

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

THE PETITIONER

Cisco Systems, Inc.,

v.

THE PATENT OWNER

UMBRA Technologies Ltd.

---

U.S. Patent No. 11,146,632

Issue Date: October 21, 2021

Title: DATA BEACON PULSER(S) POWERED BY INFORMATION  
SLINGSHOT

---

*Inter Partes* Review No.: Unassigned

---

**DECLARATION OF DR. A.L. NARASIMHA REDDY IN SUPPORT OF  
PETITION FOR *INTER PARTES* REVIEW OF  
U.S. PATENT NO. 11,146,632**

***Mail Stop “PATENT BOARD”***

Patent Trial and Appeal Board

U.S. Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

## **TABLE OF CONTENTS**

|                                                                               | <b>Page</b> |
|-------------------------------------------------------------------------------|-------------|
| I. INTRODUCTION .....                                                         | 1           |
| II. MY BACKGROUND AND QUALIFICATIONS .....                                    | 3           |
| III. RELEVANT LEGAL STANDARDS.....                                            | 6           |
| IV. BACKGROUND.....                                                           | 10          |
| A. The '632 Patent .....                                                      | 10          |
| B. Claim Construction.....                                                    | 12          |
| C. Priority Date of Claims .....                                              | 13          |
| D. Person of Ordinary Skill in the Art (POSITA).....                          | 13          |
| V. STATE OF THE ART .....                                                     | 14          |
| 1. Parallel File Systems.....                                                 | 15          |
| 2. Write and Read Queues .....                                                | 16          |
| VI. OVERVIEW .....                                                            | 20          |
| VII. CHALLENGES .....                                                         | 21          |
| A. Identification of Challenges .....                                         | 21          |
| B. Challenge 1: Argarwala alone, Teaches Every Element of<br>Claims 1-7. .... | 21          |
| 1. The Agarwala Reference.....                                                | 21          |
| 2. Detailed Application to the Challenged Claims .....                        | 28          |
| a. Claim 1 .....                                                              | 28          |
| [1.0] A network system for providing data<br>beacons, comprising: .....       | 28          |

|       |                                                                                                                                                                                                                                                                                       |    |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| [1.1] | a first node comprising a first read queue, a first write queue, and a first parallel file system; .....                                                                                                                                                                              | 30 |
| [1.2] | a second node comprising a second read queue, a second write queue, and a second parallel file system; .....                                                                                                                                                                          | 42 |
| [1.3] | wherein the first node writes first data from the first write queue to the second parallel file system; and .....                                                                                                                                                                     | 43 |
| [1.4] | wherein the second node reads the first data from the second parallel file system and places the first data in the second read queue. ....                                                                                                                                            | 46 |
| b.    | Claim 2 .....                                                                                                                                                                                                                                                                         | 47 |
| [2]   | The network system of claim 1, wherein the second node writes second data from the second write queue to the first parallel file system; and wherein the first node reads the second data from the first parallel file system and places the second data in the first read queue..... | 47 |
| c.    | Claim 3 .....                                                                                                                                                                                                                                                                         | 48 |
| [3]   | The network system of claim 1, wherein the first data is a carrier file comprising a header, a body, and a footer. ....                                                                                                                                                               | 48 |
| d.    | Claim 4 .....                                                                                                                                                                                                                                                                         | 50 |
| [4]   | The network system of claim 3, wherein the first nodes subsequently writes additional data to the first parallel file system.....                                                                                                                                                     | 50 |
| e.    | Claim 5 .....                                                                                                                                                                                                                                                                         | 51 |
| [5]   | The network system of claim 4, wherein the additional data is written at a set frequency. ....                                                                                                                                                                                        | 52 |

|                                                                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| f. Claim 6 .....                                                                                                                                                          | 53 |
| [6] The network system of claim 4, wherein the<br>additional data only contains information that has<br>changed since the first data was written. ....                    | 53 |
| g. Claim 7 .....                                                                                                                                                          | 54 |
| [7.0] The network system of claim 1, further<br>comprising a third node comprising a third read<br>queue, a third write queue, and a third parallel file<br>system; ..... | 54 |
| [7.1] Wherein the first node writes first data from<br>the first write queue to the second and third parallel<br>file systems at the same time; and\.....                 | 55 |
| [7.2] wherein the third node reads the first data<br>from the third parallel file system and places the<br>first data in the third read queue. ....                       | 57 |
| CONCLUSION AND DECLARATION.....                                                                                                                                           | 1  |
| APPENDIX A – MATERIALS RELIED UPON.....                                                                                                                                   | i  |
| APPENDIX B – <i>CURRICULUM VITAE</i> .....                                                                                                                                | ii |

I, Dr. A.L. Narasimha Reddy, declare as follows:

## **I. INTRODUCTION**

1. I am over the age of eighteen (18) and otherwise competent to make this declaration.

2. I have been retained by Cisco Systems, Inc. (Petitioner) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office. Although I am being compensated at my rate of \$650 per hour for the time I spend on this matter, no part of my compensation depends on the outcome of this proceeding, and I have no other interest in this proceeding. To the best of my knowledge, I have no financial interest in Client.

3. This Petition for inter partes review involves U.S. Patent No. 11,146,632 (“the ’632 Patent”) (EX-1001), entitled “DATA BEACON PULSER(S) POWERED BY INFORMATION SLINGSHOT” and lists Joseph E. Rubenstein, as the inventor.

4. For the purposes of this *inter partes* review as discussed later, I have been instructed to assume that the effective filing date of the Claims of the ’632 Patent challenged by the Petitioner in this *inter partes* review is no earlier than April 26, 2016, the filing date of U.S. Provisional Patent Application Nos. 62/327,907, 62/327,846, and 62/327,911.

5. I understand that according to USPTO records, the '632 Patent is currently assigned to UMBRA Technologies Ltd. ("UMBRA" or "Patent Owner").

6. The '632 Patent is directed to a distributed storage system, with each node having a read/write queue and a parallel file system. I am familiar with the technology described in the '632 Patent as of the earliest possible priority date of April 26, 2016.

7. In preparing this Declaration, I have reviewed the '632 Patent (EX-1001), the file history of the '632 Patent (EX-1002), and each of the documents cited herein, and I have considered these documents in light of the general knowledge in the art as of April 26, 2016. In formulating my opinions, I have relied upon my experience in the relevant art. I have also considered the viewpoint of a person of ordinary skill in the art ("POSITA") in the relevant field, as of April 26, 2016, who I describe below (§IV.D). For convenience, the materials I considered in arriving to my opinion are listed in Appendix A.

8. The opinions and comments formulated during this assessment are based on observations and information available at the time of the investigation. The findings presented herein are made to a reasonable degree of scientific certainty. I have made every effort to accurately and completely investigate all areas of concern identified during our investigation.

9. I have been asked to provide my technical expertise, analysis, insights and opinions regarding the '632 Patent and relevant references that form the basis of the arguments set forth in the accompanying Petition for *inter partes* review of the '632 Patent. As described in detail below, I offer the following opinion in this Declaration:

- A POSITA would have found Claims 1-7 of the '632 Patent are unpatentable over *Agarwala*.

## II. MY BACKGROUND AND QUALIFICATIONS

10. I am currently the Truchard Foundation Chair Professor of Electrical and Computer Engineering at Texas A&M University in College Station, Texas. I have over 25 years of experience in a wide variety of technologies and industries relating to data communications, storage systems, and distributed systems, including packet-switched network communications.

11. My academic credentials include a Bachelor of Technology degree in Electronics and Electrical Communications Engineering from the Indian Institute of Technology, Kharagpur, India, in August 1985. I then received a Master of Science and a Ph.D. in Computer Engineering from the University of Illinois at Urbana-Champaign in May 1987 and August 1990, respectively.

12. I have worked for over 25 years in the field of Electrical Engineering. My primary focus and research interest has been on computer networks, storage

systems, multimedia systems, and computer architecture. I have authored and co-authored over a hundred technical papers and several book chapters related to several of these interests, including on such topics as multipath routing, route control, high-speed networks, network congestion, packet management, quality of service regulation, network security, network modeling, differentiated services, storage system enhancements, caching strategies, and multimedia system enhancements to name a few examples. I am listed as an inventor on six U.S. Patents in the field of multi-node communication networks and computer systems.

13. My employment history following my graduation from the University of Illinois at Urbana-Champaign began at the IBM Almaden Research Center in San Jose, California in 1990. At IBM, I worked on projects related to disk arrays, multiprocessor communication, hierarchical storage systems and video servers.

14. In 1995, I joined the faculty of the department of Electrical Engineering at Texas A&M University initially as an Associate Professor and was later promoted to full, tenured Professor. At Texas A&M, during 2016-21, I was the Associate Agency Director for Strategic Initiatives and Centers for the Texas A&M Engineering Experiment Station (TEES), which engages in engineering and technology-oriented research and educational collaborations. Further, I also served as Associate Dean for Research in the College of Engineering during 2016-21.

15. At Texas A&M, I have taught dozens of courses related to computer networking and communications, as well as computer architecture, multimedia systems and networks, topics in networking security, multimedia storage and delivery, as well as networking for multimedia applications. I have also served on various committees for the benefit of the scientific community and the Texas A&M University community.

16. I am a member of a number of professional societies, including the Institute of Electrical and Electronic Engineers (IEEE), where I have been elevated to an IEEE Fellow, and the Association for Computing Machinery (ACM). I have been responsible for chairing or co-chairing numerous conferences and programs, as well as presenting research at major IEEE and ACM conferences. For example, I served as program co-chair for the 2008 5<sup>th</sup> International Conference on Broadband Communications, Networks and Systems, panels co-chair for the 2008 3<sup>rd</sup> International Conference on Communication Systems Software & Middleware, and panel chair of the IEEE Conference of High Performance Computer Architecture.

17. My presentations include a Keynote speech at International Conference on Information Technology-New Generations in 2013, a Keynote speech at IEEE International Symposium on Computers and Communications 2010, and several invited talks including at Georgia Tech (2013), COMSNETS

Conference (2013), Int. Conf. on Networking and Communications (2012), Samsung (2011), Korea University (2011), Aijou University (2011), Catedra Series at University of Carlos III, Madrid (2009), Thomson Research, Paris (2009), Telefonica Research, Barcelona (2009), and a Distinguished Seminar at IBM Austin Research Lab (2008).

18. I have received multiple awards in the field of networks and computer architecture. I received the NSF Career Award from 1996-2000. I received an outstanding professor award by the IEEE student branch at Texas A&M during 1997-1998, an outstanding faculty award by the department of Electrical and Computer Engineering during 2003-2004, a Distinguished Achievement award for teaching from the former students' association of Texas A&M University, and a citation "for one of the most influential papers from the 1st ACM Multimedia conference.

19. My curriculum vitae, which includes a more detailed summary of my background and experience, is attached as Appendix B.

### **III. RELEVANT LEGAL STANDARDS**

20. I am not a lawyer and do not provide any legal opinions in this Declaration. Although I am not a lawyer, I have been informed and understand that certain legal standards are to be applied by technical experts in forming opinions regarding the meaning and validity of patent claims. I have been asked to

provide my opinions regarding whether the claims of the '632 Patent are anticipated or would have been obvious to a person having ordinary skill in the art at the time of the alleged invention, in light of the prior art.

