

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

**DECLARATION OF KEVIN C. ALMEROTH, PH.D. UNDER 37 C.F.R. §
1.68 IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW**

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	8
II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE	9
III. LEVEL OF ORDINARY SKILL IN THE ART	22
IV. RELEVANT LEGAL STANDARDS	23
V. OVERVIEW OF THE '139 PATENT	25
VI. CLAIM CONSTRUCTION	30
VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE....	30
VIII. THE PRINCIPAL PRIOR ART	32
A. Summary of <u>Srinivasan</u> (Ex-1005).....	32
B. Summary of <u>Taylor</u> (Ex-1006).....	37
IX. DETAILED ANALYSIS THAT CLAIMS 1-18 ARE OBVIOUS OVER <u>SRINIVASAN</u> IN VIEW OF <u>TAYLOR</u>	40
A. <i>Claim 1</i>	52
1. <i>[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:”</i>	52
2. <i>[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;”</i>	57
3. <i>[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”</i>	61
4. <i>[1c]: “(c) combining the delta image with the active transaction redo log at the standby site, to produce a standby archive transaction redo log.”</i>	66
B. <i>Claim 2: “A method as in claim 1, further comprising the step of storing the standby archive transaction redo log on the standby site.”</i>	67

C.	<i>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.”</i>	69
D.	<i>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.”</i>	71
E.	<i>Claim 5:</i>	73
1.	<i>[5preamble]: “A method for creating archive redo logs for a standby database at a remote location, comprising the steps of:”</i>	73
2.	<i>[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;”</i>	75
3.	<i>[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;”</i>	76
4.	<i>[5c]: “(c) transmitting the delta image to a standby archive log replicator contained in the standby database; and”</i>	78
5.	<i>[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.”</i>	80
F.	<i>Claim 6: “The data redundancy system according to claim 1, wherein the first redundancy appliance is disjoint from the second redundancy appliance.”</i>	81
G.	<i>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.”</i>	84
H.	<i>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.”</i>	86

- I. *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.” 88*
- J. *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.” 90*
- K. *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.” 94*
- L. *Claim 12: 95*
 - 1. *[12preamble]: “A method for creating archive redo logs from a primary database, for a standby database at a remote location, comprising the steps of:” 95*
 - 2. *[12a]: “(a) trapping file system transactions to create an online redo log at the standby database;” 96*
 - 3. *[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;” 96*
 - 4. *[12c]: “(c) transmitting the delta image to a standby database; and” 96*
 - 5. *[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.” 96*
- M. *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.” 97*
- N. *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.” 97*

O.	<i>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.”</i>	97
P.	<i>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.”</i>	97
Q.	<i>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.”</i>	98
R.	<i>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.”</i>	98
X.	CONCLUSION.....	99

EXHIBIT LIST

Ex- No.	Brief Description
1001	U.S. Patent No. 6,691,139
1002	File History of U.S. Patent No. 6,691,139
1003	Declaration of Kevin C. Almeroth, Ph.D.
1004	<i>Curriculum Vitae</i> of Kevin C. Almeroth, Ph.D.
1005	U.S. Pat. No. 6,823,336 (“ <u>Srinivasan</u> ”)
1006	U.S. Pat. No. 6,658,589 (“ <u>Taylor</u> ”)
1007	Ramez Elmasri and Shamkant B. Navathe, FUNDAMENTALS OF DATABASE SYSTEMS (2d ed.1994)
1008	Original Complaint for Patent Infringement in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1009	Original Complaint for Patent Infringement in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 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 <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1012	Motion to Dismiss for Lack of Standing (Public Version) in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)

Ex- No.	Brief Description
1013	Motion to Reconsider Ruling on Motion to Dismiss for Lack of Standing (Public Version) in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1014	Plaintiffs' Notice Regarding <i>Valtrus II</i> in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)
1015	SAP's Invalidity Contentions (Cover Pleading) in <i>Valtrus Innovations Ltd. v. SAP America, Inc. et al</i> , 2:24-cv-00021-JRG (E.D. Tex.)

I, Kevin C. Almeroth, do hereby declare as follows:

I. INTRODUCTION

1. I am making this declaration at the request of Petitioner in the matter of the *Inter Partes* Review of U.S. Patent No. 6,691,139 (“’139 Patent”) to Ganesh et al.

2. I am being compensated for my work in this matter at my standard hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether the subject matter of claims 1-18 (“the Challenged Claims”) of the ’139 Patent are anticipated or otherwise would have been obvious to a person having ordinary skill in the art (“POSITA”) at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

- a. The ’139 Patent, Ex-1001.
- b. The prosecution history of the ’139 Patent (“’139 File History”), Ex-1002.
- c. U.S. Pat. No. 6,823,336 (“Srinivasan”), Ex-1005.
- d. U.S. Pat. No. 6,658,589 (“Taylor”), Ex-1006.

e. The additional references I refer to below and/or are identified in the exhibit list.

5. In forming the opinions expressed below, I have considered: the documents listed above; the relevant legal standards as explained to me, including the standards for anticipation and obviousness; and my own knowledge and experience based upon my work in the field of network communications and security as described below.

6. Unless otherwise noted, all emphasis in any quoted material has been added. Claim terms are *italicized*. Exhibit titles are underlined.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

7. My complete qualifications and professional experience are described in my Curriculum Vitae, a copy of which can be found in Ex.1004. The following is a brief summary of my relevant qualifications and professional experience.

8. I am currently a Professor Emeritus in the Department of Computer Science at the University of California, Santa Barbara (UCSB). While at UCSB, I held faculty appointments and was a founding member of the Computer Engineering (CE) Program, Media Arts and Technology (MAT) Program, and the Technology Management Program (TMP). I also served as the Associate Director of the Center for Information Technology and Society (CITS) from 1999 to 2012. I have been a faculty member at UCSB since July 1997.

9. I hold three degrees from the Georgia Institute of Technology: (1) a Bachelor of Science degree in Information and Computer Science (with minors in Economics, Technical Communication, and American Literature) earned in June 1992; (2) a Master of Science degree in Computer Science (with specialization in Networking and Systems) earned in June 1994; and (3) a Doctor of Philosophy (Ph.D.) degree in Computer Science (Dissertation Title: Networking and System Support for the Efficient, Scalable Delivery of Services in Interactive Multimedia System, minor in Telecommunications Public Policy) earned in June 1997. During my education, I took a wide variety of courses as demonstrated by my minors. My undergraduate degree also included a number of courses more typical of a degree in electrical engineering including digital logic, signal processing, and telecommunications theory.

10. One of the major concentrations of my research over the past 30+ years has been the delivery of multimedia content and data between computing devices, including through various network architectures. In my research, I have studied large-scale content delivery systems, and the use of servers located in a variety of geographic locations to provide scalable delivery to hundreds or thousands of users simultaneously. I have also studied smaller-scale content delivery systems in which content is exchanged between individual computers and portable devices. My work has emphasized the exchange of content more efficiently across computer networks,

including the scalable delivery of content to many users, mobile computing, satellite networking, delivering content to mobile devices, and network support for data delivery in wireless networks.

11. In 1992, the initial focus of my research was on the provision of interactive functions (e.g., VCR-style functions like pause, rewind, and fast-forward) for near video-on-demand systems in cable systems; in particular, how to aggregate requests for movies at a cable head-end and then how to satisfy a multitude of requests using one audio/video stream broadcast to multiple receivers simultaneously. This research has continually evolved and resulted in the development of techniques to scalably deliver on-demand content, including audio, video, web documents, and other types of data, through the Internet and over other types of networks, including over cable systems, broadband telephone lines, and satellite links.

12. An important component of my research has been investigating the challenges of communicating multimedia content, including video, between computers and across networks including the Internet. Although the early Internet was used mostly for text-based, non-real time applications, the interest in sharing multimedia content, such as video, quickly developed. Multimedia-based applications ranged from downloading content to a device to streaming multimedia content to be instantly used. One of the challenges was that multimedia content is

typically larger than text-only content, but there are also opportunities to use different delivery techniques since multimedia content is more resilient to errors. I have worked on a variety of research problems and used a number of systems that were developed to deliver multimedia content to users. One content-delivery method I have researched is the one-to-many communication facility called “multicast,” first deployed as the Multicast Backbone, a virtual overlay network supporting one-to-many communication. Multicast is one technique that can be used on the Internet to provide streaming media support for complex applications like video-on-demand, distance learning, distributed collaboration, distributed games, and large-scale wireless communication. The delivery of media through multicast often involves using Internet infrastructure, devices and protocols, including protocols for routing and TCP/IP.

13. Starting in 1997, I worked on a project to integrate the streaming media capabilities of the Internet together with the interactivity of the web. I developed a project called the Interactive Multimedia Jukebox (IMJ). Users would visit a web page and select content to view. The content would then be scheduled on one of a number of channels, including delivery to students in Georgia Tech dorms delivered via the campus cable plant. The content of each channel was delivered using multicast communication.

14. In the IMJ, the number of channels varied depending on the capabilities of the server including the available bandwidth of its connection to the Internet. If one of the channels was idle, the requesting user would be able to watch their selection immediately. If all channels were streaming previously selected content, the user's selection would be queued on the channel with the shortest wait time. In the meantime, the user would see what content was currently playing on other channels, and because of the use of multicast, would be able to join one of the existing channels and watch the content at the point it was currently being transmitted.

15. The IMJ service combined the interactivity of the web with the streaming capabilities of the Internet to create a jukebox-like service. It supported true Video-on-Demand when capacity allowed, but scaled to any number of users based on queuing requested programs. As part of the project, we obtained permission from Turner Broadcasting to transmit cartoons and other short-subject content. We also connected the IMJ into the Georgia Tech campus cable television network so that students in their dorms could use the web to request content and then view that content on one of the campus's public access channels.

16. More recently, I have also studied issues concerning how users choose content, especially when considering the price of that content. My research has examined how dynamic content pricing can be used to control system load. By

raising prices when systems start to become overloaded (i.e., when all available resources are fully utilized) and reducing prices when system capacity is readily available, users' capacity to pay as well as their willingness can be used as factors in stabilizing the response time of a system. This capability is particularly useful in systems where content is downloaded or streamed on-demand to users.

17. As a parallel research theme, starting in 1997, I began researching issues related to wireless devices and sensors. In particular, I was interested in showing how to provide greater communication capability to "lightweight devices," i.e., small form-factor, resource-constrained (e.g., CPU, memory, networking, and power) devices. Starting in 1998, I published several papers on my work to develop a flexible, lightweight, battery-aware network protocol stack. The lightweight protocols we envisioned were similar in nature to protocols like Bluetooth, Universal Plug and Play (UPnP) and Digital Living Network Alliance (DLNA).

18. From this initial work, I have made wireless networking—including ad hoc, mesh networks and wireless devices—one of the major themes of my research. My work in wireless networks spans the protocol stack from applications through to the encoding and exchange of data at the data link and physical layers.

19. At the application layer, even before the large-scale "app stores" were available, my research looked at building, installing, and using apps for a variety of purposes, from network monitoring to support for traditional computer-based

applications (e.g., content retrieval) to new applications enabled by ubiquitous, mobile devices. For example, my research has looked at developing applications for virtually exchanging and tracking “coupons” through “opportunistic contact” among mobile wireless devices (i.e., communication among devices moving into communication range with each other). In many of the courses I have taught there is a project component. Through these projects I have supervised numerous efforts to develop new “apps” for download and use across a variety of mobile platforms.

