

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of:	Vipul Pabari	Attorney Docket No. 42342-0157IP1
U.S. Patent No.:	12,079,667	
Issue Date:	September 3, 2024	
Appl. Serial No.:	18/135,007	
Filing Date:	April 14, 2023	
Title:	VIRTUAL SYSTEMS MANAGEMENT	

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES
PATENT NO. 12,079,667, PURSUANT TO 35 U.S.C. §§ 311–319,
37 C.F.R. § 42

TABLE OF CONTENTS

I.	REQUIREMENTS FOR IPR	1
A.	Grounds for Standing.....	1
B.	Challenge and Relief Requested.....	1
C.	Claim Construction	2
1.	“Virtual Asset”	3
2.	“Resources”/”Physical Resources”	3
3.	“Software Instantiations of Computer Systems”	3
4.	“Allocation of Data”	4
5.	“Control Computer”	5
6.	“Logical”	5
7.	“Middleware”	6
8.	“Configuration Management”	7
9.	“Application Routing for Virtual Assets”	8
D.	Level of Ordinary Skill in the Art.....	9
II.	THE ’667 PATENT	9
A.	Brief Description.....	9
III.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	10
A.	GROUND 1	11
1.	Hashimoto	11
2.	Lubbers	14
3.	The Hashimoto-Lubbers Combination.....	17
4.	Claim 1	22
5.	Claim 2	39
6.	Claim 3	41
7.	Claim 4	41
8.	Claim 5	43
9.	Claim 7	44
10.	Claim 8	46
B.	GROUND 2	49
1.	Hirschfeld	49
2.	The Hashimoto-Lubbers-Hirschfeld Combination	50
3.	Claim 9	55
4.	Claim 10	63
5.	Claim 11	64
6.	Claim 12	69
7.	Claim 13	69

8.	Claim 14	69
9.	Claim 15	69
10.	Claim 16	70
11.	Claim 17	70
12.	Claim 18	71
13.	Claim 19	75
14.	Claim 20	75
C.	GROUND 3	75
1.	The Hashimoto-Lubbers-Khandekar Combination.....	75
2.	Claim 6	78
IV.	FEES	79
V.	MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1).....	80
A.	Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....	80
B.	Related Matters Under 37 C.F.R. § 42.8(b)(2).....	80
C.	Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3).....	80
D.	Service Information	81

LIST OF EXHIBITS

EX1001	U.S. Patent No. 12,079,667 to Vipul Pabari (“the ‘667 Patent”)
EX1002	Excerpts from the Prosecution History of the ‘667 Patent (“the Prosecution History”)
EX1003	Declaration and Curriculum Vitae of Dr. Michael D. Mitzenmacher
EX1004	Complaint, <i>ToutVirtual, Inc. v. Microsoft Corporation</i> , 2-25-cv-00943 (EDTX) filed September 8, 2025
EX1005	U.S. Pub. No. 2006/0174087 to Akiyoshi Hashimoto <i>et al.</i> (“Hashimoto”)
EX1006	U.S. Pub. No. 2003/0084241 to Clark E. Lubbers <i>et al.</i> (“Lubbers”)
EX1007	U.S. Pub. No. 2003/0051021 to Robert A. Hirschfeld <i>et al.</i> (“Hirschfeld”)
EX1008	U.S. Pat. No. 8,799,431 to Vipul Pabari (’431 Patent).
EX1009	[RESERVED]
EX1010	Internet Archive Capture of https://en.wikipedia.org/wiki/Middleware from Jul. 29, 2005 (available at https://web.archive.org/web/20050729234922/https://en.wikipedia.org/wiki/Middleware)
EX1011	U.S. Pub. No. 2006/0294519 to Naoya Hattori <i>et al.</i>
EX1012	U.S. Pat. No. 7,577,722 to Dilip Khandekar <i>et al.</i> (“Khandekar”)
EX1013	U.S. Pat. No. 8,335,794 to John Goodson <i>et al.</i> (“Goodson”)
EX1014	U.S. Pub. No. 2006/0136667 to Steven Shultz <i>et al.</i> (“Shultz”)
EX1015	U.S. Pub. No. 2003/0225801 to Murthy Devarakonda <i>et al.</i> (“Devarakonda”)

LISTING OF CHALLENGED CLAIMS

Claim 1	
[1pre]	A method for data storage across one or more virtualization system resources over a network, the method comprising:
[1.1.a]	implementing a virtualization system that includes a control center application, one or more physical resources, and one or more virtual assets on each of the physical resources,
[1.1.b]	wherein the virtual assets are managed by virtualization layer instances running on each of the physical resources, and
[1.1.c]	wherein at least two of the virtual assets on one or more of the physical resources include virtual storage assets;
[1.2]	obtaining a set of data to be stored the virtual storage assets on any of the physical resources;
[1.3]	selecting one or more of the virtual assets for storage of the set of data based on an allocation of data on the virtual storage assets; and
[1.4]	causing storage of the set of data at the selected virtual storage assets on any of the physical resources.
Claim 2	
[2]	The method of claim 1, wherein the at least one virtual storage asset is virtualized via storage virtualization layers running on each of the physical resources.
Claim 3	
[3]	The method of claim 1, wherein at least two virtual assets each comprise software instantiations of computer systems.
Claim 4	
[4]	The method of claim 2, wherein at least one of the virtual assets comprise middleware applications running on each of the physical resources to access the virtual storage assets via the storage virtualization layer.
Claim 5	
[5]	The method of claim 1, wherein the control center application is configured to perform configuration management or forecasting services via the virtualization layer instances.

Claim 6	
[6]	The method of claim 5, wherein the configuration management includes application routing for virtual assets or implementation of application rules.
Claim 7	
[7]	The method of claim 1, wherein the at least one virtual storage asset is a logical unit with a logical unit number (LUN).
Claim 8	
[8.1]	The method of claim 1, further comprising: obtaining a request from a user device to obtain at least a portion of the set of stored data;
[8.2]	retrieving at least the portion of the set of stored data from the virtual storage asset; and
[8.3]	transmitting at least the portion of the set of stored data to the user device via a network.
Claim 9	
[9pre]	A virtualization system comprising:
[9.1]	a control computer running a control center application; and
[9.2.a]	one or more physical resources, wherein one or more physical resources are configured to execute one or more virtual assets,
[9.2.b]	the virtual assets managed by virtualization layer instances on each of the physical resources,
[9.2.c]	wherein at least two of the virtual assets on the one or more of the physical resources include virtual storage assets,
[9.3]	wherein the control center application is configured to: obtain a request from a user device for data stored at any of the virtual storage assets;
[9.4]	retrieve the data from one or more virtual storage assets that are storing the data; and
[9.5]	transmit the retrieved data to the user device.
Claim 10	
[10]	The virtualization system of claim 9, wherein the virtual storage assets are virtualized via a storage virtualization layer.

