ex-1003, ¶ 210.

VIII. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. The Board Should Not Deny Institution Under 35 U.S.C. § 325(d)

Denial under § 325(d) is not warranted because the challenges presented in this Petition are neither cumulative nor redundant to the prosecution of the '139 Patent, as Srinivasan and Taylor were neither cited nor considered by the Examiner during prosecution of the '139 Patent. The references in this Petition were neither

relied upon during prosecution nor are cumulative of art relied upon during prosecution. Further, the '139 Patent has not been previously challenged in an IPR. In addition, the Examiner also did not have the benefit of Dr. Almeroth's detailed expert testimony and the other evidence of record.

Even if it were asserted that any of these references were substantially the same or cumulative of the art and arguments raised during prosecution, this Petition's Grounds identify reference combinations that are not substantially the same as any considered references, specifically by relying on Srinivasan (a reference never considered by the Examiner) as disclosing the majority of the claim limitations. *See, e.g.*, *Sony v. MZ Audio Scis., LLC*, No. IPR2022-01544, Paper 12, at 7 (Apr. 21, 2023) (finding Section 325(d) discretion improper where "the Examiner did not consider the specific combination of references asserted during prosecution"). The Board "did not expressly consider" any art with substantially the same teachings as the art in this Petition, rendering it "impossible . . . to determine the consideration the Examiner gave" to such references, and demonstrating that "the Office erred in a manner material to patentability." *Ecobee Techs. v. Causam Enters., Inc.*, No. IPR2022-01339, Paper 20, at 20–21 (declining to exercise Section 325(d) discretion).

Thus, the Board should not exercise discretion to deny institution.

B. The Board Should Not Deny Institution Under 35 U.S.C. § 314(a)

The merits of Petitioner's arguments are compelling and the evidence in support is substantial. That "alone demonstrates that the PTAB should not discretionarily deny institution under *Fintiv*." Ex-1010 at 4-5. But in any event, the six *Fintiv* factors do not justify denying institution.

Factor 1 (stay) is **neutral**. SAP has not yet moved to stay pending this IPR.

Factor 2 (proximity to trial) is **neutral or weighs against denial**. In the originally filed litigation by Patent Owner Valtrus (*Valtrus I*), the trial date is not currently set (though jury selection is set for September 22, 2025. Ex-1011. However, the Board should not rely on the current schedule in *Valtrus I* because Patent Owner Valtrus filed suit alone (without the other named Patent Owner, Key Patent Innovations) and thus did not have Article III Standing. Ex-1014; Ex-1013. Petitioner filed a Motion to Dismiss for Lack of Standing in *Valtrus I*, reconsideration of which remains pending and casts serious doubt on whether *Valtrus I* will actually proceed along the current schedule. Indeed, in response to Petitioner's Motion to Dismiss for Lack of Standing, Patent Owners essentially concede that Valtrus alone lacked standing because Patent Owners together filed a second, duplicative lawsuit (*Valtrus II*) in an effort to rectify the Article III standing deficiencies in *Valtrus I*. Ex-1009; Ex-1014. This conduct confirms that Patent Owners themselves doubt that the trial schedule in *Valtrus I* will hold, further

supporting that the Board should not rely on that schedule at all. Additionally, the other pending litigations involving the challenged patent (*Valtrus II* and a declaratory judgment action filed by Petitioner) do not have any trial date currently set, meaning there is no reliable trial date for the Board to consider in relation to this Petition.

Even if the Board assumes the *Valtrus I* trial date applies, this factor is neutral because there will be other claims not litigated at trial that are covered by this Petition. Therefore, a final written decision will issue first at least on the claims not litigated at trial but covered by this Petition.

Factor 3 (investment in parallel proceedings) **weighs against denial**. Importantly, claims 2, 4, 6, and 13 are not at issue in the related litigations. *Samsung Bioepis Co. Ltd v. Regeneron Pharmaceuticals, Inc.,* IPR2023-00739, Paper 9 at 57 (Oct. 20, 2023) ("[B]ecause certain of the challenged claims, . . . are not expressly at issue in the district court litigation, *Fintiv* factor 3 favors institution.") The related litigations are in its early stages of discovery, motions to transfer and dismiss have been filed, and the investment by the parties has been relatively minimal. All litigations involving the challenged patent are at an early stage, with major deadlines several months away. Ex-1011. Patent Owners may argue that the parties have invested substantial resources on procedural motions, but the resources expended have been directed toward resolving procedural missteps perpetuated by Patent

Owner Valtrus's choice to file alone (without Article III standing) and Patent Owners' later attempts to fix that with procedurally improper duplicative litigation. These procedural disputes (caused by Patent Owners' actions) should not be held against Petitioner under this factor.