20. Toward the middle of the protocol stack, my research has also looked to build wireless infrastructure support to enable communication among a set of mobile devices unaided by any other kind of network infrastructure. These kinds of networks are useful either in challenged network environments (e.g., when a natural disaster has destroyed existing infrastructure) or when suitable support for network communication never existed. The deployment of such networks (or even the use of traditional network support) are critical to support services like disaster relief, catastrophic event coordination, and emergency services deployment.

21. Yet another theme is monitoring wireless networks, in particular different variants of IEEE 802.11 compliant networks, to (1) understand the operation of the various protocols used in real-world deployments, (2) use these measurements to characterize use of the networks and identify protocol limitations and weaknesses, and (3) propose and evaluate solutions to these problems. I have

successfully used monitoring techniques to study wireless data link layer protocol operation and to improve performance by enhancing the operation of such protocols. For wireless protocols, this research includes functions like network acquisition and channel bonding.

22. One theme in my wireless network research has been cross-layer solutions and innovations. As mentioned above, with greater wireless device use and network support, we envisioned new application paradigms and services, for example, when mobile devices come into contact with each other. Instead of relying on existing infrastructure to relay communication, the devices are able to discover each other and communicate directly. Other examples include discovering and using location information to enhance users' experiences. Network support and novel applications span use a variety of network architectures supporting users on foot, in vehicles, and across varying terrains and environments. Finally, we studied how communication efficiency can be supported through intelligent handoffs as well as location and movement prediction.

23. Protecting networks, including their operation and content, has been an underlying theme of my research almost since the beginning of my research career. Starting in 2000, I have been involved in several projects that specifically address security, network protection, and firewalls. After significant background work, a team on which I was a member successfully submitted a \$4.3M grant proposal to the

Army Research Office (ARO) at the Department of Defense to propose and develop a high-speed intrusion detection system. Key aspects of the system included associating streams of packets and analyzing them for viruses and other malware. Once the grant was awarded, we spent several years developing and meeting the milestones of the project. A number of my students worked on related projects and published papers on topics ranging from intrusion detection to developing advanced techniques to be incorporated into firewalls. I have also used firewalls, including their associated malware detection features, in developing techniques for the classroom to ensure that students are not distracted by online content.

24. My recent work ties some of the various threads of my past research together. I have investigated content delivery in online social networks and proposed reputation management systems in large-scale social networks and marketplaces. On the content delivery side, I have looked at issues of caching and cache placement, especially when content being shared and the cache has geographical relevance. We were able to show that effective caching strategies can greatly improve performance and reduce deployment costs. Our work on reputation systems showed that reputations have economic value, and as such, creates a motivation to manipulate reputations. In response, we developed a variety of solutions to protect the integrity of reputations in online social networks. The techniques we developed for content delivery and reputation management were

particularly relevant in peer-to-peer communication and recommendations for downloadable “apps.”

25. As an important component of my research program, I have been involved in the development of academic research into available technology in the market place. One aspect of this work is my involvement in the Internet Engineering Task Force (IETF). The IETF is a large and open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. I have been involved in various IETF groups including many content delivery-related working groups like the Audio Video Transport (AVT) group, the MBone Deployment (MBONED) group, Source Specific Multicast (SSM) group, the Inter-Domain Multicast Routing (IDMR) group, the Reliable Multicast Transport (RMT) group, the Protocol Independent Multicast (PIM) group, etc. I have also served as a member of the Multicast Directorate (MADDOGS), which oversaw the standardization of all things related to multicast in the IETF. Finally, I was the Chair of the Internet2 Multicast Working Group for seven years.

26. My involvement in the research community extends to leadership positions for several academic journals and conferences. I am the co-chair of the Steering Committee for the ACM Network and System Support for Digital Audio and Video (NOSSDAV) workshop and on the Steering Committees for the

International Conference on Network Protocols (ICNP), ACM Sigcomm Workshop on Challenged Networks (CHANTS), and IEEE Global Internet (GI) Symposium. I have served or am serving on the Editorial Boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Network, ACM Computers in Entertainment, AACE Journal of Interactive Learning Research (JILR), and ACM Computer Communications Review. I have co-chaired a number of conferences and workshops including the IEEE International Conference on Network Protocols (ICNP), IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), International Conference on Communication Systems and Networks (COMSNETS), IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS), the International Workshop On Wireless Network Measurement (WiNMee), ACM Sigcomm Workshop on Challenged Networks (CHANTS), the Network Group Communication (NGC) workshop, and the Global Internet Symposium, and I have served on the program committees for numerous conferences.

27. Furthermore, in the courses I taught at UCSB, a significant portion of my curriculum covered aspects of the Internet and network communication including the physical and data link layers of the Open System Interconnect (OSI) protocol stack, and standardized protocols for communicating across a variety of physical media such as cable systems, telephone lines, wireless, and high-speed

Local Area Networks (LANs). The courses I have taught also cover most major topics in Internet communication, including data communication, multimedia encoding, and mobile application design. My research and courses have covered a range of physical infrastructures for delivering content over networks, including cable, Integrated Services Digital Network (ISDN), Ethernet, Asynchronous Transfer Mode (ATM), fiber, and Digital Subscriber Line (DSL). For a complete list of courses I have taught, see my curriculum vitae (CV).

28. In addition, I co-founded a technology company called Santa Barbara Labs that was working under a sub-contract from the U.S. Air Force to develop very accurate emulation systems for the military's next generation internetwork. Santa Barbara Labs' focus was in developing an emulation platform to test the performance characteristics of the network architecture in the variety of environments in which it was expected to operate, and, in particular, for network services including IPv6, multicast, Quality of Service (QoS), satellite-based communication, and security. Applications for this emulation program included communication of a variety of multimedia-based services, including video conferencing and video-on-demand.

29. In addition to having co-founded a technology company myself, I have worked for, consulted with, and collaborated with companies for nearly 30 years. These companies range from well-established companies to start-ups and include

IBM, Hitachi Telecom, Turner Broadcasting System (TBS), Bell South, Digital Fountain, RealNetworks, Intel Research, Cisco Systems, and Lockheed Martin.

30. Through my graduate education, leadership with CITS, involvement in TMP, role in the development of the Internet2 infrastructure, and consulting with ISPs, I have gained a strong understanding in the role of the Internet in our society and the challenges of deploying large-scale production networking infrastructure. CITS, since its inception, has looked at the role of the Internet in society, including how the evolution of technology have created communication opportunities and challenges, including, for example through disruptive technologies like P2P. TMP looks to focus on non-purely technical issues, including, for example, state-of-the-art business methods, strategies for successful technology commercialization, new venture creation, and best practices for fostering innovation. Through my industry collaborations and Internet2 work, I have developed significant experience in the challenges of deploying, monitoring, managing, and scaling communication infrastructure to support evolving Internet services like streaming media, conferencing, content exchange, social networking, and e-commerce.

31. I am a Member of the Association of Computing Machinery (ACM) and a Fellow of the Institute of Electrical and Electronics Engineers (IEEE).

III. LEVEL OF ORDINARY SKILL IN THE ART

32. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

33. A Person of Ordinary Skill in the Art (“POSITA”) as of January 31, 2001, would have had 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 could compensate for work experience, and vice versa.

34. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my own experience and what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the priority date of the ’139 Patent.

IV. RELEVANT LEGAL STANDARDS

35. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the '139 Patent, I am relying on certain basic legal principles that counsel has explained to me.

36. I understand that prior art to the '139 Patent includes patents and printed publications in the relevant art that predate the priority date of the '139 Patent. For purposes of this Declaration, I am applying January 31, 2001, as the priority date of the '139 Patent, but I take no position on that issue. Even if the Patent Owners contend the date of invention is earlier than January 31, 2001, that would not change my opinions to the extent that would not disqualify any of the applied references as prior art.

37. I have been informed by counsel that a claimed invention is invalid if it is “anticipated” by the prior art. I understand that, for a patent claim to be “anticipated” by the prior art, each and every requirement of the claim must be found, expressly or inherently, in a single prior art reference as recited in the claim. I understand that claim limitations that are not expressly described in a prior art reference may still be there if they are “inherent” to the thing or process being described in the prior art.

38. I understand that if a reference incorporates other documents by reference in their entirety, the incorporating reference and the incorporated reference(s) should be treated as a single prior art reference.

39. I understand that it is acceptable to consider evidence other than the information in a particular prior art document to determine if a feature is necessarily present in or inherently described by that reference.

40. I have been informed by counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been informed by counsel that the obviousness analysis considers factual inquiries, including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

41. I have been further informed by counsel that there are several recognized rationales for combining references or modifying a reference to show obviousness. These rationales include: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement

to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art or to combine prior art teachings to arrive at the claimed invention.

V. OVERVIEW OF THE '139 PATENT

42. The '139 Patent issued from the standalone non-provisional Application No. 09/774,839, which was filed on January 31, 2001. I have been asked to assume that January 31, 2001 is the effective filing date, but I take no position on that issue.

43. I have reviewed the file history of the 139 Patent and note that none of the activity there is particularly pertinent to my analysis.

44. 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.

45. 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.

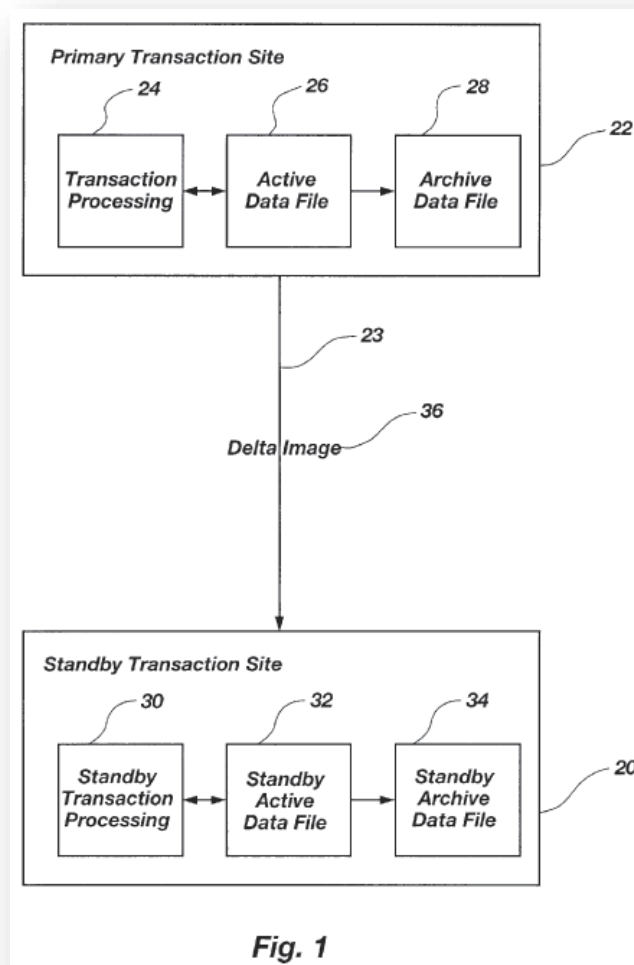
46. The '139 Patent explains that multiple systems can also safeguard against data corruption because the data can be stored in both the primary and secondary systems. Ex-1001, 1:49-55. The '139 Patent recognizes that systems keep transaction logs, such as an “active log” for tracking recent transactions, 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.