Claim 11	
[11.1]	The virtualization system of claim 9, wherein the virtualization system is further configured to: obtain the data to be stored at any of the virtual storage assets; and
[11.2]	store the data across any of the virtual storage assets running on the physical resources.
Claim 12	
[12]	The virtualization system of claim 9, wherein each of the one or more physical resources execute at least two virtual assets that each comprise software instantiations of computer systems.
Claim 13	
[13]	The virtualization system of claim 9, wherein at least one of the virtual assets comprise middleware applications accessing the virtual storage assets via the storage virtualization layer.
Claim 14	
[14]	The virtualization system of claim 9, wherein the control center application is configured to perform configuration management or forecasting via the virtualization layer instances.
Claim 15	
[15]	The virtualization system of claim 9, wherein the virtual storage assets comprise databases.
Claim 16	
[16]	The virtualization system of claim 9, the virtual storage assets are logical units with a logical unit number (LUN).
Claim 17	
[17]	The virtualization system of claim 9, wherein the virtualization system is further configured to receive, over a network by a user, one or more instructions to configure the virtual storage assets.
Claim 18	
[18pre]	A method for data storage on virtualized resources, the method comprising:
[18.1.a]	receiving a request to store data, at a control center application in communication with one or more physical resources that are part of a

	virtualization system that includes the control center application, the one or more physical resources, and one or more virtual assets on each of the physical resources, and
[18.1.b]	wherein at least two of the virtual assets on one or more of the physical resources include virtual data storage assets;
[18.2]	selecting one or more of the virtual assets for storage of the set of data based on an allocation of data on the virtual data storage assets; and
[18.3]	causing storage of the data, by the control center application, at the selected virtual data storage assets on each of the one or more physical resources,
[18.4]	wherein the virtual data storage assets are managed by storage virtualization layer instances on each of the physical resources.
Claim 19	
[19]	The method of claim 18, wherein one or more virtual assets are configured to run on each of the physical resources, wherein the virtual assets are managed by virtualization layer instances running on each of the physical resources.
Claim 20	
[20]	The method of claim 19, wherein at least two of the virtual assets comprise software instantiations of computer systems, and wherein the virtual storage assets comprise databases.

Microsoft Corporation (“Petitioner” or “Microsoft”) petitions for *Inter Partes* Review (“IPR”) of claims 1-20 (“the Challenged Claims”) of U.S. Patent No. 12,079,667 (“the ‘667 Patent”).

I. REQUIREMENTS FOR IPR

A. Grounds for Standing

Petitioner certifies that the ‘667 Patent is available for IPR. Microsoft is not barred or estopped from requesting review of the Challenged Claims on the below-identified grounds.

B. Challenge and Relief Requested

Microsoft requests an IPR of the Challenged Claims on the grounds noted below. Dr. Mitzenmacher provides supporting testimony in his Declaration (EX1003).

Ground	Claim(s)	35 U.S.C. § 103
1	1-5, 7, 8	Hashimoto, Lubbers
2	9-20	Hashimoto, Lubbers, Hirschfeld
3	6	Hashimoto, Lubbers, Khandekar

As shown in the table below, all prior art references pre-date 08/15/2005 (“Critical Date”), as identified in bold, and, thus, constitute prior art.

Reference	Filing Date	Publication Date
Hashimoto (EX1005)	05/20/2005	08/03/2006
Lubbers (EX1006)	10/22/2001	05/01/2003
Hirschfeld (EX1007)	03/18/2002	03/13/2003
Khandekar (EX1012)	04/05/2002	08/18/2009

C. Claim Construction

“[C]laim terms need only be construed to the extent necessary to resolve the controversy.” *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011). In this case, Petitioner submits that the Grounds would render the Challenged Claims obvious under any reasonable construction, including plain and ordinary meaning. For clarity, Petitioner proposes the following constructions, which are supported by the disclosure of the ’667 Patent, and therefore demonstrate subject matter that falls within the metes and bounds of these terms.

Petitioner reserves the right to respond to any constructions offered by Patent Owner or adopted by the Board. Petitioner is not conceding that each challenged claim satisfies all statutory requirements, nor is Petitioner waiving any arguments concerning claim scope or grounds that can only be raised in district court.

1. “Virtual Asset”

The ’667 Patent specifies that “virtual devices that exist only as software instantiations of equipment objects in a virtual environment will be referred to as virtual assets.” EX1001, 6:43-47. The ’667 Patent further explains that “[t]he resource-asset dichotomy will be maintained throughout this discussion.” EX1001, 6:47-48. Accordingly, accounting for context offered by the ’667 Patent, a POSITA would have understood the claimed “virtual asset” encompasses “software instantiations of equipment objects that exist only within a virtual environment.” EX1001, 6:43-48; EX1003, [46].

2. “Resources”/“Physical Resources”

The ’667 Patent specifies that, “[i]n this description, physical devices such as computers, printers, routers, and other “boxes” will be referred to as resources.” EX1001, 6:43-47. The ’667 Patent further explains that “[t]he resource-asset dichotomy will be maintained throughout this discussion.” EX1001, 6:47-48. Accordingly, accounting for context offered by the ’667 Patent, a POSITA would have understood the claimed “resources” and “physical resources” encompass “physical devices on the network, such as computers, printers, routers, and other tangible hardware.” EX1001, 6:43-48; EX1003, [47].

3. “Software Instantiations of Computer Systems”

The ’667 Patent states that “[virtual] assets...provide the functional equivalent of desktops (user interfaces), operating systems, applications, servers,

data bases, and the like.” EX1001, 3:17-21. In FIG. 3, the virtual assets are illustrated as two virtual computers, or machines, a first Virtual Machine 328 and a second Virtual Machine 330.” EX1001, 10:10-12. “Each virtual machine can include an operating system (OS) for the virtual machine, computer applications programs executing on the virtual machine OS, and virtual hardware assets for the virtual machine.” EX1001, 10:13-16. Accordingly, accounting for context offered by the ’667 Patent, a POSITA would have understood the claimed “software instantiations of computer systems” encompasses “software that can virtually perform functions of physical devices that can connect to computers.” EX1003, [48].