21. I have been informed and understand that a patent claim is not patentable under 35 U.S.C. § 103 if the differences between the patent claim and the prior art are such that the claimed subject matter as a whole would have been obvious at the time the claimed invention was made to a person having ordinary skill in the relevant art. Obviousness, as I have been informed and understand, is based on the scope and content of the prior art, the differences between the prior art and the claim, the level of ordinary skill in the art, and, to the extent that they exist, certain objective indicia of non-obviousness.

22. I understand that objective indicia can be important evidence regarding whether a patent is obvious or nonobvious, if it has an appropriate nexus to the claimed invention, i.e., is a result of the merits of a claimed invention (rather than the result of design needs or market-pressure advertising or similar activities). Such indicia include: commercial success of products covered by the patent claims; a long-felt need for the invention; failed attempts by others to make the invention; copying of the invention by others in the field; unexpected results achieved by the invention as compared to the closest prior art; praise of the invention by the infringer or others in the field; the taking of licenses under the

patent by others; expressions of surprise by experts and those skilled in the art at the making of the invention; and the patentee proceeded contrary to the accepted wisdom of the prior art.

23. I have been informed that whether there are any relevant differences between the prior art and the claimed invention is to be analyzed from the view of a person of ordinary skill in the relevant art at the time of the invention. As such, my opinions below as to a person of ordinary skill in the art are as of the time of the invention, even if not expressly stated as such; for example, even if stated in the present tense.

24. In analyzing the relevance of the differences between the claimed invention and the prior art, I have been informed that I must consider the impact, if any, of such differences on the obviousness or non-obviousness of the invention as a whole, not merely some portion of it. The person of ordinary skill faced with a problem is able to apply his or her experience and ability to solve the problem and also look to any available prior art to help solve the problem.

25. I have been informed that a precise teaching in the prior art directed to the subject matter of the claimed invention is not needed. I have been informed that one may take into account the inferences and creative steps that a person of ordinary skill in the art would have employed in reviewing the prior art at the time of the invention. For example, if the claimed invention combined elements known

in the prior art and the combination yielded results that were predictable to a person of ordinary skill in the art at the time of the invention, then this evidence would make it more likely that the claim was obvious. On the other hand, if the combination of known elements yielded unexpected or unpredictable results, or if the prior art teaches away from combining the known elements, then this evidence would make it more likely that the claim that successfully combined those elements was not obvious.

26. I have been informed and understand that there are recognized, exemplary, rationales for combining or modifying references to show obviousness of claimed subject matter. Some of the rationales include the following: combining prior art elements according to known methods to yield predictable results; simple substitution of one known element for another to yield predictable results; use of a known technique to improve a similar device (method or product) in the same way; applying a known technique to a known device (method or product) ready for improvement to yield predictable results; choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; and some teaching, suggestion, or motivation in the prior art that would have led one of

ordinary skill to modify the prior art reference or to combine prior art teachings to arrive at the claimed invention.

#### **IV. BACKGROUND**

##### **A. The '632 Patent**

27. Claims 1-7 of the '632 Patent (the "Challenged Claims") are unpatentable as obvious over the prior art. The claims are directed to a generic distributed network storage node topology that was well known in the art before the filing of the '632 Patent.

28. FIG. 13 of the '632 Patent "illustrates one example of the Data Beacon Pulser (DBP) mechanism framework and flow ... and demonstrates how DBP can utilize 'Slingshot' to make information from a source region 13-310 available to a client 13-100 in another region in as timely a fashion as possible", but the topology of the Challenged Claims is related only to what is illustrated and described in connection with the annotated bottom portion of FIG. 13 (below). *EX-1001* 16:8-13; *also* 13:21-26 (describing FIG. 7)

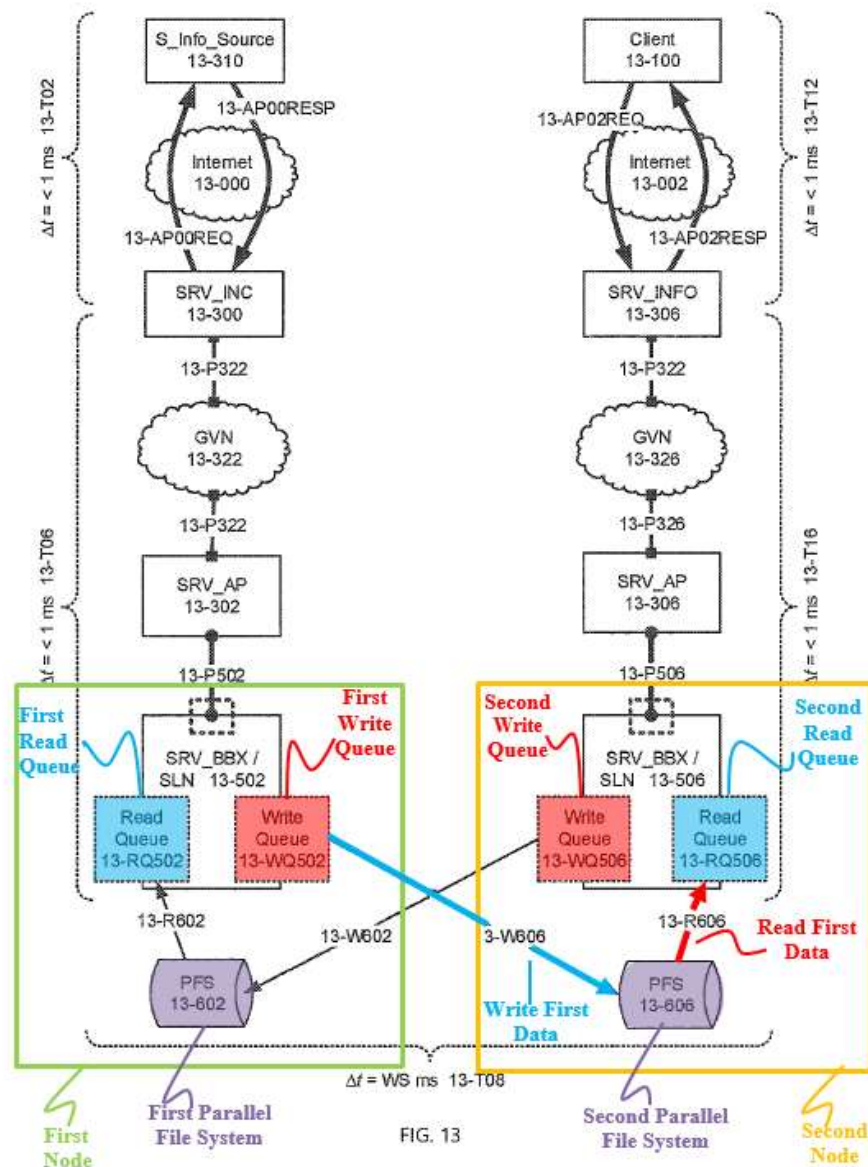


FIG. 13 shows the components that are recited in the Challenged Claims, e.g., claim 1. A first node includes a first Read Queue, a first Write Queue and a first parallel file system (PFS) device, and a second node includes a second Read Queue, a second Write Queue and a second parallel file system (PFS) device. *EX-1001* 16:23-34; 13:27-43; 16:4-5. Data is written from the first Write Queue of the

first node to the second PFS of the second node, and that data is read from the second PFS of the second node to the Read Queue of the second node. *EX-1001* 16:23-34; 13:29-43.

29. The second PFS (PFS 606) is described in the specification as “a parallel file system device in the remote region 606”. *EX-1001* 13:27-46. Additionally, all the parallel file systems are referred to as “PFS devices.” *EX-1001* FIG. 12, 16:4-5 (“In practical use, all connected nodes can concurrently connect with PFS devices in all other regions and locations”).

30. The ’632 Patent provides no details regarding the parallel file system used in its system and does not purport to offer any improvements to parallel file systems, which were known in the art well before the ’632 Patent.

31. Similarly, the ’632 Patent provides no details regarding the Write Queues and Read Queues used in its system—except that the queues hold data—and does not purport to offer any improvements to write and read queues, which were also known in the art well before the ’632 Patent.

## **B. Claim Construction**

32. I have been informed and understand that in order to properly evaluate the ’632 Patent, the terms of the claims must first be interpreted.

33. I understand that Petitioner proposes that each claim term in the Challenged Claims be given its plain and ordinary meaning in this proceeding, and

that no specific construction of any claim term is required because the prior art relied on in this Petition meets each of the claim terms under any reasonable construction. The plain meaning is the ordinary and customary meaning to those skilled in the art.

**C. Priority Date of Claims**

34. For the purpose of this *inter partes* review, I have been instructed to assume that the effective filing date of each of the Challenged Claims is no earlier than the filing date of U.S. Provisional Application Nos. 62/327,907, 62/327,846, and 62/327,911. Solely for the purpose of this IPR, I have been instructed to assume the priority date is April 26, 2016.

**D. Person of Ordinary Skill in the Art (POSITA)**

35. I am familiar with the knowledge and capabilities of one of ordinary skill in the art. Unless otherwise stated, my testimony below refers to the knowledge of one of ordinary skill in the art as of April 26, 2016, the earliest possible effective filing date of the '632 Patent.

36. I have been informed and understand that a Person of Ordinary Skill in the Art ("POSITA") is a hypothetical person who is presumed to be aware of all pertinent prior art, thinks along conventional wisdom in the art, and is a person of ordinary creativity.

37. With respect to the '632 Patent, a POSITA in April 2016 would have been familiar with network communications and distributed storage systems, including network file systems. A POSITA would have a working knowledge of techniques for accessing data (reading and writing data) in distributed network storage systems, including parallel file systems. A POSITA would have gained knowledge of these concepts through a mixture of training and work experience, such as by having a Bachelor's degree in computer science and four years of experience; or by obtaining a Master's degree in computer science, but having only one to two years of experience; or by having no formal education but experience in storage systems of at least eight years.

38. A POSITA may work as part of a multi-disciplinary team and draw upon not only his or her own skills, but also take advantage of certain specialized skills of others on the team to solve a given problem.

## **V. STATE OF THE ART**

39. The following section describes the state of the art for computer security systems as of April 2016, the earliest effective filing date for claims of the '632 Patent. The prior art references, and the discussions of what was known to a POSITA, provide the factual support for the general description of the state of the art at the time of the invention and provide additional motivation to modify or

combine the references. Accordingly, these references should be considered by the Board.

## **1. Parallel File Systems**

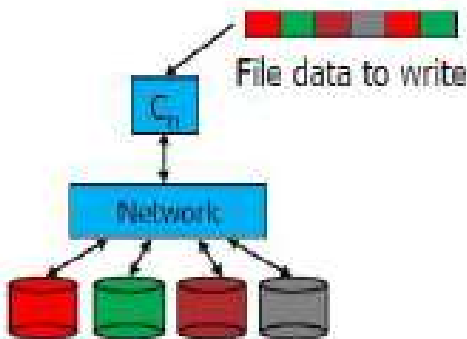
40. File systems have two key roles, including organizing and maintaining the namespace (i.e., directory hierarchy and file names) and storing the contents of files and providing an interface through which stored files can be read and data can be written. Local file systems are used by a client with direct access to the disk, and networked file systems (NFSs) provide access to clients that might not have direct access to the disk. *EX-1010* p. 228; *EX-1011* p. 31.

