

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

MICROSOFT CORPORATION,  
Petitioner,

v.

LS CLOUD STORAGE TECHNOLOGIES, LLC,  
Patent Owner.

IPR2023-00622  
U.S. Patent No. 6,549,988  
Issued: April 15, 2003  
Application No.: 09/236,409  
Filed: January 22, 1999

Title: DATA STORAGE SYSTEM COMPRISING  
A NETWORK OF PCS AND METHOD USING SAME

---

**PETITION FOR**  
**INTER PARTES REVIEW OF U.S. PATENT NO. 6,549,988**

**TABLE OF CONTENTS**

|                                                                       | Page(s) |
|-----------------------------------------------------------------------|---------|
| LIST OF EXHIBITS .....                                                | v       |
| MANDATORY NOTICES UNDER 37 C.F.R. § 42.8 .....                        | vi      |
| 1. Real Party-In-Interest.....                                        | vi      |
| 2. Related Matters .....                                              | vi      |
| 3. Lead And Back-Up Counsel, And Service Information.....             | vi      |
| I. INTRODUCTION .....                                                 | 1       |
| II. GROUNDS FOR STANDING PER SECTION 42.104(a).....                   | 1       |
| III. IDENTIFICATION OF CHALLENGE .....                                | 1       |
| A. Statement Of The Precise Relief Requested / Statutory Grounds..... | 1       |
| IV. THE '988 PATENT .....                                             | 2       |
| A. The '988 Patent's Specification .....                              | 3       |
| B. The Prosecution History.....                                       | 5       |
| V. LEVEL OF SKILL IN THE PERTINENT ART .....                          | 7       |
| VI. CLAIM CONSTRUCTION .....                                          | 7       |
| A. Volume Access Table.....                                           | 8       |
| VII. GROUND 1: CLAIMS 1 AND 3<br>ARE ANTICIPATED BY BERGSTEN .....    | 12      |
| A. Bergsten.....                                                      | 12      |
| B. Claim 1 .....                                                      | 17      |
| C. Claim 3 .....                                                      | 31      |

|       |                                                                                                                                                 |    |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------|----|
| VIII. | GROUND 2: CLAIMS 2, 4, AND 6 ARE OBVIOUS OVER<br>BERGSTEN IN VIEW OF THE KNOWLEDGE OF A POSITA .....                                            | 32 |
| A.    | Claim 2 .....                                                                                                                                   | 32 |
| B.    | Claim 4 .....                                                                                                                                   | 38 |
| C.    | Claim 6 .....                                                                                                                                   | 39 |
| IX.   | GROUND 3: CLAIMS 2, 4, AND 6 ARE<br>OBVIOUS OVER BERGSTEN IN VIEW OF BLUMENAU .....                                                             | 52 |
| A.    | Blumenau.....                                                                                                                                   | 52 |
| B.    | Rationale For Combining Bergsten And Blumenau .....                                                                                             | 54 |
| C.    | Claim 2 .....                                                                                                                                   | 57 |
| D.    | Claim 4 .....                                                                                                                                   | 60 |
| E.    | Claim 6 .....                                                                                                                                   | 61 |
| X.    | GROUND 4 & 5: CLAIM 5 IS OBVIOUS WHEN<br>SMITH IS COMBINED WITH BERGSTEN (GROUND 4) AND<br>COMBINED WITH BERGSTEN AND BLUMENAU (GROUND 5) ..... | 64 |
| A.    | Smith.....                                                                                                                                      | 65 |
| B.    | Rationale For Combining Smith With Bergsten (Ground 4) .....                                                                                    | 65 |
| C.    | Rationale For Combining Smith<br>With Bergsten And Blumenau (Ground 5) .....                                                                    | 67 |
| D.    | Claim 5 .....                                                                                                                                   | 67 |
| XI.   | NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS .....                                                                                                   | 74 |
| XII.  | THE BOARD SHOULD NOT USE ITS<br>DISCRETION TO DENY INSTITUTION OF THIS PETITION.....                                                            | 75 |
| XIII. | CONCLUSION.....                                                                                                                                 | 76 |
|       | CERTIFICATE OF COMPLIANCE.....                                                                                                                  | 77 |

**TABLE OF AUTHORITIES**

Page(s)

**Cases**

|                                                                                   |   |
|-----------------------------------------------------------------------------------|---|
| <i>Phillips v. AWH Corp.</i> ,<br>415 F.3d 1303 (Fed. Cir. 2005) .....            | 8 |
| <i>Williamson v. Citrix Online, LLC</i> ,<br>792 F.3d 1339 (Fed. Cir. 2015) ..... | 8 |

**Statutes**

|                       |               |
|-----------------------|---------------|
| 35 U.S.C. § 102 ..... | 1, 12, 52, 64 |
| 35 U.S.C. § 103 ..... | 1, 2          |
| 35 U.S.C. § 112 ..... | 8             |

**Rules**

|                          |     |
|--------------------------|-----|
| 37 C.F.R. § 42.10 .....  | vii |
| 37 C.F.R. § 42.15 .....  | 2   |
| 37 C.F.R. § 42.24 .....  | 77  |
| 37 C.F.R. § 42.8 .....   | vi  |
| 37 C.F.R. § 42.104 ..... | 1   |
| 83 Fed. Reg. 51340 ..... | 8   |

**LIST OF EXHIBITS**

| <b>No.</b> | <b>Description</b>                                                                             |
|------------|------------------------------------------------------------------------------------------------|
| 1001       | U.S. Patent No. 6,549,988 (“ <b>the ’988 Patent</b> ”)                                         |
| 1002       | File History of U.S. Patent No. 6,549,988                                                      |
| 1003       | Declaration of Jon. B. Weissman, Ph.D., dated March 23, 2023                                   |
| 1004       | U.S. Patent No. 6,073,209 (“Bergsten”)                                                         |
| 1005       | U.S. Patent Application No. 08/828,888                                                         |
| 1006       | U.S. Patent No. 6,421,711 (“Blumenau”)                                                         |
| 1007       | U.S. Patent Application No. 09/106,299                                                         |
| 1008       | Spritely NFS: Experiments with Cache-Consistency Protocols” (“Srinivasan”)                     |
| 1009       | R.H. Katz, Network-Attached Storage Systems, 1992 (“Katz”)                                     |
| 1010       | Gang Ma, et al., An evaluation of storage systems based on network-attached disks, 1998 (“Ma”) |
| 1011       | Jim Handy, The Cache Memory Book, 1998 (excerpts) (“Handy”)                                    |
| 1012       | Microsoft Computer Dictionary, 2d ed., 1994 (excerpts)                                         |
| 1013       | Kinariwala et al., Programming in C, 1993 (“Kinariwala”)                                       |
| 1014       | Parsaye, et al., Intelligent Database Tools & Applications, 1993 (“Parsaye”)                   |
| 1015       | Alan J. Smith, Cache Memories, 1982 (“Smith”)                                                  |
| 1016       | U.S. Patent No. 4,500,954 (“Duke”)                                                             |
| 1017       | U.S. Patent No. 4,433,374 (“Hanson”)                                                           |
| 1018       | Smith Copyright Registration                                                                   |

**MANDATORY NOTICES UNDER 37 C.F.R. § 42.8**

**1. Real Party-In-Interest**

Microsoft Corporation is the sole real party-in-interest.

**2. Related Matters**

The '988 patent (EX1001) has been asserted in the following litigations:

- *LS Cloud Storage Technologies, LLC v. Google LLC*, Case No. 1:22-cv-00853 (W.D. Tex.), filed March 25, 2022;
- *LS Cloud Storage Technologies, LLC v. Microsoft Corporation*, Case No. 1:22-cv-00974 (W.D. Tex.), filed March 25, 2022;
- *LS Cloud Storage Technologies, LLC v. Amazon.com, Inc. et al.*, Case No. 1:22-cv-01167 (W.D. Tex.), filed March 25, 2022;
- *LS Cloud Storage Technologies, LLC v. Cisco Systems, Inc.*, Case No. 6:22-cv-00319 (W.D. Tex.), filed August 9, 2022

**3. Lead And Back-Up Counsel, And Service Information**

| <b>Lead Counsel</b>                                           | <b>Back-up Counsel</b>                                                                                                                                                                                              |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Todd M. Siegel (Reg. No. 73,232)<br>todd.siegel@klarquist.com | Cameron D. Clawson (Reg. No. 73,509)<br>cameron.clawson@klarquist.com<br><br>Claire E. Green (Reg. No. 75,290)<br>claire.green@klarquist.com<br><br>Andrew M. Mason (Reg. No. 64,034)<br>andrew.mason@klarquist.com |
| KLARQUIST SPARKMAN, LLP<br>121 SW Salmon Street, Suite 1600   |                                                                                                                                                                                                                     |

|                                                                       |
|-----------------------------------------------------------------------|
| Portland, Oregon, 97204<br>503-595-5300 (phone)<br>503-595-5301 (fax) |
|-----------------------------------------------------------------------|

Petitioner consents to service via email at the above email addresses.

Pursuant to 37 C.F.R. § 42.10(b), concurrently filed with this Petition is a Power of Attorney executed by Petitioner and appointing the above counsel.

## **I. INTRODUCTION**

Microsoft Corporation (“Petitioner”) respectfully requests *inter partes* review (“IPR”) of claims 1-6 of U.S. Patent No. 6,549,988 (EX1001, “’988 patent”), allegedly owned by LS Cloud Storage Technologies, LLC, (“Patent Owner”). The ’988 patent claims a computer system that uses software for managing cached data stored across a distributed storage system. As discussed below, this subject matter was well-known to skilled artisans before the patent’s filing date, rendering the Challenged Claims unpatentable as anticipated and/or obvious.

## **II. GROUNDS FOR STANDING PER SECTION 42.104(a)**

Petitioner certifies that the ’988 patent is available for IPR and that Petitioner is not barred or estopped from requesting an IPR challenging the patent claims on the grounds identified in this petition.

## **III. IDENTIFICATION OF CHALLENGE**

### **A. Statement Of The Precise Relief Requested / Statutory Grounds**

Petitioner requests *inter partes* review of claims 1-6 (the “Challenged Claims”) of the ’988 patent, on the following statutory grounds:

|                 | Reference(s)                         | Basis           | Claims  |
|-----------------|--------------------------------------|-----------------|---------|
| <b>Ground 1</b> | U.S. Pat. No. 6,073,209 (“Bergsten”) | 35 U.S.C. § 102 | 1, 3    |
| <b>Ground 2</b> | Bergsten                             | 35 U.S.C. § 103 | 2, 4, 6 |

|                 |                                                                                                               |                 |         |
|-----------------|---------------------------------------------------------------------------------------------------------------|-----------------|---------|
| <b>Ground 3</b> | Bergsten in view of U.S Pat. No. 6,421,711 (“Blumenau”)                                                       | 35 U.S.C. § 103 | 2, 4, 6 |
| <b>Ground 4</b> | Bergsten in view of “Cache Memories,” Alan Jay Smith, Computing Surveys, Vol. 14, No. 3, Sept. 1982 (“Smith”) | 35 U.S.C. § 103 | 5       |
| <b>Ground 5</b> | Bergsten in view of Blumenau and further in view of Smith                                                     | 35 U.S.C. § 103 | 5       |

For each ground, in Sections VII-X below, the petition presents evidence of unpatentability and establishes a reasonable likelihood that the Petitioner will prevail in establishing that each Challenged Claim is unpatentable.

With the filing of this petition an electronic payment of \$41,500 for the requisite fees is being charged to deposit account no. 02-4550. 37 C.F.R. § 42.15(a). Any fee adjustments may be debited/credited to the deposit account.

#### **IV. THE '988 PATENT**

The '988 patent, titled “DATA STORAGE SYSTEM COMPRISING A NETWORK OF PCS AND METHOD USING SAME” issued on April 15, 2003

from U.S. Patent Application No. 09/236,409 (“’409 Application”), filed on January 22, 1999. EX1001.

**A. The ’988 Patent’s Specification**

The ’988 patent is directed at a “data storage system that is assembled out of standard modules, using off-the-shelf hardware components and a standard general-purpose operating system that supports standard network software and protocols.” *Id.*, 3:23-26. The alleged invention is further directed at “a cached data storage system that permits independent data accesses from I/O channel attached local hosts, network attached remote hosts, and network-attached remote data storage systems.” *Id.*, 3:27-30.

Figure 1, shown below, illustrates “[t]he PC data storage system 131 [that] services a plurality of channel attached host processors 111, 112 using channels 121, 122, and a plurality of network attached host processors 106, 107 using network link 151, and a plurality of network attached data storage systems 132, 133 using network links 152, 153. PC storage system 132 services channel attached hosts 157, 158.” EX1001, 4:67-5:6; EX1003, ¶46.

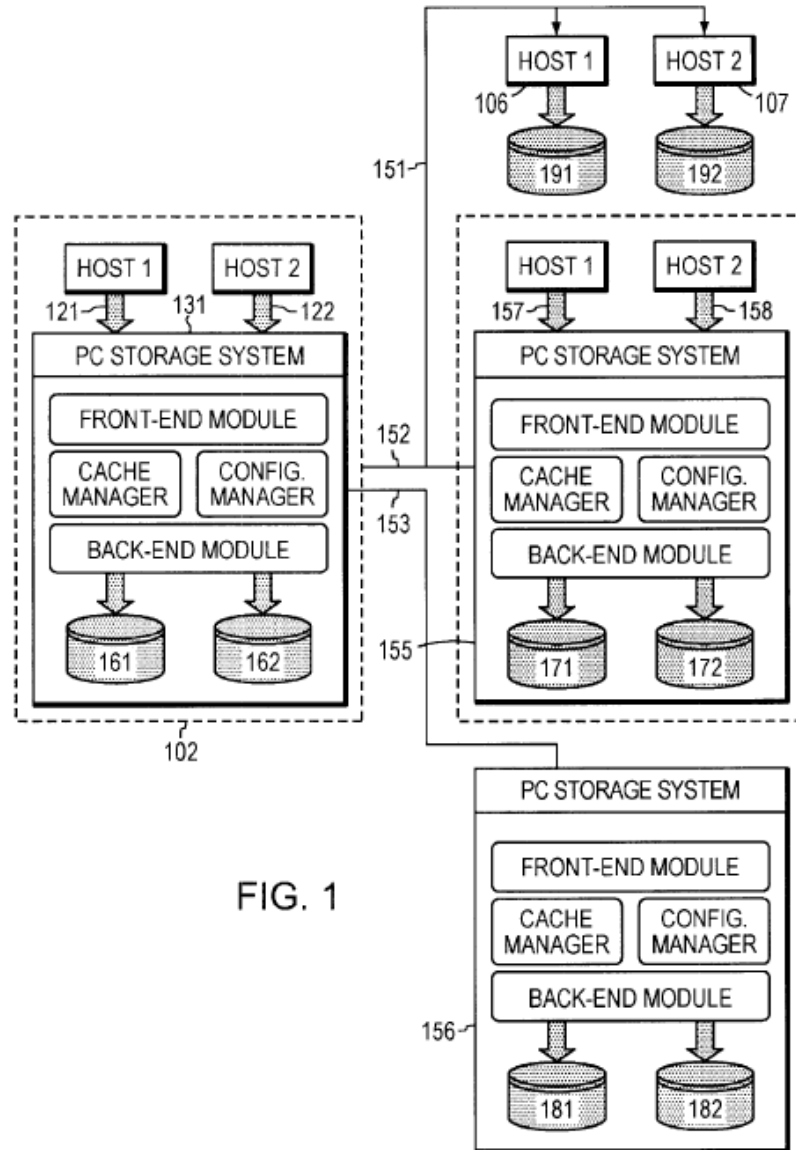


FIG. 1

### EX1001, '988 Patent Fig. 1

The preferred embodiment of the alleged invention uses well-established technologies such as SCSI channels for I/O traffic and Ethernet link for network traffic. In FIG. 2, the alternate embodiment of the alleged invention uses fiber

channel technology for both I/O traffic and network traffic. The fiber channel connects computers and hard disks into one logical network.