4. “Allocation of Data”

The ’667 Patent states that “FIG. 12 is a block diagram representation of how storage in physical servers can be allocated by the virtual server using the system management of the present invention.” EX1001, 20:58-60. FIG. 12 “compris[es] an association or mapping between actual Physical Resources comprising a collection of disk drives or storage devices 1204 to a collection of Logical Units 1206, such as network virtual storage drives, in accordance with a set of application critical Rules 1208.” EX1001, 20:60-67. The ’667 Patent further describes that “[t]he configuration of the Logical Units 1206 and their corresponding virtual drives will be governed by the set of Rules 1208.” EX1001,

21:7-9. Accordingly, accounting for context offered by the '667 Patent, a POSITA would have understood that the claimed “allocation of data” encompasses “the mapping of physical storage capacity to logical storage units according to specific application critical rules.” EX1003, [49].

5. “Control Computer”

The '667 Patent states that “[t]he Control Center computers are represented in FIG. 1 as physical devices rather than virtual assets.” EX1001, 6:35-37. “Each Control Center computer 110, 112 that is equipped with the virtualization management application described herein will have management of one or more physical assets within a domain or other subnet arrangement associated with the respective Control Center.” EX1001, 7:13-16. Accordingly, accounting for context offered by the '667 Patent, a POSITA would have understood that the term “control computer” encompasses a “computing device that is separate and distinct from the physical host machines being managed and is not itself deployed as a virtual asset. EX1003, [50].

6. “Logical”

The '667 Patent states that “FIG. 12 depicts the configuration management 120 of the network computer system 100 (FIG. 1) as comprising an association or mapping between actual Physical Resources comprising a collection of disk drives or storage devices 1204 to a collection of Logical Units 1206, such as network

virtual storage drives, in accordance with a set of application critical Rules 1208.”

EX1001, 20:60-67. “That is, the storage servers (physical resources comprising disks) are mapped to virtual assets (logical units).” EX1001, 20:67-21:2.

Accordingly, accounting for context offered by the ’667 Patent, a POSITA would have understood the claimed “logical” encompasses “software-defined abstraction that requires mapping to underlying physical hardware. EX1003, [51].

7. “Middleware”

The ’667 Patent provides multiple examples of “middleware.” EX1003, [52]. In one example, the ’667 Patent states that “[a] Virtual Platform Abstraction Layer 416 permits communications between the host machine 302 resources and a set of multiple middleware adapters 418.” EX1001, 11:48-52. “The middleware adapters communicate with the virtualization environments of the host machine 302.” EX1001, 14:39-41.

In another example, the ’667 Patent describes that “applications can be supported by middleware software through an application virtualization layer 616.” EX1001, 16:63-65. “The middleware applications 618, 620, 622 can access data stores through a storage virtualization layer 624.” EX1001, 16:67-17:1.

Accordingly, accounting for context offered by the ’667 Patent, a POSITA would have understood the claimed “middleware” encompasses “software that acts as a bridge between an operating system or database and applications, especially

on a network.” EX1003, [52]-[54]; EX1010.

8. “Configuration Management”

The ’667 Patent states that “the Control Center application can perform configuration management and forecasting services for the host computer and the available virtual assets through each of the various virtualization layers.” EX1001, 17:3-8. “FIG. 11 is a block diagram representation of a server virtualization configuration management scheme that can be implemented from a computer such as the Control Center computers 110, 112 illustrated in FIG. 1.” EX1001, 20:32-35. “FIG. 11 depicts the virtualization management 1102 of the network computer system 100 (FIG. 1) as comprising an association or mapping between the actual Physical Resources, such as a collection of computer hardware machines 1104, to a collection of Virtual Machine Assets 1106, or applications, in accordance with a set of application critical Rules 1108.” EX1001, 20:35-41. “If configuration changes are called for,...then the Control Center issues commands to the virtualization layer software to implement the desired configuration changes.” EX1001, 23:16-19. The ’667 Patent describes various management functions, including “Create VM,” “Start VM,” “Create Virtual Disk,” “Consolidate_VM,” “Configure CPU,” “Configure disk,” “Configure memory,” “configure network,” etc. EX1001, 15:42-16:44. Accordingly, accounting for context offered by the ’667 Patent, a POSITA would have understood the claimed “configuration

management” encompasses “the creation, modification, and management of associations or mappings between physical resources and virtual assets, performed by the control center application in accordance with a defined set of application rules. EX1003, [55].

9. “Application Routing for Virtual Assets”

In the '667 Patent, “[t]he Control Center can [] provide Adaptive Application Routing (AAR) for virtual assets based on [Application Criticality Rules] ACRs.” EX1001, 18:3-4. “The ACRs allow a user to specify business rules for controlling priority settings for applications.” EX1001, 18:8-9. To this end, the Control Center includes a “Dynamic Application Router.” EX1001, 7:51-55, 8:54-9:7. “This component can be used to move applications to different virtual assets as global business need changes, provide dynamic allocation of global virtual assets for workload optimization, enable zero down time in upgrading virtual environments, provide an optimizer for balancing resource and asset inventory against needs, a scheduler to auto schedule actions and provide reports, and provide alarms and triggers to notify users of out of balance inventory items.” EX1001, 8:59-67. Further, this component “can compare business policy against real time asset usage and then make load balance changes.” EX1001, 8:67-9:7. In one example, “if a rule specifies that a particular application has a high priority at a particular time of day, then a respective one of the Control Center

computers 110, 112 (FIG. 1) will adjust the deployment of applications among the Virtual Machines 1106 under its management at the appropriate times.”

Accordingly, accounting for context offered by the ’667 Patent, a POSITA would have understood the claimed “application routing for virtual assets” encompasses “the dynamic deployment, movement, or allocation of applications to or between virtual assets based on business policies, workload conditions, or application criticality rules, as a function of configuration management performed by the control center application.” EX1003, [56].