41. A parallel file system (PFS) is a file system designed to store data across multiple networked servers and to facilitate high-performance access through simultaneous, coordinated input/output operations between clients and storage nodes.

42. The PFS breaks up a data set into blocks and distributes, or “stripes”, the blocks to multiple storage drives, which can be located in local and/or remote servers. A global namespace is used to facilitate data access. Reads and writes of data to distributed storage devices can be done through multiple I/O paths concurrently, which provides a significant performance benefit. Capacity and bandwidth can be scaled to accommodate large quantities of data.

43. One such well-known parallel file system is Lustre ([www.lustre.org](http://www.lustre.org)), which was created back in 1999. Another well-known parallel file system is the Parallel Virtual File System (PVFS), which was developed in 1993 and changed its name to OrangeFS ([OrangeFS.org](http://OrangeFS.org)) as of its 2011 release. *EX-1010* pp. 227-28; *EX-1009* p. 11; *EX-1012* pp. 142-43, 146.

44. As noted above, striping is the basic mechanism used in parallel files systems to improve performance where a file is split into fixed-size blocks that are written to separate disks in order to facilitate parallel access, as illustrated below.



Principally, this technique allows multiple servers, disks and network links to be leveraged during concurrent I/O operations. This eliminates bottlenecks and can improve performance. *EX-1008* ¶ 21; *EX-1010* p. 228; *EX-1012* pp. 143.

## 2. Write and Read Queues

45. Storage systems have long used write and read queues for holding data being written to and read from file systems. Write queues are used as a staging area for data that is to be written into storage. Read queues are used to

hold data read from storage for access by applications and clients, for example, and to allow for speedy access to the read data and for subsequent read requests.

46. U.S. Patent Application Publication No. 2016/0100027 to Haghighi (“’027 Publication”) (*EX-1007*) discloses a storage system that includes local servers 105 and remote servers 110 connected by a network 130. Each server includes a storage device 185, 190 (e.g., solid state devices or drives (SSDs) or hard disk drives (HDDs)) and memory 165, 170, including queues 175, 180 for accessing (reading/writing) data on the storage devices 185, 190. *EX-1007* ¶¶ 0023, 0028, 0029.

47. Information can be structured, stored, and accessed in the storage system via files, blocks, logical block address (LBA), logical unit number (LUN), key-value pairs, objects, or the like. *EX-1007* ¶ 24, 27.

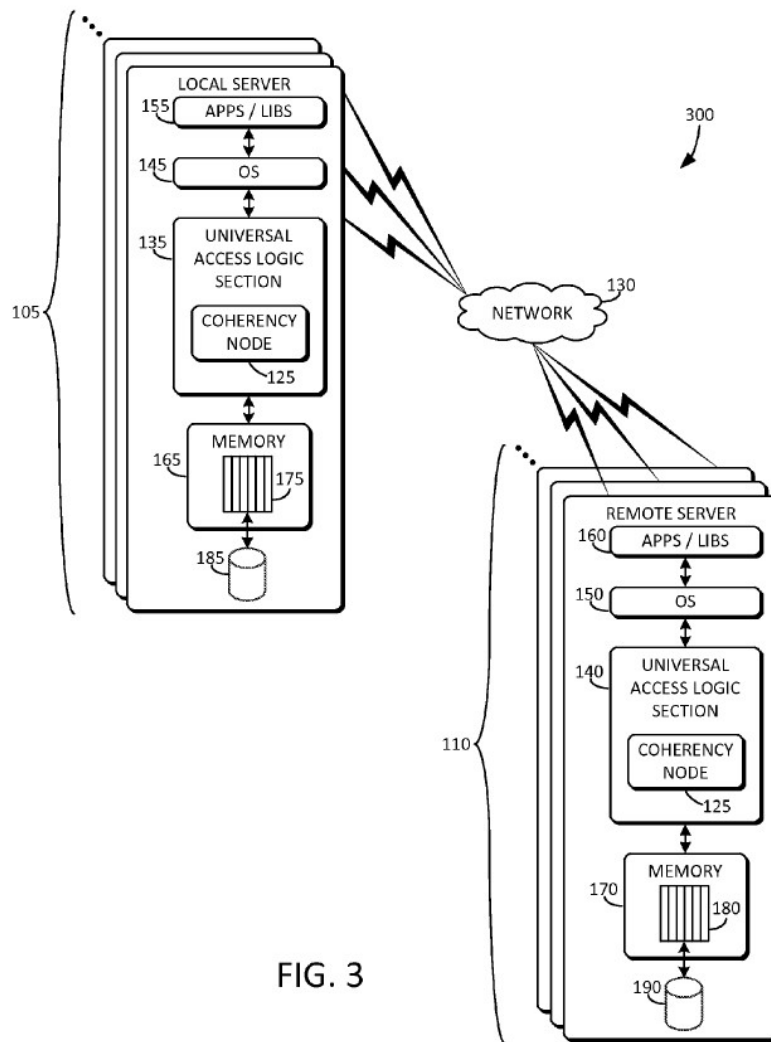


FIG. 3

Each local and remote server in the system can read data stored on any one of the other individual servers, and each server can write data to storage devices of any other server in the system. *EX-1007 ¶ 0036.*

48. Each of the servers has write data queues that queue local write data to be written to the local storage devices and incoming data queues that queue data received from a remote server for writing to the local storage device. *EX-1007 ¶¶ 0040, 00041.*

49. Each of the servers also includes queues for data that is being read from storage, both for local and remote applications. *EX-1007* ¶¶ 0040, 0043.

50. For example, with reference to local server 105 shown in FIG. 6 (below), the '027 Publication expressly discloses that each local memory array 625 includes multiple storage queues 621, including (i) multiple local write storage queues (e.g., 626, 628, and 622) and (ii) multiple local read storage queues (e.g., 627, 629, and 623). *EX-1007* ¶¶ 0044, 0049.

51. Network interface card (NIC) 645 of local server 105 also includes outgoing data queues 677 for sending write data to remote servers 110 for writing to remote storage devices 640, and incoming data queues 667 for receiving remote read data from remote servers 110. *EX-1007* ¶ 0050.

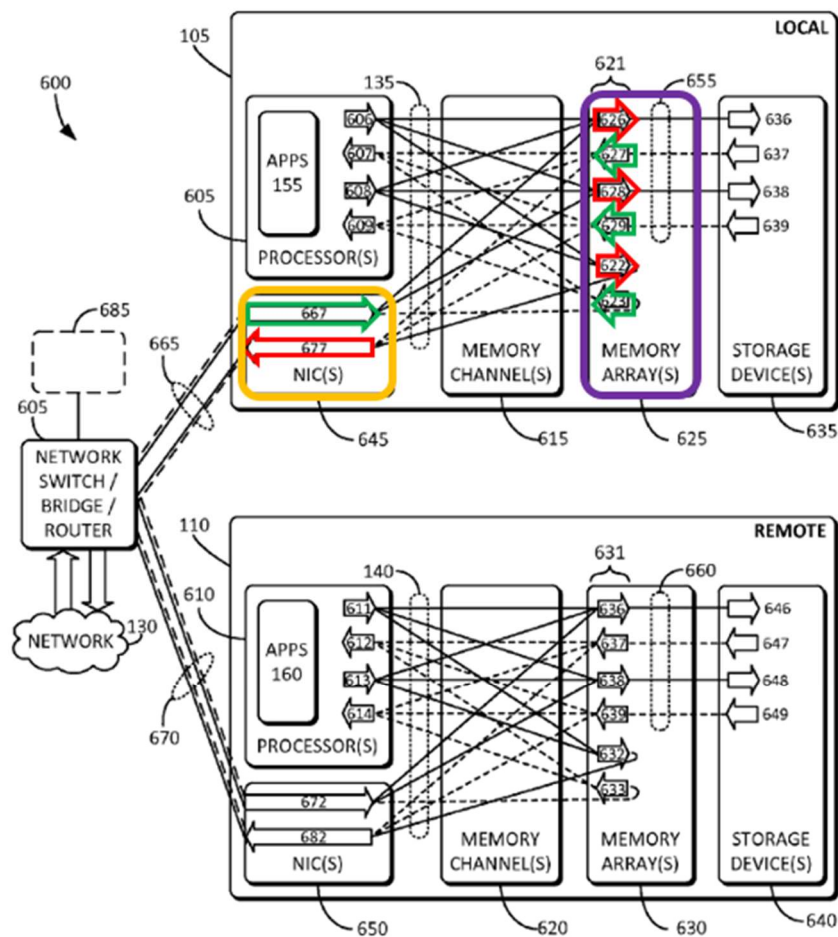


FIG. 6

## VI. OVERVIEW

52. As detailed below, I explain how each limitation recited in the Challenged Claims are found in the prior art. The Challenged Claims are directed to a distributed storage system, with each node having a read/write queue and a parallel file system. The read/write queue of a first node writes data from the write queue to a parallel file system of a second node, and the second node reads the data

from the parallel file system of the second node and places it in the read queue of the second node.

## **VII. CHALLENGES**

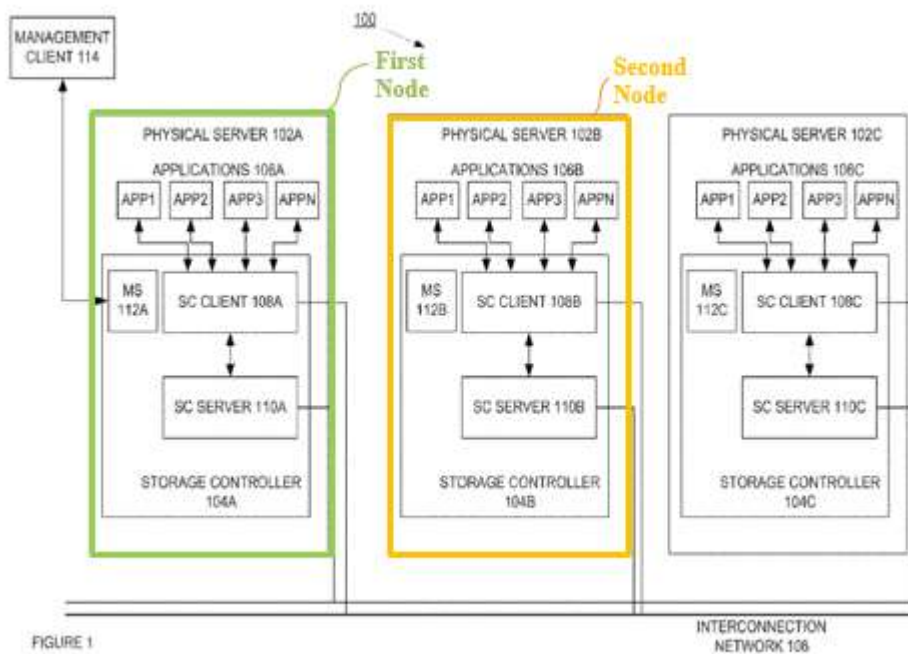
### **A. Identification of Challenges**

53. Based on my understanding of the law of obviousness discussed above, Claims 1-7 of the '632 Patent are invalid as being obvious over Agarwala.