47. The primary and backup (standby) systems are remotely located from one another to improve reliability in the event one system goes down. Ex-1001,

Ex-1003

1:37-47; 1:67-2:1 (explaining the “standby system is always ready to take over” and that the “standby system is located in another building or in a geographically remote area”). 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.

48. 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):



Ex-1001, Fig. 1.

49. 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.

50. The system of the '139 Patent includes a first primary cluster node 50, a second primary cluster not 52, each cluster 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):

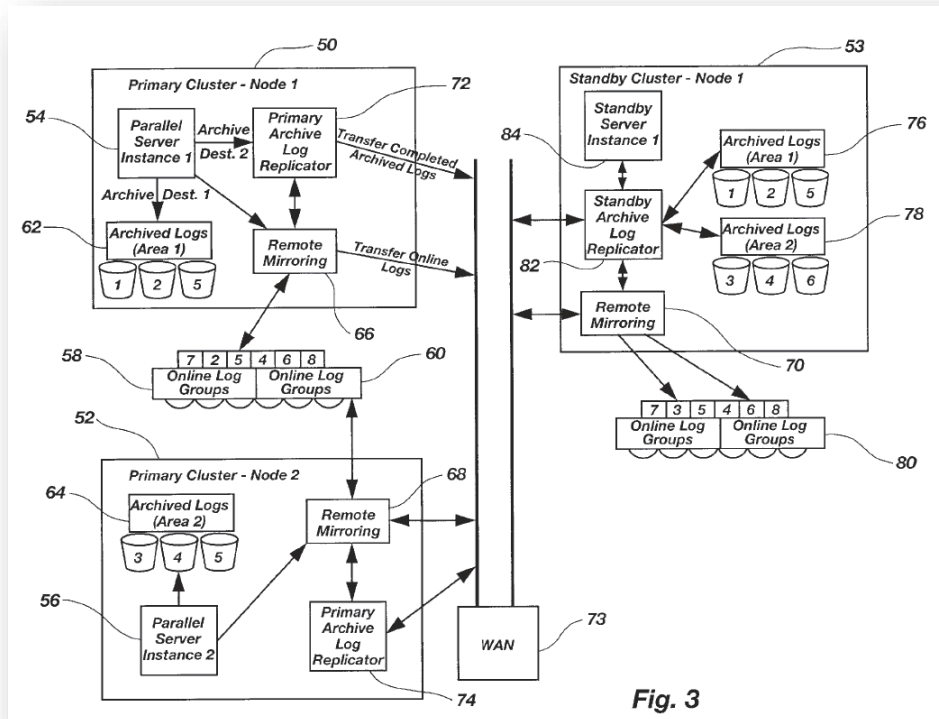


Fig. 3

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

51. 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

Ex-1003

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 (describing transmission of archive files using “a technique wherein only the difference between the archive log file and the corresponding online redo file is sent across to the standby site”).

VI. CLAIM CONSTRUCTION

52. It is my understanding that in order to properly evaluate the '139 Patent, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this *inter partes* review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the unpatentability inquiry.

53. I do not believe that any claim constructions would affect my analysis or conclusion that the claims are obvious over the asserted prior art.

VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

54. The discussion in this Declaration provides a detailed analysis of how the asserted prior art would have rendered obvious each limitation of the Challenged Claims.

55. As part of my analysis, I have considered, and discuss in detail, the scope and content of the prior art and any differences between the alleged invention and the prior art. As detailed below, the scope and content of the Srinivasan prior art reference alone or in combination with the Taylor prior art reference includes every element of the asserted claims without any meaningful difference between the teachings of those references and the elements of the asserted claims.

56. I am not aware of any secondary considerations. The Patent Owners have not identified any secondary considerations of nonobviousness so I have not considered any in my analysis.

57. I have considered the teachings in the prior art from the perspective of a POSITA and it is my conclusion that a POSITA would have found it obvious to combine the teachings of those known elements in the way they are recited in the Challenged Claims as of January 31, 2001.

58. Regardless of whether secondary considerations exist or not, they would not rise to the level to overcome the strong obviousness arguments presented below.

59. I do not believe any secondary considerations would alter my overall conclusion that the claims would have been obvious to a POSITA as of January 31, 2001. Nevertheless, should the Patent Owners provide secondary

considerations and argue those considerations rise to the level of maintaining the validity of the Challenged Claims, I reserve the right to respond.

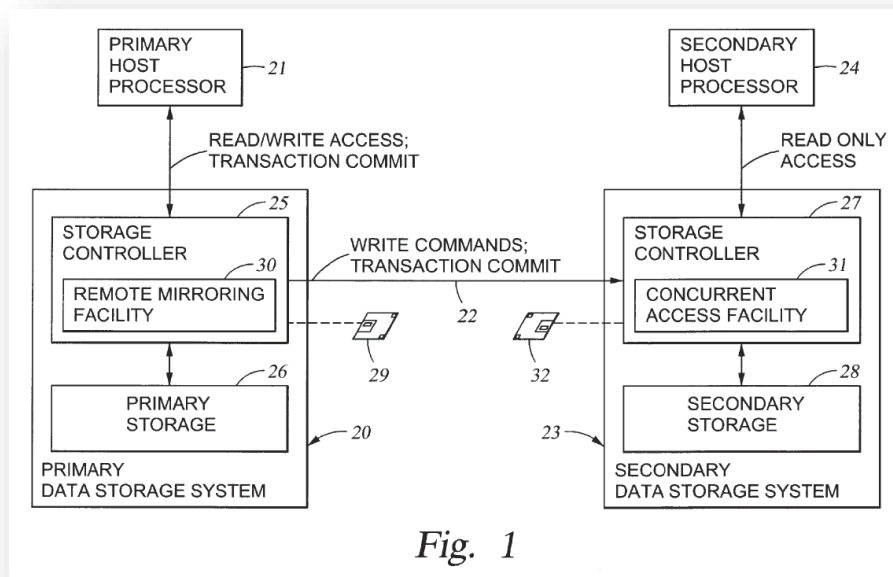
VIII. THE PRINCIPAL PRIOR ART

A. Summary of Srinivasan (Ex-1005)

60. The application underlying U.S. Patent No. 6,823,336 to Srinivasan et al. ("Srinivasan") was filed on September 26, 2000, and published on November 23, 2004. I understand that Srinivasan is therefore prior art to the '139 Patent at least under pre-AIA 35 U.S.C. § 102(e).

61. Srinivasan "relates generally to data storage systems," that are "for transaction processing applications where one host processor has read-only access to a dataset concurrent with read-write access by another host processor." Ex-1005, 1:11-15.

62. 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):



Ex-1005, 5:16-20, Fig. 1.

63. Srinivasan's Figure 8 shows a preferred construction of the data processing system of Figure 1:

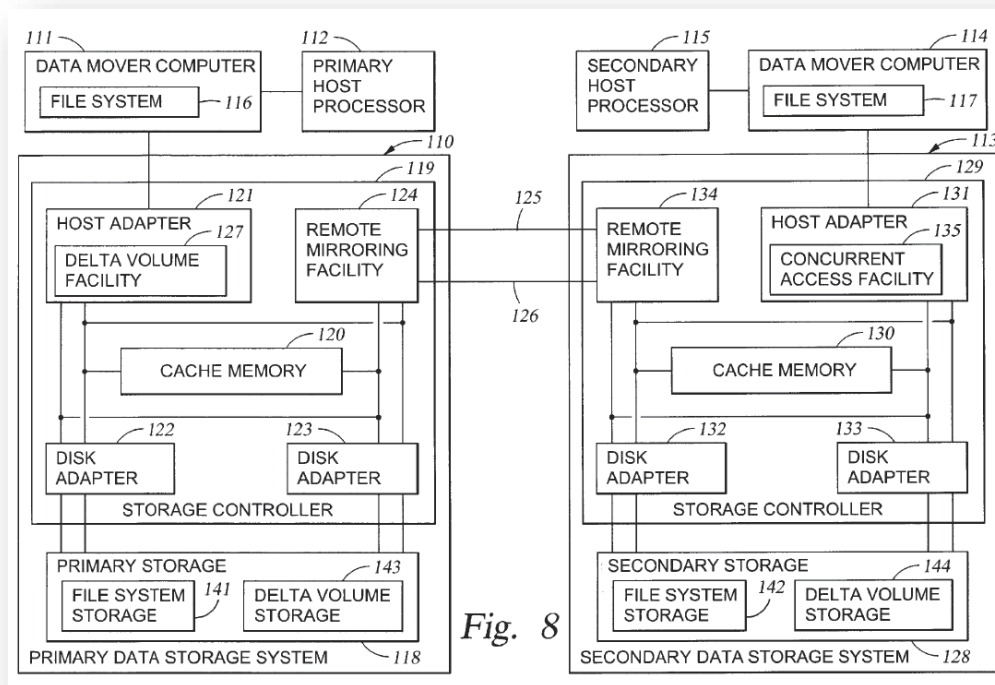


Fig. 8

Ex-1005, 4:52-53, 9:51-54; Fig. 8 (annotated).

64. 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.

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

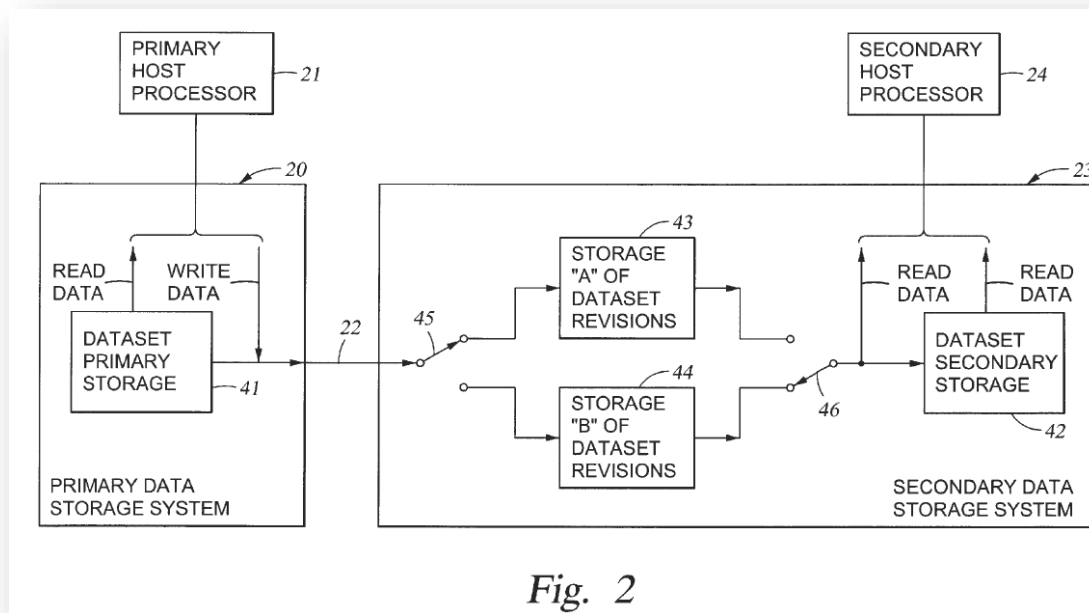
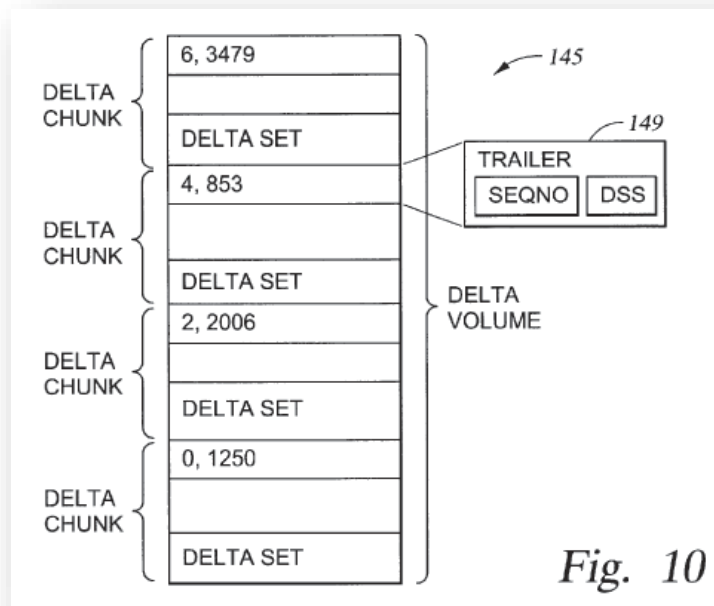


Fig. 2

Ex-1005, 5:56-57; Fig. 2.