D. Level of Ordinary Skill in the Art

A person of ordinary skill in the art (“POSITA”) relating to the subject matter of the ’667 Patent as of the Critical Date would have had a combination of (1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related field, and (2) at least two years of industry experience with distributed virtualization systems. EX1003, [20]-[22]. Additional education could substitute for professional experience, and professional experience could substitute for education. *Id.*

II. THE ’667 PATENT

A. Brief Description

The ’667 Patent is allegedly directed to “management of one or more virtual environments regardless of any underlying central processing unit (CPU) specification and regardless of any underlying operating system (OS) or

virtualization environment.” EX1001, 3:53-57. However, the ’667 Patent claims make no mention of CPU specification or underlying OS. EX1001, 24:10-26:30.

Rather, the ’667 Patent claims relate to simple storage and retrieval of data from virtualized storage assets. EX1001, 24:10-26:30. For example, claim 1 recites:

1. A method for data storage across one or more virtualization system resources over a network, the method comprising: - implementing a virtualization system that includes a control center application, one or more physical resources, and one or more virtual assets on each of the physical resources, wherein the virtual assets are managed by virtualization layer instances running on each of the physical resources, and wherein at least two of the virtual assets on one or more of the physical resources include virtual storage assets; - obtaining a set of data to be stored the virtual storage assets on any of the physical resources; - selecting one or more of the virtual assets for storage of the set of data based on an allocation of data on the virtual storage assets; and - causing storage of the set of data at the selected virtual storage assets on any of the physical resources.

EX1001, 24:10-27. Unsurprisingly, these basic storage features were well known in the art, as demonstrated by the Grounds of this petition. EX1003, [58].

III. THE CHALLENGED CLAIMS ARE UNPATENTABLE

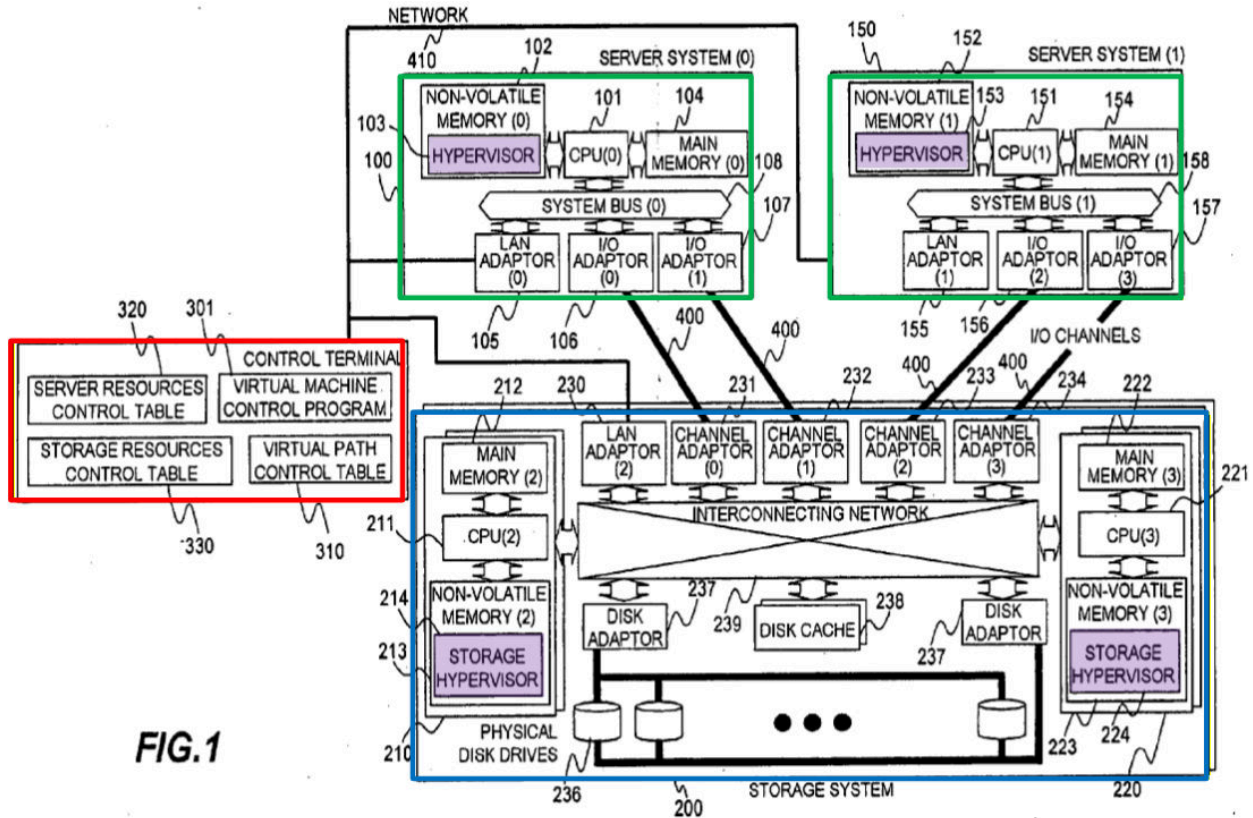
Hashimoto, Lubbers, Hirschfeld, and Khandekar are all analogous art, each being in the field of endeavor (virtualization systems) and reasonably pertinent to the problems said to be addressed by the ’667 Patent because they each seek to improve users’ and administrators’ ability to efficiently manage a virtualization

environment. EX1003, [68]-[88]; EX1001, 3:47-49, Title, Abstract; EX1005, Title, Abstract, [0014]-[0016]; EX1006, Title, Abstract, [0012]-[0014]; EX1007, Title, Abstract, [0008], [0067]; EX1012, Title, Abstract, 1:7-10, 22:14-33, 13:44-46; *infra*, §IV.

A. GROUND 1

1. Hashimoto

Hashimoto “relates to a computer system, and more particularly to a technique of logically partitioning a computer system including a storage system.” EX1005, [0002]. Hashimoto’s system includes **one or more server systems 100** logically partitioned “to create virtual machines that operate independently” (EX1005, [0052], [0060]), **one or more storage systems 200** that can be logically partitioned “to create virtual storage systems that operate independently” (EX1005, [0052], [0070]), and **control terminal 300** running a virtual machine control program that “controls virtual configuration of the computer system” (EX1005, [0052], [0083]-0084). EX1003, [63].