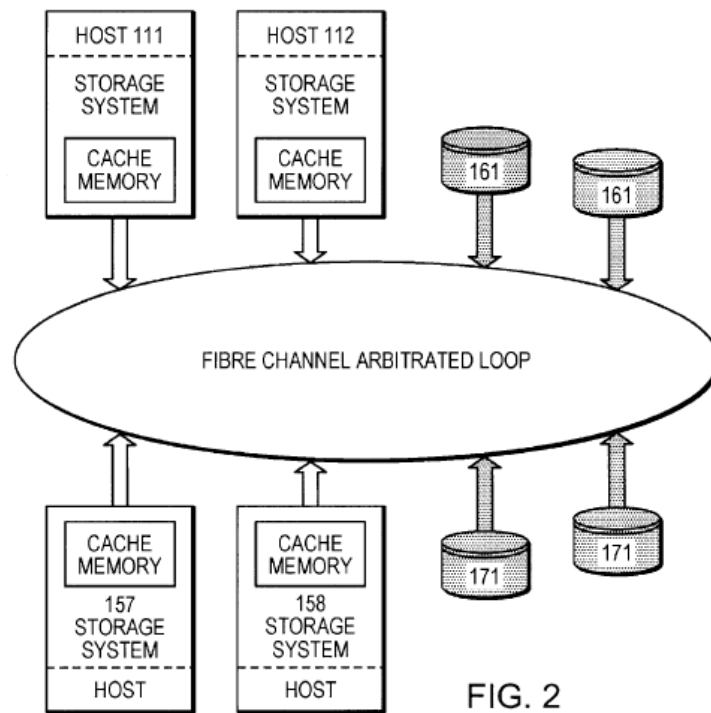


FIG. 2

EX1001, '988 Patent Fig. 2

## B. The Prosecution History

The '409 application was prosecuted pro se. EX1002. On November 20, 2000, the Examiner rejected claims 2 and 3 as failing to satisfy the enablement requirement and claims 1 and 4 as anticipated by U.S. Patent No. 6,044,438 ("Olnowich"). *Id.*, 80-85.

The Applicant amended all claims and amended the specification with additional disclosures. *Id.*, 93-101. Following a subsequent office action, the

Applicant amended claim 1 to recite a “volume access table” and further amended the specification by adding disclosure regarding the use of the “volume access table,” which corresponds to column 6, lines 15-42 of the ’988 patent. *Id.*, 149-160.

On June 28, 2002, the Examiner issued a final office action, rejecting all claims as indefinite for the “volume access table” limitations and claims 1 and 4 as anticipated by Olnowich. *Id.*, 161-169.

The Applicant then amended claim 1 to remove the “volume access table” and to recite “back-end software, responsive to said configuration manager software, for handling reads and writes to disks corresponding to the I/O requests but without communication over the I/O channel adapter, thereby separating disk operations from network and I/O traffic.” *Id.*, 175-176. The response also added new claims 5 and 6. *Id.*, 178.

The Examiner subsequently issued a Notice of Allowance. *Id.*, 225-230. The Examiner provided reasons for allowance, particularly emphasizing the following limitation of claim 1 as purportedly not taught by the prior art of record:

back-end software, responsive to the configuration manager software, for handling reads and writes to disks corresponding to the I/O requests but without communication over the I/O channel adapter, thereby separating disk operations from network and I/O traffic.

*Id.*, 227-28.

## **V. LEVEL OF SKILL IN THE PERTINENT ART**

As discussed herein and in the accompanying Declaration of Jon Weissman (EX1003), the pertinent technical concepts were well known in the art long before the '409 application was filed. For example, computer scientists recognized the advantages of using multiple storage controllers across a distributed network. EX1003, ¶¶31-35 (citing EX1009). Computer scientists also recognized the value of storing and accessing data from cache memory. EX1003, ¶¶36-40 (citing EX1010, EX1011). In order to avoid data consistency issues across the network, computer scientists implemented known caching protocols. EX1003, ¶¶41-45 (citing EX1008, EX1011).

The person of ordinary skill in the art in January 1999 (“POSITA”) would have a bachelor’s degree in Computer Science, Computer Engineering, or Electrical Engineering, or an equivalent course of study, as well as two or more years of academic or industry experience in the field of computer networking, computer architecture and storage management. EX1003, ¶30. Lack of work experience could have been offset by additional education, and vice versa. *Id.*

## **VI. CLAIM CONSTRUCTION**

For *inter partes* review, claim terms should be given the ordinary meaning that the terms would have to a POSITA on the earliest effective filing date, in view of the

specification and file history. 83 Fed. Reg. 51340; *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-1313 (Fed. Cir. 2005) (*en banc*).

Unless otherwise expressly discussed, Petitioner applies the plain and ordinary meaning of all claim terms below. Petitioner does not waive any argument in any litigation that claim terms in the '988 patent are indefinite or otherwise invalid nor does Petitioner waive its right to raise additional issues of claim construction in any litigation.

There is a rebuttable presumption against invoking pre-AIA 35 U.S.C. § 112 ¶6 when the word “means” is not in the claim. *Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1348 (Fed. Cir. 2015) (*en banc* in relevant part). Here, the Challenged Claims do not use “means” language, and Petitioner is unaware of any assertion by Patent Owner that the claims nonetheless invoke pre-AIA 35 U.S.C. § 112 ¶6. Thus, Petitioner applies the plain and ordinary meaning of the claim language except where otherwise noted.

**A. Volume Access Table**

Challenged Claims 2, 4, 5, and 6 of the '988 patent recite a “volume access table.” “Volume access table” was not a commonly known term when the '409 application was filed. EX1003, ¶85. A search of patent and publication references at [www.google.com/patents](http://www.google.com/patents) shows a single reference (EX1006) filed prior to the '409 application to use the term in its entirety. Accordingly, a POSITA would understand

this term based on the plain and ordinary meaning of the individual words—“volume,” “access,” and “table”—in the context of the patent specification. *Id.* The Microsoft Computer Dictionary defines these words as follows (EX1012, 414, 8, 383):

- “volume” – “Another name for a disk or tape that stores computer data. Each separate disk or tape is a volume, with a unique volume name or number. In some instances, a large hard disk can be divided into separate volumes, each of which is treated as a separate disk, even though all volumes reside physically in the same drive.”
- “access” – “To obtain entry to, or to locate, read into memory, and make ready for, some operation. Access is used with regard to disks, files, records and network entry procedures.”
- “table” – “In programming, a data structure usually consisting of a list of entries, each entry being identified by a unique key and containing a set of related values. A table is often implemented as an array of records, a linked list, or (in more primitive languages) several arrays of different data types, all using a common indexing scheme. See also array, list, record. In relational databases, a data structure characterized by rows and columns, with data occupying or potentially occupying

each cell formed by a row column intersection. The table is the underlying structure of a relation.”

The as-filed ’409 application included no definition for “volume access table.” The extent of the as-filed specification’s description of a “volume access table” is contained in the following three passages:

The flowchart uses a volume access table of Fig. 5 is [sic] controlled by the configuration manager.

EX1002, 7.

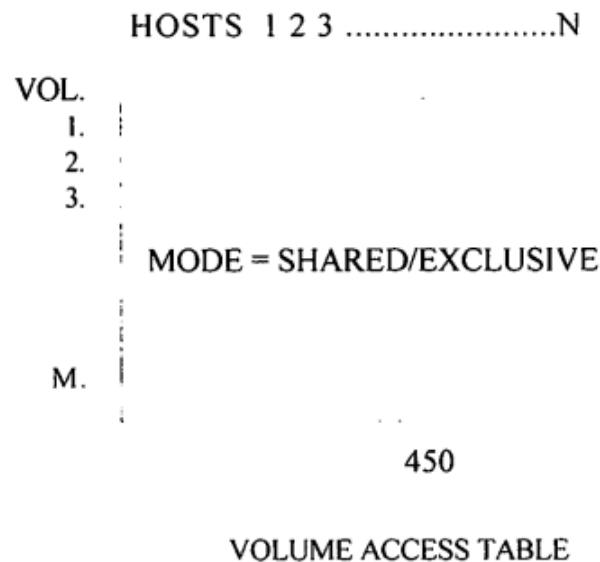
Depending upon the status of the value in a volume access table 450 the requests are routed through either 4A for write exclusive mode, 4B for read exclusive, 4C for write shared and 4D for read shared.

*Id.*, 8.

The configuration manager uses a volume access table 450 that has been shown in Fig. 4.

*Id.*, 10.

The as-filed application also depicts such a volume access table in Fig. 4. *Id.*, 20.



**EX1002, '409 Application As-Filed from '988 FH, Fig. 4 (Excerpt)**

Therefore, the as-filed application discloses only that the volume access table is used by configuration manager software to determine an access mode for a read/write request. EX1003, ¶84. Thus, a POSITA would understand “volume access table” to mean a table used to store access characteristics of storage volumes. *Id.*, ¶85.

While the '988 patent provides no expressed definition for a “volume access table,” the prior art Blumenau reference “specifies a correspondence between hosts and respective lists of volumes accessible to the hosts.” EX1006, 13:4-6. To the extent the Board does not adopt Petitioner’s proposed construction and instead adopts a construction requiring such a table to specify a correspondence between

hosts and respective volumes accessible to the hosts, the below Grounds explain why the challenged claims are unpatentable under such a construction.

**VII. GROUND 1: CLAIMS 1 AND 3 ARE ANTICIPATED BY BERGSTEN**

As explained below, claims 1 and 3 are anticipated by U.S. Patent No. 6,073,209 (“Bergsten”).

**A. Bergsten**

Bergsten, titled “Data Storage Controller Providing Multiple Hosts With Access To Multiple Storage Subsystems,” issued from U.S. Patent Application No. 08/828,888, filed on March 31, 1997. EX1005. The original Bergsten application, filed more than twenty months before the ’409 application, discloses the subject matter of the issued Bergsten patent relied on herein. *Id.*; EX1003, ¶49. Thus, Bergsten qualifies as prior art under pre-AIA Section 102(e).

Bergsten is directed to “a storage controller which allows multiple host computer systems at different locations to access any of multiple copies of stored data.” EX1004, 3:2-4; EX1003, ¶50. The “storage controller is not dependent upon any particular hardware or software” and “it emulates a local storage array” and a “host computer system” for the storage array. EX1004, 3:11-17. The host computer accesses “stored data using virtual device addresses, which are mapped to real device addresses by the storage controller.” *Id.*, 3:17-19.

Figure 1 illustrates multiple storage controllers (3-1 through 3-M) providing multiple host computers (2-1 through 2-M) access to multiple storage arrays (4-1 through 4-M). EX1003, ¶51.

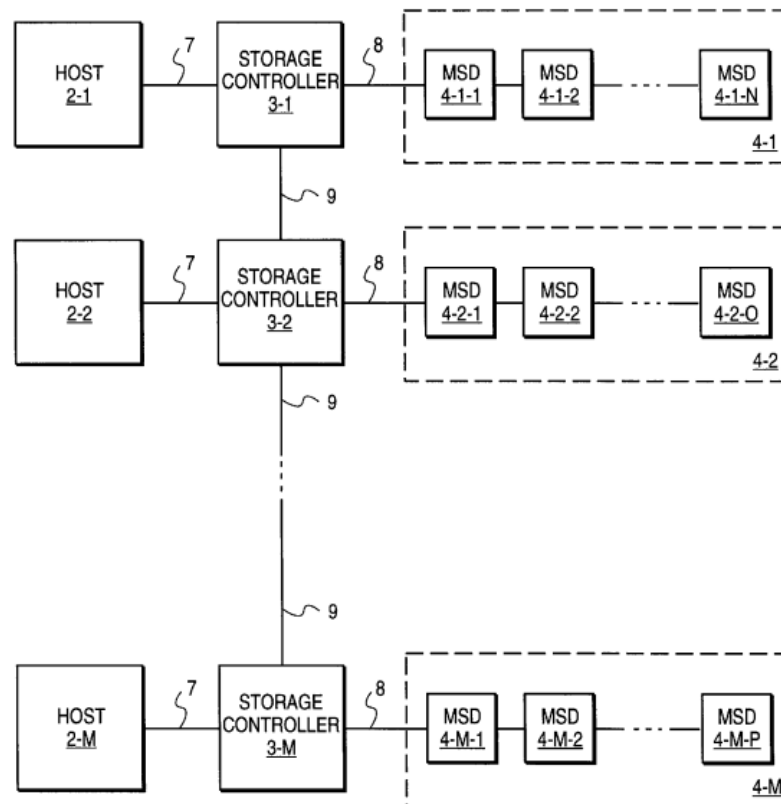


**FIG. 1**

**EX1004, Bergsten Fig. 1**

Bergsten explains that each storage array (4-1 through 4-M) includes multiple mass storage devices. EX1004, 3:28-30. Bergsten explains that the communication paths between hosts and storage controllers may be implemented via any “suitable

type of link,” including local area network or Ethernet. EX1003, ¶52; EX1004, 3:35-45.

Figure 3 illustrates the hardware architecture of a storage controller. Notably “[t]he host device interface 14 connects communication path 7 to the bus 17 in order to connect a local host computer to the storage controller 3.” *Id.*, 6:41-43. Bergsten provides that the host device interface can be a SCSI adapter. *Id.*, 6:46-52. Bergsten explains that “[t]he controller device interface 16 connects communication path 9 to the bus 17 in order to connect a remote storage controller to a storage controller 3.” *Id.*, 6:56-61; EX1003, ¶54.

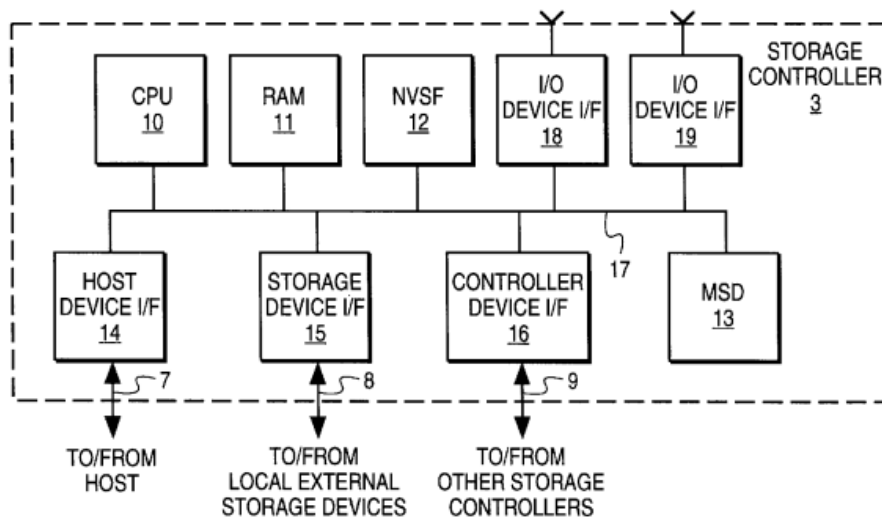


**FIG. 3**

**EX1004, Bergsten Fig. 3**

Figure 4, illustrates the functional modules within a storage controller, including “an operating system (OS) 20 coupled to emulation drivers 21, physical drivers 22, communication drivers 23, and local memory 24.” EX1004, 7:24-26.

The OS 20 controls the overall operation of the storage controller, including emulation, data flow, caching of data in the local memory 24, control of data mirroring, and error recovery. Emulation drivers 21 are controlled by the OS 20 to provide communication of data between the storage controller 3 and the local host computer 2 via communication path 7. In particular, the emulation drivers 21 receive read and write requests from the host computer 2 and ... transfer any data received from the local storage array or a remote storage array to the local host computer. The communication drivers 23 are controlled by the OS 20 to provide communication of data between the storage controller 3 and remote storage controllers via communication path 9. The physical drivers 22 are controlled by the OS 20 to transfer data between the storage controller 3 and its local storage array.

*Id.*, 7:29-46; EX1003, ¶55.