66. 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.

67. 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):

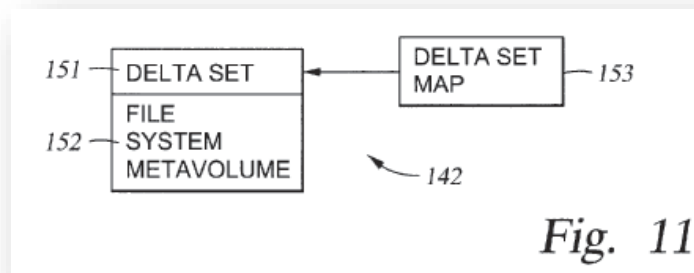


Ex-1005, 11:22-33; Fig. 10.

68. 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.

69. 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):



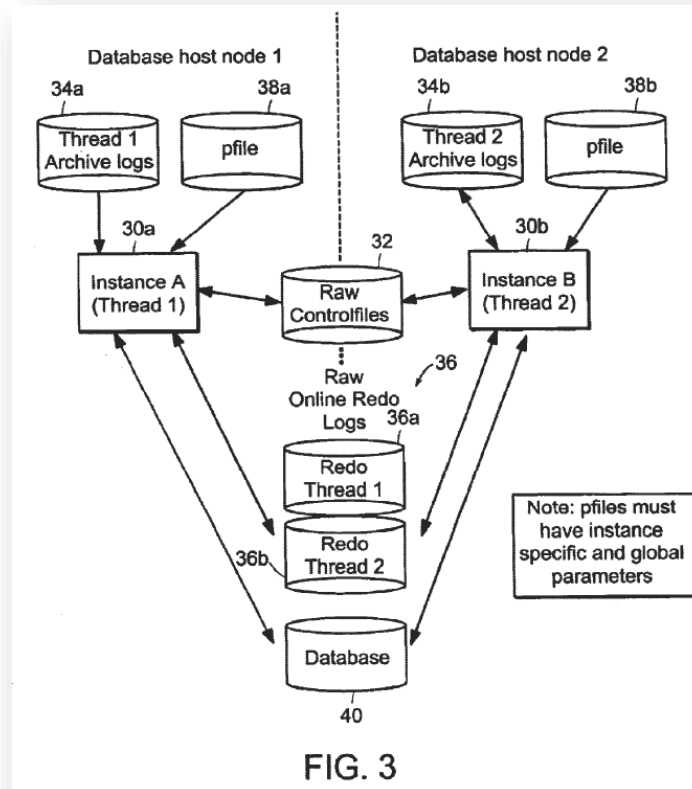
Ex-1005, 11:57-61; Fig. 11, 4:52-58 (explaining Figure 11 is a block diagram of data structures in the secondary storage of the secondary data storage system in Figure 8).

B. Summary of Taylor (Ex-1006)

70. The application underlying U.S. Patent No. 6,658,589 to Taylor (“Taylor”) was filed on December 20, 1999, and published on December 2, 2003. I understand that Taylor is therefore prior art to the ’139 Patent under at least 35 U.S.C. § 102(e).

71. Taylor teaches “[a] system and method for safe and effective backup and restore of parallel server databases stored in data storage systems.” Ex-1006,

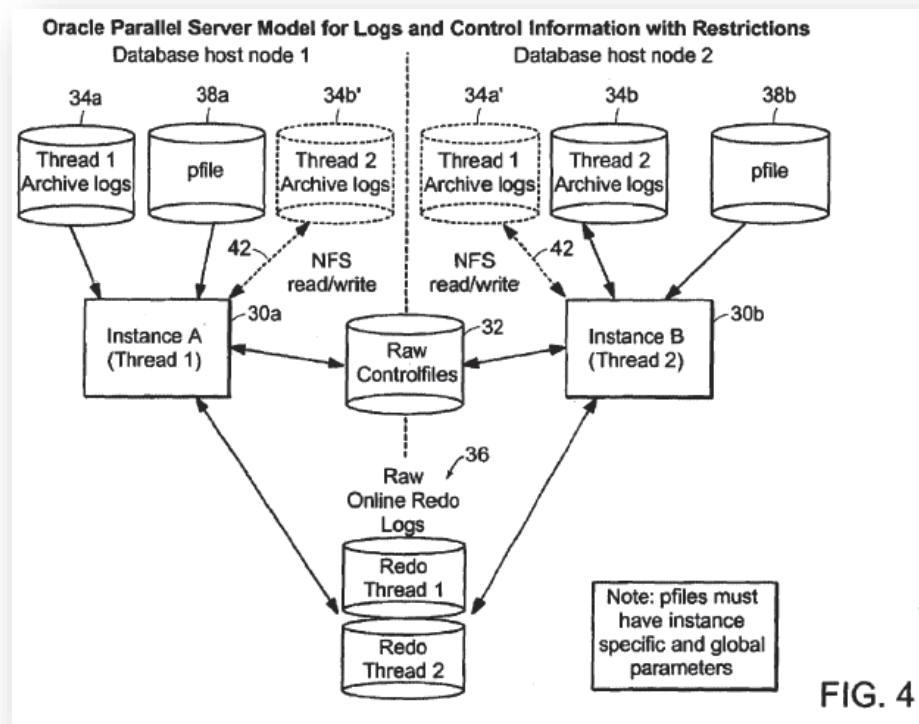
Abstract. As shown in Figure 3 below, a parallel server database architecture generally includes instances (30a, 30b) that can independently access a database 40, archive logs 34 and online redo logs 36:



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

72. 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):



Ex-1006, 3:3-11, 5:27-28, 44-46; FIG. 4 (annotated).

73. 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.

IX. DETAILED ANALYSIS THAT CLAIMS 1-18 ARE OBVIOUS OVER SRINIVASAN IN VIEW OF TAYLOR

74. For the reasons below, Srinivasan and Taylor would have rendered obvious claims 1-18.

75. 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.

76. It would have been obvious to organize the transaction logs of Srinivasan into “online” redo logs, and “archive” redo logs, as taught by Taylor. 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”).

77. 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-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-1006, 2:67-3:8.

78. 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-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-1006, 3-11.

79. 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-1001, 2:5-7, Ex-1005, 6:51-64, 10:62-11:2, Ex-1006, 4:38-42.

80. 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-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), 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.

81. 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. *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. 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-1007, 71-76 (describing latency associated with the storage type, including (1) disk storage involving a “seek time,” followed by a “rotational delay” or “latency,” and then “block transfer time,” and (2) magnetic tape storage that “accesses data blocks in sequential order”), 466-469 (explaining criteria for physical database design, including “Response time: ... the elapsed time between submitting a database transaction for execution and receiving a response,” which is “influenced by ... system load, operation system scheduling, or communication delays” and “Transaction throughput: ... the average number of transactions that can be processed per minute by the database system ... measured at peak conditions”).

82. 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. 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. *See, e.g.*, Ex-1007, at 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). 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-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. *See, e.g.,* Ex-1007, 527-553 (describing transaction processing concepts), 577-592 (explaining recovery techniques). 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.

83. 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-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*”).

84. 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.

85. First, Srinivasan teaches how to avoid issues with data coherence. 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-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. 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.

86. 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-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.

87. 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*”). For example, Srinivasan teaches that the 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). 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-1005, 11:57-61 (explaining overlaying delta volume delta set directory on transaction log data structure reduces computational overhead).

88. The ’139 Patent also 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 also 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.

89. 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. 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-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).

90. 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-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).

91. 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-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).

92. 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, 15:63-16:1 (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),

17:34-38 (explaining each data storage system can be directly linked to multiple host processors).

93. 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. Thus, a POSITA would have reasonably expected to succeed in combining the teachings of Srinivasan and the teachings of Taylor.

A. Claim 1

1. ***[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:”***

94. For purposes of this analysis, I have been told to assume that the preamble is limiting for this claim.

95. 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”).

96. 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):

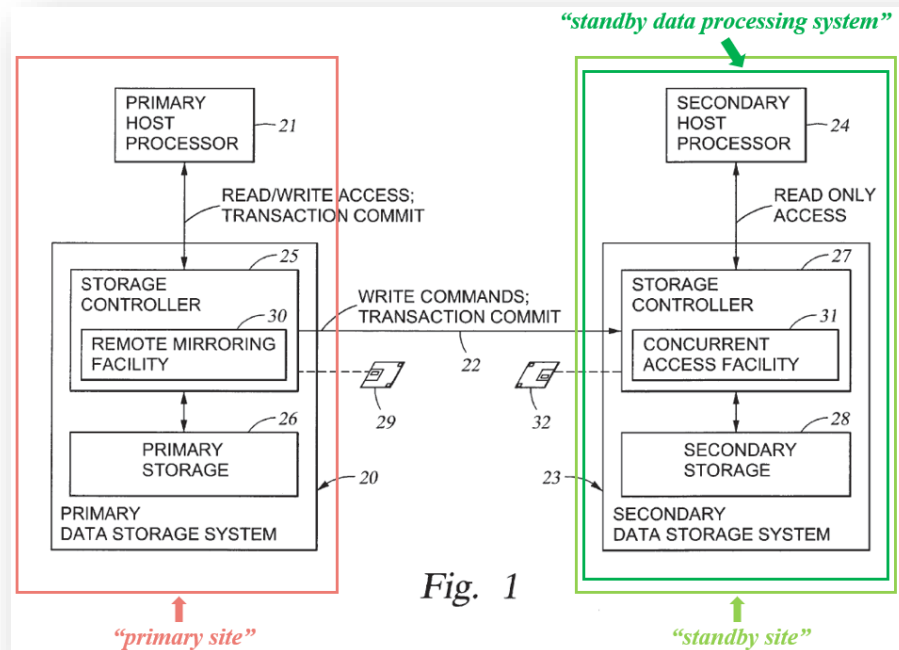
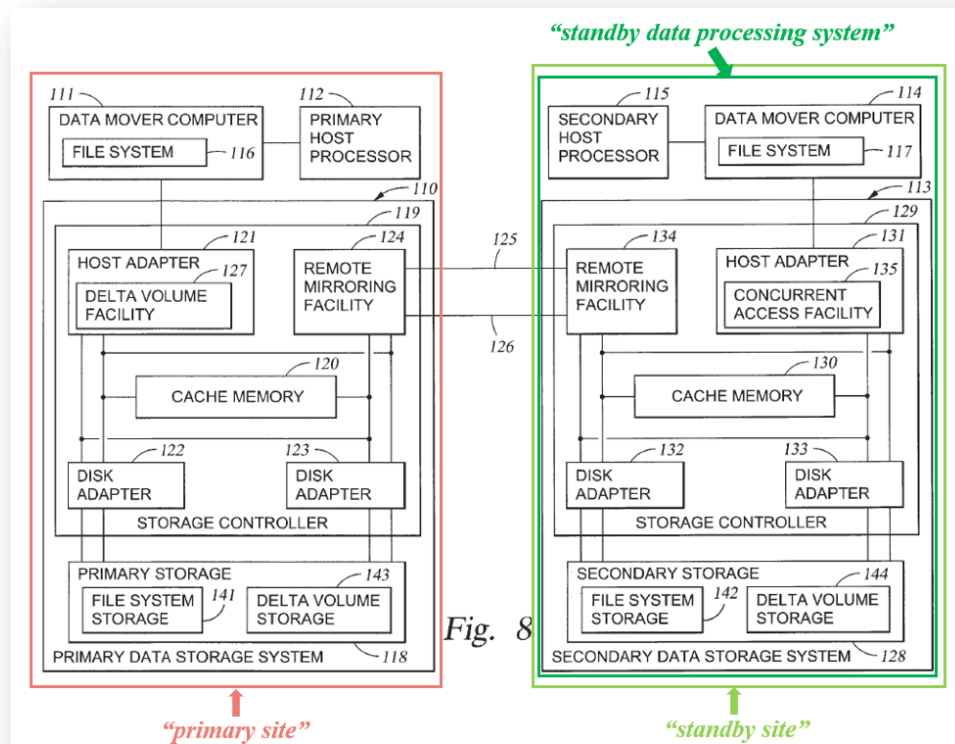