EX1005, Fig. 1 (annotated)

The **control terminal 300** controls the operation of the virtualization system through communication with programs called “**hypervisors**” that are stored and run on each of the server systems and storage systems. EX1005, [0059]-[0060], [0069]-[0070]; EX1003, [64]. For example, to create a virtual machine on a server system 100, an “administrator operates the control terminal 300 to set allocation of server resources of the server system (0) 100 and gives an instruction to create a virtual machine,” and the “server system (0) 100 receives the virtual machine creation instruction from the control terminal 300 and then the hypervisor 103 creates a virtual machine by allocating server resources of the server system (0)

100.” EX1005, [0178]-[0179]. Once the virtual machine has been created, the control terminal 300 “checks whether a virtual storage system should be created in the storage system 200,” and if so, goes through a similar process with the storage hypervisor to allocate storage resources to a virtual storage system. EX1005, [0181]-[0185]; EX1003, [64].

“[I]n the virtual storage system (0) 240 and the virtual storage system (1) 250, the storage hypervisors 214 and 224 may partition a physical disk drive 236 into a plurality of virtual disks 244 and 254, or may unite a plurality of physical disk drives 236 into a single virtual disk 244 or 254.” EX1005, [0114]. “The storage system 200 selects a single virtual disk or a plurality of virtual disks from the virtual disks 244 and 254 and provides the virtual disk or disks to the virtual machines 100 and 150 as storage areas.” EX1005, [0115]. Application programs run on the virtual machines and the virtual disks store data required for operations of the virtual machines. EX1005, [0052], [0102]; EX1003, [65].

“When the virtual storage system (0) 240 receives a disk I/O command from the virtual machine (0) 130, the virtual storage system (0) 240 obtains the virtual disk number that is the target of the access by the disk I/O command to specify the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. “In this process, the storage hypervisor 214 translates the virtual block address of the accessed virtual disk 244 to a physical block address of

the physical disk drive 236.” EX1005, [0230]. “Then, the storage hypervisor 214 accesses the physical disk drive 236 and reads/writes data.” EX1005, [0230]

2. Lubbers

Lubbers relates to “software, systems and methods for providing and managing virtualized data storage.” EX1006, [0002]. “Storage virtualization generally refers to systems that provide transparent abstraction of storage at the block level.” EX1006, [0014]. “[V]irtualization separates out logical data access from physical data access, allowing users to create virtual disks from pools of storage that are allocated to network-coupled hosts as logical storage when needed.” EX1006, [0014]-[0018]; EX1003, [70].

In Lubbers’s system, “[p]hysical storage is carved into units called physical segments,” called PSEGs. EX1006, [0019]. “Logical storage is implemented in atomic logical units called RStores comprising a range of virtual address space that when allocated, is bound to a particular group of PSEGs.” EX1006, [0019]. “A plurality of RStores make up a logical disk that is presented to a user.” EX1006, [0019]. Lubber’s system includes “a storage pool 101 comprising an arbitrarily large quantity of storage space from which logical disks (also called logical units or LUNs) 102 are allocated.” EX1006, [0034]. “Each LUN 102 is essential[ly] a contiguous range of logical addresses that can be addressed by host devices 105, 106, 107 and 109 by mapping requests from the connection protocol used by the

hosts to the uniquely identified LUN 102.” EX1006, [0035]. “Storage access requests expressed in terms of logical disk addresses are mapped to PSEGs [physical segments of physical storage] containing data represented by the logical addresses through a split-directory representation of the logical unit.” EX1006, [0019].

The memory mapping data structures proposed by Lubbers are illustrated in Fig. 5. EX1006, [0024], [0054]; EX1003, [72].

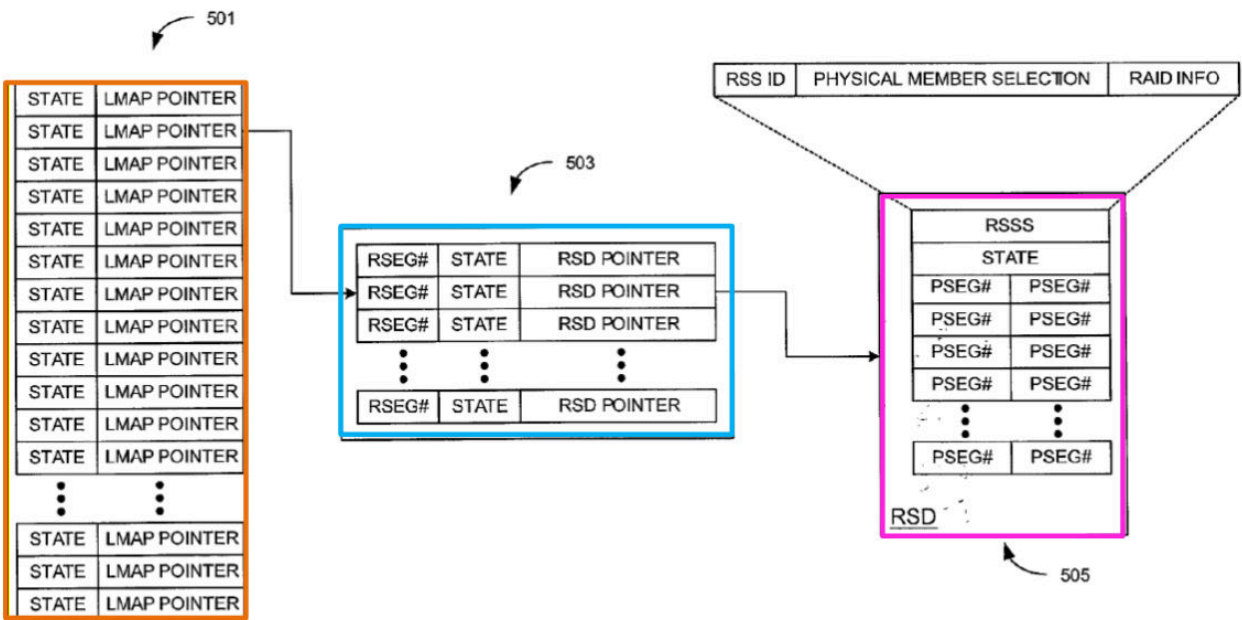


FIG. 5

EX1006, Fig. 5

The mapping data structures illustrated in Fig. 5 include **level 2 maps (L2MAP) 501**, **LMAPs 503**, and **redundancy set descriptors (RSDs) 505**. EX1006, [0056]-[0057]; EX1003, [73]. “A single L2MAP handles the entire LUN

102.” EX1006, [0057]. “L2MAP 501 includes a plurality of entries where each entry represents 2 Gbyte of address space.” EX1006, [0058]. “An LMAP 503 exists only for each 2 Gbyte increment of allocated address space.” EX1006, [0060]. In other words, “[e]ach LUN 102 is represented by multiple LMAPs 503 where the particular number of LMAPs 503 depend on the actual address space that is allocated at any given time.” EX1006, [0057].