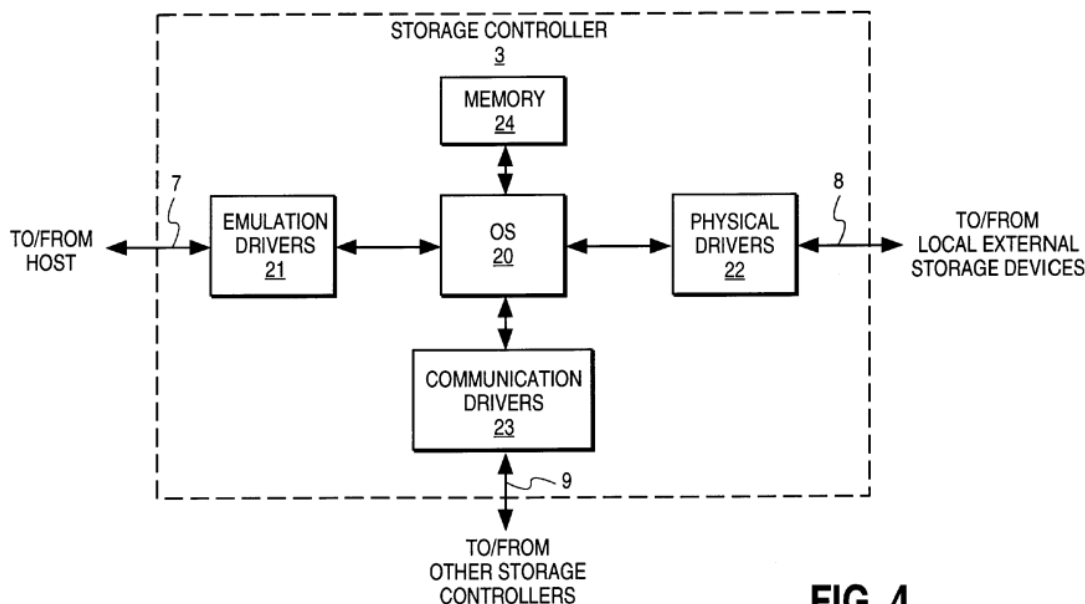


**FIG. 4**

**EX1004, Bergsten Fig. 4**

Finally, Bergsten discloses using cache storage:

At various times, the storage controller must read data from one or more MSDs into its internal memory (“the cache”) or transfer data from the cache to one or more MSDS, these procedures are known as “staging” and “destaging,” respectively. Staging is generally performed, for example, when executing a host read request. Destaging is generally performed when executing a host write request.

EX1004, 10:23-47; EX1003, ¶56. Bergsten depicts these read, write, staging and destaging routines performed by a storage controller at Figs. 14, 15, 10, and 12, respectively, *Id.*, 2:29-31, 2:34-36, 2:39-44.

**B. Claim 1**

[1.1]<sup>1</sup> A computer suitable for use in a data storage system comprising a network interconnecting a plurality of such computers, the computer comprising:

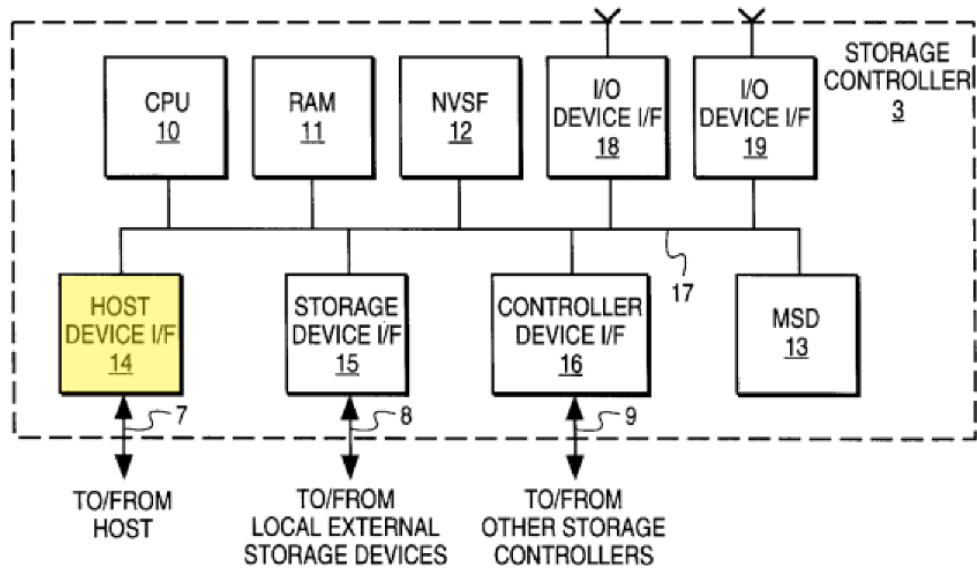
To the extent the preamble is limiting, Bergsten discloses this limitation by disclosing a network of interconnected storage controllers (plurality of computers), each coupled to a storage array. EX1003, ¶58; EX1004, Abstract; *see also* 3:20-39, Fig. 1.

[1.2] an I/O channel adapter for accepting an incoming I/O request from a host;

Bergsten's networked storage controllers receive I/O requests from the hosts. EX1003, ¶59. Specifically, Bergsten includes an I/O adapter accepting incoming I/O requests from a host in the form of host device interface 14 of a storage controller, which receives I/O requests from the host over communication path 7. EX1004, 6:41-55; *see also Id.*, 8:24-26 (discussing a storage controller receiving I/O read or write requests from a host). Bergsten expressly discloses that the host device interface can be a "well-known" standard-protocol SCSI adapter. *Id.*, 6:41-55; EX1003, ¶59. The host device interface connecting the storage controller "to/from host" via communication path 7 is shown below. EX1004, 2:13, Fig. 3.

---

<sup>1</sup> Reference numbers in the format of [claim#.limitation#] are added throughout for ease of reference.



**EX1004, Bergsten Fig. 3 (highlighting added)**

The claimed “I/O channel adapter” corresponds to the ’988 patent’s similar recitation of “I/O adapter driver software [that] has been modified to accept target SCSI I/O requests from host 111 and 112.” EX1001, 5:52-54; EX1003, ¶60. Thus, this limitation is broad enough to encompass Bergsten’s disclosure of using an SCSI adapter as the host device interface. EX1003, ¶60.

[1.3] configuration manager software for enabling said I/O channel adapter to decide whether (i) to route said request to cache, (ii) to route said request to disk, or (iii) to reject said request;

As discussed above, Bergsten discloses functional modules within the storage controllers, including controlling the flow and caching of data. EX1004, 7:29-35; EX1003, ¶61. Specifically, the storage controller of Bergsten includes configuration manager software as part of its operating system (OS 20), which “controls the overall operation of the storage controller, including emulation, *data flow, caching of data*

in the local memory 24, control of data mirroring, and error recovery.” EX1004, 7:29-32 (emphasis added). EX1003, ¶61. The functional modules of Bergsten, including OS 20, are depicted below.

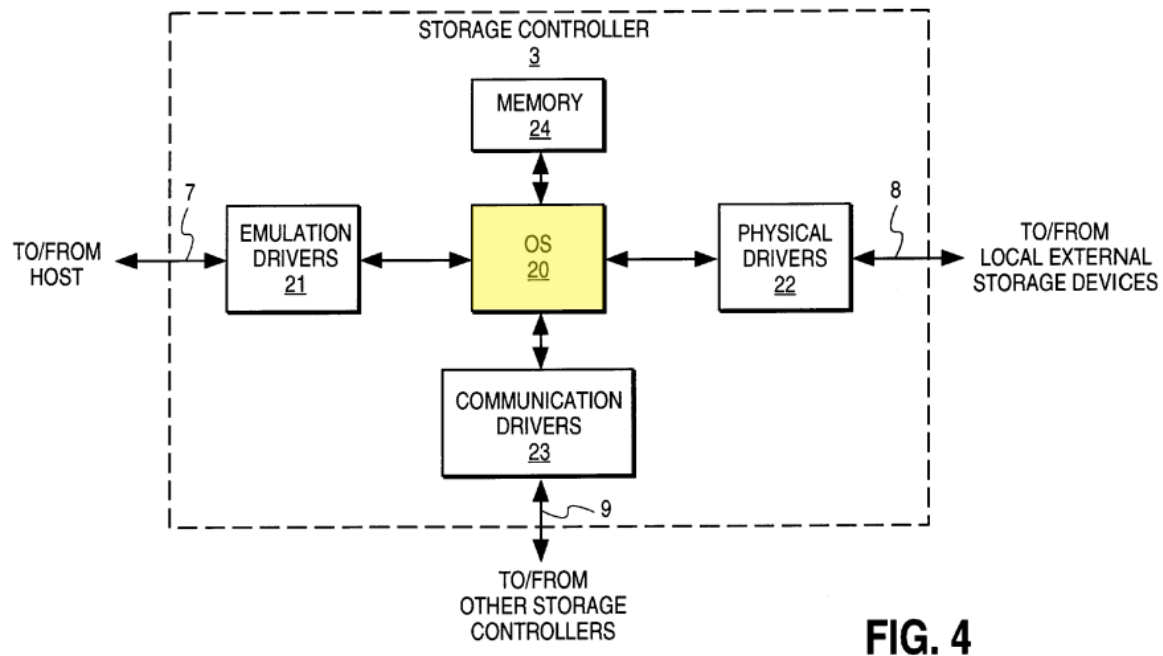


**FIG. 4**

**EX1004, Bergsten Fig. 4 (highlighting added)**

Bergsten discloses routines that satisfy each of the claimed alternative routing decisions (i.e., route request to cache, route request to disk, or reject request) and because the storage controller’s OS controls the operations of data flow and memory caching a POSITA would understand software components of the OS module to control how the controller’s host device interface routes requests received from a host. EX1003, ¶62.

With regard to routing a host request to cache, Bergsten discloses performing a standard read operation in response to a read request (EX1004, 12:37-38), including deciding to route the request to cache if the block is cached, and if so, transferring that cached data to the host. EX1004, 12:47-56, Fig. 14 at 1404-1406; EX1003, ¶62. Bergsten's standard read operation performed by the storage controller is depicted below. EX1004, 2:39-41, Fig. 10.

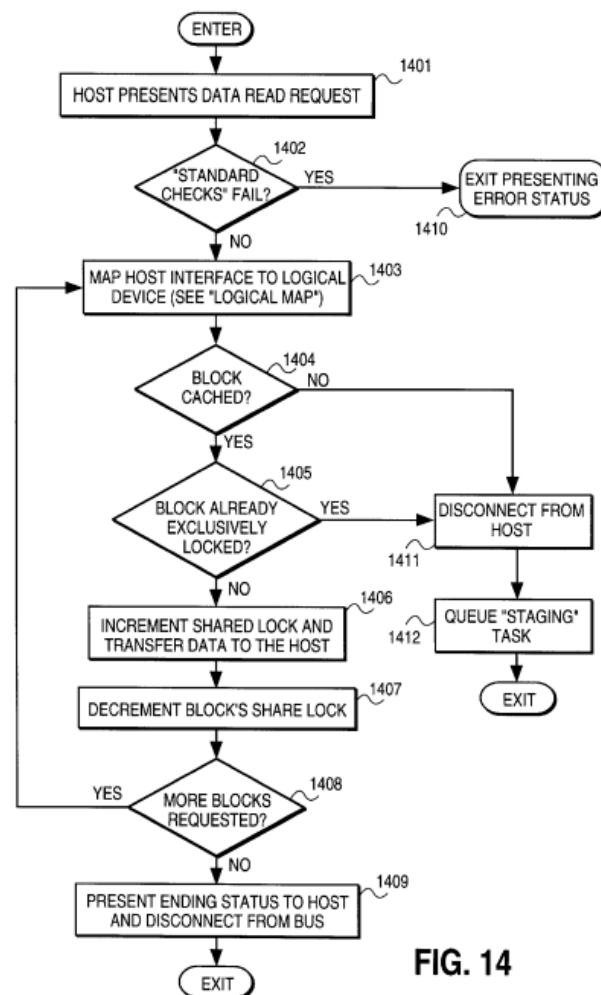


FIG. 14

EX1004, Bergsten Fig. 14

Bergsten further discloses routing the request to disk where, if a requested data block to be read is not cached, the storage controller initiates a staging routine. *Id.*, 12:47-53, Fig. 14 at 1404, 1411-1412; EX1003, ¶63. During the staging routine, the storage controller determines the nearest free storage drive housing the requested data – “nearest” refers to the mass storage device (local or remote) having the smallest access time or earliest availability– and routes the request to that drive to read the requested block and transfer the read data back to the host. EX1004, 10:30-45, 55-61; EX1003, ¶63. Figure 10, depicting the staging routine performed by the storage controller, is reproduced below. EX1004, 2:29-31, Fig. 10.

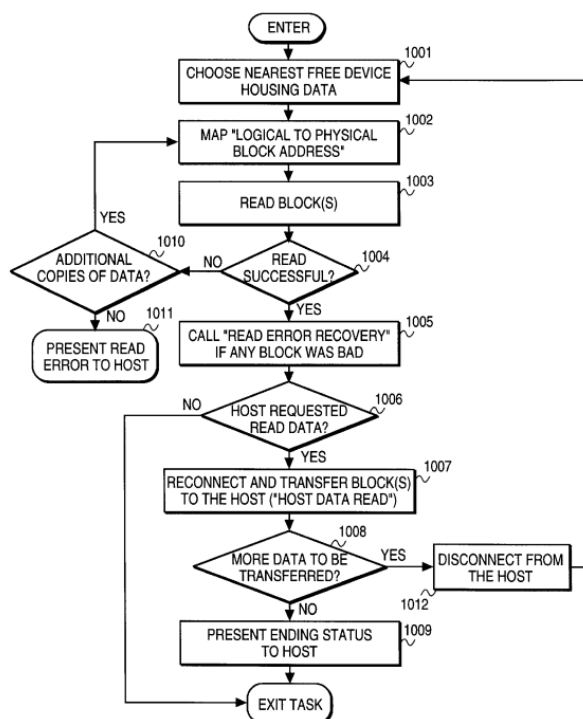


FIG. 10

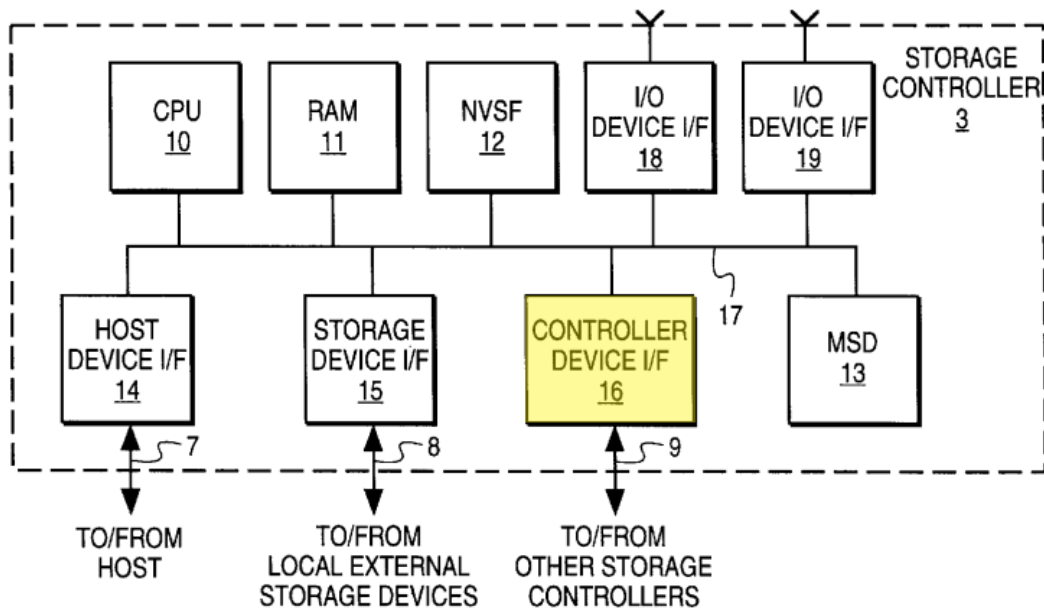
EX1004, Bergsten Fig. 10

Finally, with regard to determining whether to reject a request, “[i]n response to each I/O request (read or write) from a host, the storage controller performs a series of standard checks before carrying out the operation.” *Id.*, 8:24-26. For example, when performing a standard read routine, “in step 1402 the standard checks routine (see FIG. 6) is performed. If any of the standards checks fail, then the routine exits in step 1410, presenting an error status message to the host.” *Id.*, 12:40-43. Figure 6 of Bergsten shows a list of the standard checks performed that may result in an error message presented to the host, including checking if the device is off-line, ready, or busy, or if the requested data block exists on the device, is reserved, or exceeds the device capacity. *Id.*, 8:24-61, Fig. 6; EX1003, ¶64. A POSITA would understand that, because the storage controller exits its routine and sends an error message to the host in response to any of the checks failing, this results in the storage controller rejecting the request by not completing the request. EX1003, ¶64.