### **B. Challenge 1: Argarwala alone, Teaches Every Element of Claims 1-7.**

#### **1. The Agarwala Reference**

54. Agarwala discloses “distributed multi-layer caching in a distributed storage system.” *EX-1005* 1:11-13; *EX-1004* ¶0001 (“This invention relates generally to a storage system and more particularly to software defined distributed storage system.”). Agarwala refers to the storage system as “StorFS,” and FIG. 1 illustrates the StorFS system including physical servers 102A-C (which Agarwala also calls “storage nodes 102A-C”) interconnected by network 106. *EX-1005* 3:59-62; *EX-1004* ¶¶ 0013, 0047.



EX-1005 FIG. 1; EX-1004 FIG. 1. A first storage node (e.g., storage node 102A) and a second storage node (e.g., storage node 102B) map to the *first node* and *second node* of the Challenged Claims. Each storage node 102 includes a storage controller server 110 that manages the underlying storage (e.g., hard disk drive (HDD), solid state devices (SSD), PCIe flash, etc.), where the storage controllers collectively provide storage functionality to storage controller (SC) clients 108. EX-1005 3:67-4:6; EX-1004 ¶0047. Each SC client 108 processes I/O requests from applications that are local to its physical storage node and routes them to the appropriate local or remote SC server 110 for processing. EX-1005 4:7-12; EX-1004 ¶0047.

55. Agarwala further discloses write and read queues. As show in FIG. 2 below, each storage controller 200 includes a write cache 208 and a read cache (DOCL) 210, which map to the write and read queues, respectively, of the Challenged Claims. *EX-1005* 6:10-17; *EX-1004* ¶¶0052.

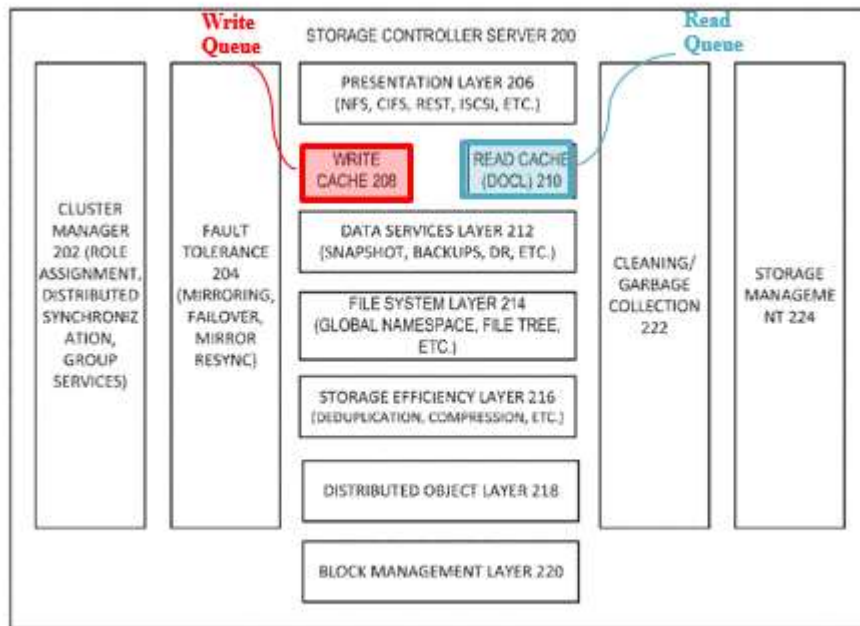


FIGURE 2

56. A POSITA understood that Agarwala's read cache is a read queue. First, objects are initially read from persistent storage (e.g., SSD 412 or the HDD 410 of FIG. 4) and added to the read cache where they are held for reading. *EX-1005* 8:27-28, 8:27-55; *EX-1004* ¶¶0061-0062.

57. Second, Agarwala discloses in this read cache items can be moved through and out of the read cache, as in a queue. For example, cached items can be promoted to the Most Recently Used (MRU) list based on an initial read of an

object and the Most Frequently Used (MFU) list depending on frequency of access, as well as be evicted to create room for new contents:

FIG. 3B is a block diagram of one embodiment of a DOCL layer 1 (L1) cache 350. In one embodiment, the L1 cache 350 includes two types of list: Recency-based eviction lists 352; and frequency-based eviction lists 354... The recency-based eviction lists 352 are used to track objects being added to the cache and include an eviction list for data objects 360 and metadata objects 358... As the objects are read, they are initially promoted to the recency list (e.g., a Most Recently Used (MRU) list). In one embodiment, the frequency-based eviction lists are used to track accesses to objects in the cache and include an eviction list for data objects 364 and metadata objects 362. If this object is accessed within a certain time period again, the object is promoted to the frequency list (e.g., a Most Frequently Used (MFU) list)... In one embodiment, a hotness/coldness of an object measures how recently the object was added or accessed by the StorFS system. In this embodiment, an object that has not been recently added or accessed can be a candidate for eviction from the L1 cache. In one embodiment, an object that has not been recently added or accessed can be considered “cold” or “stale” and is an object that is one that can be evicted from the L1 cache.

In one embodiment, eviction from and updates to the L1 cache is at the object granularity.

EX-1005 7:5-47; EX-1004 ¶0058.

58. A POSITA would also understand that Agarwala's write cache is a *write queue*. Agarwala's write cache includes a write log for holding or queuing data to be written to the persistent storage at a later time. *EX-1005* 6:12-14; *EX-1004* ¶¶0004, 00006, claim 16 ("if the active log partition is full after the writing of the data, transferring the data to persistent storage"). When the storage control server receives a write storage request, it first "stores this data in a write log by appending the data at the end of the write log. This can lead to a fast write for the data that allows a quick write storage fulfillment for the requesting application." *EX-1004* ¶0041. When the write log is full, the SC server "flushes the write log to persistent storage." *EX-1004* ¶0041.

59. Data is also written from the write log of Agarwala's write cache in one storage node to other storage nodes both synchronously and asynchronously to "efficiently replicate data across the cluster". *EX-1004* ¶¶0041, 0043, 0059.

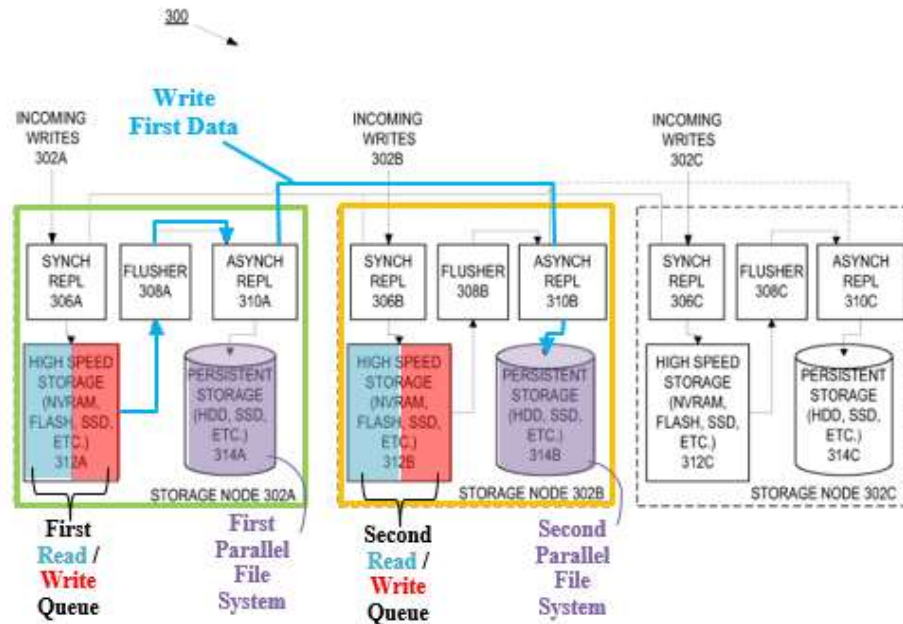


FIGURE 3

EX-1004 FIG. 3.

60. A POSITA would also understand that Agarwala’s storage system implements a *parallel file system*, including in its file system and persistent storage.

61. Agarwala discloses storage controller (SC) servers “collectively” provide storage for the storage clients across multiple distributed nodes:

The SC servers 110A-C manage their underlying storage (e.g., Hard disk drive (HDD), Solid state drives (SSD), PCIe flash, etc.) and collectively provide reliable and unified storage functionality to the SC clients. The SC client 108A-C processes input/output (I/O) requests from the applications that are local to its physical storage node

and routes them to the appropriate SC servers for processing.

*EX-1005* 4:3-9; *EX-1004* ¶0047. Across the multiple distributed nodes, the workload is partitioned, distributed, and parallelized. *EX-1005* 5:13-27, 5:39-41; *EX-1004* ¶¶ 0050-0051.

62. Agarwala discloses that the storage controllers of the StorFS system implement a file system, including a global distributed object namespace and logical global namespace:

In one embodiment, the Distributed Object Layer 218 of the storage control server uses an API of the Block Management Layer 220 to provide a global distributed object namespace that is accessible over the network...

In one embodiment, the File System Layer 214 provides a logical global namespace abstraction to organize and locate data in the cluster... In one embodiment, the Presentation Layer 206 provides an interface to access the StorFS storage using well-known standard protocols like NFS, CIFS, REST, iSCSI, etc.

*EX-1005* 5:63-6:21; *EX-1004* ¶0052. Lastly, like a parallel file system, “the StorFS system stripes file and file system data across multiple storage nodes in our cluster.” *EX-1004* ¶122. “[E]ach logical file (or entity) is partitioned into equal sized ‘stripe units’ [where] [t]he location of a stripe unit is determined based on a mathematical placement function.” *EX-1005* 4:30-33; *EX-1004* ¶0040, 0048.

Further, the workload is distributed and parallelized across multiple storage nodes, as is also the characteristic of a parallel file system. *EX-1005* 5:13-27, 5:39-41; *EX-1004* ¶¶ 0050-0051.

## 2. Detailed Application to the Challenged Claims

### a. Claim 1

***[1.0] A network system for providing data beacons, comprising:***

63. To the extent the preamble is limiting, Agarwala discloses this limitation. *EX-1004*; *EX-1005*.

64. Agarwala discloses “the StorFS system is a storage system that includes a plurality of storage nodes coupled together by an interconnection network” (*a network system*). *EX-1004* ¶0039; *EX-1005* 3:59-62. With respect to *providing data beacons*, the body of claim 1 does not refer to “data beacons” and instead only recites that “first data” is written (e.g., [1.3] *from the first write queue to the second parallel file system*) and read (e.g., [1.4] *from the second parallel file system to the second read queue*). As such, “data” that is written or read across nodes is an example of a “data beacon”, and Agarwala discloses that its StorFS system writes and read data across nodes (*providing data beacons*). For example:

In one embodiment, the input/out request can be a request to read content from the StorFS system, write content to the StorFS system, and/or request status of content stored in the StorFS system.