Fig. 1

Ex-1005, Fig. 1 (annotated).

97. 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):



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

98. 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-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

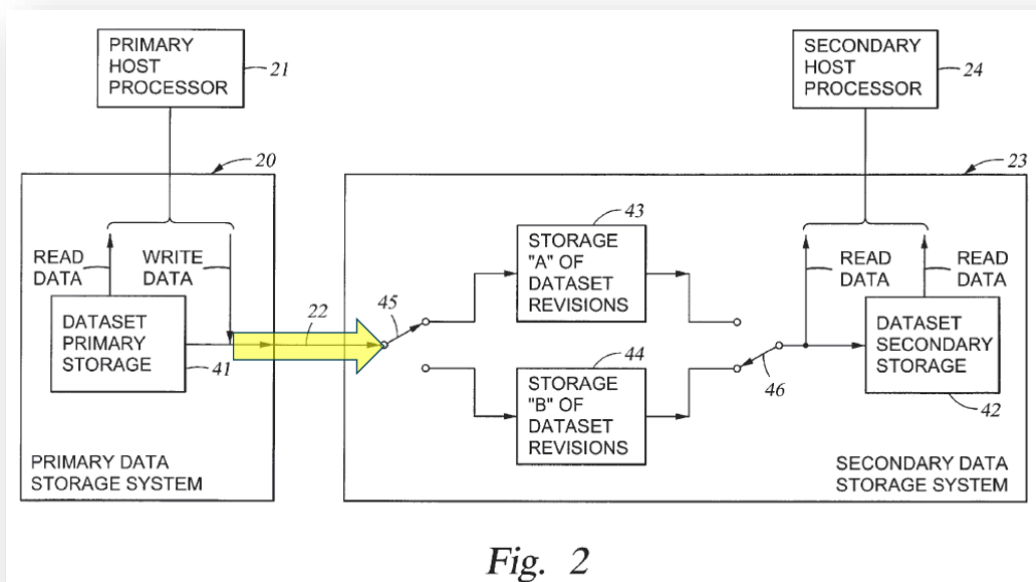
(explaining “the combination of the secondary data storage system 113 and the data mover computer 113 also functions as a file server”).

99. The “*standby site*” of the ’139 Patent can include a secondary or backup site that is remotely located from a primary site. 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).

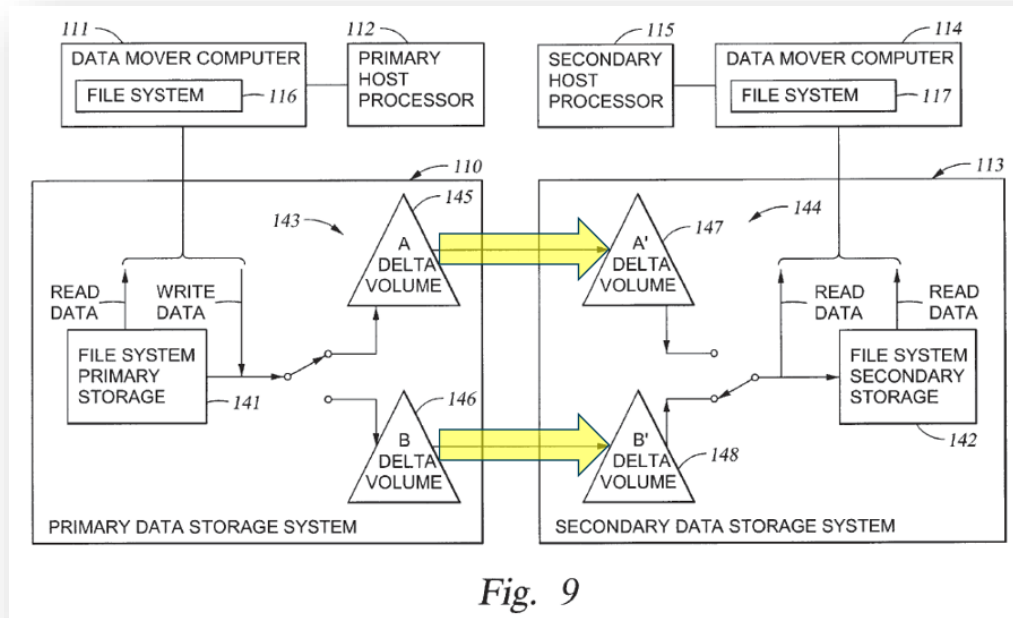
100. The “*standby data processing system*” described in Srinivasan is “located at a *standby site*, which is remotely located from a *primary site*,” consistent with how a “*standby site*” is used in the ’139 Patent, because the write commands received at the “*primary site*” are remotely mirrored, via one or more transmission links to the “*standby data processing system*” at the “*standby site*.” Ex-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*”).

101. 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-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).

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



Ex-1005, Fig. 2 (annotated).



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

103. Accordingly, Srinivasan taught or suggested the preamble.

2. ***[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;”***

104. This element would have been obvious based on the combined teachings of Srinivasan and Taylor. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

105. As discussed in ¶¶74-93, it would have been obvious to organize the transaction logs of Srinivasan into online redo logs, and archive redo logs, as taught by Taylor.

106. 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-1001, 1:56-60 (explaining an “active transaction log is kept which tracks recent transactions” and an “archive log is then kept to store information from the active log after the active log has been filled or a certain time period has passed”), 4:47-49 (explaining “each node will have its own copy of the online redo logs 58, 60 (i.e., an active data file)”), 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)”).

107. 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*

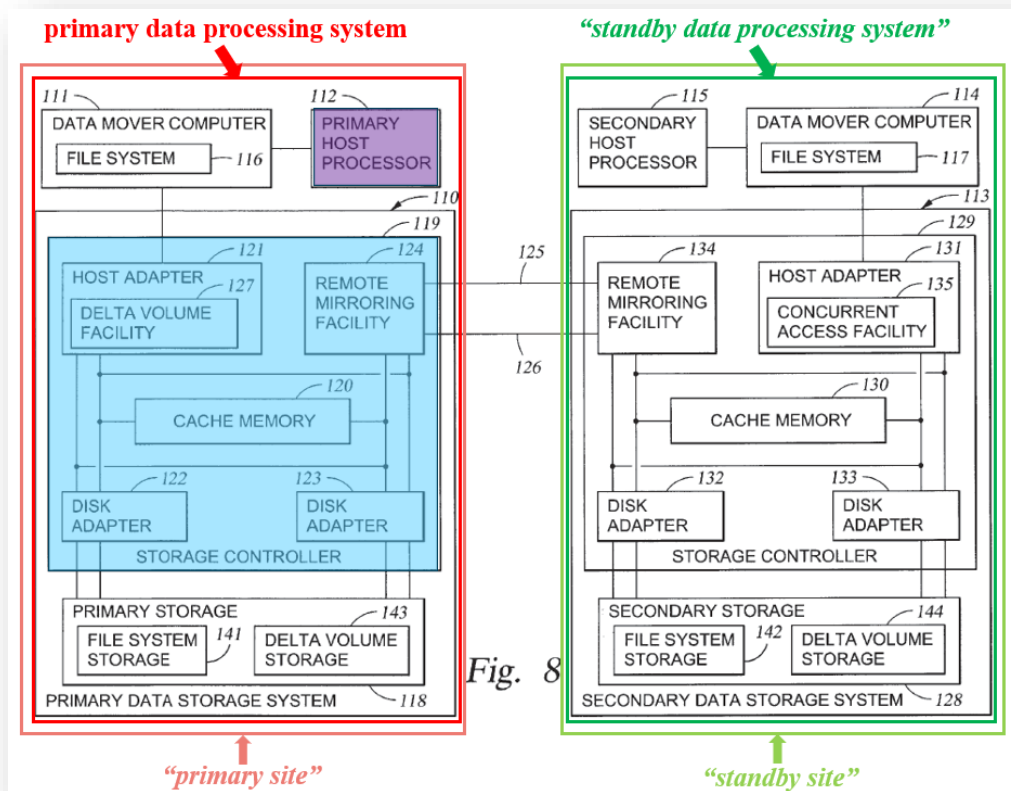
transaction redo logs”) are redo logs that have been archived by being spooled to disk. Ex-1006, 2:67-3:8.

108. 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-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).