[1.4] a network adapter for handling network control traffic;

Bergsten discloses a network adapter for handling network control traffic in the form of its “controller device interface 16,” which “connects communication path 9 to the bus 17 in order to connect a remote storage controller to storage controller 3.” EX1004, 6:56-58. “Controller device interface 16 may be an Ethernet, ATM, T1, T3, or FDDI adapter, or any other suitable device, depending upon the nature of the communication link 9.” *Id.*, 6:56-61. A POSITA would understand this

to include the claimed network adapter for handling network control traffic because it enables communications between the networked storage controllers of Bergsten. EX1003, ¶65. The controller device interface connecting the storage controller “to/from other storage controllers” via communication path is shown below. EX1004, 2:13, Fig. 3.



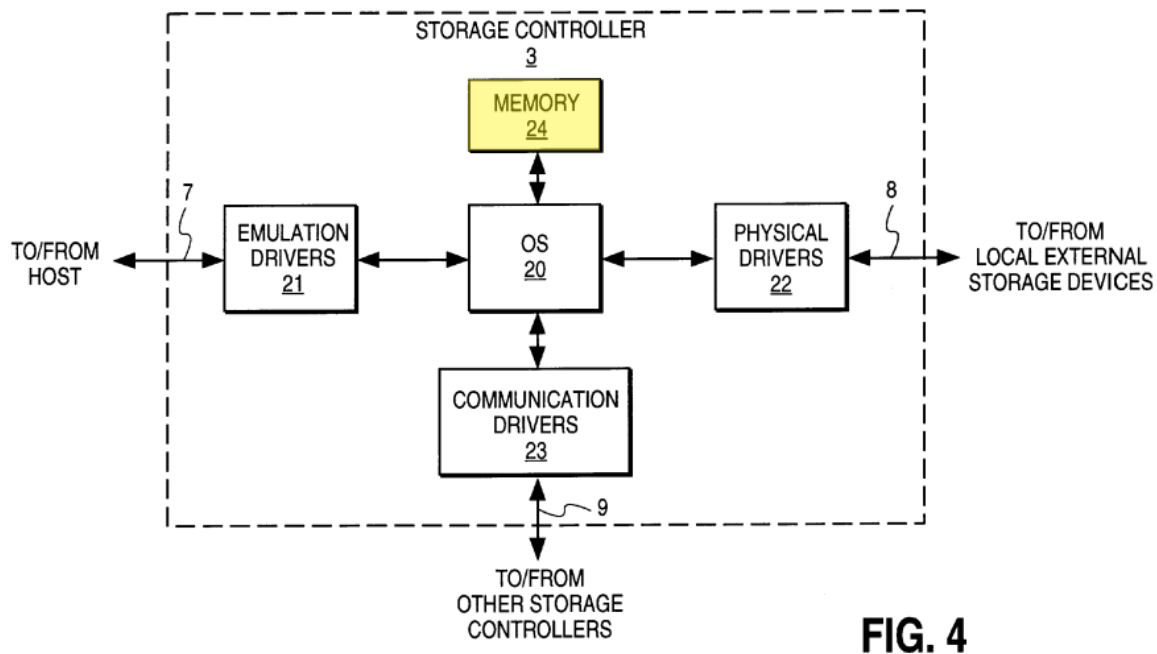
**EX1004, Bergsten Fig. 3 (highlighting added)**

This corresponds to the '988 patent's disclosure of using either an “Ethernet link” or fiber channel technology for network traffic. EX1001, 5:25-29; EX1003, ¶66.

[1.5] a cache memory;

Bergsten discloses a cache memory in the form of its local memory 24 (alternatively called cache 24 and internal memory 24). EX1003, ¶67. This local

memory “may include any combination of RAM 11, NVSF 12, or MSD 13 (see FIG. 3)” and is depicted as a functional block in Fig. 4, reproduced below. EX1004, 7:20-28, Fig. 4.



**EX1004, Bergsten Fig. 4 (highlighting added)**

In performing various operations in response to a host request, such as a standard read or write operation, the storage controller connected to the host checks whether the requested block is cached by determining whether the block is stored in the storage controller’s cache memory 24. *Id.*, 12:47-51, 13:20-23; EX1003, ¶68. In addition, “[a]t various times the storage controller must read data from one or more MSDs into its internal memory (‘the cache’) or transfer data from the cache to one or more MSDs,” i.e., staging and destaging. EX1004, 10:23-29.

[1.6] front-end software for handling I/O requests arriving at the I/O channel adapter or the network adapter;

A POSITA understood “front-end” to refer to the local, user-facing aspects of a networked computer system, e.g., the host processors of the ’988 patent or Bergsten. EX1003, ¶69. Accordingly, a POSITA would recognize a storage controller’s emulation drivers and communication drivers, in conjunction with OS, to constitute front-end software because, as explained below, they include software for handling requests from the system’s front-end components, namely a host directly connected to the storage controller or a host indirectly connected via a separate storage controller. *Id.*, ¶69.

For handling I/O requests at its I/O channel adapter (i.e., host device I/F 14), Bergsten includes front-end software in the form of emulation drivers 21, which “are controlled by the OS 20 to provide communication of data between the storage controller 3 and the local host computer 2 via communication path 7.” EX1004, 7:32-35; EX1003, ¶70. In particular, “[t]he emulation drivers 21 receive read and write requests from the host computer 2 and convert the read and write requests into a format recognized by the OS 20.” EX1004, 7:35-38.

For handling I/O requests arriving at its network adapter (i.e., controller device I/F), Bergsten’s “communication drivers 23 are controlled by the OS 20 to provide communication of data between the storage controller 3 and remote storage

controllers via communication path 9.” *Id.*, 7:40-43; EX1003, ¶70. Emulation drivers handling I/O requests “to/from host” and communication drivers handling I/O requests “to/from other storage controllers” are depicted as functional block in Fig. 4, reproduced below. EX1004, 7:20-28, Fig. 4.

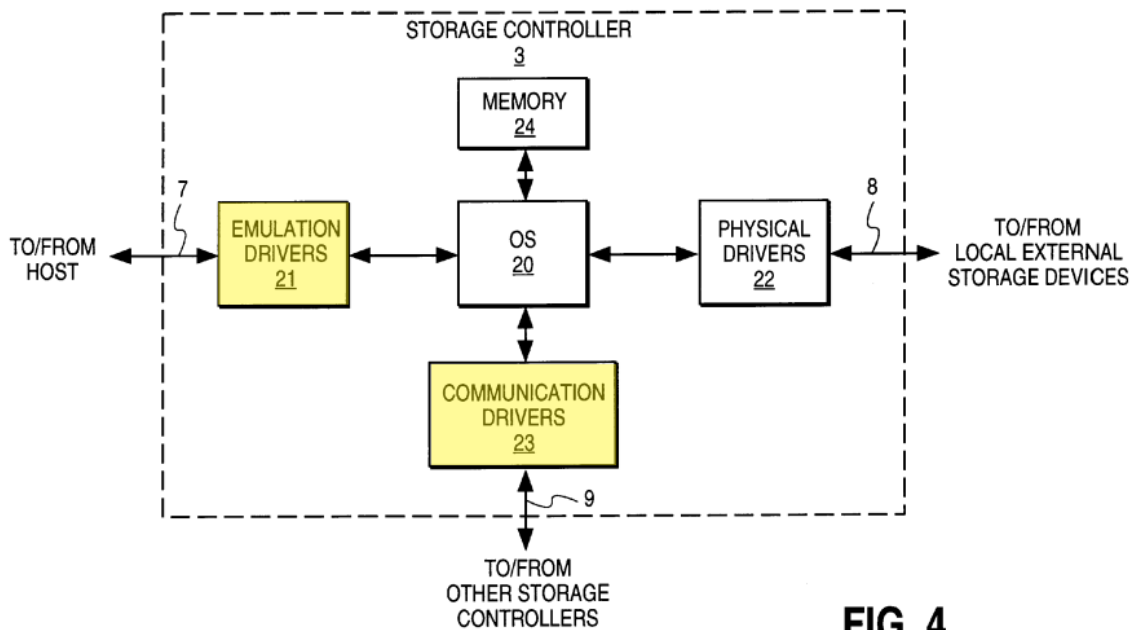


**FIG. 4**

**EX1004, Bergsten Fig. 4 (highlighting added)**

[1.7] cache manager software, responsive to said front-end software, for handling data stored in said cache memory; and

Bergsten discloses this limitation because its “OS 20 controls the overall operation of the storage controller, including emulation, data flow, *caching of data in the local memory 24*, control of data mirroring, and error recovery.” EX1004, 7:29-35 (emphasis added). A POSITA would understand that Bergsten’s OS of a

storage controller includes software for managing the storage controller's cache, i.e., "cache manager software." EX1003, ¶71.

One example of Bergsten's cache manager software responsive to front-end software is the routine performed by the storage controller in "performing a standard read operation in response to a request by a host." EX1004, 12:37-40; EX1003, ¶72. As part of this routine, the storage controller determines whether the requested block of data is stored in cache, and if not, initiates a staging task to retrieve the requested block from a mass storage drive. EX1004, 12:47-58. If the block is stored in cache, "the data is transferred to the host from the cache." *Id.*, 12:56.

Bergsten's standard read operation is analogous to read operations described in the '988 patent. For example, Fig. 4B shows a flowchart for a cache manager of a storage system processing a read request for a volume that it holds in an exclusive mode (EX1001, 6:65-66, Fig. 4B), and Fig. 4D shows a similar flowchart for a cache manager of a storage system processing a read request for a volume it holds in a shared mode (*id.*, 7:46-47, Fig. 4D). As with Bergsten, both of the '988 patent's read routines performed by the software of the cache manager handle data stored in the cache memory by first checking whether the requested read data is stored in the cache memory (steps 420, 450), and either initiating a disk read in response to a cache miss (steps 425, 455) or transmit cached data to the host in response to a cache hit (steps 424, 451). *Id.*, 6:65-7:8, 7:46-61, Figs. 4B, 4D; EX1003, ¶73.

Another example of Bergsten's cache manager is the routine performed by the storage controller in "performing a standard write operation in response to a request from a host." EX1004, 13:6-7. As part of this write routine the storage controller performs a cache check, determining "in step 1506 whether the block is already cached in the storage controller's internal memory 24." *Id.*, 13:20-23, Fig. 15; EX1003, ¶73. If the block is not in cache (i.e., a cache miss), "in step 1520 an empty block is allocated in the internal cache" (EX1004, 13:43-45, Fig. 15) and, subsequently at step 1508, the storage controller transfers write data from the host to the allocated block in the cache (EX1004, 13:51-53, Fig. 15). EX1003, ¶73. In case of a cache hit, the necessary block is already cached so no new cache block needs to be allocated, so the storage controller proceeds directly (after exclusively locking the block) to step 1508 to transfer data from the host to the appropriate block in cache. EX1004, 13:23-30, Fig. 15; EX1003, ¶73.

As with read operations, the '988 patent's disclosure of the handling of data stored in cache memory for write operations is analogous to Bergsten. For example, Fig. 4A shows a flowchart for a cache manager of a storage system processing a write request for a volume that it holds in an exclusive mode (EX1001, 6:43-44, Fig. 4A), and Fig. 4C shows a similar flowchart for a cache manager of a storage system processing a write request for a volume it holds in a shared mode (*id.*, 7:18-19, Fig. 4C). EX1003, ¶75. As with Bergsten, both of the '988 patent's write routines

performed by the software of the cache manager handle data stored in the cache memory by first checking whether the requested buffer is stored in the cache memory (steps 411, 430), allocating a new buffer in the cache in response to a cache miss (steps 412, 431), and finally transferring data from the host to the appropriate cache buffer for both a cache hit or miss (steps 413, 432). EX1001, 6:45-50, 7:19-25, Figs. 4A, 4C; EX1003, ¶75.

A POSITA would understand that because OS controls the operations of caching of data in the storage controller, the OS would contain software instructions for performing the described read and write routines responsive to the front-end software receiving a host read or write request. EX1003, ¶76.

[1.8] back-end software, responsive to said configuration manager software, for handling reads and writes to disks corresponding to the I/O requests but without communication over the I/O channel adapter, thereby separating disk operations from network and I/O traffic.

As explained above (*supra* Section IV.B), the Examiner emphasized this limitation in particular as not taught in the prior art of record. But such back-end software is disclosed by Bergsten, which was not cited during prosecution. Bergsten discloses back-end software in the form of “physical drivers 22 [that] are controlled by the OS 20 to transfer data between the storage controller 3 and its local storage array. Such transfer includes the transformation of commands and control data from the format used by the storage controller 3 to the format recognized by the local

storage array (e.g., SCSI).” EX1004, 7:44-49. As seen in Figure 4, reproduced below, the physical driver operations, which handle reads and writes to local storage disks, are separate from the communication drivers and emulation drivers, that handle the network and I/O traffic, respectively. Because the physical drivers communicating with local storage drives are separate from the emulation drivers communicating with a host computer, a POSITA would understand that the back-end software of Bergsten does not communicate over Bergsten’s I/O channel adapter (i.e., host device I/F 14) that receives I/O requests from the host. EX1003, ¶77.

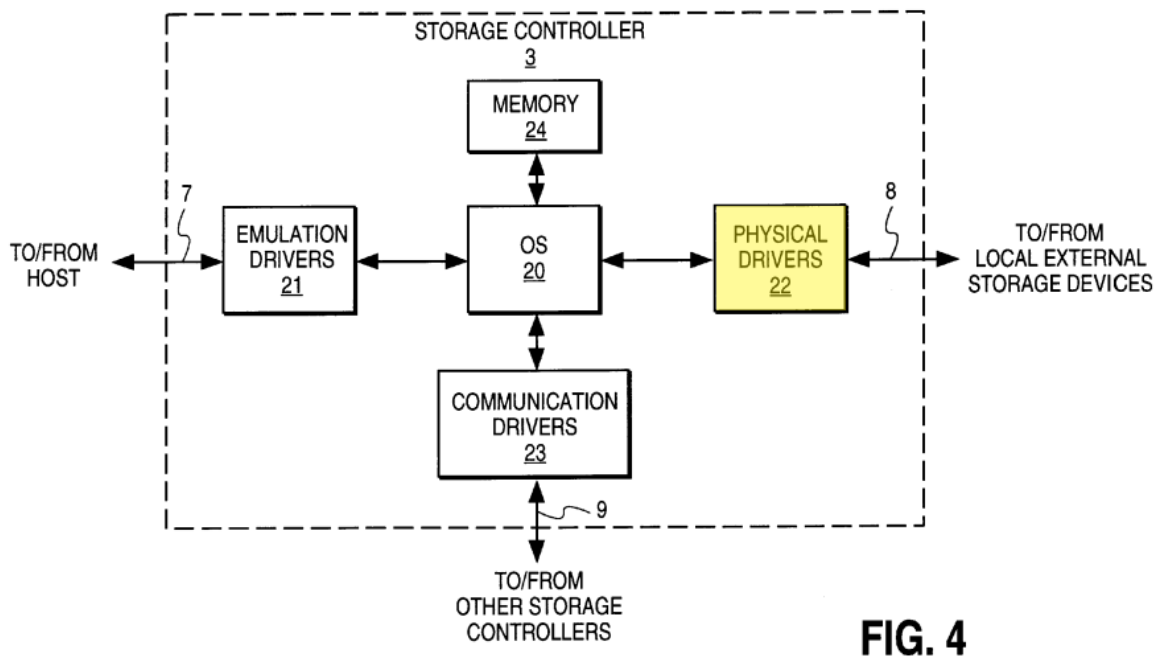


**FIG. 4**

**EX1004, Bergsten Fig. 4 (highlighting added)**

Thus, Bergsten discloses the claimed back-end software that handles the reads and writes to disks without communicating over the I/O channel adapter and a

POSITA would understand that the disk operations of Bergsten are separate from network and I/O traffic. EX1003, ¶78.

Claim 1 is therefore unpatentable as anticipated because Bergsten satisfies each of the limitations.

**C. Claim 3**

3. The system of claim 1 wherein the cache memory comprises a portion of a distributed cache memory stored in the computers interconnected by the network.