*EX-1004* ¶0053, *EX-1005* 4:50-53. Data is mirrored from the write log in the write cache of one node across the network to other nodes for storage in persistent storage. *EX-1004* ¶0043 (describing asynchronous replication across nodes).

65. Further, a POSITA understood that data of different types can be stored in Agarwala's StorFS system based on the applications. For example, the '027 Publication discloses that "information can be structured, stored, and accessed in the storage system via files, blocks, logical block address (LBA), logical unit number (LUN), key-value pairs, objects, or the like." *EX-1007* ¶24, 27.

66. I also note that data that is written or read across nodes is an example of a "data beacon" and I am informed that this position is also consistent with Plaintiff's assertions in the Related Litigation. *See EX-1006* p. 3 ("The HX Data Platform controller handles all read and write requests for volumes that the hypervisor accesses and thus intermediates all I/O from the virtual machines and containers....', i.e., Cisco HyperFlex HX-Series nodes in a Cisco HyperFlex System are a network system for providing data beacons."), p. 9 ("[Node2] commits the write to the active local write log on the caching SSD [A1] and sends

duplicate copies to other nodes, e.g., '[A2] on Node 1', i.e. the first node writes first data from the first write queue to the second parallel file system").<sup>1</sup>

67. Therefore, a POSITA understood that Agarwala's StorFS system is *a network system for providing data beacons*.

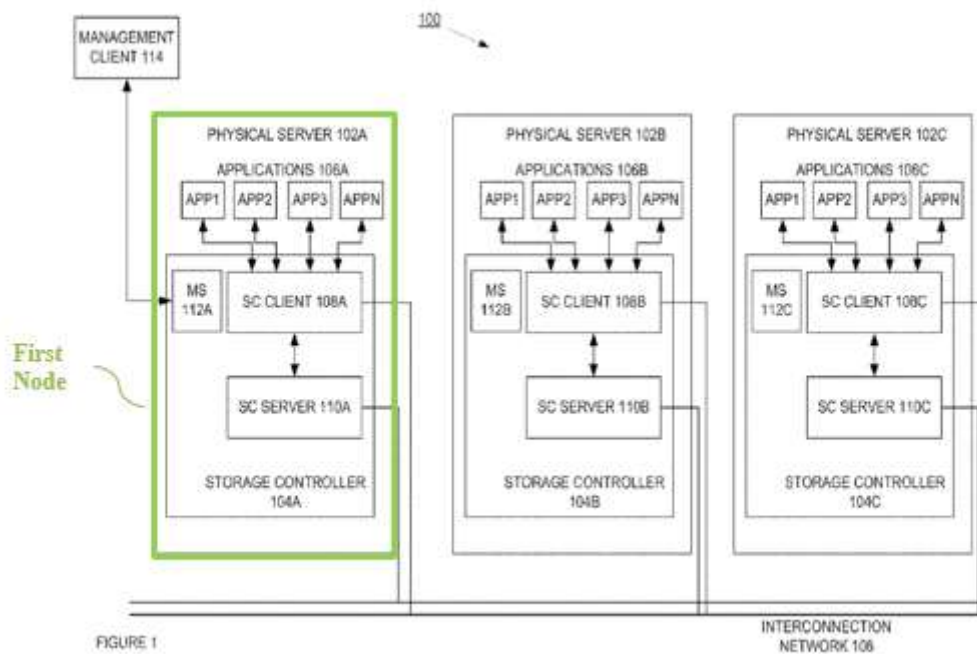
***[1.1] a first node comprising a first read queue, a first write queue, and a first parallel file system;***

*First Node*

68. Agarwala discloses that its storage system—the "StorFS system"—includes storage nodes 102A-C coupled by an interconnection network 116 (a first storage node of which is a *first node*). *EX-1005* 3:59-62. FIG. 1 illustrates the StorFS system:

---

<sup>1</sup> Citations to *EX-1006* are to the internal page number of the chart attached to the Exhibit. For example, *EX-1006* p. 3 is a citation to *EX-1006* p. 3 of 31 as reflected in the footer of the exhibit.



EX-1005 3:59-62, FIG. 1; EX-1004 ¶¶0013, 0047, FIG. 1.

69. Each storage node contains a storage controller server (110A – C in FIG. 1 above) that is used to manage the storage request of the local and/or remote applications using the storage. EX-1004 ¶0039.

The SC servers 110A-C manage their underlying storage (e.g., Hard disk drive (HDD), Solid state drives (SSD), PCIe flash, etc.) and collectively provide reliable and unified storage functionality to the SC clients. The SC client 108A-C processes input/output (I/O) requests from the applications that are local to its physical storage node and routes them to the appropriate SC servers for processing.

EX-1005 4:3-9; EX-1004 ¶0047.

### First Write Queue/ First Read Queue

70. FIG. 2 below shows each controller server contains a write cache (write queue (red)) and a read cache (read queue (blue)):

[T]he Write Cache 208 and the Read Cache 210 Layers provide acceleration for write and read I/O respectively using fast storage devices. In one embodiment, the Write Cache Layer 208 includes the write log as described below. In one embodiment, the Read Cache Layer 210 includes a deduplicated object cache layer (DOCL) that is used to cache recently added and accessed objects for the StorFS system.

EX-1005 6:10-17; EX-1004 ¶ 0052.

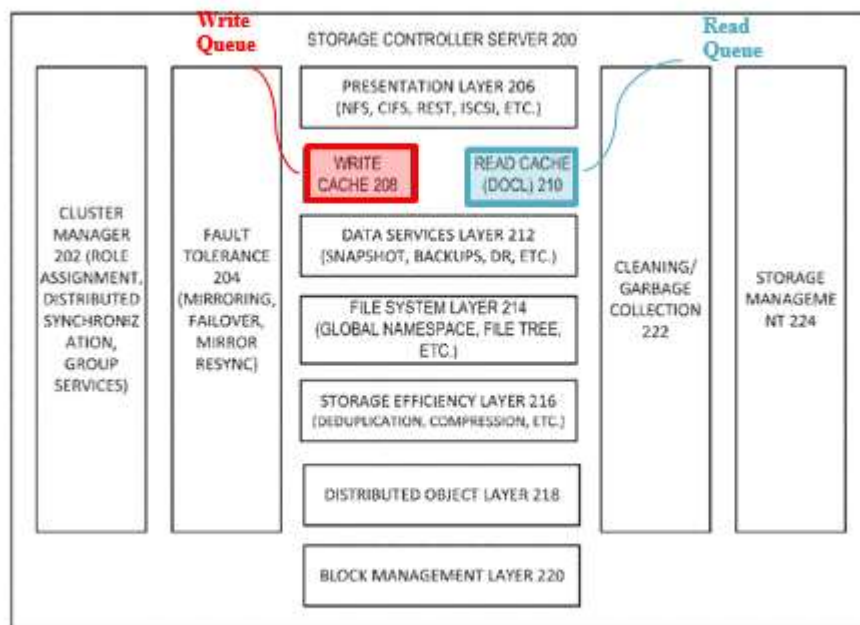


FIGURE 2

71. A POSITA understood that Agarwala's read cache is a *read queue*.

First, as discussed above, objects are initially read from persistent storage (e.g., SSD 412 or the HDD 410 of FIG. 4) and added to the read cache where they are held for reading. *EX-1005* 8:27-28, 8:27-55; *EX-1004* ¶¶0061-0062.

72. Second, in this read cache, items can be moved through and out of the read cache, as in a queue. For example, cached items can be promoted to the Most Recently Used (MRU) list based on an initial read of an object and the Most Frequently Used (MFU) list depending on frequency of access, as well as be evicted to create room for new contents:

FIG. 3B is a block diagram of one embodiment of a DOCL layer 1 (L1) cache 350. In one embodiment, the L1 cache 350 includes two types of list: Recency-based eviction lists 352; and frequency-based eviction lists 354... The recency-based eviction lists 352 are used to track objects being added to the cache and include an eviction list for data objects 360 and metadata objects 358... As the objects are read, they are initially promoted to the recency list (e.g., a Most Recently Used (MRU) list). In one embodiment, the frequency-based eviction lists are used to track accesses to objects in the cache and include an eviction list for data objects 364 and metadata objects 362. If this object is accessed within a certain time period again, the object is promoted to the frequency list (e.g., a Most Frequently

Used (MFU) list.) In one embodiment, a hotness/coldness of an object measures how recently the object was added or accessed by the StorFS system. In this embodiment, an object that has not been recently added or accessed can be a candidate for eviction from the L1 cache. In one embodiment, an object that has not been recently added or accessed can be considered “cold” or “stale” and is an object that is one that can be evicted from the L1 cache. In one embodiment, eviction from and updates to the L1 cache is at the object granularity.

*EX-1005* 7:5-47; *EX-1004* ¶0058.

73. The '632 Patent discloses that the financial industry examples pertaining to time sensitive data it provides are only illustrative and the invention can be used more broadly “in many other industries and/or applications where information from one location is required to be transmitted to other locations.”

*EX-1001* 10:21-25.

74. If the read queue must be a specific type of queue (e.g., a first in first out (FIFO) queue for time sensitive data such as financial data), it would have been obvious to a POSITA to configure Agarwala’s read cache as a FIFO queue, which was a known structure for queues.

75. Like a FIFO queue, Agarwala discloses that older objects (e.g., objects that are “cold” or “stale”) are evicted from the cache in favor of more

recently accessed or added objects, which is consistent with FIFO processing. *EX-1005* 7:5-47; *EX-1004* ¶0058. For example, the MRU and MFU operate as FIFO queues for data that are accessed only once.

76. Second, a POSITA would have been motivated to implement the cache as a FIFO queue to the extent the data being read is time sensitive.

77. In addition, I am informed that the read cache of Agarwala is a *read queue* is also consistent with Plaintiff's assertions in the Related Litigation. *See EX-1006* p. 5 (identifying a "RC Read Cache" as a *read queue*), p. 11 (asserting read cache is *second read queue*).

78. A POSITA likewise understood that Agarwala's write cache is a *write queue*. Agarwala's write cache includes a write log for holding or queuing data to be written to the persistent storage. *EX-1005* 6:12-14; *EX-1004* ¶¶0004, 00006, claim 16.

79. When the storage control server receives a write storage request, it first "stores this data in a write log by appending the data at the end of the write log. This can lead to a fast write for the data that allows a quick write storage fulfillment for the requesting application." *EX-1004* ¶0041. When the write log is full, the SC server "flushes the write log to persistent storage." *EX-1004* ¶0041.

80. If the write queue must be a specific type of queue (e.g., a first in first out (FIFO) queue for time sensitive data such as financial data), it would have been

obvious to a POSITA to configure Agarwala's write cache as a FIFO queue, which was a known structure for queues.

81. First, as noted above, like a FIFO queue, Agarwala discloses that data is appended at the end of the write cache, which is consistent with FIFO processing. *EX-1004* ¶0041. A POSITA understood that a log maintains writes in the order in which the writes are completed, and to maintain data consistency the log is written in order to the storage system i.e., the log is used as a FIFO write queue.