“An LMAP 503 exists only for each 2 Gbyte increment of allocated address space.” EX1006, [0060]. “Each LMAP 503 includes a plurality of entries where each entry optionally corresponds to a redundancy segment (RSEG).” EX1006, [0060]. “An RSEG is an atomic logical unit that is roughly analogous to a PSEG in the physical domain—akin to a logical disk partition of an RStore.” EX1006, [0060]. “Each LMAP entry also includes state information about the particular RSEG, and an RSD pointer.” EX1006, [0060]. “The RSD pointer points to a specific RSD 505 that contains metadata describing the RStore in which the corresponding RSEG exists.” EX1006, [0062].

“An RStore is analogous to a RAID disk set, differing in that it comprises PSEGs rather than physical disks.” EX1006, [0068]. “An RStore is smaller than conventional RAID storage volumes, and so a given LUN 102 will comprise multiple RStores as opposed to a single RAID storage volume in conventional systems.” EX1006, [0068]. A primary benefit of the RStore/RSEG logical

structure is that Lubbers's system "can readily form a nested RAID set using, for example, increments of RStores rather than increments of six drives." EX1006, [0075].

3. The Hashimoto-Lubbers Combination

As noted in §III.A.1, *supra*, Hashimoto does not limit the manner in which the physical storage is partitioned into virtual storage or the manner in which virtual addresses are translated into physical addresses. EX1005, [0113]-[0116]; [0228]-[0230]; EX1003, [89]. Rather, Hashimoto is more focused on managing the pathways and communication between the virtual machines and virtual storage systems. EX1005, [0014], [0052], [0084], [0116], [0132]-[0134]; EX1003, [89]. Hashimoto leaves the details of how to partition the physical storage into virtual storage to the POSITA implementing the system. EX1003, [89].

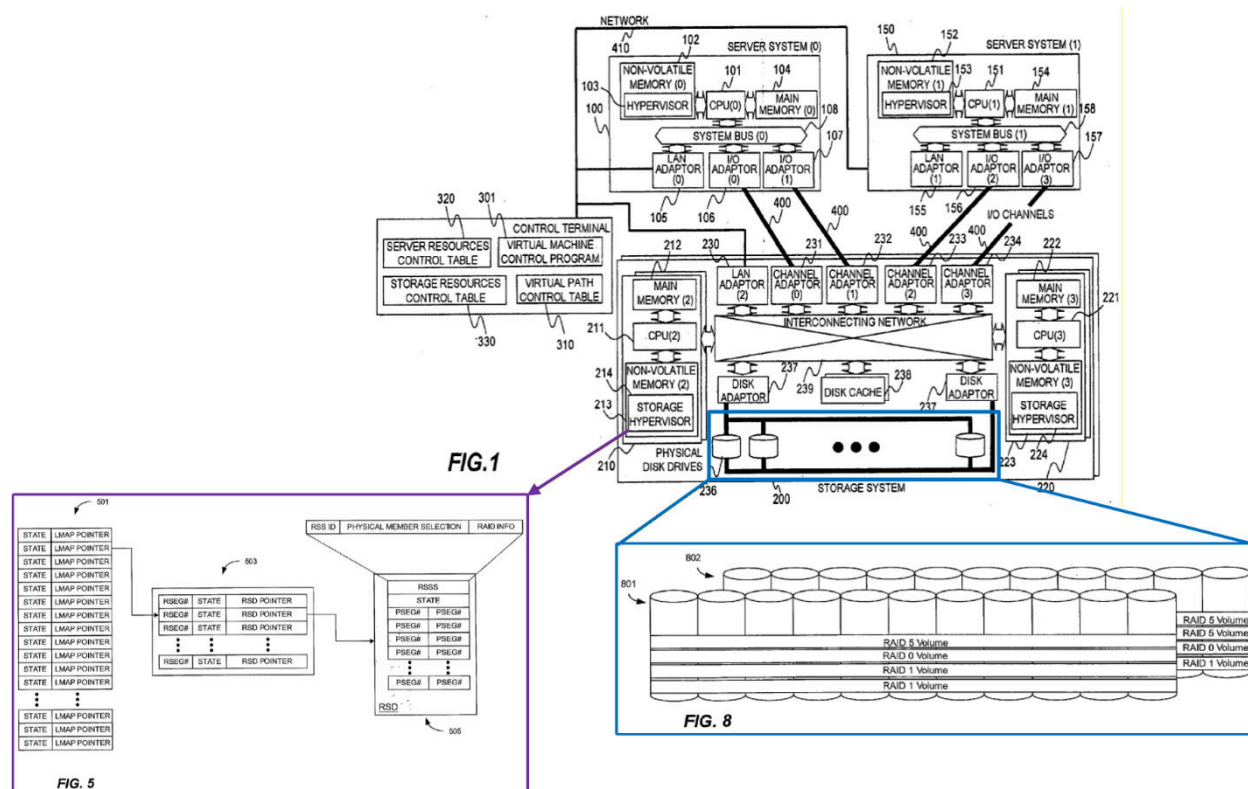
Lubbers provides an advantageous example of how to partition the physical storage into virtual storage in Hashimoto's virtualization environment. EX1003, [90]. Specifically, a POSITA would have found it obvious to incorporate the partitioning techniques described by Lubbers into Hashimoto's system. EX1003, [90]. For example, Lubbers describes a SAN manager appliance 109 that manages a "quorum space 113" in which metadata about the overall storage system is maintained. EX1006, [0040]-[0041]. A POSITA would have recognized that Lubbers's SAN manager appliance 109 is similar in structure and purpose to the

control terminal 300 described by Hashimoto, which similarly maintains various data tables used to manage Hashimoto's virtualization environment, including a storage resources control table 330 that "stores the associations of the resources of the storage system 200." EX1005, [0083]-[0085], [0116], [0161]; EX1003, [90].