Bergsten discloses the added limitation of claim 3. A POSITA understood “distributed cache memory” to refer to cache memory allocated among multiple devices over a network. EX1003, ¶79. For example, the ’988 patent, describes multiple PC storage systems connected over a network with each system including its own cache memory. EX1001, 5:38-45, Fig. 3. Thus, each cache of each PC storage system constitutes a portion of a distributed cache memory across the network of storage systems. EX1003, ¶79. Bergsten teaches that each of its storage controllers includes a portion of distributed cache memory in the same way as described in the ’988 patent. As discussed above, Bergsten discloses multiple storage controllers interconnected by a network, with each storage controller including its own cache memory (local memory 24). EX1004, 3:20-39, 7:23-28, Fig. 1. Accordingly, each storage cache of each storage controller constitutes a portion of total cache memory distributed across the interconnected network of storage

controllers. Further, Bergsten recognizes the importance and “common practice” of storing back-up documents on separate back-up devices. Accordingly, a POSITA would understand that back-up copies of documents would be distributed across multiple storage caches belonging to multiple storage controllers in the system of Bergsten. EX1003, ¶79.

Thus, claim 3 is unpatentable as anticipated by Bergsten.

**VIII. GROUND 2: CLAIMS 2, 4, AND 6 ARE OBVIOUS OVER  
BERGSTEN IN VIEW OF THE KNOWLEDGE OF A POSITA**

As explained below, claims 2, 4, and 6 are obvious over Bergsten in view of the knowledge of one of ordinary skill in the art.

**A. Claim 2**

[2.1] The system of claim 1 wherein the computers comprise off-the-shelf hardware and operating systems and further comprise:

Bergsten discloses all limitations of claim 1 as discussed above for Ground 1.

A POSITA would understand that Bergsten can be implemented with off-the-shelf hardware and operating systems. EX1003, ¶82. Indeed, during prosecution the Applicant stated that “[i]t is well known to those skilled in the art that PCS refer to standard off-shelf computers that can be purchased at the retail store.” EX1002, 98. Similarly, Bergsten explains that its computer systems, storage controller components, and storage devices are all conventional systems and components implementing standard computing protocols. EX1004, 3:50-53 (explaining that host

computers may be any conventional computer system), 4:18-25 (discussing standard communications protocols for host-storage controller), 4:29-37 (“[D]ata paths 7 and 8 are conventional interfaces.”), 6:32-34 (“MSD 13 may be any conventional device that is suitable for non-volatile storage of large volumes of data, such as any of those discussed above”).

Moreover, a POSITA would be motivated to implement the system of Bergsten using off-the-shelf hardware and operating systems because such systems would be widely available to a POSITA and a POSITA would recognize an off-the-shelf solution to be more economical and convenient than building the system of Bergsten *ad hoc* from the ground up. EX1003, ¶82. Nothing in Bergsten would indicate to a POSITA that it could not be implemented using off-the-shelf components and operating systems. EX1003, ¶82.

[2.2] an adapter I/O software for accepting incoming I/O requests from a host;  
and

Bergsten discloses this limitation for the reasons discussed above for limitation [1.6]. Specifically, Bergsten discloses emulation drivers “controlled by the OS 20 to provide communication of data between the storage controller 3 and the local host computer 2 via communication path 7. In particular, the emulation drivers 21 receive read and write requests from the host computer 2 and convert the

read and write requests into a format recognized by the OS 20.” EX1004., 7:32-38; EX1003, ¶83.

[2.3] a volume access table employed by the configuration manager to ensure consistency of data stored on the network.

Dependent claim 2 introduces the “volume access table” feature. As discussed above, a POSITA would understand that a volume access table is a table used to store access characteristics of storage volumes. *See also* EX1003, ¶85. In the context of computer storage, a “volume” refers to the “disk or tape that stores computer data.” EX1012, 414. Though a disk may be partitioned “into several volumes, each of which is treated as a separate disk.” *Id.*

Bergsten recognizes the need to maintain consistency of multiple copies of data that may be located throughout a distributed storage system. EX1004, 5:61-62, 12:62-64, 14:42-43. A POSITA would be motivated to implement the storage controller of Bergsten, utilizing the conventional knowledge in the art, to include a volume access table to further ensure consistency of data stored on the network. EX1003, ¶86. Bergsten suggests such an approach in that it already discloses mapping volume access mode values between a host and volumes that the host has access to. Specifically, Bergsten teaches implementing “Exclusive Lock” and “Shared Lock” access modes to ensure data consistency as follows:

Associated with each block of data are the two single-bit binary quantities, Exclusive Lock and Shared Lock. Only a single device is permitted access to a given block of data for purposes of performing a write operation; this constraint is implemented using Exclusive Lock.... When a write operation on a block is initiated, Exclusive Lock is set for that block, so that no other device may access that block during the write operation. In contrast, multiple devices are permitted to read from a given block of data at the same time.... When a read is initiated on a given block, Shared Lock for that block is set, so that no device may write to that block during the read operation.

EX1004, 11:23-42. Thus, Bergsten discloses storing data structures in the form of “two single-bit binary quantities” (i.e., a binary numeral expressing a 0 or 1 (EX1012, 42, 44-45)) associated with each block of stored data, indicating whether that block is subject to either an Exclusive Lock or a Shared Lock. A POSITA understands a “block” in the context of disk storage to constitute a “collection of consecutive bytes of data that are read from or written to [a] disk as a group.” EX1012, 48; EX1003, ¶86.

A POSITA would recognize that Bergsten’s disclosed Exclusive Lock and Shared Lock access modes are implemented to ensure consistency of data across a network. EX1003, ¶87. Specifically, the use of the Exclusive Lock mode ensures that only a single device may write to a given block at any one time, while the use

of the Shared Lock allows for multiple devices to read a given block at the same time while also preventing devices from writing to that block while others are reading it. *Id.*

While Bergsten discloses that its access modes are binary quantities associated with each data block, Bergsten does not explicitly disclose how the binary access modes are structured and stored in the system of Bergsten. Thus, Bergsten leaves it to a POSITA to implement its access modes according to solutions known in the art. EX1003, ¶88. A POSITA would therefore find it obvious to implement the Exclusive/Shared Lock access mode values of Bergsten using a table mapping each data block with the access mode value associated with that block, because a table is a common, long used, and well-understood data structure for associating one element with another (e.g., for associating a data block and access mode). EX1003, ¶88.

Specifically, a table is a basic data structure consisting of rows and columns creating cells and is particularly useful for storing related data. EX 1003, ¶88; EX1012, 383. In computing, tables are one of the most fundamental data structures for storing data. For example, in programming an array is an implementation of table. *Id.*, 26, 383; EX1003, ¶88 (citing EX1013 and explaining that arrays are data structures for storing base data). As known to a POSITA, “[t]here is no best data structure for all tasks; suitable data structures must be selected for the specific task.”

EX1013, 293; *see also* EX1003, ¶88 (discussing EX1014, “Parsaye,” and the benefits of the table data structure.)

A POSITA would implement a table as a particularly suitable data structure for storing the association of data blocks to the Exclusive/Shared Locks of Bergsten. EX1003, ¶88. One reason for the suitability of a table would be its simplicity and ease of implementation, consisting only of specified rows and columns holding the associated data. *Id.* Another reason a POSITA would recognize such a table to be suitable is because it would allow for the fast and efficient search for relevant information via a table lookup routine. *Id.* Thus, such a table would allow a storage controller receiving a request for a specific data block to easily access the table for the Exclusive/Shared Lock value associated with that block. *Id.*

Providing the above-described association between stored data blocks and Exclusive/Shared Lock values constitutes a volume access table. *Id.*, ¶89. A POSITA would understand that the data blocks stored in a mass storage system are stored in volumes, i.e., the physical disks comprising the mass storage. *Id.* Therefore, by controlling how hosts can access specific stored blocks based on the Exclusive/Shared Lock value, the table holding such values also controls how the host can access the specific volume on which that block is stored. *Id.* For example, where a set Exclusive/Shared Lock permits a host to perform a read or write

operation for a specific data block, it likewise permits the host to perform the operation to the specific volume to which the data block is stored. *Id.*

However, to the extent the Board determines that a “volume access table” requires a mapping between hosts and volumes, a POSITA would also find it obvious to include in such a table what host or hosts hold a specific lock on a given block. *Id.*, ¶90. A POSITA would do so in order for the system to know which devices are currently writing to or reading from a data block and also to know which devices are currently preventing other devices in the network from writing to or reading to a block. *Id.* (explaining why information would be represented at volume level). Finally, a POSITA would implement the storage controllers of Bergsten to utilize such a table because the storage controller’s “OS 20 controls the overall operation of the storage controller, including emulation, data flow, caching of data in the local memory 24, control of data mirroring, and error recovery,” (ex. 1004, 7:29-35) and would therefore control access by a host to a given block based on the access value for that block. EX1003, ¶90.

Thus, claim 2 is unpatentable as obvious over Bergsten.

**B. Claim 4**

4. The system of claim 3 further comprising a volume access table employed by the configuration manager to ensure consistency of data stored in the distributed cache.

Bergsten discloses all limitations of claims 1 and 3 as discussed above for Ground 1.

Bergsten discloses the necessity of ensuring consistency of data stored in the distributed cache. EX1004, 5:61-62, 12:62-64, 14:42-43, Fig. 18. As explained above for claim 2, it would have been obvious to a POSITA to implement the storage controllers of Bergsten to include a volume access table including Exclusive/Shared Lock access values to ensure consistency. EX1003, ¶92. Further, Bergsten discloses that data blocks may be located at either mass storage drives or in a storage controllers cache storage. *Id.* A POSITA would recognize that Bergsten's use of exclusive and shared access modes would ensure data consistency by preventing writes to data blocks by a device while other devices are writing to or reading from that block and a POSITA would recognize that such access modes and resulting data consistency would be desirable regardless of whether the data block is stored in cache or in a mass storage drive. *Id.*

Thus, claim 4 is unpatentable as obvious over Bergsten.

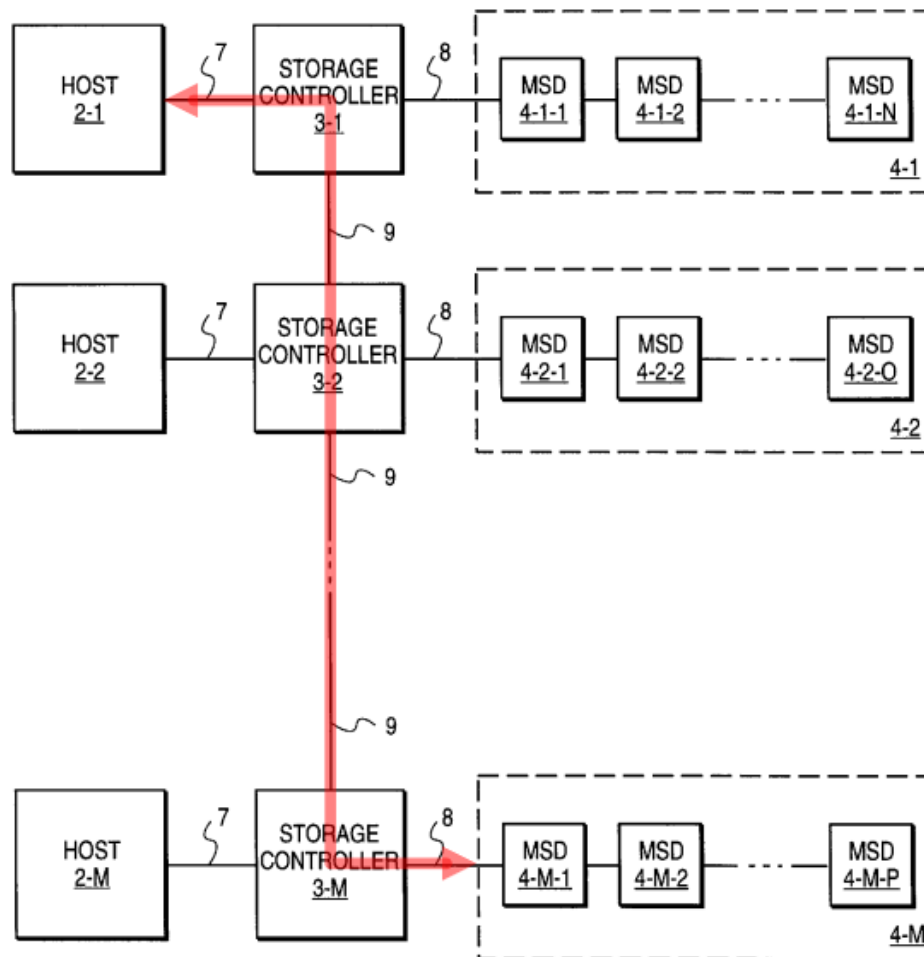
**C. Claim 6**

[6.1] A method of accessing a remote disk over a computer network without incurring network overhead, the method comprising the steps of:

To the extent the preamble is limiting, Bergsten discloses this limitation by disclosing a computer network of interconnected storage controllers (plurality of

computers), each coupled to a storage array, and each storage controller coupled to at least one host processing system and to at least one other storage controller to control access of the host processing systems to the storage device. EX1004, Abstract; see also 3:20-39, Fig. 1; EX1003, ¶93.

This computer network allows a local host computer to access any connected remote disk as “[t]he storage controllers in the computing system cooperate to allow any of the host computer systems to access data located in any of the storage arrays.” *Id.*, 4:39-41. With reference to Fig. 1, Bergsten provides the example of host computer system 2-1 accessing data located in remote storage array 4-M “by communicating with remote storage controller 3-M via storage controllers 3-1, 3-2, etc.” *Id.*, 4:31-47. Such communication between the local computer system and remote storage array via intermediary storage controllers is depicted in annotated Fig. 1, below. EX1003, ¶94.



**EX1004, Bergsten Fig. 1 (annotated)**

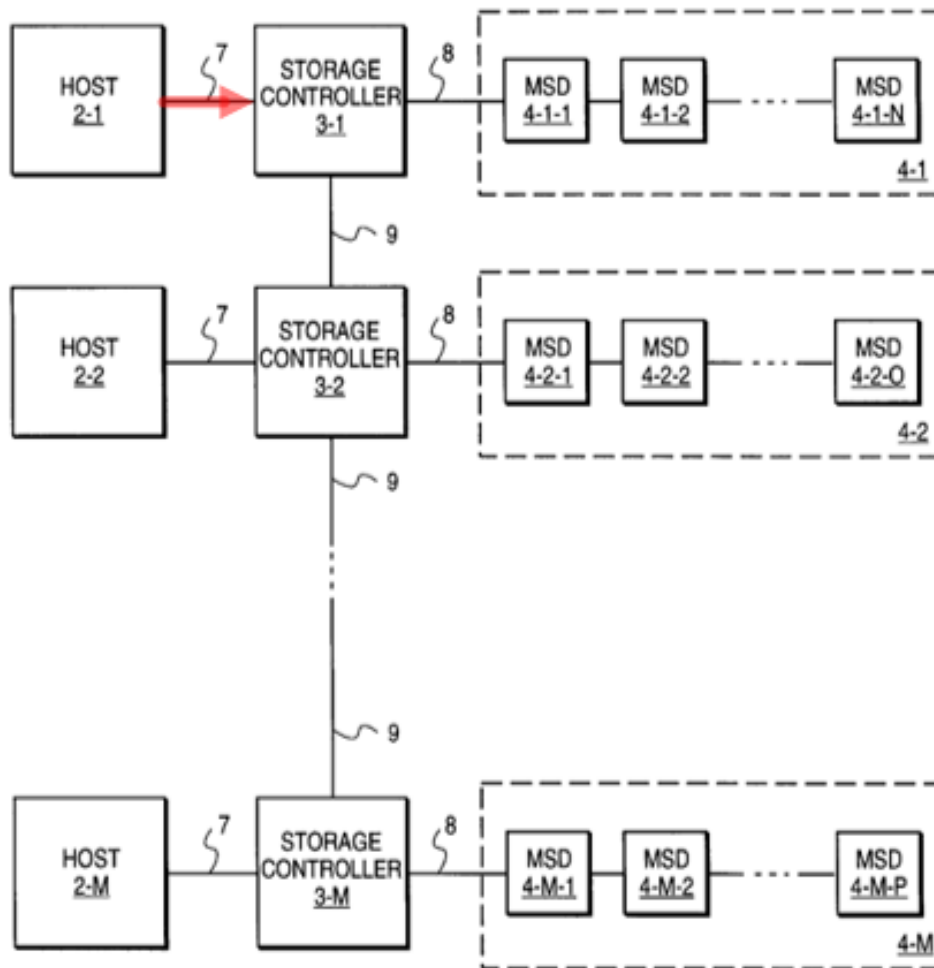
In addition, the local host computer of Bergsten accesses a remote disk without incurring network overhead in the same way as described in the '988 patent. EX1003, ¶95. The '988 patent describes a computer network that includes, *inter alia*, a data storage system 131 including host processors 111, 112 using channels 121, 122. EX1001, 4:66-5:2. Data storage system 131 connects to network attached data storage system 132 using network link 152, with storage system 132 servicing hosts 157, 158. *Id.*, 5:2-6. Thereby, “[h]osts 157 and 158 access a data storage system 131

indirectly via network attached data storage system 132, thereby off loading communications protocol overhead from remote hosts 157, 158.” *Id.*, 5:7-10. On the other hand, separate hosts 106, 107 are directly connected to storage system 131 via network link 151. *Id.*, 5:10-14, Fig. 1. This direct access to storage system 131 “incurs communications protocol overhead on hosts 106, 107 and therefore decreasing performance of applications running on said hosts.” *Id.*

Bergsten performs remote access without incurring network overhead at the host in the same as the ’988 patent. EX1003, ¶96. Specifically, as demonstrated in Bergsten’s annotated Figure 1 above, a local host processor indirectly connects to a remote disk storage array 4-M via a series of storage controllers 3-1 through 3-M. EX1004, 4:39-50. As with the ’988 patent, Bergsten does not incur network overhead for a host computer, because network communications protocols are handled by the intermediary storage controllers. Bergsten explains “[c]ommunication between individual storage controllers takes place on communication links 9 using a common protocol implemented by all storage controllers in the system.” *Id.*, 4:55-58. Such communications between a local storage controller and remote storage controllers are provided by communication driver and controlled by OS of the storage controller. Thus, processing of network communications in Bergsten are off-loaded to the storage controllers. EX1003, ¶96.