82. Second, flushing the write cache when full results in the older data (i.e., first data stored "in" the write cache) being the first data "out" of the write cache to make room for new data, which is consistent with FIFO processing. *EX-1004* ¶0041.

83. Third, a POSITA would have been motivated to implement the cache as a FIFO queue to the extent the data being written is time sensitive, such that the most recently added data remains accessible in the write cache and older data is stored in the persistent storage upon receipt of new data.

84. In addition, I am informed that the write cache of Agarwala is a *write queue* is also consistent with Plaintiff's assertions in the Related Litigation. *See EX-1006* p. 5 (identifying a "primary active write log" of the "caching layer" as a

*write queue*), p. 9 (accusing “active local write log on caching SSD” as *first write queue*).

*First Parallel File System*

85. A POSITA also understood that Agarwala’s storage system implements a parallel file system, including in its file system and persistent storage.

86. First, Agarwala discloses storage controller (SC) servers “collectively” provide storage for the storage clients across multiple distributed nodes:

The SC servers **110A-C** manage their underlying storage (e.g., Hard disk drive (HDD), Solid state drives (SSD), PCIe flash, etc.) and collectively provide reliable and unified storage functionality to the SC clients. The SC client **108A-C** processes input/output (I/O) requests from the applications that are local to its physical storage node and routes them to the appropriate SC servers for processing.

*EX-1005* 4:3-9; *EX-1004* ¶¶0047. Across the multiple distributed nodes, the workload is partitioned, distributed, and parallelized. *EX-1005* 5:13-27, 5:39-41; *EX-1004* ¶¶ 0050-0051.

87. Second, as described in connection with FIG. 2, the storage controllers of Agarwala's StorFS system implement a file system, including a global distributed object namespace and logical global namespace:

In one embodiment, the Distributed Object Layer 218 of the storage control server uses an API of the Block Management Layer 220 to provide a global distributed object namespace that is accessible over the network... In one embodiment, the File System Layer 214 provides a logical global namespace abstraction to organize and locate data in the cluster... In one embodiment, the Presentation Layer 206 provides an interface to access the StorFS storage using well-known standard protocols like NFS, CIFS, REST, iSCSI, etc.

*EX-1005 5:63-6:21; EX-1004 ¶¶0052, 0122.*

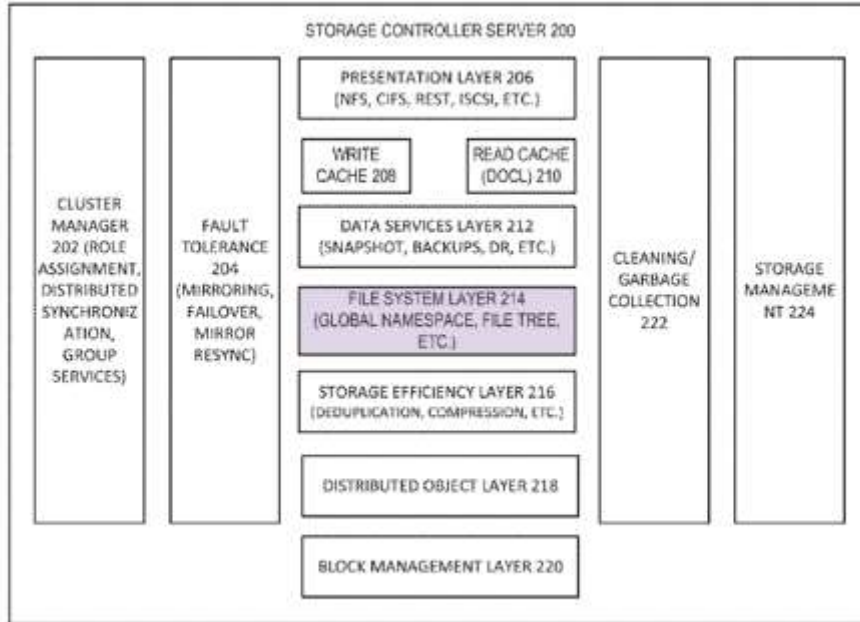


FIGURE 2

88. Third, as is characteristic of a parallel file system, “the StorFS system stripes file and file system data across multiple storage nodes in our cluster.” *EX-1004* ¶122. “[E]ach logical file (or entity) is partitioned into equal sized ‘stripe units’ [where] [t]he location of a stripe unit is determined based on a mathematical placement function.” *EX-1005* 4:30-33; *EX-1004* ¶¶0040, 0048. Further, the workload is distributed and parallelized across multiple storage nodes, as is also the characteristic of a parallel file system:

In one embodiment, the “Stripe\_Unit\_Per\_Stripe” attribute determines the number of total virtual nodes that an entity can be spread across. This enables distributing and **parallelizing** the workload across multiple storage

nodes (e.g., multiple SC servers 110A-C) In one embodiment, the StorFS system 100 uses a two-level indexing scheme that maps the logical address (e.g. offset within a file or an object) to a virtual block address (VBA) and from the VBAs to physical block address (PBA). In one embodiment, the VBAs are prefixed by the ID of the vNode in which they are stored. This vNode identifier (ID) is used by the SC client and other StorFS system 100 components to route the I/O to the correct cluster node. The physical location on the disk is determined based on the second index, which is local to a physical node.

*EX-1005* 5:13-27, 5:39-41; *EX-1004* ¶¶ 0050-0051.

89. Fourth, Agarwala discloses that its StorFS system manages storage requests of local and remote applications using its storage. *EX-1004* ¶ 39.

Agarwala also discloses that storage nodes can be located remote from one another. *EX-1004* ¶¶ 47, 59, 87. That Agarwala's persistent storage at storage node 302A is a *parallel file system* is consistent with the disclosure in the '632 Patent that data from a first node in a first region can be written to a parallel file system device at a node in a remote region. *EX-1001* 13:27-36; 16:4-5 ("In

practical use, all connected nodes can concurrently connect with PFS devices in all other regions and locations.”).

90. In addition, I am informed that Agarwala’s persistent storage is a *parallel file system* is also consistent with Plaintiff’s assertions in the Related Litigation. *See EX-1006* p. 5 (“[T]he IO Visor intercepts workload traffic and stripes the block across available nodes in the cluster... The data platform implements a log-structured file system...”), p. 11 (reading data from “Cache Drive” or “Capacity Drive” to “read cache” is reading “the first data from the second parallel file system and plac[ing] the first data in the second read queue”).

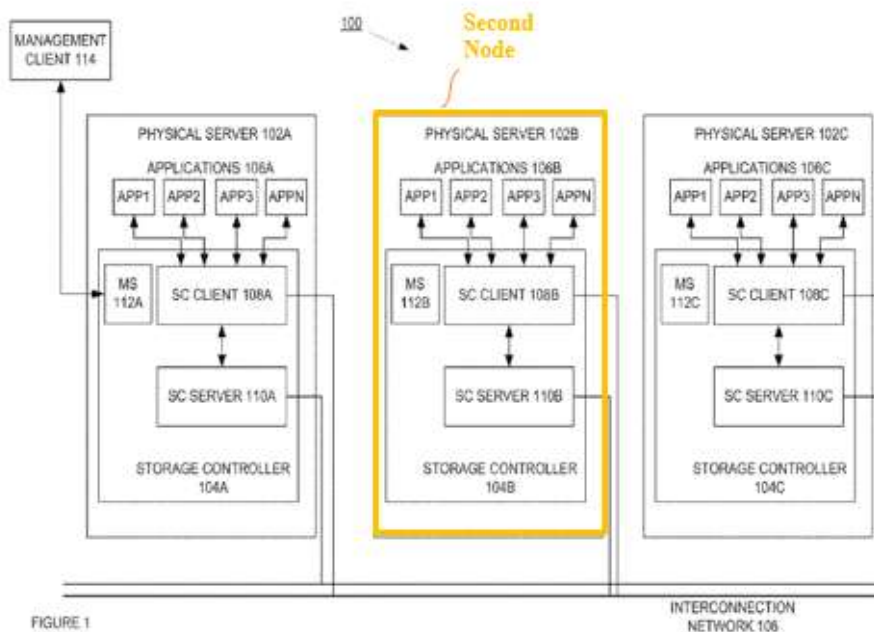
91. Agarwala discloses that it implements the read/write queues using high speed storage, SSDs and multiple levels of caches, to hide the latency of slower HDDs from the users. In one embodiment, the StorFS system writes data sequentially on SSD because random writes incur higher latency and reduce the life of SSD. In addition to taking data blocks’ recency and frequency into account, the StorFS system considers SSD’s erase block size for eviction of “cold” data that may no longer be required by the application. *EX-1004* ¶¶0058.

92. For example, the synchronous replicate module 306A receives the incoming write 304A and stores the contents of the write in the high-speed storage 312A (e.g., NVRAM, Flash, SSD, etc.) *EX-1004* ¶¶ 0059, 0081. With respect to read operations, in one embodiment, an object is initially read from the persistent

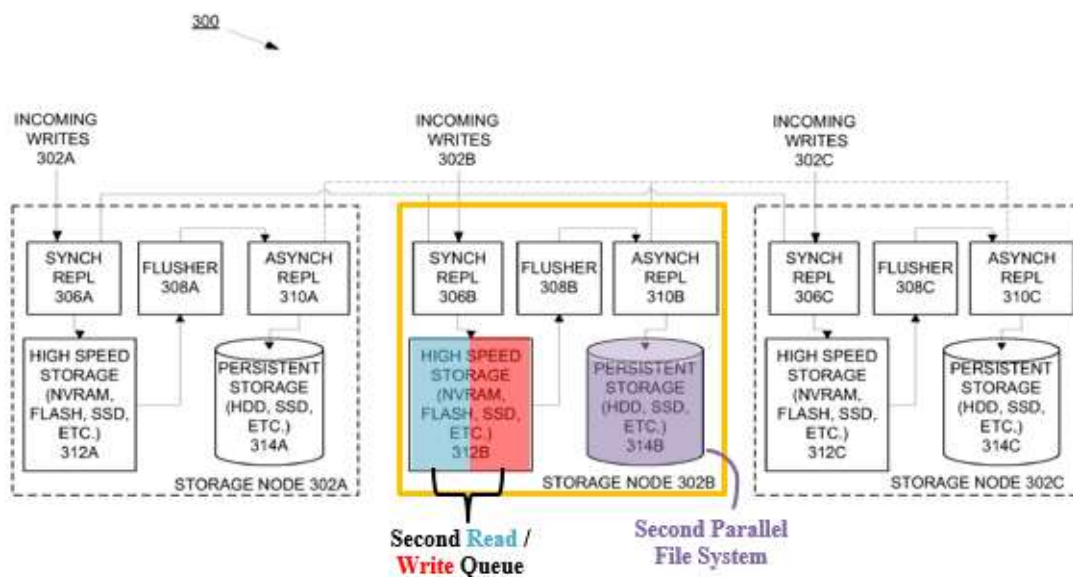
storage and added to cache which includes a most recently used (MRU) list and a most frequently used (MFU) list. Upon access, the object is added to the MRU list. If the object is not accessed again while in the MRU list, the object will go stale and be evicted from the MRU list. If the object is accessed the object is moved to the MFU list. *EX-1005* 8:27-45.