As another example, Lubbers's storage cells 203 each include a redundant pair of NSCs 301, which "implement[] a network interface to receive storage access requests from hosts as well as fibre channel arbitrated loop ports to connect to storage device in cabinet 303." EX1006, [0049]-[0051]; EX1003, [90]. A POSITA would have recognized that Lubbers's storage cell 203 and NSCs 301 are similar in structure and purpose to the storage system 200 and redundant control blocks 210, 220 described by Hashimoto, which similarly "control data input/output to and from physical disk drives 236 and manages operations of the storage system 200." EX1005, [0065]-[0066]; EX1003, [90]. Indeed, Hashimoto teaches that "one virtual storage system can be configured by extracting resources of a plurality of storage systems," like the collection of storage cell 203 in Lubbers, and that its invention applicable "to such...virtual storage system configurations." EX1005, [0374]-[0375]. Accordingly, Hashimoto's system already includes much of the underlying structure necessary to employ Lubbers's partitioning techniques. EX1003, [90].

In an aspect of the proposed combination (a diagram of which is included

below), Hashimoto's virtualization system would implement the storage schemes, various data structures, and control functionality described by Lubber to virtualize Hashimoto's storage system 200 and "enable[] translation of a request expressed in terms of a logical block address (LBA) from host 301 into a read/write command addressed to a particular portion of a physical disk 405 having the desired information." EX1005, [0230]; EX1006, [0030]-[0033], [0046], [0054], [0127]-[0128]; EX1003, [91]. In the example implementation shown below, Hashimoto's physical disk drives 236 would have stored the **storage volumes** described by Lubbers, and Hashimoto's storage hypervisors would store the **mapping structures of the LUNs** described in FIG. 5 of Lubbers. *Id.* A POSITA would have been motivated to combine Hashimoto and Lubbers for several reasons. EX1003, [91].



Amalgamation of Fig. 1 of EX1005 and Figs. 5 and 8 of EX1006

First, Lubbers describes that its method for virtualizing a storage system “atomizes storage capacity in the physical domain in the form of PSEGs and in the logical domain in the form of RStores to enable fluid movement of capacity.” EX1006, [0128]. “The relationship of particular disks can be modified to provide various levels of protection and allow for construction of virtual disks with variable levels of data protection.” EX1006, [0128]. A POSITA would have recognized that these express benefits of Lubbers’s methodology were directly applicable to Hashimoto’s similarly structured storage system 200. EX1003, [92].

Second, incorporating Lubbers's method for virtualizing a storage system into Hashimoto's virtualization system would have advanced Hashimoto's stated goals. EX1003, [93]. One of the goals of Hashimoto's system is to "realize a high-availability configuration of a computer system including virtual machines and virtual storage systems." EX1005, [0016]. Lubber explains that its method of virtualization "provide[s] high performance, high data availability, fault tolerance, and efficient storage management." EX1006, [0030].

Third, Lubbers provides one example of known storage virtualization techniques not otherwise specified in Hashimoto. EX1003, [94]. As a result, the combination of Hashimoto and Lubbers would have been no more than the obvious use of a known technique (*i.e.*, Lubbers's techniques for virtualizing networked storage systems) to improve similar devices (*i.e.*, Lubbers's storage cells 203 that include a set of physical disks and Hashimoto's storage system 200 that also includes a set of physical disks) in the same way. *KSR Intern. Co. v. Teleflex Inc.*, 550 US 398, 401 (2007); EX1003, [94].

Likewise, a POSITA would have reasonably expected success implementing the combination, especially since the resulting system would be implemented with conventional software and hardware techniques (*e.g.*, general-purpose processors and storage systems interconnected on well-known networks) using virtualization and data storage techniques (*e.g.*, RAID) that were well known by the Critical

Date. *KSR* at 401; EX1003, [95]. Further, the techniques integrated from Lubbers in the Hashimoto-Lubbers combination are fully compatible with Hashimoto's computer system of FIG. 1 and would not disturb the ability of Hashimoto's system to "manage physical paths between [] server system[s] and [] storage system[s] and information about routing between virtual machines and virtual storage systems in an integrated fashion." EX1005, Abstract; EX1003, [95].

4. Claim 1

[1pre]

Hashimoto's system and methods "manage physical paths between [] server system[s] and [] storage system[s] and information about routing between virtual machines and virtual storage systems in an integrated fashion." EX1005. [0014]. More specifically, Hashimoto describes a computer system that includes server systems 100 and 150, at least one storage system 200, and a control terminal 300, all of which are connected via a network 410. EX1005, [0052]; *Id.*, [0371]-[0375]; EX1003, [104]. "The storage system 200 stores data required for operations of the server systems 100 and 150." EX1006, [0052]. The storage system 200 runs a program called a "storage hypervisor" that "creates logical partitions in the storage system 200 to create virtual storage systems that operate independently." EX1005, [0070].

As discussed in §III.A.3, *supra*, in the Hashimoto-Lubbers combination,

Hashimoto's virtualization system would implement the various data structures and control functionality described by Lubber to virtualize Hashimoto's storage system 200. EX1003, [105]. And Lubbers's disclosure relates to "software, systems and methods for providing and managing virtualized data storage." EX1006, [0002].

[1.1.a]

As discussed for [1.pre], *supra*, and shown below in FIG. 2, Hashimoto describes a computer system that includes server systems 100 and 150, at least one storage system 200, and a control terminal 300, all of which are connected via a network 410. EX1005, [0052]; *Id.*, [0371]-[0375]; EX1003, [106].