[6.2] causing a local host to issue a request over an I/O channel to a local computer;

Bergsten discloses a local host issuing a request over an I/O channel to a local computer. *Id.*, ¶97. As explained above at [1.2], Bergsten’s storage controller includes a host device interface to receive I/O channel requests from a local host to the storage controller. *Id.* The system of Bergsten causes the local host to issue I/O requests over the data communication path 7 to its local storage controller 3-1. EX1004, 8:24-26 (“In response to each I/O request (read or write) from a host, the storage controller performs a series of standard checks before carrying out the operation.”) This operation of a local host sending a read request to its local storage controller is demonstrated below. *Id.*, ¶97.



**EX1004, Bergsten Fig. 1 (annotated)**

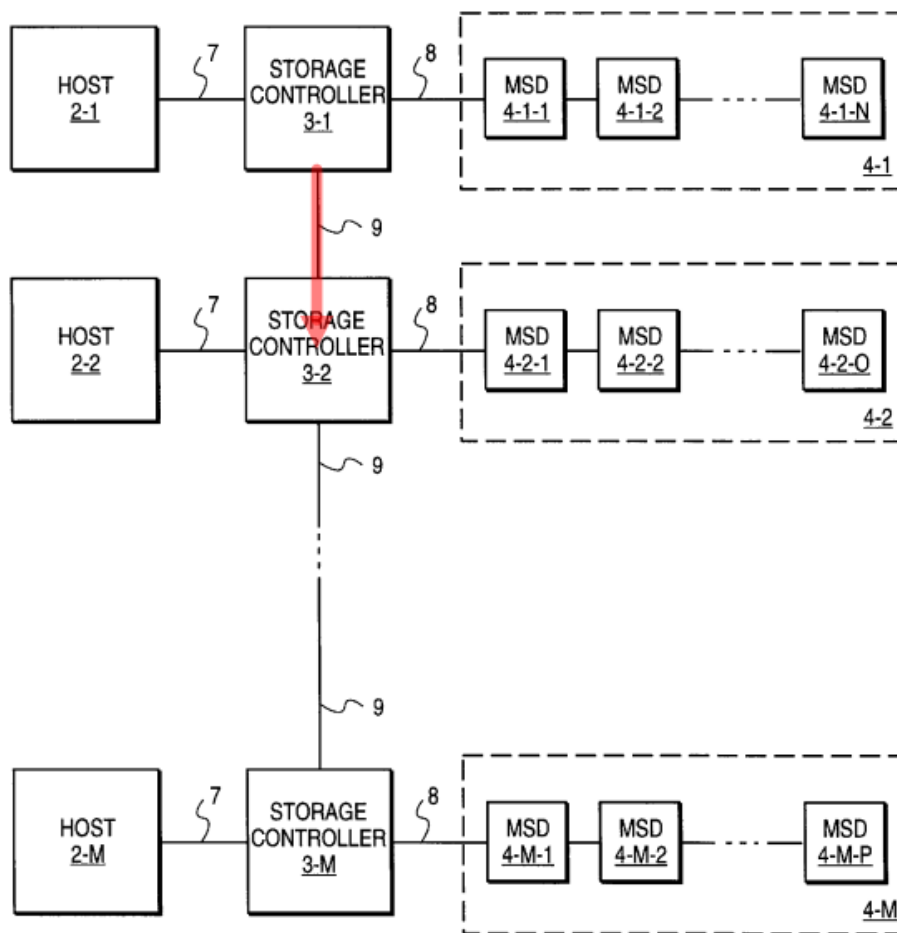
[6.3] providing a configuration manager on the local computer, the configuration manager routing the request to a remote computer via the computer network;

As discussed above at [6.2], Bergsten teaches a local computer in the form of storage controller 3-1. EX1003, ¶98. Bergsten discloses functional modules within the storage controllers, including controlling the flow and caching of data. EX1004, 7:29-35. Specifically, the storage controller of Bergsten includes a configuration manager as part of its operating system, which “controls the overall operation of the

storage controller, including emulation, *data flow*, caching of data in the local memory, control of data mirroring, and error recovery.” *Id.*, 7:29-32 (emphasis added); EX1003, ¶98. As part of controlling data flow, OS controls communication drivers “to provide communication of data between the storage controller 3 and remote storage controllers via communication path 9.” EX1004, 7:40-43; EX1003, ¶98.

As Bergsten explains, a local storage controller will read data from or write data to a remote mass storage device connected to a remote storage controller according to a staging/destaging routine. EX1004, 10:23-29; EX1003, ¶99. Bergsten’s Figure 10 demonstrates the staging procedure performed by a storage controller upon receiving a read request from its local host. EX1004, 2:29-31, 10:30-31, Fig. 10. At the beginning of the destaging process, at step 1001, the storage controller determines the “nearest” free mass storage device storing the data to be read, where “nearest” refers to the device with the smallest overall access time or earliest availability, which may be “an MSD within a remote storage array.” *Id.*, 10:32-41. Next, at steps 1002 and 1003, the logical block address of the requested data is mapped to the physical block address holding that data in the identified storage device and the data is read from that device. *Id.*, 10:41-43. A POSITA would recognize that, in the case of a local storage controller reading requested data from a remote storage drive, the local storage controller would route the read request for

the data through the remote storage controller associated with the storage drive, because, as shown in Fig. 1, the local storage controller (e.g., 3-1) connects with the remote storage array (e.g., 4-2) holding the remote drive indirectly via an intermediary remote storage controller (e.g., 3-2). *Id.*, Fig. 1; EX1003, ¶100. Thus, in Bergsten the configuration manager controlling the flow of data in a local storage controller routes a read request to a remote storage controller to read data from a remote storage device, as demonstrated in annotated Fig. 1, below. EX1003, ¶100.



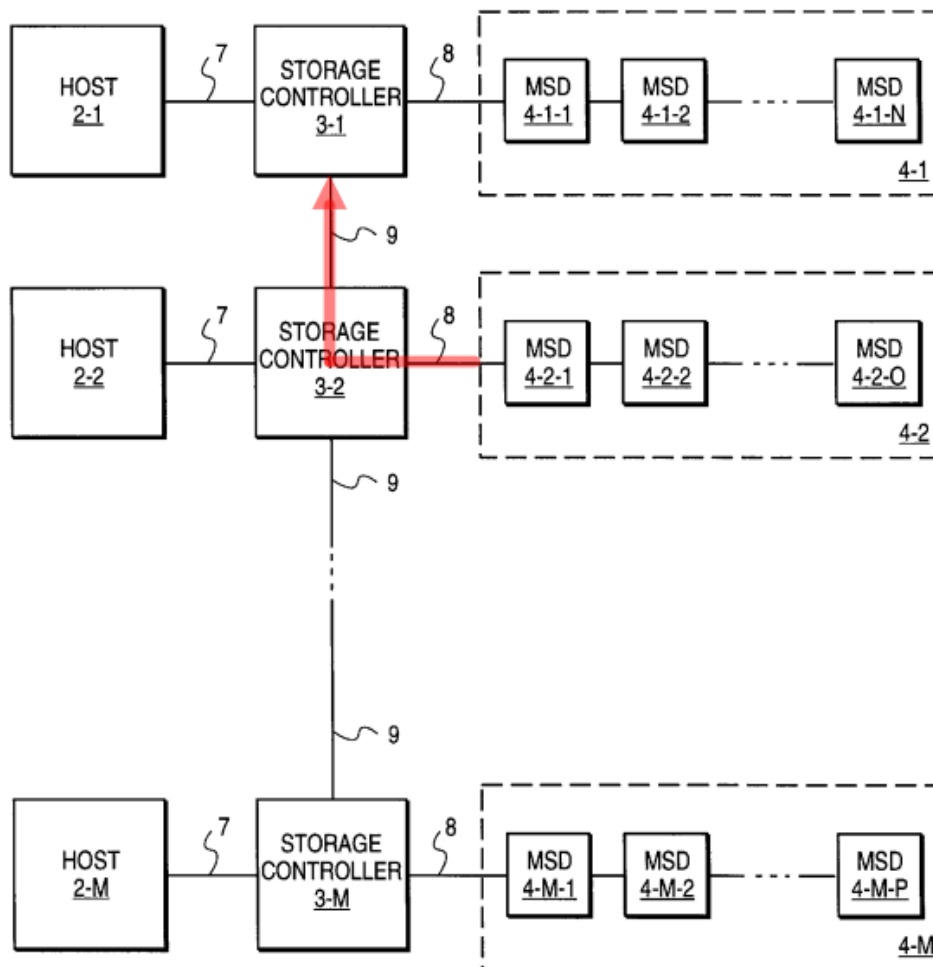
**EX1004, Bergsten Fig. 1 (annotated)**

[6.4] causing the remote computer to check the request against a volume access table;

For the reasons discussed above at for claims 2 and 4, a POSITA would have been motivated to implement Bergsten with a volume access table to ensure consistency of data stored on the network, including causing the remote storage controller to check the request against the volume access table. EX1003, ¶¶98-101. Specifically, in Bergsten, a read on a data block “is performed by obtaining a Shared Lock on that block” and may be obtained by any device “provided Exclusive Lock for that block is not already set.” EX1004, 11:35-40. Accordingly, a remote storage controller (e.g., storage controller 3-2) receiving a remote write request pursuant to Bergsten’s destaging procedure, as discussed above for claim limitation [6.3], would determine whether the requested data block in its connected storage array (e.g., storage array 4-2) has an Exclusive Lock. EX1003, ¶101. The storage array would do so by checking the corresponding Shared/Lock access value in the volume access table associated with the requested block. This procedure would ensure operation consistent with Bergsten’s prescription of allowing data reads only for blocks not exclusively locked by another device. EX1004, 11:35-40; EX1003, ¶101.

[6.5] causing the remote computer to perform an I/O operation on a disk located on the remote computer and to return data to the local computer;

Bergsten discloses this limitation because, as explained at [6.3], in order to read data from a remote storage drive, a local storage controller routes a read request to the intermediary remote storage controller associated with the remote storage drive. EX1003, ¶102. Bergsten explains, “[t]he storage controllers in the computing system cooperate to allow any of the host computer systems to access data located in any of the storage arrays. . . . For example, host computer system 2-1 may access data located in remote storage array 4-M by communicating with remote storage controller 3-M via storage controllers 3-1, 3-2, etc.” EX1004, 4:38-47. Storage controllers perform I/O operations with their associated MSDs over communication path 8 using a standard communications protocol such as SCSI. *Id.*, 4:15-28. Accordingly, a POSITA would understand that in Bergsten in response to a remote read request (e.g., from host 2-1 via local storage controller 3-1), the remote storage controller (e.g., storage controller 3-2) would perform an I/O operation over communication path 8 to read the requested data from its associated mass storage array holding the data (e.g., 4-2) and then transmit the read data back to the local storage controller over communication path 9 in order for the data to be routed to the local host requesting it. EX1003, ¶102. This operation of the remote storage controller retrieving read data from its mass storage array and transmitting that data to the storage controller requesting it is demonstrated in annotated Fig. 1, below. *Id.*



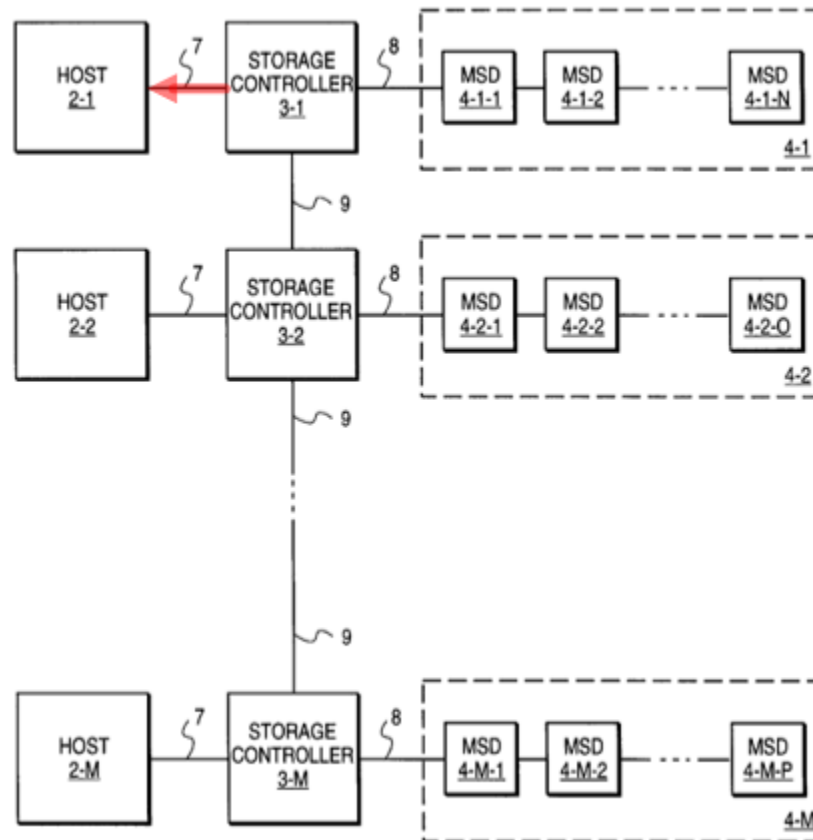
**EX1004, Bergsten Fig. 1 (annotated)**

[6.6] causing the local computer to provide the returned data to the local host via the I/O channel; and

Bergsten discloses this limitation because, after receiving the requested read data, at step 1007 of the destaging process “the storage controller reconnects to the host (i.e., acquires exclusive use of communication link 8, which may be a SCSI bus) and transfers the block or blocks to the host in step 1007.” EX1004, 10:55-61, Fig. 10; EX1003, ¶103. The step of the storage controller transferring the requested

data blocks to the requesting host is demonstrated in annotated Fig. 1, below.

EX1003, ¶103.



**EX1004, Bergsten Fig. 1 (annotated)**

[6.7] causing the local computer to check the data against the volume access table to ensure consistency of the data on the local and the remote computers.

For the same reasons as discussed above at ¶¶86-90, 98-100, a POSITA would have been further motivated to implement Bergsten to cause the local computer (i.e., the local storage controller) to check the data against the volume access table to ensure consistency with the remote computers (remote storage controllers). EX1003,

¶104. Specifically, in Bergsten, a write operation is performed by obtaining an Exclusive Lock on the block of data to be accessed. EX1004, 11:25-2. A POSITA would therefore recognize the need for a local storage controller, when receiving retrieved data, to check the volume access table for the data block associated with that data to determine that no host computer has obtained an Exclusive Lock on the associated data block because such an Exclusive Lock would indicate a potential write to the block and an editing of the data in that block. EX1003, ¶104. A check of the lock value in the volume access table for the data block associated with the retrieved data constitutes a check of that data against the volume access table, the presence of an Exclusive Lock would indicate a potential change to the block from which the data was retrieved. *Id.* In such a case a POSITA would recognize a potential inconsistency between the data received at the local storage controller and the newly written data for that block and would prevent the data to be transferred to the requesting host in order to ensure data consistency. *Id.* Conversely, where the local storage controller determines from its volume access table that no new Exclusive Lock has been attained the storage controller would be assured of consistency of the data across computers and transfer the received data to the requesting host. *Id.*

**IX. GROUND 3: CLAIMS 2, 4, AND 6 ARE  
OBVIOUS OVER BERGSTEN IN VIEW OF BLUMENAU**

As explained below, claims 2 and 4 are obvious over Bergsten in view of U.S. Patent No. 6,421,711 (“Blumenau”). EX1003, ¶105.