Factor 4 (overlap) **weighs against denial** because Petitioner herby stipulates that, if the IPR is instituted, it will not pursue the IPR grounds in the district court. Indeed, neither Srinivasan nor Taylor are at issue in the District Court Litigation. Ex-1015. Thus, "[i]nstituting trial here serves overall system efficiency and integrity goals by not duplicating efforts and by resolving materially different patentability issues." *Apple, Inc. v. SEVEN Networks, LLC*, IPR2020-00156, Paper 10 at 19 (June 15, 2020); *see also Sand Revolution II, LLC v. Continental Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 12 (June 16, 2020). In addition, Petitioner challenges more claims than Patent Owners will ultimately assert at trial. Patent Owner Valtrus agreed to narrow to 30 total asserted claims by April 30, 2025, and a maximum of six asserted claims per asserted patent. Ex-1011. Thus, Patent Owner will be limited to asserting up to six of the 14 asserted claims of the '139 Patent, merely a subset of the claims challenged in this Petition. *See, e.g.*, *Apple*, No. IPR2020-00156, Paper 10, at 17, 21-22.

While Factor 5 (parties) may weigh slightly in favor of denial because the parties are the same as in district court, under the "holistic view" of the remaining

factors and whether integrity of the system and efficiency is best served, institution is favored. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

Factor 6 (other circumstances) **weigh against denial**. Here, the merits of the Petition are particularly compelling—for example, Srinivasan is a strikingly similar disclosure to that of the '139 Patent, is directed at the same problem and proposes the same solutions as the '139 Patent and discloses nearly identical architecture, including several nearly identical figures. The evidence presented, if unrebutted in trial, would plainly lead to a conclusion that one or more claims are unpatentable by a preponderance of the evidence.

IX. CONCLUSION

Petitioner respectfully requests that the Board institute IPR and enter a final written decision finding the Challenged Claims unpatentable.

Dated: January 14, 2025

Respectfully submitted,

/Louis L. Campbell/
Louis L. Campbell (Lead Counsel)
llcampbell@winston.com
USPTO Reg. No. 59,963
Winston & Strawn LLP
255 Shoreline Dr., Ste. 520
Redwood City, CA 94065
T: (650) 858-6500

Lead Counsel for Petitioner

CERTIFICATE OF COMPLIANCE

This Petition complies with the word count limits set forth in 37 C.F.R. § 42.24(a)(1)(i), because this Petition contains 13,876 words, excluding the parts of the Petition exempted by 37 C.F.R. § 42.24(a)(1), determined using the word count provided by Microsoft Word, which was used to prepare this Petition.

Dated: January 14, 2025

Respectfully submitted,

/Louis L. Campbell/
Louis L. Campbell (Lead Counsel)
llcampbell@winston.com
USPTO Reg. No. 59,963
Winston & Strawn LLP
255 Shoreline Dr., Ste. 520
Redwood City, CA 94065
T: (650) 858-6500

Lead Counsel for Petitioner

CERTIFICATE OF SERVICE

Under 37 C.F.R. §§ 42.6(e) and 42.105(a), this is to certify that on January 14, 2025, I caused to be served a true and correct copy of the foregoing "**PETITION *INTER PARTES* REVIEW OF U.S. PATENT NO. 6,691,139**" and accompanying exhibits by Federal Express priority overnight mail on the following:

Botos Churchill IP Law LLP
430 Mountain Avenue
Suite 401
New Providence, NJ 07974

With a courtesy copy of this Petition and supporting material was also served on litigation counsel for Patent Owners via email:

Matt Berkowitz
mberkowitz@reichmanjorgensen.com
Patrick R. Colsher
pcolsher@reichmanjorgensen.com
Aaron Morris
amorris@reichmanjorgensen.com
Navid Cyrus Bayar
nbayar@reichmanjorgensen.com
Reichman Jorgensen Lehman & Feldberg LLP
100 Marine Parkway
Suite 300
Redwood Shores, CA 94065

Khue Van Hoang
khoang@reichmanjorgensen.com
Reichman Jorgensen Lehman & Feldberg LLP
400 Madison Avenue
Suite 14d
New York, NY 1001

Philip Eklem

peklem@reichmanjorgensen.com
Savannah Harman Carnes
scarnes@reichmanjorgensen.com
Reichman Jorgensen Lehman & Feldberg LLP
1909 K Street, NW
Suite 800
Washington, DC 20006

/Louis L. Campbell/
Louis L. Campbell (Lead Counsel)
llcampbell@winston.com
USPTO Reg. No. 59,963
Winston & Strawn LLP
255 Shoreline Dr., Ste. 520
Redwood City, CA 94065
T: (650) 858-6500

Lead Counsel for Petitioner