***[1.2] a second node comprising a second read queue, a second write queue, and a second parallel file system;***

93. Agarwala discloses this limitation for the reasons discussed in limitation [1.1]. As illustrated in FIG. 3, a *second storage node* (orange) contains a *write queue* (red), *read queue* (blue), and *parallel file system* (purple):



*EX-1005* FIG. 1; *EX-1004* FIG. 1.



EX-1004 FIG. 3. See also EX-1004 FIGS. 1-2; EX-1005 FIGS. 1-2.

**[1.3] wherein the first node writes first data from the first write queue to the second parallel file system; and**

94. Agarwala discloses a *first node*, as described above in [1.1], that sends data from the write log (*writes first data from the first write queue*) of the first node to persistent storage in another storage controller server (*to the second parallel file system*).

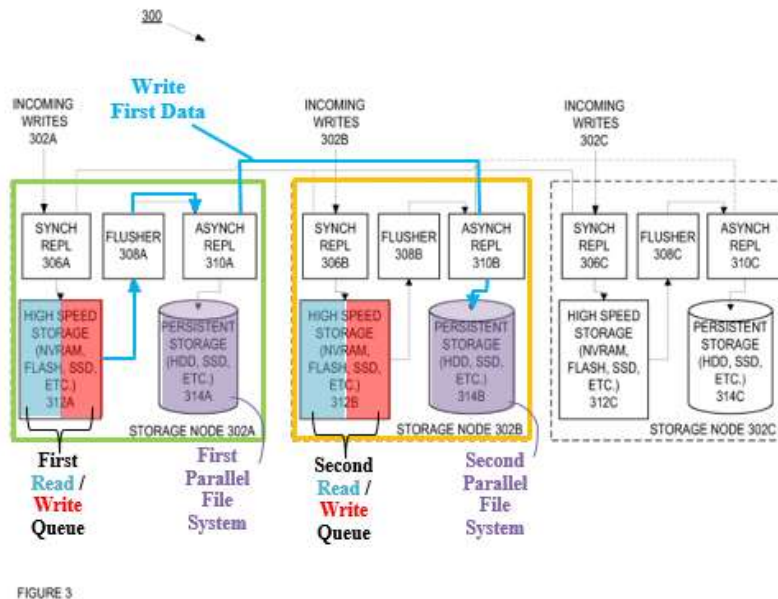
95. For example, this write occurs at least as part of Agarwala's asynchronous replication of data to mirror nodes, where data is written from the write log of Agarwala's write cache in one storage node to other storage nodes to "efficiently replicate data across the cluster". EX-1004 ¶¶0041, 0043, 0059. With reference to FIG. 3 (below):

Figure 3 is an illustration of one embodiment of an efficient distributed replication. In Figure 3, the system 300 uses a combination of synchronous and asynchronous techniques to efficiently replicate data across the cluster. In one embodiment, storage nodes of SC servers 302A-C can each receive an incoming client write 304A-C. In one embodiment, the storage node 302A receives the incoming client write with synchronous replicate module 306A. The synchronous replicate module 306A receives the write 304A and stores the contents of the write in the high-speed storage 312A (e.g., NVRAM, Flash, SSD, etc.). In one embodiment, the synchronous replicate module 306A stores the write contents at the backend of the high-speed storage 312A as described above. A flusher module 308A flushes the contents of the write to the asynchronous replication module 310A...

In one embodiment, the asynchronous replication module 310A stores the content to persistent storage 314A. In this embodiment, the asynchronous replication module 310A can additionally replicate this content to the other storage nodes 302B-C via the asynchronous replication modules 310B-C, respectively. These receiving asynchronous replication modules 310B-C, each store the content in the corresponding local perspective storage 314B-C, respectively.

EX-1004 ¶¶0081-0082.

96. FIG. 3, annotated below, further illustrates data from the first node (green) written to the second node's parallel file system (purple):



EX-1004 FIG. 3.

97. In addition, I am informed that Agarwala's disclosure of data being written from a write cache in one storage node to another storage node corresponds to the first node writes first data from the first write queue to the second parallel file system is also consistent with Plaintiff's assertions in the Related Litigation. See EX-1006 p. 9 ("[Node2] commits the write to the active local write log on the caching SSD [A1] and sends duplicate copies to other nodes, e.g., '[A2] on Node 1', i.e. the first node writes first data from the first write queue to the second parallel file system").

***[1.4] wherein the second node reads the first data from the second parallel file system and places the first data in the second read queue.***

98. Agarwala discloses that after data is written into the persistent storage of a second storage node, it will be read from persistent storage and added to the read cache (*the second node reads the first data from the second parallel file system and places the first data in the second read queue*). A POSITA understood that this read occurs following replication of the data from a first node to a second mirror node as part of an asynchronous replication. See [1.3].

99. For example, Agarwala discloses that when there is a read cache miss for data object, the data object is read from the persistent storage (*second parallel file system*) into the read cache (*second read queue*):

In one embodiment the StorFS system uses the L2 cache in tandem with the L1 cache for a multi-layer DOCL. In one embodiment, the L2 cache works as follows: On L1 cache miss, the object is looked in the L2 cache. If there is hit, the object is read and promoted to the MFU list in the L1 cache. **If there is a miss the object is read from StorFS persistent store and promoted to the MRU list in the L1 cache.**

*EX-1005* 10:44-51. The read process is also disclosed in FIG. 9B, where at block 950 content is read from persistent storage upon a cache miss and placed in the cache storage at block 966. *EX-1004 ¶¶0061-62; FIG. 9.*

100. In addition, I am informed that Agarwala's reading from persistent storage and placing the read data into a read cache corresponds to the *second node reads the first data from the second parallel file system and places the first data in the second read queue* is also consistent with Plaintiff's assertions in the Related Litigation. See EX-1006 p. 11 (reading data from "Cache Drive" or "Capacity Drive" to "read cache" is *reading the first data from the second parallel file system and placing the first data in the second read queue*).

**b. Claim 2**

***[2] The network system of claim 1, wherein the second node writes second data from the second write queue to the first parallel file system; and wherein the first node reads the second data from the first parallel file system and places the second data in the first read queue.***

101. Agarwala discloses or at least renders obvious wherein the second node writes second data from the second write queue to the first parallel file system; and wherein the first node reads the second data from the first parallel file system and places the second data in the first read queue.

102. First, as set forth above (see [1.3] and [1.4]), Agarwala *discloses the first node writes first data from the first write queue to the second parallel file system and the second node reads the first data from the second parallel file system and places the first data in the second read queue.*

103. Agarwala also discloses that any of the storage nodes can write to a different storage node within the network. *EX-1004 ¶¶0007-0008, 0039* (mirroring across multiple nodes); *EX-1005 4:7:6-12; FIG. 2* (Fault Tolerance 204 (mirroring...)).

104. A POSITA understood that just as first data is written from the first write queue of the first node of Agarwala to the second parallel file system of the second node of Agarwala, and that first data is placed in the second read queue (*see* [1.3] and [1.4]), other (second) data can also be written from the second write queue of the second node of Agarwala to the first parallel file system of the first node of Agarwala, and that second data is then placed in the second read queue.

105. Therefore, Agarwala discloses or renders obvious that *the second node writes second data from the second write queue to the first parallel file system; and wherein the first node reads the second data from the first parallel file system and places the second data in the first read queue.*

**c. Claim 3**

***[3] The network system of claim 1, wherein the first data is a carrier file comprising a header, a body, and a footer.***

106. Agarwala discloses *wherein the first data is a carrier file comprising a header, a body, and a footer.*

107. As seen in Plaintiff's infringement allegations, Plaintiff states:

| Claim #3                                                                                                                 | Accused Instrumentalities                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 3.<br>The network system of claim 1, wherein the first data is a carrier file comprising a header, a body, and a footer. | Network packets typically include a header, and a payload including a footer, thus a "first data" as "carrier file" can be inferred. |

*EX-1006* p. 28.

108. Agarwala's storage nodes are coupled together by an interconnection network. *EX-1004* ¶¶0039, 0116, FIG. 1; *EX-1004* 3:59-62, FIG. 1. A POSITA understood that data when transmitted over the interconnection network would be in packet form, and thus the data would have a header, body and footer, which are common portions of network packets.

109. The '632 Patent discloses that header information can contain information about the host origin, host destination, timestamp, and other information. *EX-1004* 12:64-66. A POSITA understood that it is well known for network packets to include headers with origin and destination information.

110. The '632 Patent also discloses that the footer can include payload checksum information to validate the integrity of the body data. *EX-1005* 12:64-66, FIG. 6, FIG. 9. A POSITA understood that it is well-known for network packets to include footers with checksum information for the packet payload.

111. Additionally, a POSITA understood that data of different types could be stored in Agarwala's StorFS system based on the applications. Also, a POSITA understood that a common way to move data around is using carrier files, and thus it would be obvious to modify the teachings of Agarwala to use carrier files.

112. For example, the '027 Publication discloses that "information can be structured, stored, and accessed in the storage system via files, blocks, logical block address (LBA), logical unit number (LUN), key-value pairs, objects, or the like." *EX-1007* ¶ 24, 27.

113. A POSITA understood that a common structure of a carrier file included a header, footer and body.

114. A POSITA would be motivated to use carrier files for sending the first data from the cache of the first node to the second node for storage in persistent storage because it would have provided an efficient method of transferring data in bulk. Indeed, Agarwala discloses that in embodiments its systems transfer data as a large bulk transfer so that this data transfer is more efficient. *EX-1004* ¶43.

**d. Claim 4**

***[4] The network system of claim 3, wherein the first nodes subsequently writes additional data to the first parallel file system.***

115. Agarwala discloses *wherein the first nodes<sup>2</sup> subsequently writes additional data to the first parallel file system.*

116. Agarwala discloses that the first node writes first data to the persistent storage of the first node (*first parallel file system*). For example, Agarwala discloses that “when the write log is full, the SC server flushes the write log to persistent storage.” *EX-1004 ¶¶ 0042; EX-1005 11:64-67.* “[A] ‘flusher’ moves the logged content asynchronously from the high-speed storage to its permanent location for persistence.” *EX-1004 ¶¶ 0059, 0069-0070* (describing asynchronous flush of data to persistent storage of target node and mirror nodes); FIGS. 3, 11B.

117. A POSITA understood that Agarwala discloses that there would be writes of data beyond the first data (i.e., *additional data*) after (*subsequently*) to the write of the first data during normal system operation in Agarwala, because that is the purpose of moving the first data from the write log to persistent storage, i.e., to allow additional data to be stored in the write log and then also be written from the write log to persistent storage.

118. Therefore, Agarwala discloses *wherein the first nodes subsequently writes additional data to the first parallel file system.*

**e. Claim 5**

---

<sup>2</sup> Claim 1 refers to only one “first node”.