**A. Blumenau**

Blumenau, titled “Virtual Ports For Data Transferring Of A Data Storage System,” issued from U.S. Patent Application No. 09/106,299, filed on June 29, 1998. EX1007, 1. Blumenau’s original application, filed more than six months before the earliest possible effective filing date of the ’988 patent, discloses the subject matter of the issued Blumenau patent relied on herein. *Id.* Thus, Blumenau qualifies as prior art under pre-AIA Section 102(e).

Blumenau is directed to a data storage system, including a storage controller that provides virtual ports for storage access and a virtual switch for routing storage access requests from the physical port to the virtual ports. EX1006, Abstract.

Like Bergsten, Blumenau discloses multiple host computers accessing multiple storage systems. EX1003, ¶108. More specifically, Blumenau discloses a cached storage subsystem, including storage volumes and a storage controller for controlling access of the hosts to the storage volumes. “The storage volumes are logical units of storage distributed over one more storage devices.” *Id.*, 7:2-7:3. Blumenau further discloses “stor[ing] information defining a correspondence

between hosts in the data processing system and the set of volumes accessible to each host through the port adapter. For example, a volume access table and volume lists are stored in the memory. EX1003, ¶108. The volume access table specifies a correspondence between hosts and respective lists of volumes accessible to the hosts.” EX1007, 12:66-13:6; *see also e.g.*, Figs. 8-10, shown below.

Fig. 8

| VOLUME GROUP NAME | HOST CONTROLLER WWN | PRIVATE /SHARED FLAG | VOLUME BITMAP | VOLUME LIST | PORT ADAPTER INDEX LIST |
|-------------------|---------------------|----------------------|---------------|-------------|-------------------------|
| —                 | —                   | —                    | —             | —           | —                       |
| —                 | —                   | —                    | —             | —           | —                       |
| ⋮                 | ⋮                   | ⋮                    | ⋮             | ⋮           | ⋮                       |

88

Fig. 9

| HOST CONTROLLER S_ID | PRIVATE /SHARED FLAG | VOLUME BITMAP | PERSISTENT TABLE INDEX |
|----------------------|----------------------|---------------|------------------------|
| —                    | —                    | —             | —                      |
| —                    | —                    | —             | —                      |
| ⋮                    | ⋮                    | ⋮             | ⋮                      |

80

Fig. 10

| SID MODULUS | LISTS OF S_IDs AND TRANSIENT TABLE INDICES |
|-------------|--------------------------------------------|
| 0           | (S_ID, INDEX) , (S_ID, INDEX ) , ...       |
| 1           | (S_ID, INDEX) , (S_ID, INDEX ) , ...       |
| ⋮           | ⋮                                          |
| ⋮           | ⋮                                          |

97

EX1006, Blumenau Figs. 8, 9, 10

Referring to Figure 8 of Blumenau, the volume access table includes columns for volume group name, a host controller WWN, and a private/shared flag. *Id.*, 15:34-42, Fig. 8. The volume group name “provides a stable and unique identification number for a group of logical storage volumes to be accessed from a

host port” (*id.*, 13:38-40), with the specific volume group names and the volumes in each volume list defined by a system administrator (*id.*, 13:58-59). Each host computer of Blumenau includes a host controller (*id.*, Fig. 4) and each such controller includes a World Wide Name (“WWN”) (*id.*, 9:58-59), which is listed in the volume access table. Thus, Blumenau’s volume access table provides a mapping of each host and the data volumes to which the host as access. EX1003, ¶109. With regard to the private/shared flag of the volume access table, Blumenau explains that in data storage, “[r]estricted access is desirable for security of private data. For example, private volumes should be assigned to each host for storage of private data, and other hosts should not be permitted to see or modify the private volumes of other hosts.” EX1007, 1:48:-51. Therefore, when the table’s private/shared flag is set, this “indicates that all of the logical volumes in the volume list are private and no permission is needed from a lock manager before the host controller port can access[] any logical volume in its assigned volume list.” *Id.*, 13:53-57.

**B. Rationale For Combining Bergsten And Blumenau**

For at least the following reasons, Bergsten and Blumenau would have been a natural combination to a POSITA in January, 1999. EX1003, ¶110.

First, Bergsten and Blumenau relate to the same technology, each disclosing distributed storage systems in which multiple host computers make requests to the storage system. *Id.*, ¶111. Bergsten discloses, among other things, an architecture

similar to the architecture of the '988 patent, consisting of multiple network-connected host processors, storage control systems (i.e., storage controllers), and mass storage disks (see *e.g.*, EX1001, Fig. 1; EX1004, Fig. 1). Blumenau similarly discloses hosts connected over a network to a storage controller and remote storage volumes (EX1006, Fig. 1) and also recognizes multi-host, multi-storage environments (*id.*, 41:34-39). Thus, the teachings in these references relate to similar technology. EX1003, ¶111.

Second, a POSITA reading Bergsten would understand the need to improve security. EX1003, ¶112. Bergsten recognizes that “[a] storage controller of the present invention further allows data blocks to be write protected, so that a block cannot be modified from any host computer. Write protection may be desirable for purposes such as virus protection or implementation of security firewalls.” EX1004, 15:40-46. Blumenau provides techniques related to improving security in these types of storage systems. Blumenau recognizes that “[i]t may be possible for any host to have access to any volume in a data storage subsystem to which the host has access. However, it may be desirable to restrict the set of volumes that can be seen by any one host. Restricted access is desirable for security of private data. For example, private volumes should be assigned to each host for storage of private data, and other hosts should not be permitted to see or modify the private volumes of other hosts.” EX1006, 1:43-51.

Third, as noted above, each of Bergsten and Blumenau track information regarding host and how that information can be mapped to corresponding logical devices. Thus, a POSITA implementing Bergsten would be motivated to look to similar references such as Blumenau for ways to improve performance, including incorporating teachings related to security that Bergsten recognizes is needed. EX1003, ¶113. Blumenau describes the use of a table to manage access modes such as private access (EX1006, 15:34-42), an efficient data structure (as described earlier), consistent with Bergsten's goals of improving system performance (e.g., EX1004, 6:15-17).

Fourth, combining the teachings of Blumenau's volume access table with Bergsten would be one way in which a POSITA would improve security in Bergsten's storage system. Specifically, as taught by Blumenau, using a volume access table with specified access modes would restrict access to data, which, as stated by Blumenau, "is desirable for security of private data." EX1003, ¶114.

Finally, the techniques taught by Bergsten were entirely compatible with the volume access table taught by Blumenau, and a POSITA would have had a reasonable expectation of success to combine these teachings. EX1003, ¶115. Specifically, a POSITA would have combined the distributed storage network architecture as depicted in Fig. 1 of Bergsten with the volume access table software solution taught by Blumenau. *Id.* A POSITA would recognize that Bergsten's

storage controller already include memory 24 (EX1004, 7:23-28) capable of storing the volume access table as well as an operating system (OS 20) that controls the operation of the storage controller (*id.*, 7:29-32) and that could easily be programmed by a POSITA to access the volume access table and control data flow based on data stored in the table. EX1003, ¶115. It was well within the skill of a POSITA to program the storage controller of Bergsten to implement the volume access table of Blumenau. *Id.*

**C. Claim 2**

[2.1] The system of claim 1 wherein the computers comprise off-the-shelf hardware and operating systems and further comprise:

Bergsten discloses this limitation for the same reasons discussed above in Ground 3 for limitation [2.1]. *See also* EX1003, ¶¶116-117.

[2.2] an adapter I/O software for accepting incoming I/O requests from a host; and

Bergsten discloses this limitation for the reasons discussed above in Ground 1 at [1.6] and Ground 2 at [2.2]. EX1003, ¶118.

[2.3] a volume access table employed by the configuration manager to ensure consistency of data stored on the network.

Notwithstanding Petitioner's proposed interpretation discussed above regarding volume access table, it should be noted that the applicant added a

definition for “volume access table” during prosecution of the patent that was not included in the original specification as-filed:

Volume access table (450) in FIG. 4 contains a mapping between hosts and volumes specifying an access mode value. If the access mode is set to neither shared nor exclusive configuration manager forwards I/O requests directly to disk. In addition to the access mode said volume access table may contain other values that help to manager and improve performance of said data storage system.

EX1002, 160. Thus, to the extent the phrase is interpreted more narrowly by the Board to mean a mapping between hosts and volumes specifying an access mode value, a POSITA would further understand that such a volume access table was known in the prior art. EX1003, ¶¶120-122.

As discussed above, Blumenau expressly discloses a volume access table that includes access values for host computers and their corresponding volumes. EX1006, 13:4-13:6 (“The volume access table specifies a correspondence between hosts and respective lists of volumes accessible to the hosts.”); *see, also*, Figs. 5, 8-10.

82

| VOLUME GROUP NAME | HOST CONTROLLER WWN | HOST CONTROLLER S_ID | PRIVATE /SHARED FLAG | VOLUME LIST POINTER |
|-------------------|---------------------|----------------------|----------------------|---------------------|
| HOST 22-1         | 0276ADF1            | 0000B8               | 1                    | 0472                |
| HOST 23-2         | 0276ADC2            | 0000B9               | 0                    | 0476                |
| HOST 24-2         | 0276AE32            | 0001A7               | 0                    | 048E                |
| HOST 25-1         | 0276AE51            | 0001A6               | 1                    | 04AO                |
| ⋮                 | ⋮                   | ⋮                    | ⋮                    | ⋮                   |

*Fig. 5*

**EX1006, Blumenau Fig. 5**

88

*Fig. 8*

| VOLUME GROUP NAME | HOST CONTROLLER WWN | PRIVATE /SHARED FLAG | VOLUME BITMAP | VOLUME LIST | PORT ADAPTER INDEX LIST |
|-------------------|---------------------|----------------------|---------------|-------------|-------------------------|
| —                 | —                   | —                    | —             | —           | —                       |
| —                 | —                   | —                    | —             | —           | —                       |
| ⋮                 | ⋮                   | ⋮                    | ⋮             | ⋮           | ⋮                       |

*Fig. 9*

80

| HOST CONTROLLER S_ID | PRIVATE /SHARED FLAG | VOLUME BITMAP | PERSISTENT TABLE INDEX |
|----------------------|----------------------|---------------|------------------------|
| —                    | —                    | —             | —                      |
| —                    | —                    | —             | —                      |
| ⋮                    | ⋮                    | ⋮             | ⋮                      |

*Fig. 10*

97

| SID MODULUS | LISTS OF S_IDs AND TRANSIENT TABLE INDICES |
|-------------|--------------------------------------------|
| 0           | (S_ID, INDEX), (S_ID, INDEX), ...          |
| 1           | (S_ID, INDEX), (S_ID, INDEX), ...          |
| ⋮           | ⋮                                          |

**EX1006, Blumenau Figs. 8, 9, 10**

Moreover, Blumenau's volume access table includes a private/shared flag for an identified mapping of data volumes and host, and a POSITA would understand

that such a flag is used to ensure data consistency of data stored in a network. EX1003, ¶122. Blumenau states that “[c]ertain ‘private’ volumes should be assigned to each host and other hosts should not be permitted to see or modify the private volumes of other hosts.” EX1006, 12:2-5. Therefore, setting a private/shared flag to indicate a private volume ensures the consistency of that data across the network by only allowing it to be modified by a single host. EX1003, ¶122.

For the reasons discussed above, a POSITA would have been motivated to incorporate Blumenau’s teachings of a volume access table into Bergsten to combine Bergsten’s disclosed network architecture including a configuration manager with Bergsten’s access information in a table structure such as Blumenau’s Fig. 8. *Id.*, ¶123. Thus, claim 2 is unpatentable as obvious over Bergsten in view of Blumenau.

**D. Claim 4**

4. The system of claim 3 further comprising a volume access table employed by the configuration manager to ensure consistency of data stored in the distributed cache.

Bergsten discloses the limitations of claim 3 as explained for Ground 1 above. Bergsten combined with Blumenau discloses the added limitation of claim 4.

Bergsten discloses the necessity of ensuring consistency of data stored in the distributed cache. EX1004, 5:61-62, 12:62-64, 14:42-43, Fig. 18. As explained above, it would have been obvious to a POSITA to combine the distributed storage system including a configuration manager as taught by Bergsten with the volume

access table of Blumenau. EX1003, ¶125. Further, Bergsten discloses that data blocks may be located at either mass storage drives or in a storage controllers cache storage. *Id.* A POSITA would recognize that the use of a private access mode as disclosed in Blumenau would ensure data consistency by ensuring only a single device can modify a data block flagged as private and a POSITA would recognize that such an access mode and resulting data consistency would be desirable regardless of whether the data block is stored in cache or in a mass storage drive. *Id.*

Thus, claim 4 is unpatentable as obvious over Bergsten in view of Blumenau.

**E. Claim 6**

[6.1] A method of accessing a remote disk over a computer network without incurring network overhead, the method comprising the steps of:

Bergsten discloses this limitation for the reasons discussed above in Grounds 2 for limitation [6.1]. EX1003, ¶¶126-129.

[6.2] causing a local host to issue a request over an I/O channel to a local computer;

Bergsten discloses this limitation for the reasons discussed above in Grounds 2 for limitation [6.2]. *Id.*, ¶130.

[6.3] providing a configuration manager on the local computer, the configuration manager routing the request to a remote computer via the computer network;

Bergsten discloses this limitation for the reasons discussed above in Grounds 2 for limitation [6.3]. *Id.*, ¶¶131-133.

[6.4] causing the remote computer to check the request against a volume access table;

For the reasons discussed above, a POSITA would have been motivated to implement Bergsten with a volume access table as taught by Blumenau to ensure consistency of data stored on the network, including causing the remote storage controller to check the request against the volume access table. *Id.*, ¶134. A POSITA would utilize Bergsten's storage controller, which controls system data flows (EX1004, 7:29-32), to check the stored private/shared flag (in the volume access table as taught by Blumenau) for a volume sought to be accessed by a host read/write request to determine whether the host is permitted access to that volume and thereby ensure consistency of data across the network. EX1003, ¶134. It would have been obvious for a POSITA to implement such a check of the volume access table to be performed at the remote storage controller in response to a read request for data stored on the storage array directly connected to that remote storage device. *Id.* A POSITA would recognize a defined, limited number of options for implementation (e.g., the private/shared flag could be checked in response to the request at either the local storage controller 3-1 or at the remote storage controller 3-2) and would therefore find it an obvious implementation to perform the check at the remote storage controller (i.e, the remote computer). *Id.*

[6.5] causing the remote computer to perform an I/O operation on a disk located on the remote computer and to return data to the local computer;

Bergsten discloses this limitation for the reasons discussed above in Grounds 2 for limitation [6.5]. *Id.*, ¶135.

[6.6] causing the local computer to provide the returned data to the local host via the I/O channel; and

Bergsten discloses this limitation for the reasons discussed above in Grounds 2 for limitation [6.6]. *Id.*, ¶136.

[6.7] causing the local computer to check the data against the volume access table to ensure consistency of the data on the local and the remote computers.