109. 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 ¶¶74-93. 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*”)

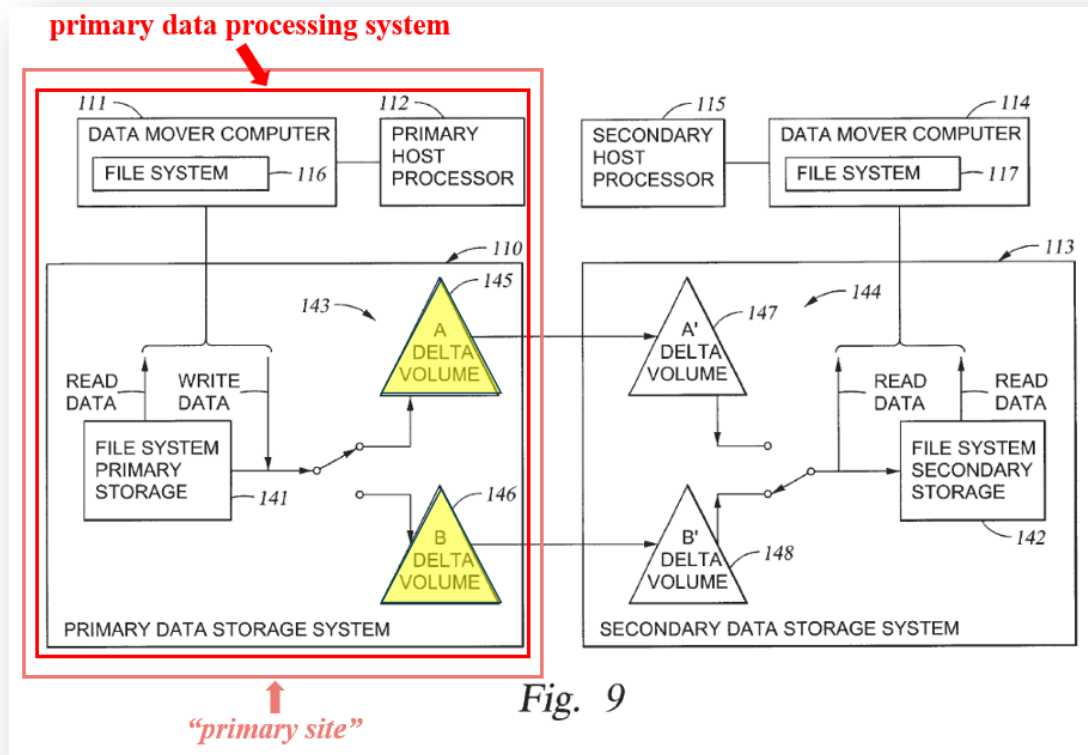
containing changes (updates) to the system at the “*primary site*” by comparing the differences between online redo logs (“*active transaction redo logs*”) and archive redo logs (“*archive transaction redo logs*”). See ¶¶74-93. And it would have been obvious to do so for the reasons previously discussed.

110. 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:



Ex-1005, Fig. 8 (annotated).

111. 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).

112. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this claim element.

3. ***[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”***

113. This element would have been obvious based on the combined teachings of Srinivasan and Taylor. A POSITA would have been motivated to

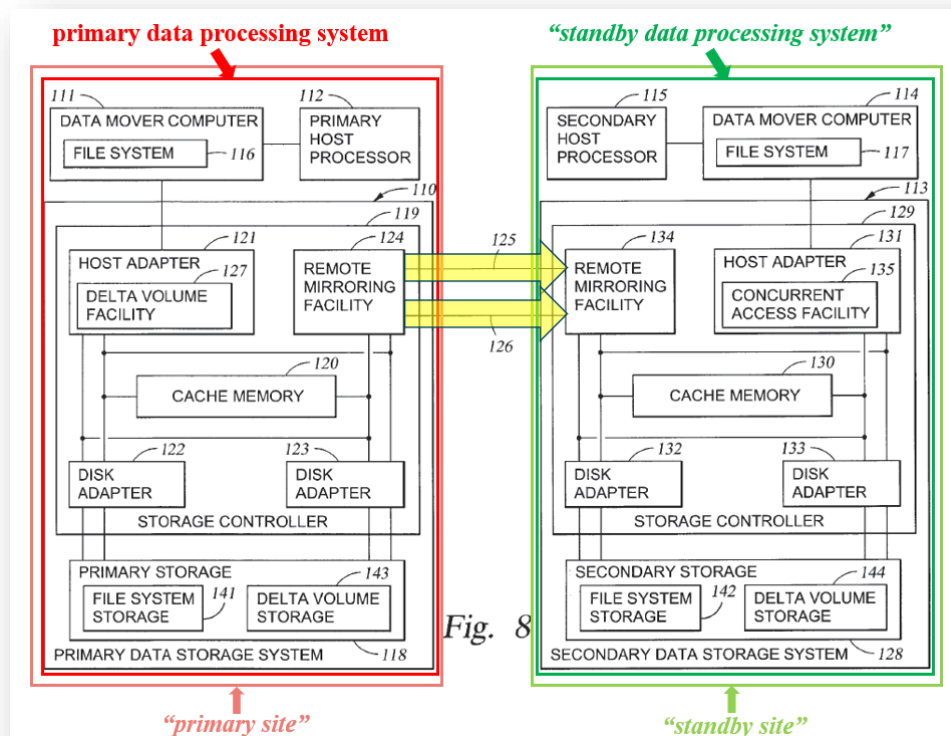
combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

114. 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 ¶¶74-93.

115. 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*.”

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



Ex-1005, Fig. 8 (annotated).

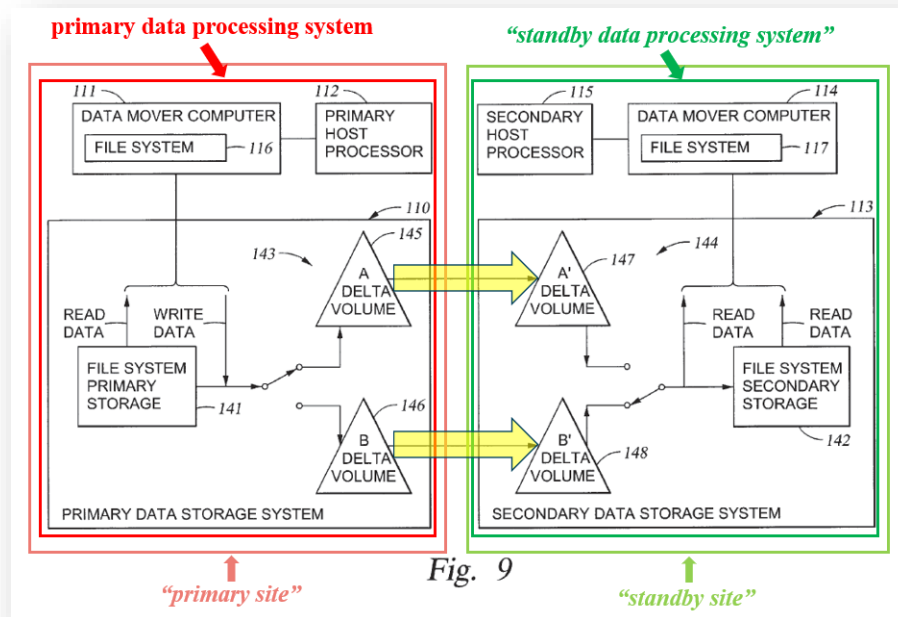
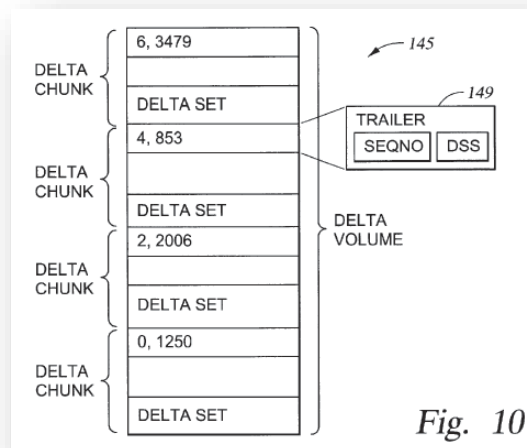


Fig. 9

Ex-1005, Fig. 9 (annotated).

117. 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:



Ex-1005, Fig. 10.

118. 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-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).

119. 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*.”

120. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this claim element.

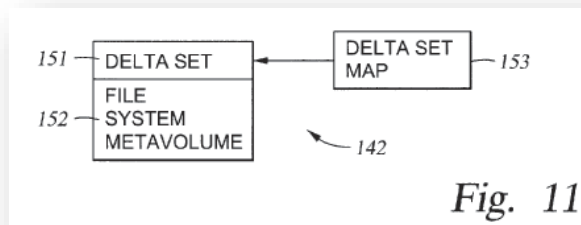
4. ***[1c]: “(c) combining the delta image with the active transaction redo log at the standby site, to produce a standby archive transaction redo log.”***

121. This element would have been obvious based on combining the teachings of Srinivasan and Taylor. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

122. 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*”).

123. Figure 11 of Srinivasan shows combined data structures at the “*standby site*”:



Ex-1005, Fig. 11.

124. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this claim element.

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

125. 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*.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

126. 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).

127. 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.

128. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 2.

C. 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.”

129. 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*.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

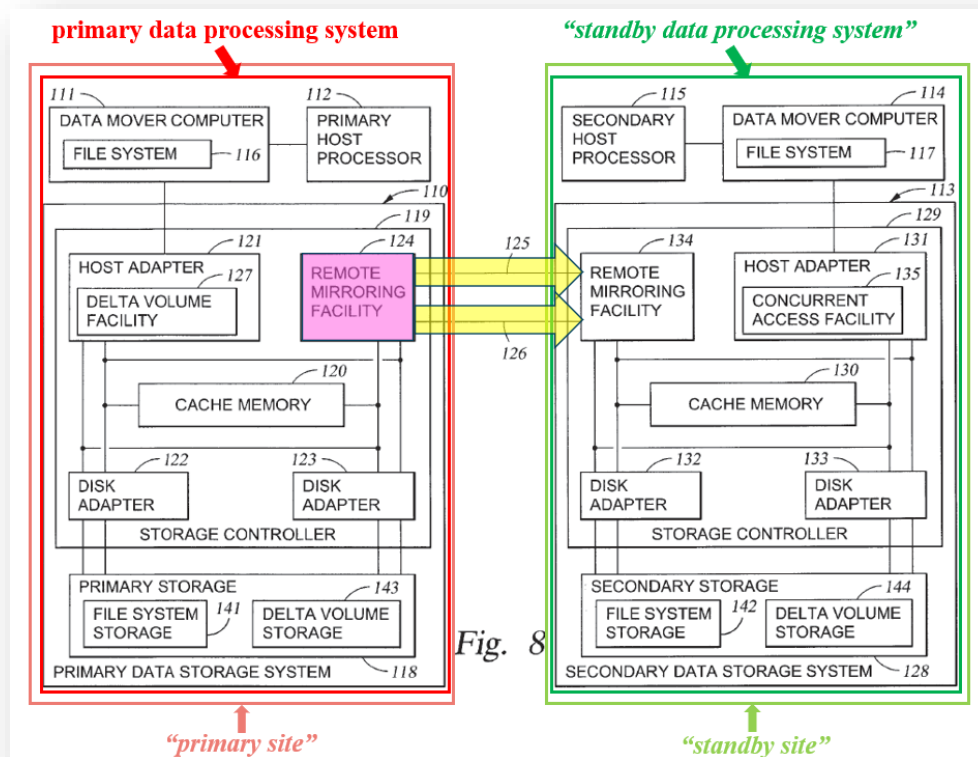
130. The ’139 Patent describes mirroring the online redo log using asynchronous remote mirroring technology that works by trapping all operating system writes for a storage device and mirroring those writes across disks on remote systems. Ex-1001, 5:30-35 (explaining remote mirroring technology that “can be a UNIX device driver that works by trapping all operating system writes for a storage device and mirroring these writes across disks on remote systems”), 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).

131. 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).

132. 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.

133. 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:



Ex-1005, Fig. 8 (annotated).

134. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 3.

D. 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.”

135. 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.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

136. 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 phase 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).

137. 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.

138. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 4.

E. Claim 5:

1. ***[5preamble]: “A method for creating archive redo logs for a standby database at a remote location, comprising the steps of:”***

139. For purposes of this analysis, I have been told to assume that the preamble is limiting for this claim.

140. As discussed above for claim 1, the combined teachings of Srinivasan and Taylor would have rendered obvious an “*archive redo log*.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

141. As discussed above in the claim [1preamble], 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*”).

142. 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*.”