***[5] The network system of claim 4, wherein the additional data is written at a set frequency.***

119. Agarwala discloses *wherein the additional data is written at a set frequency.*

120. As discussed with respect to Claim 4, the “flusher” of Agarwala moves data from the high speed cache storage to the persistent storage. Agarwala further discloses that the “flusher” operates “periodically” to move (*write*) the data to persistent storage:

New incoming writes are logged and synchronously replicated in SSDs or other high-speed storage medium.

**Periodically**, a ‘flusher’ 308A-C moves the logged content asynchronously from the high-speed storage to its permanent location for persistence.

*EX-1004 ¶0083.*

121. A POSITA understood that Agarwala uses “synchronous” to refer to writes from the client to the write log of fast storage and replication to its mirror nodes, and uses “asynchronous” to refer to the operation where data is moved or “flushed” from high-speed storage to persistent storage of the node and replicated to its mirror nodes. *EX-1004 ¶¶ 0007-0008, 0041, 0043, 0059, 0069, 0083.*

122. These “asynchronous” operations occur on a scheduled basis, i.e., *at a set frequency*, as noted above. *EX-1004 ¶0083.*

123. Therefore, a POSITA understood that Agarwala discloses *wherein the additional data is written at a set frequency.*

**f. Claim 6**

***[6] The network system of claim 4, wherein the additional data only contains information that has changed since the first data was written.***

124. Agarwala discloses *wherein the additional data only contains information that has changed since the first data was written.*

125. Agarwala discloses a deduplication module located within the flusher described above in connection with Claim 4. *EX-1004* ¶00104; FIG. 6. The deduplication module contains deduplication analyzer that “generates the dedup hint table by analyzing the streams of new objects that are written to the storage”. *EX-1004* ¶00104, FIG. 6. The dedup hint table is used “to determine whether data is being duplicated”. *EX-1004* ¶0045. “[T]he writelog flusher checks the dedup table to avoid storing duplicates to the persistent storage.” *EX-1004* ¶00104.

In one embodiment, during the flushing operation, the StorFS system, determines if an entity is stored elsewhere by using a deduplication hints table. In one embodiment, the hints table stores characteristics of the top-K storage entities in the StorFS system. If there is match with the entity being flushed, the StorFS system updates the metadata for this entity, but does not flush the entity to

the persistent storage. If not, the StorFS system flushes the entity to persistent storage.

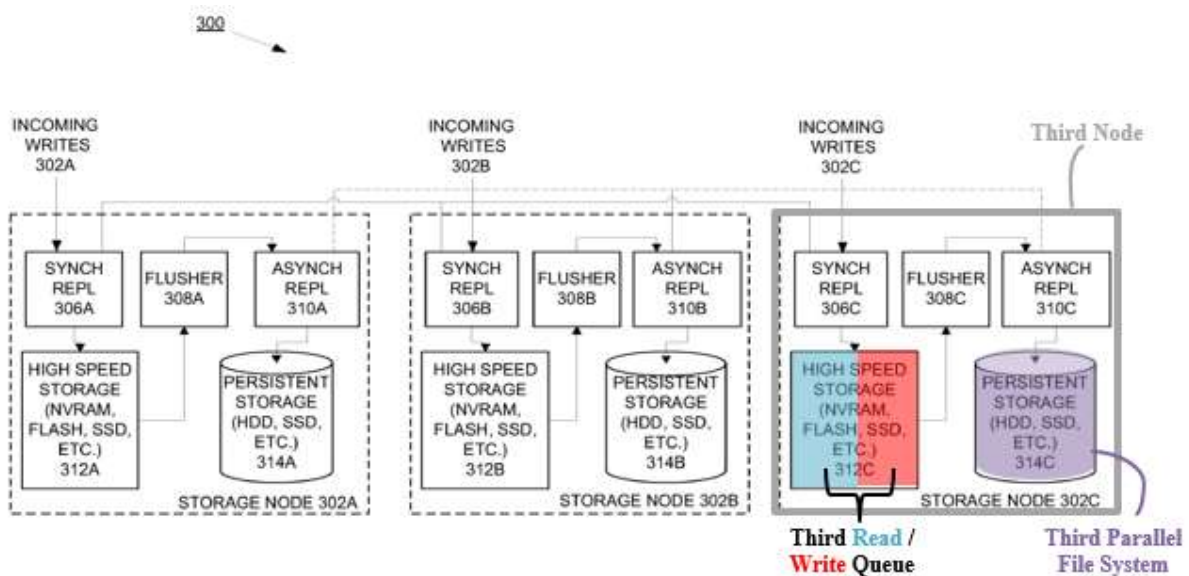
EX-1004 ¶0042.

126. By not writing duplicate data to persistent storage, a POSITA understood that Agarwala discloses that *additional data only contains information that has changed since the first data was written.*

**g. Claim 7**

**[7.0] The network system of claim 1, further comprising a third node comprising a third read queue, a third write queue, and a third parallel file system;**

127. Agarwala discloses this limitation for the same reasons as recited with respect to Claim [1.1]. Agarwala shows a *third storage node* (grey) contains a *write queue* (red), *read queue* (blue), and *parallel file system* (purple):



EX-1004 FIG. 3. See also EX-1004 FIGS. 1-2; EX-1005 FIGS. 1-2.

***[7.1] wherein the first node writes first data from the first write queue to the second and third parallel file systems at the same time; and***

128. Agarwala discloses *wherein the first node writes first data from the first write queue to the second and third parallel files system at the same time.*

129. As analyzed with respect to Claim [1.3], Agarwala's first storage node (*first node*) writes data (*first data*) from the write log of the write cache (*first write queue*) to the persistent storage of the second storage node (*second parallel file system*) as part of asynchronous replication. *See* Claim [1.3].

130. Second, Agarwala discloses that this asynchronous replication involves replication to multiple storage nodes. For example:

In this embodiment, the asynchronous replication module 310A can additionally replicate this content to the other storage nodes 302B-C via the asynchronous replication modules 310B-C, respectively. These receiving asynchronous replication modules 310B-C, each store the content in the corresponding local perspective storage 314B-C, respectively.

*EX-1004 ¶0082.*

131. A POSITA understood that one of the multiple mirror nodes would be the second node and the other, the third node.

132. A POSITA understood that Agarwala uses “asynchronous” to refer to the operation where data is moved or “flushed” from high-speed storage to persistent storage of the node and simultaneously replicated to its mirror nodes.

*EX-1004 ¶¶ 0008, 0043, 0059, 0069, 0083.*

133. These “asynchronous” operations occur on a scheduled basis, i.e., *at a set frequency*, as noted above. *EX-1004 ¶0083.*

134. Thus, a POSITA would have understood that periodically, asynchronous replication module 310A of the first node would replicate data to the second and third storage nodes, and that this replication to the second and third storage nodes would occur *at the same time* because it is efficient to make multiple copies at the same time at the first node to send to the second and third nodes. *EX-1004 ¶0070, FIG. 11B (one step 1156 of “Asynchronously Send the Data and Metadata to the Vnode and Mirror Nodes in the AMS).*

135. Additionally, a POSITA understood that a common way to replicate data to multiple locations was to do so at the same time, and thus it would be obvious to modify the teachings of Agarwala so that asynchronous replication module 310A is programmed to replicate first data at the second and third nodes at the same time because doing would be a more efficient use of resources and was a common and well-known technique that had predictable results. A POSITA

understood that writing to multiple nodes at the same time provided the benefit of keeping replicas consistent across the system.

136. Therefore, a POSITA understood that Agarwala discloses that a first node writes first data from the first write queue to the second and third parallel file system at the same time.

***[7.2] wherein the third node reads the first data from the third parallel file system and places the first data in the third read queue.***

137. Agarwala discloses *wherein the third node reads the first data from the third parallel file system and places the first data in the third read queue.*

138. As discussed above, Agarwala discloses that the second node reads the first data from the second parallel file system and places the first data in the second read queue. *See [1.4].*

139. Agarwala also discloses that the first data is replicated to the third parallel file system of the third node. *See [7.1].*

140. For the same reasons as set forth in connection with [1.4], a POSITA understood that the first data would also be read from the third parallel file system to the third read queue, e.g., upon a cache miss.

141. Thus Agarwala discloses or renders obvious *wherein the third node reads the first data from the third parallel file system and places the first data in the third read queue.*

## **CONCLUSION AND DECLARATION**

The findings and opinions set forth in this declaration are based on my research and analysis to date. I may continue my examinations in view of additional documents or other factual evidence over the course of this proceeding that may necessitate supplementing and/or refining my opinions. I reserve the right to add to, alter, or delete my opinions and my Declaration upon discovery of additional information. I reserve the right to make such changes as necessary.

In signing this declaration, I understand that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

I declare that all statements made herein on 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 the 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.

[SIGNATURE ON FOLLOWING PAGE]

Dated: 25<sup>th</sup> January 2024

By:   
A.L. Narasimha Reddy

Dr. A.L. Narasimha Reddy

## **APPENDIX A – MATERIALS RELIED UPON**

| <i><b>Exhibit #</b></i> | <i><b>Description</b></i>                                                                                                                                                                                                       |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1001</b>             | U.S. Patent No. 11,146,632 (“the ’632 Patent”)                                                                                                                                                                                  |
| <b>1002</b>             | Select portions of prosecution history of the ’632 Patent (“File History”)                                                                                                                                                      |
| <b>1004</b>             | U.S. Provisional Patent Application No. 61/739,685                                                                                                                                                                              |
| <b>1005</b>             | U.S. Patent No. 9,582,421 to Agarwala et al.                                                                                                                                                                                    |
| <b>1006</b>             | UMBRA Technologies Ltd.’s First Infringement Analysis of Claims 1-4 & 7 of U.S. Patent No. 11,146,632, <i>UMBRA Technologies Ltd. (UK) et al v. Cisco Sys., Inc.</i> , E.D. Tex. Case No. 1-23-cv-00903, Dkt 1-8 (Aug. 2, 2023) |
| <b>1007</b>             | U.S. Patent Application Publication No. 2016/0100027 to Haghighi (“’027 Publication”)                                                                                                                                           |
| <b>1008</b>             | U.S. Patent Application Publication No. 2011/0145367 to Ananthanarayanan et al. (“’367 Publication”)                                                                                                                            |
| <b>1009</b>             | Kassick et al., “Dynamic I/O Reconfiguration for NFS-Based Parallel File System.” In 2011 19 <sup>th</sup> International Euromicro International Conference on Parallel, Distributed and Network-Based Processing (IEEE, 2011)  |
| <b>1010</b>             | Li et al., “Small Files Problem in Parallel File System.” In 2011 International Conference on Network Computing and Information Security (NCIS) (IEEE, 2011)                                                                    |
| <b>1011</b>             | Marc Farley, <i>Storage Networking Fundamentals: An Introduction to Storage Devices, Subsystems, Applications, Management, and File Systems</i> (Cisco Press 2005)                                                              |
| <b>1012</b>             | James O’Reilly, <i>Network Storage: Tools and Technologies for Storing Your Company’s Data</i> (Elsevier Inc. 2017)                                                                                                             |

## **APPENDIX B – CURRICULUM VITAE**