A POSITA would have been further motivated to implement Bergsten to cause the local computer (i.e., the local storage controller) to check the data against the private/shared flag in Blumenau's volume access table. *Id.*, ¶137. As explained above, determining whether a host has private or shared access to a data volume ensures data consistency across the network, including between local and remote computers (storage controllers). Specifically, a POSITA would implement the local storage controller (e.g., storage controller 3-1) to check the private/shared flag for the volume associated with the retrieved data prior to transferring the data to the requesting host processor (e.g., host 2-1). *Id.* By doing so, the system would ensure that the private/shared flag has not changed for the requesting host prior to transmitting the data. *Id.* For example, if the flag has changed (e.g., via a system

administrator) to indicate that the volume associated with the requested data has been set to be privately held by a different host than the requesting host, then the requesting host would no longer be permitted access to that data and the transmission of the data should not be completed. *Id.* A POSITA would further understand that performing such a check of the volume access table at the local storage controller (before transferring data to the requesting host) and the remote storage controller (in response to a request, as explained for claim element 6.4) would ensure system redundancy and further promote system security. *Id.*

**X.     **GROUNDS 4 & 5: CLAIM 5 IS OBVIOUS WHEN  
SMITH IS COMBINED WITH BERGSTEN (GROUND 4) AND  
COMBINED WITH BERGSTEN AND BLUMENAU (GROUND 5)****

As discussed below, claim 5 is obvious over Bergsten in view of “Cache Memories,” Alan Jay Smith, Computing Surveys, Vol. 14, No. 3, Sept. 1982 (EX1015, “Smith”). Smith was a printed publication as late as the end of 1982 (qualifying as prior art under pre-AIA Section 102(b)), as indicated by its printing in an ACM periodical, dates printed on its face, copyright registration, and recognition by the Applicant in an information disclosure statement. EX1015; EX1002, 66; EX1018; EX1003, ¶¶139-141.

To the extent the Board determines that Bergsten alone does not render obvious a “volume access table,” Claim 5 is further obvious over Bergsten and Blumenau in view of Smith.

**A. Smith**

Smith, cited on the face of the '988 patent, provides a summary of various aspects of cache memory design and operation. Smith discusses using cache memory in “modern computer systems” to temporarily hold contents of main memory that are currently in use. EX1015. According to Smith, the advantage of cache memory is that less time is used to access information stored in cache memory than is stored in main memory. *Id.* Smith describes various common design techniques used in cache memory systems, including broadcasting invalidate messages to maintain cache consistency (*id.*, 505) and bypassing the cache for operations unlikely to result in the reuse of data (*id.*, 503).

**B. Rationale For Combining Smith With Bergsten (Ground 4)**

Smith is a background reference generally discussing cache memory and techniques used in such memories. A POSITA would be very familiar with ACM Computing Surveys publications such as Smith. EX1003, ¶115; EX1003, ¶141. ACM has been publishing articles related to computing for over 50 years. EX1003, ¶115; EX1003, ¶141. As Bergsten discloses a system using cache memory, a POSITA would naturally use Smith as a reference paper for understanding the details of cache memory when implementing the Bergsten system. EX1003, ¶141

Smith specifically describes multicache systems. EX1015, 504-55. Smith describes “[l]arge modern computer systems...hav[ing] several independent

processors, consisting sometimes of several CPUs” where each processor may have its own assigned cache and “with a main memory behind the cache.” *Id.*, 503-05. A POSITA would understand this description to be applicable to the system of Bergsten, which also constitutes a system of networked independent processors (hosts), with each individual host assigned its own cache (storage controller) and main memory (storage array). EX1003, ¶141. A POSITA would therefore understand Smith’s teachings to be applicable to and compatible with Bergsten’s networked multicache system. *Id.*

Smith also teaches various design techniques to improve the function of cache memories and multicache systems. *Id.* Therefore, applying these techniques to the multicache system of Bergsten constitute nothing more than use of known techniques to improve a similar device in the same way, as well as applying known techniques to a known device ready for improvement to yield predictable results. *Id.* For example, one technique taught by Smith is the broadcast of invalidate messages, and as explained in detail below Smith provides an express motivation for a POSITA to use the technique to ensure consistency of multiple copies of information stored across a distributed storage system. *Id.* Another technique taught by Smith is the bypass of the cache, and as explained below Smith provides an express motivation for implementing the technique in order to avoid displacing from the cache data that is likely to be reused in favor of non-reused data. *Id.*

**C. Rationale For Combining  
Smith With Bergsten And Blumenau (Ground 5)**

As explained above for Ground 3 (*supra* Section IX), a POSITA would have been motivated to combine the volume access table taught by Blumenau with the distributed storage architecture disclosed by Bergsten. EX1003, ¶150. For the same reasons discussed above, a POSITA would have been further motivated to combine the same teachings of Smith with the proposed combination of Bergsten and Blumenau for the same reasons as explained for Ground 4. EX1003, ¶¶141, 150. The addition of Blumenau's volume access table to the architecture of Bergsten would not dissuade a POSITA from implementing the additional teachings of Smith in order to attain their known benefits. EX1003, ¶150.

**D. Claim 5**

[5.1] The system of claim 4, wherein the configuration manager includes software that checks an access mode in the volume access table and

**Ground 4** - As explained above for Ground 2, claim 4 is obvious over Bergsten. Claim 5 is obvious over the further combination with Smith. As explained above in Sections VIII.A & VIII.B, it would have been obvious for a POSITA to implement Bergsten to contain a volume access table storing Exclusive Lock and Shared Lock signals. The storage controller OS of Bergsten, in managing data caching and data flow between networked host computers and storage devices (EX1004, 7:29-31), checks the Exclusive Lock/Shared Lock access signals stored in

this table to determine an appropriate access mode for a request from a given host. EX1003, ¶142. For example, Bergsten's storage controller, in response to a read request (Fig. 14), at step 1405 checks the stored access signal associated with the requested data to determine whether Exclusive Lock is set. EX1004, 12:47-51. Similarly, in response to a write request (Fig. 15), the storage controller at step 1507 determines whether Exclusive Lock is possible by checking that neither Shared Lock nor Exclusive Lock are set. *Id.*, 13:23-27; EX1003, ¶142.

**Ground 5** - As explained above for Ground 3, claim 4 is met by the combination of Bergsten and Blumenau. Claim 5 is obvious over the further combination with Smith.

The storage controller OS of Bergsten, in managing data caching and data flow between networked host computers and storage devices (EX1004, 7:29-31), checks the stored Exclusive Lock/Shared Lock access signal. Bergsten does not explicitly disclose how the access signals are structured and stored in the system of Bergsten. Thus, Bergsten leaves it to a POSITA to implement its access modes according to solutions known in the art. EX1003, ¶152. A POSITA, having implemented the volume access table of Blumenau into the distributed storage system of Bergsten, would be motivated to store the Exclusive/Shared lock values in this volume access table. As explained above for claim [2.3] in Ground 2 (*supra* Section VIII.A), a POSITA would recognize a table as a suitable data structure for

storing the mode values and a POSITA would implement the values in the already available volume access table of Blumenau rather than generate and store an entirely new data structure. *Id.*

[5.2] (i) if the access mode is set to an exclusive mode, causes both reads and writes to be stored in the cache memory, and causes invalidate messages to be sent to remote storage systems;

**Ground 4** - Bergsten discloses that if the access signal associated with requested data is set to Exclusive lock, then both reads and writes are stored in cache memory. EX1003, ¶143. Specifically, Bergsten teaches both reads stored in cache memory and writes stored in cache memory. *Id.*, Fig. 14 at step 1404, Fig. 15 at 1506. Regarding write operations, Bergsten teaches that “[o]nly a single device is permitted access to the given block of data.” *Id.*, 11:25-28. Therefore, in order to perform a write of requested data, the associated Exclusive Lock signal must be set “so that no other device may access that block during the write operation. *Id.*, 13:28-34. For read operations, by contrast, there is no need to allow only a single device to access at a given time and, therefore, “multiple devices are permitted to read from a given block of data at the same time” because there is no concern about data being overwritten while being accessed by other devices. *Id.*, 11:34-36. In contrast, a POSITA would recognize that while a host holds a given volume in an exclusive access mode then that host would be allowed to perform both read and write operations as disclosed in Figures 14 and 15 of Bergsten, because no other device

would be able to overwrite data in the volume due to the Exclusive Lock held by the host. EX1003, ¶143.

Bergsten does not explicitly disclose causing invalidate messages to be sent to remote storage systems. However, invalidate messages were well-known procedures associated with distributed storage systems and are explicitly taught by Smith. EX1003, ¶144. As discussed above, a POSITA would understand Smith to be applicable to the system of Bergsten, which also constitutes a system of networked independent processors (hosts), with each individual host assigned its own cache (storage controller) and main memory (storage array). *Id.*

Bergsten distributes data throughout its system, as “the storage controllers automatically provide multiple back-up copies of stored data.” EX1004, 5:24-26. “For example, when a given data file is saved by a local host computer to its local storage array, the local storage controller causes one or more back-up copies of the data file to be stored in one or more of the remote storage arrays, without any intervention from or interruption to the local host computer.” *Id.*, 5:27-33. And because copies of the same file may exist in multiple storage arrays, a POSITA would also understand that copies may also be present in the storage cache of each associated storage controller to facilitate efficient access to that data. EX1003, ¶145 (explaining advantages of storing multiple copies of the same data in multiple caches).

Smith describes a known problem with such distributed storage systems:

Unfortunately, in such a multiple processor system, a given piece of information may exist in several places at a given time, and it is important that all processors have access (as necessary) to the same, unique (at a given time) value.

EX1015, 505. That is, it is important that each copy of a file distributed across the system is the same, so that different processors accessing the same file are truly operating with the same data. EX1003, ¶146.

Smith describes sending invalidate messages as a solution to this problem of potential inconsistencies among multiple file copies across a distributed storage system. Smith explains as follows:

2. *Broadcast Writes*. Every time a CPU performs a write to the cache, it also sends that write to all other caches in the system. If the target of the store is found in some other cache, it may be either updated or invalidated. Invalidation may be less likely to create inconsistencies.... Updates also require more data transfer.

EX1015, 505. Thus, Smith describes a specific solution to maintain data consistency for multiple file copies across a distributed storage system in response to a write request. EX1003, ¶147. Bergsten already recognizes the need to maintain data consistency (EX1004, 5:61-62, 12:62-64, 14:42-43), and a POSITA would therefore

be motivated to implement the known solution of invalidate messages as taught by Smith in order to further ensure distributed data consistency. EX1003, ¶147. Therefore, a POSITA would modify Bergsten to cause invalidate messages to be sent in response to a write operation editing a stored file copy. *Id.*

**Ground 5** - The above analysis applies equally to this limitation for Ground 5 where the Exclusive Lock access signal of Bergsten is stored in a volume access table as taught by Blumenau. EX1003, ¶¶143-147, 153-155.

|                                                                                                         |
|---------------------------------------------------------------------------------------------------------|
| [5.3] (ii) if the access mode is set to shared, causes only reads to be stored in the cache memory; and |
|---------------------------------------------------------------------------------------------------------|

**Ground 4** - Bergsten teaches that when its Shared Lock access mode is set, only reads are stored in cache memory. As explained above for limitation [5.2], Fig. 14 of Bergsten, at step 1404, discloses reads stored in cache memory and write operations are only allowed for an Exclusive Lock mode. For read requests, however, “[w]hen a read is initiated on a given block, Shared Lock for that block is set, so that no device may write to that block during the read operation.” *Id.*, 11:40-42. Therefore, a POSITA would understand that where the access value table contains a Shared Lock access signal, this would allow reads to be stored in the cache memory according to the read operation of Bergsten but would not allow write operations. EX1003, ¶148.

**Ground 5** - The above analysis applies equally to this limitation for Ground 5 where the Shared Lock access signal of Bergsten is stored in a volume access table as taught by Blumenau. EX1003, ¶¶148, 156.

[5.4] (iii) if the access mode is set to a value other than exclusive or shared, causes reads and writes to be performed directly to a disk without using the cache memory.

**Ground 4** – Smith further discloses the “idea of bypassing the cache for operations unlikely to result in the reuse of data. For example, long vector operations such as a very long move character (e.g., IBM MVCL) could bypass the cache entirely . . . and thereby avoid displacing other data more likely to be reused.” EX1015, 503. A POSITA was aware of various types of operations, including read and write operations, that would be unlikely to result in the reuse of data and should therefore bypass the cache. See EX1003, ¶149 (discussing EX1016 at 2:19-22, which provides “it is not always in the best interest of total data processing performance and integrity to always use a caching buffer interposed between a using unit and a backing store”); see also EX1003, ¶149 (discussing EX1017 at 1:35-42 as further disclosing bypassing cache memory. Accordingly, a POSITA would have been motivated to modify the Bergsten to further incorporate the teaching of Smith to provide a cache bypass functionality for data that a POSITA would understand to be unlikely to be reused, and therefore prevent the displacement in the cache of more likely reused data. EX1003, ¶149. A POSITA would further find it obvious to

implement such a solution by including an additional access mode in the volume access table of Bergsten to indicate that the reads or writes of data associated with a given volume in the table should bypass the cache, i.e., they should be performed directly to disk without using the cache. *Id.*

Thus, claim 5 is unpatentable as obvious over Bergsten in view of Smith.

**Ground 5** - The above analysis applies equally to this limitation for Ground 5 and, for the same reasons discussed, a POSITA would include an additional access mode in the volume access table of Blumenau storing the access values of Bergsten to indicate that the reads or writes of data associated with a given volume in the table should bypass the cache, i.e., they should be performed directly to disk without using the cache. EX1003, ¶¶149, 157-58.

Thus, claim 5 is unpatentable as obvious over Bergsten and Blumenau in view of Smith.

## **XI. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS**

Petitioner is unaware of any objective indicia of non-obviousness as to any challenged claim.

**XII. THE BOARD SHOULD NOT USE ITS  
DISCRETION TO DENY INSTITUTION OF THIS PETITION**

Patent Owner asserts the '988 patent against Petitioner in a parallel district court litigation, which was assigned to Judge Robert Pitman on September 26, 2022. The District Court has not set a trial date.

Petitioner stipulates that, if the Board institutes IPR, it will not pursue in the parallel proceeding the grounds raised, or which reasonably could have been raised, in this petition as to the Challenged Claims. When a petitioner presents such a stipulation, “the PTAB will not discretionarily deny institution in view of parallel district court litigation.” USPTO Director’s Memorandum On Interim Procedure For Discretionary Denials In AIA Post-Grant Proceedings With Parallel District Court Litigation at 3, 7 (June 21, 2022). Accordingly, the Board should not discretionarily deny institution in view of the parallel district court litigation.

This is the first post-grant proceeding to challenge the '988 patent and, the grounds presented in this Petition are not cumulative or duplicative of any prior art or arguments previously presented to the Patent Office.

To the extent Patent Owner makes any arguments regarding discretionary denial, Petitioner reserves the right to further address them.

**XIII. CONCLUSION**

Petitioner requests institution of IPR on claims 1-6, based on the grounds presented in this petition.

Respectfully submitted,

Dated: March 23, 2023

By: /Todd M. Siegel/

Todd M. Siegel (Reg. No. 73,232)

todd.siegel@klarquist.com

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600

121 S.W. Salmon Street

Portland, Oregon 97204

Tel: 503-595-5300

Fax: 503-595-5301

Counsel for Petitioner

**CERTIFICATE OF COMPLIANCE WITH  
TYPE-VOLUME LIMITATION PURSUANT TO 37 C.F.R. § 42.24**

This brief complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(1)(i).

The brief contains 13,963 words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24(a).

The brief has been prepared in a proportionally spaced typeface using Microsoft Word 2013 in a 14-point Times New Roman font.

Dated: March 23, 2023

By: /Todd M. Siegel/  
Todd M. Siegel (Reg. No. 73,232)  
todd.siegel@klarquist.com  
KLARQUIST SPARKMAN, LLP  
One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Tel: 503-595-5300  
Fax: 503-595-5301

Counsel for Petitioner

**CERTIFICATE OF SERVICE  
IN COMPLIANCE WITH 37 C.F.R. § 42.6(e)(4)**

The undersigned certifies that the **Petition for *Inter Partes* Review of U.S. Patent No. 6,549,988 and Exhibits 1001 – 1018** were served on March 23, 2023, via **Express Mail** on the Patent Owner at the following address of record as listed on the USPTO Patent Center:

Toler Law Group  
8500 Bluffstone Cove  
Suite A201  
Austin, TX 78759

The undersigned also certifies that a true and correct copy of the **Petition and Exhibits 1001 – 1018** were made available on OneHub, with a link for downloading them sent via electronic mail to counsel for LS Cloud Storage Technologies, LLC at the below email address:

William P. Ramey, III – [wramey@rameyfirm.com](mailto:wramey@rameyfirm.com)

By: /Todd M. Siegel/  
Todd M. Siegel (Reg. No. 73,232)  
[todd.siegel@klarquist.com](mailto:todd.siegel@klarquist.com)  
KLARQUIST SPARKMAN, LLP  
One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Tel: 503-595-5300  
Fax: 503-595-5301

Counsel for Petitioner