143. The “*standby database*” at the “*remote location*” is shown in annotated Figure 8:

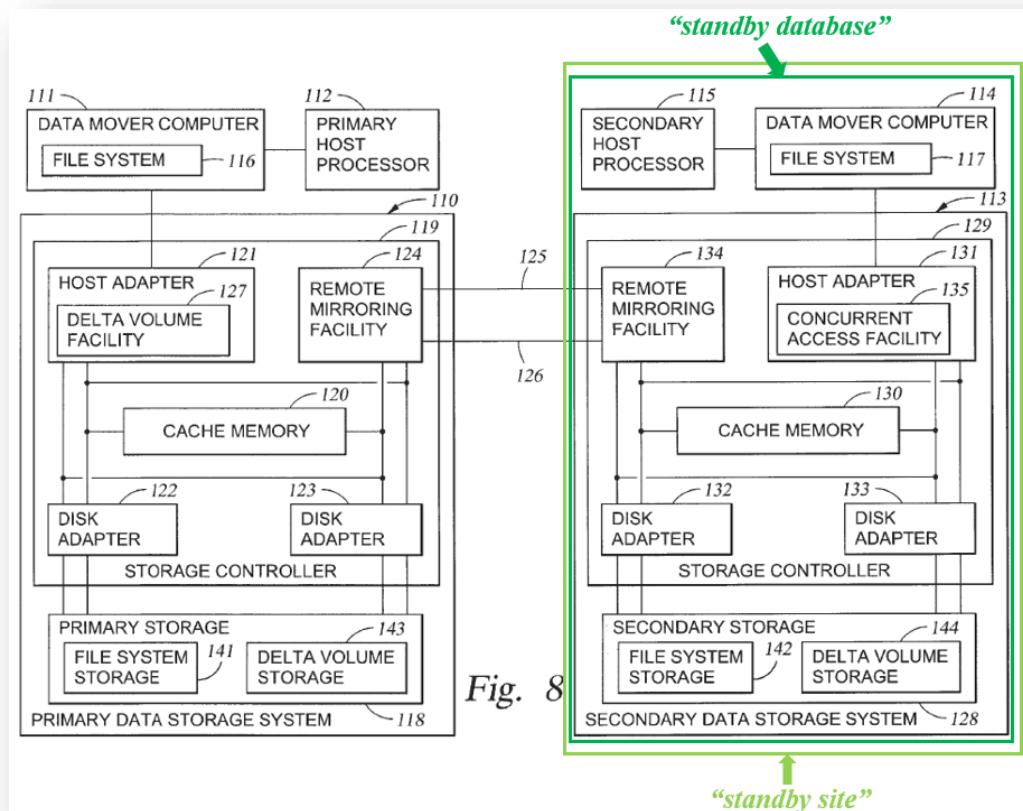


Fig. 8

Ex-1005, Fig. 8 (annotated).

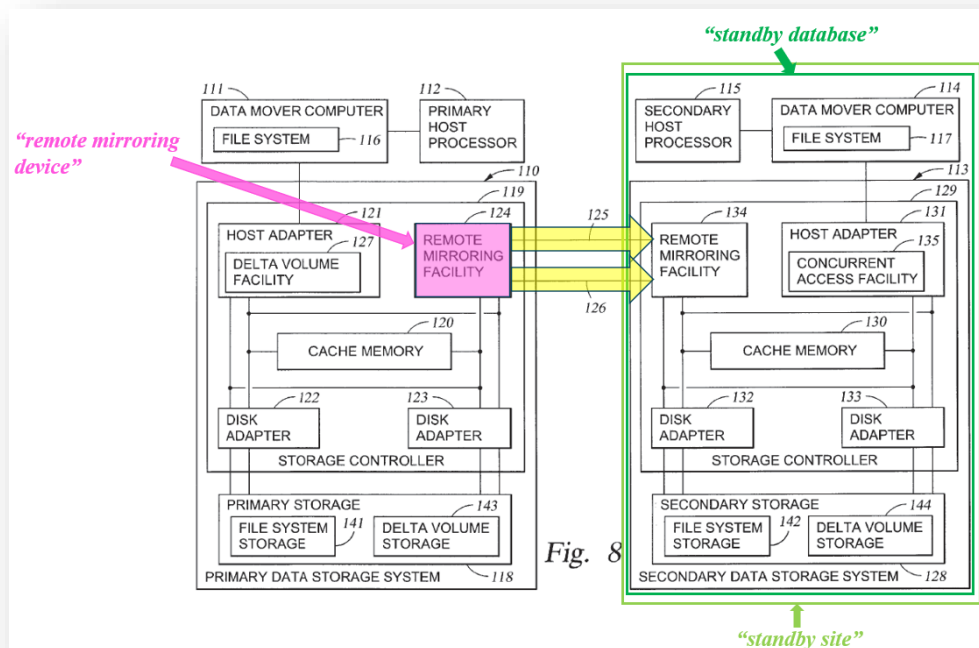
144. Additionally, as discussed above in claim [1c], the combined teachings of Srinivasan and Taylor would have rendered obvious creating archive redo logs (“archive redo logs”) as a result of the creation of “the *standby data processing system*.”

145. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this preamble.

2. [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;”

146. As discussed for claim 1, the combined teachings of Srinivasan and Taylor would have rendered obvious an “online redo log.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

147. 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):



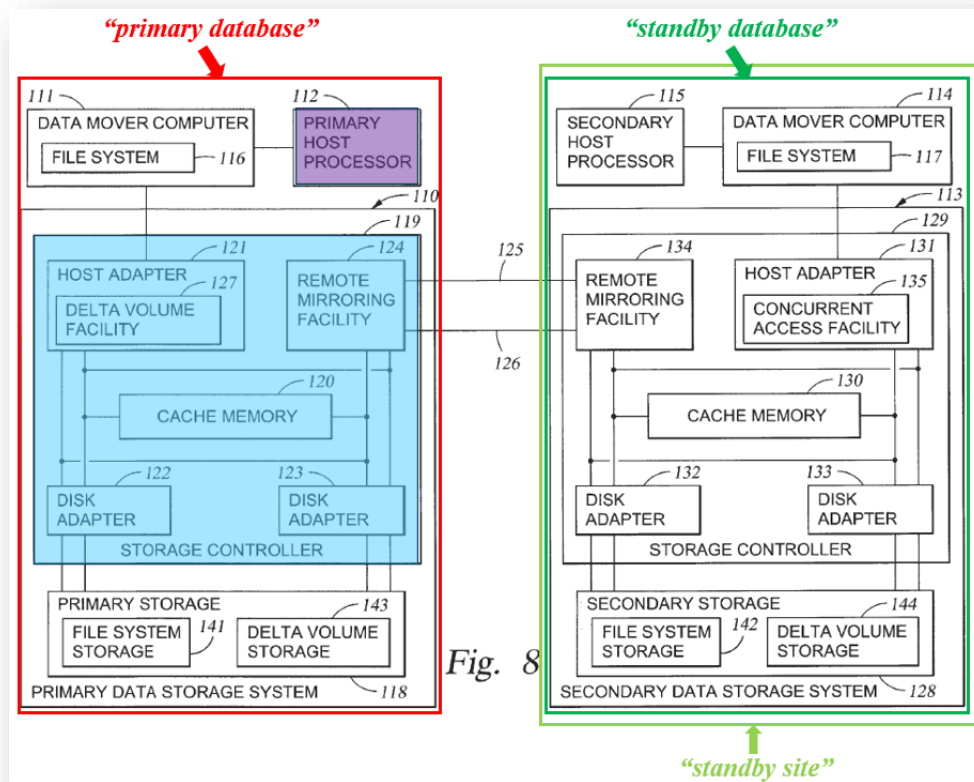
Ex-1005, Fig. 8 (annotated); *see* claim [3].

148. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this element.

3. ***[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;”***

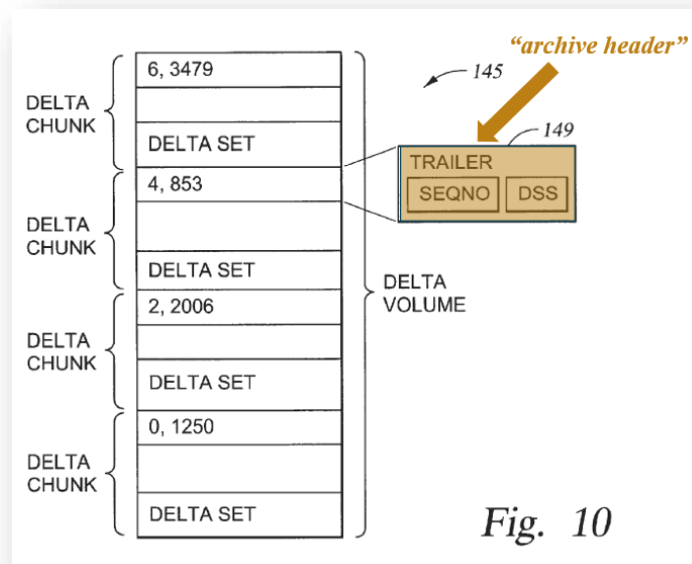
149. As discussed above for claim [1a], the combined teachings of Srinivasan and Taylor would have rendered obvious “*computing a difference between an online redo log and archive redo log to create a delta image at a **primary database**.*” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

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



Ex-1005, Fig. 8 (annotated).

151. 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*”):



Ex-1005, Fig. 10 (annotated).

152. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this element.

4. [5c]: “(c) transmitting the delta image to a standby archive log replicator contained in the standby database; and”

153. 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*.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

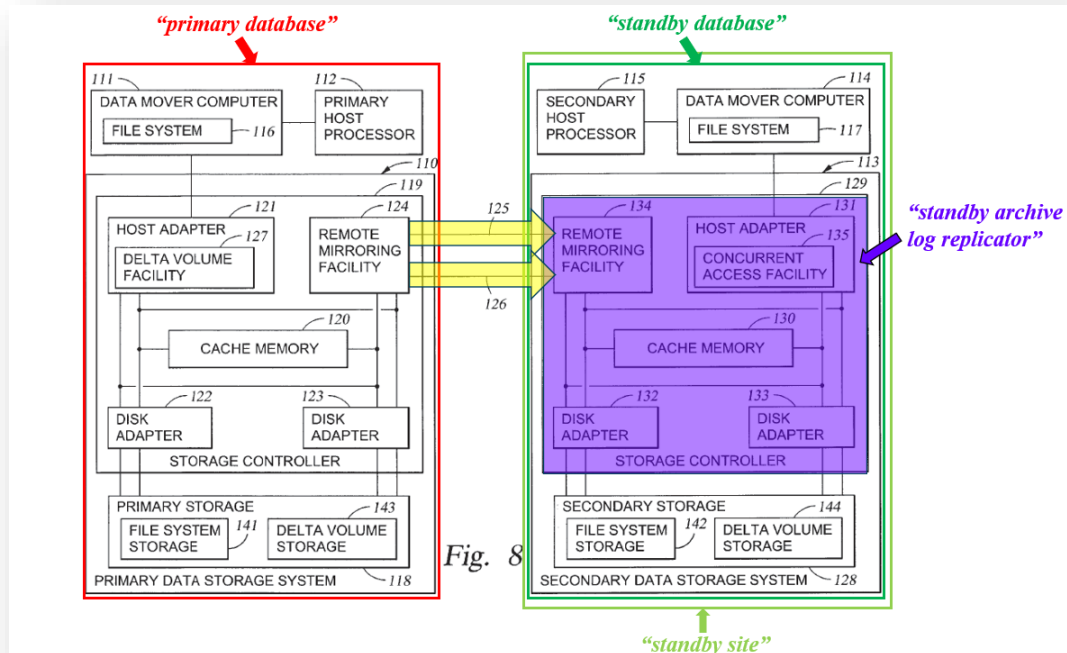
154. The '139 Patent explains that the standby archive log replicator 82 communicates with the remote mirroring component 70 using replication handles (sequence numbers) to confirm whether the corresponding online redo log has been

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.

155. 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).

156. 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.

157. Figure 8 of Srinivasan shows the storage controller 129 (“*standby archive log replicator*”):



Ex-1005, Fig. 8 (annotated).

158. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious this element.

5. **[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.”**

159. 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.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

160. 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].

161. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 5.

F. Claim 6: “The data redundancy system according to claim 1, wherein the first redundancy appliance is disjoint from the second redundancy appliance.”

162. 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.*” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

163. 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 logs*” as that term is used in the ’139 Patent.

164. As discussed above in ¶¶74-93, it would have been obvious to combine Taylor’s teachings for “*creating links to the standby archive redo logs*” (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).

165. 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).

166. 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.

167. 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.

168. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 6.

- G. 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.”**

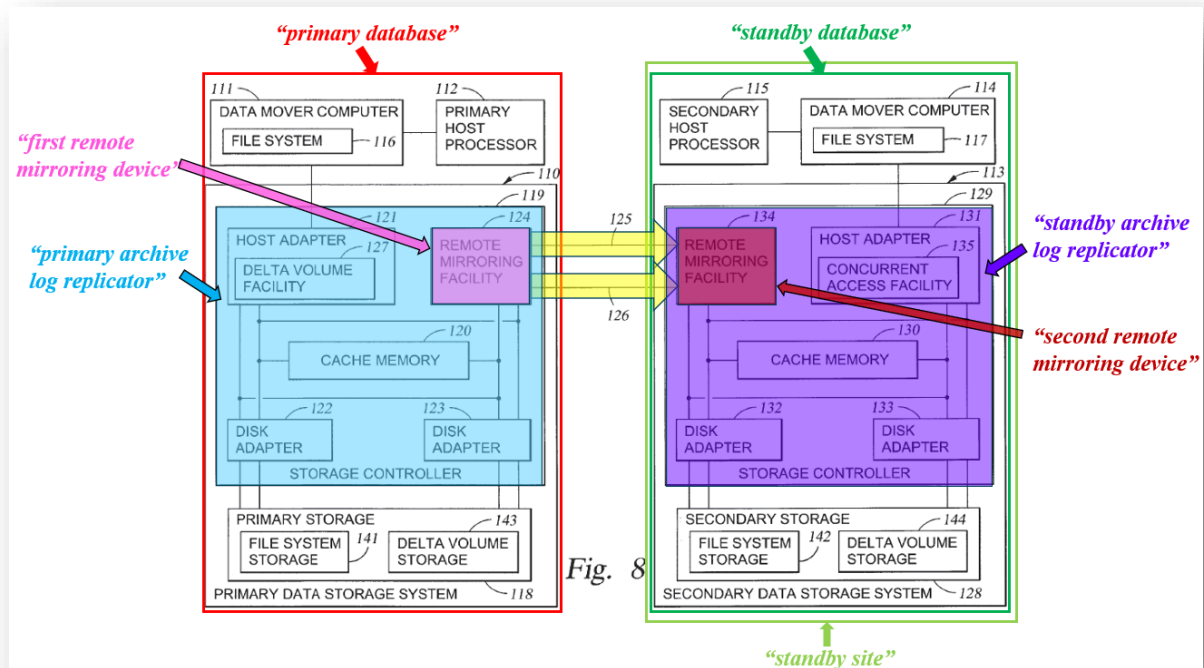
169. As discussed above for claim [5c], the combined teachings of Srinivasan and Taylor would have rendered obvious “transmitting the delta image to a *standby archive log replicator* ... in the *standby database*.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

170. 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).

171. 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.

172. 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*”):



Ex-1005, Fig. 8 (annotated).

173. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 7.

H. 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.”*

174. 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*”). A POSITA would

have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

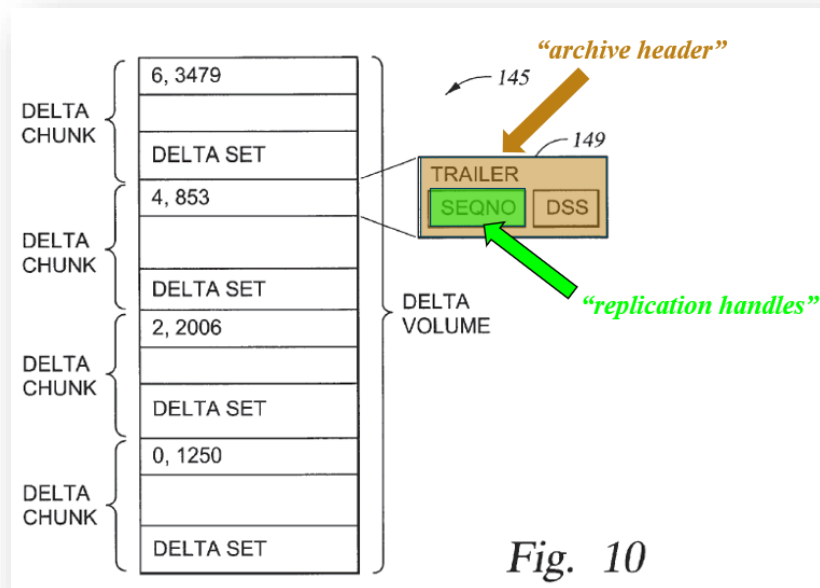
175. The '139 Patent explains that replication handles are sequence numbers used to help track the progress of the mirror writes on the standby site as compared to the writes on the primary site. Ex-1001, 5:55-62.

176. 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).

177. 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.

178. Figure 10 of Srinivasan shows the sequence number (SEQNO) (“*replication handle*”):



Ex-1005, Fig. 10 (annotated).

179. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 8.

- I. ***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.”***

180. 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. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

181. 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.

182. 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).

183. 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.

184. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 9.

J. *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.”*

185. 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. A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

186. 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.

187. 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*”).

188. 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.

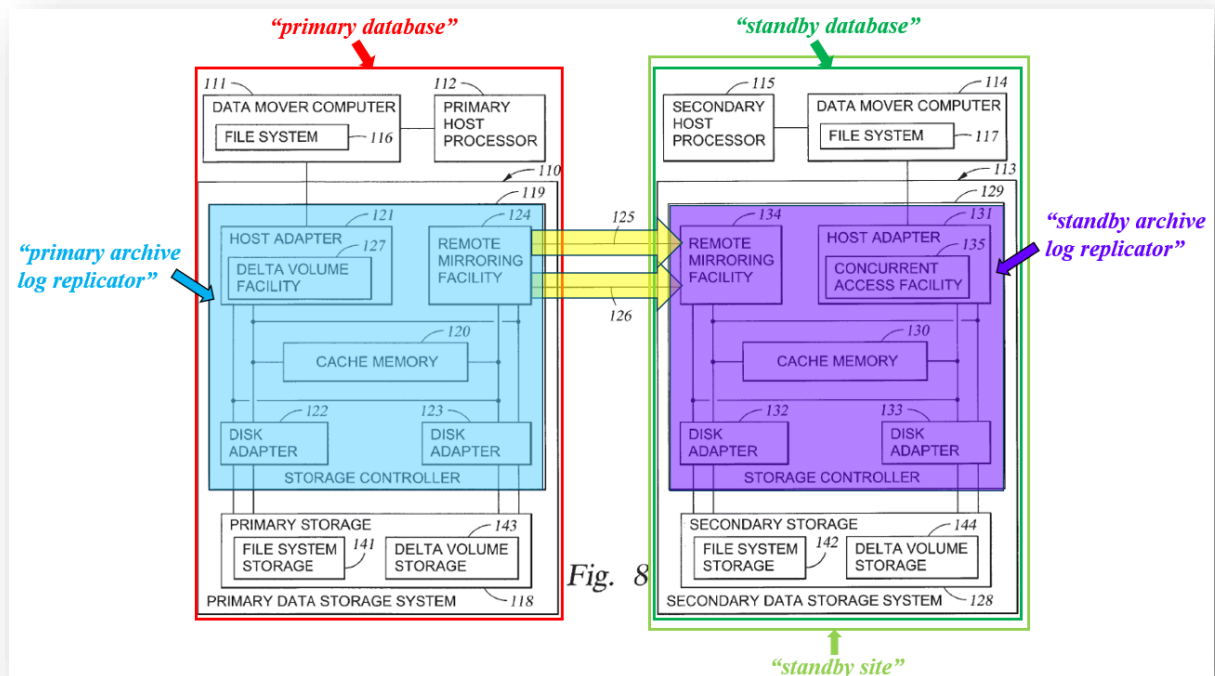
189. As discussed for claim [5c], the storage controller 129 is a “*standby archive log replicator*.” See ¶¶152-157.

190. 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).

191. 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.

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



Ex-1005, Fig. 8 (annotated).

193. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 10.

- K. 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.”**

194. 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.” A POSITA would have been motivated to combine the teachings of Srinivasan and Taylor for the reasons described above in ¶¶74-93.

195. 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-1006, 3:8-11, 5:29-31.

196. 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).

197. 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.

198. Accordingly, the combined teachings of Srinivasan and Taylor would have rendered obvious claim 11.

L. Claim 12:

1. ***[12preamble]: “A method for creating archive redo logs from a primary database, for a standby database at a remote location, comprising the steps of:”***

199. For purposes of this analysis, I have been told to assume that the preamble is limiting for this claim.

200. The combined teachings of Srinivasan and Taylor would have rendered obvious this preamble for the reasons discussed for claims [1preamble] and [5preamble]. See ¶¶94-103, 139-145.

2. [12a]: “(a) trapping file system transactions to create an online redo log at the standby database;”

201. The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5a]. See ¶¶146-148.

3. [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;”

202. The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5b]. See ¶¶149-151.

4. [12c]: “(c) transmitting the delta image to a standby database; and”

203. The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5c]. See ¶¶152-157.

5. [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.”

204. The combined teachings of Srinivasan and Taylor would have rendered obvious this element for the reasons discussed in claim [5d]. See ¶¶158-160.

- M. 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.”*

205. The combined teachings of Srinivasan and Taylor would have rendered obvious claim 13 for the reasons discussed in claim 6. See ¶¶161-167.

- N. 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.”*

206. The combined teachings of Srinivasan and Taylor would have rendered obvious claim 14 for the reasons discussed in claim 7. See ¶¶168-172.

- O. 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.”*

207. The combined teachings of Srinivasan and Taylor would have rendered obvious claim 15 for the reasons discussed in claim 8. See ¶¶173-178.

- P. 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.”*

208. The combined teachings of Srinivasan and Taylor would have rendered obvious claim 16 for the reasons discussed in claim 9. See ¶¶179-183.

Q. 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.”

209. The combined teachings of Srinivasan and Taylor would have rendered obvious claim 17 for the reasons discussed in claim 10. See ¶¶184-192.

R. 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.”

210. The combined teachings of Srinivasan and Taylor would have rendered obvious claim 18 for the reasons discussed in claim 11. See ¶¶193-197.

X. CONCLUSION

211. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

January 13, 2025
Date: _____



Kevin C. Almeroth, Ph.D.