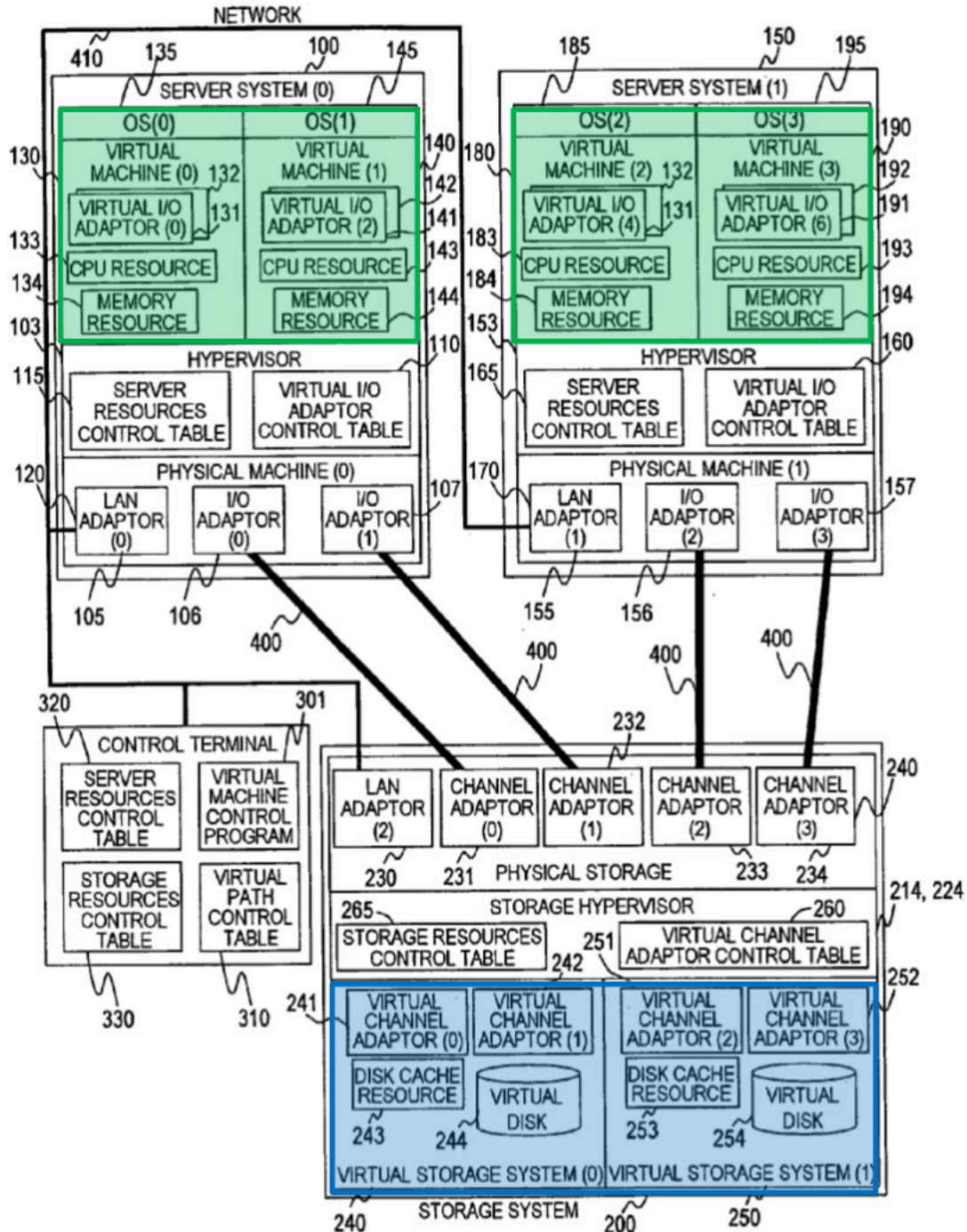


FIG. 2

EX1005, FIG. 2 (Annotated)

Each server system is “a computer having physical resources including a CPU (0) 101, a non-volatile memory (0) 102, a main memory (0) 104, a LAN

adaptor (0) 105, an I/O adaptor (0) 106, an I/O adaptor (1) 107, and a system bus 108.” EX1006, [0053]. “The non-volatile memory (0) 102 stores a hypervisor 103.” EX1006, [0059]. “The hypervisor 103, implemented by the execution of processing by the CPU (0) 101, is a means that enables logical partitioning of physical resources of the server system (0) 100.” EX1006, [0059]. The “hypervisor 103 is a control program that creates logical partitions in the server system (0) 100 to create virtual machines that operate independently.” EX1006, [0060]; EX1003, [107].

Similarly, “[t]he storage system 200 has a storage control block 210 including a CPU (2) 211, a main memory (2) 212, and a non-volatile memory (2) 213.” EX1005, [0065]. A second storage control block 220 includes similar components. EX1005, [0065]; EX1003, [108]. “The storage control blocks 210 and 220 control data input/output to and from physical disk drives 236 and manages operations of the storage system 200.” EX1005, [0066]. Storage hypervisor 214 and 224 are “implemented by the execution of processing by the CPUs 211 and 221, is a means that enables logical partitioning of physical resources of the storage system 200.” EX1005, [0069]. “The storage hypervisor 214 is implemented with a control program that creates logical partitions in the storage system 200 to create virtual storage systems that operate independently.” EX1005, [0070].

“The control terminal 300 manages information for controlling the computer system and controls operations of the entire computer system.” EX1005, [0052].

“A virtual machine control program 301 runs in the CPU of the control terminal 300.” EX1005, [0083]. “The virtual machine control program 301 controls virtual configuration of the computer system by using the virtual path control table 310, the server resources control table 320, and the storage resources control table 330.” EX1005, [0084]. The control program 301 communicates with the hypervisors 103 and 153 and storage hypervisors 214 and 224 to control the partitioning and maintain the control tables. EX1005, [0060], [0070], [0179]-[0180], [0184], [0193]; EX1003, [109].

FIG. 2 (above) shows each of server systems 100 and 150 and storage system 200 divided into a physical layer, a hypervisor layer, and a virtual machine/storage layer. EX1005, [0090], [0104]. Server systems 100 and 150 have each been partitioned into two virtual machines (130, 140, 180, and 190). EX1005, [0100], [0103], [0164]-[0167]. Similarly, storage system 200 has been partitioned into two virtual storage systems (240 and 250). EX1005, [0113]-[0114].

Accordingly, Hashimoto-Lubbers *implement[s] a virtualization system* (e.g., Hashimoto’s computer system of FIGS. 1 and 2) *that includes a control center application* (e.g., virtual machine control program 301), *one or more*

physical resources (e.g., the physical resources of server systems 100 and 150, and storage system 200), *and one or more virtual assets on each of the physical resources* (e.g., virtual machines 130, 140, 180, and 190 and virtual storage systems 240 and 250). EX1003, [111].

[1.1.b]

As discussed for [1.1.a], *supra*, the server systems 100 and 150 run a hypervisor 103, 153, and the storage control blocks 210 and 220 of storage system 200 run a storage hypervisor 214, 224. EX1005, Fig. 1-2, [0059]-[0060], [0069]-[0070], [0092], [0103], [0106]; EX1003, [112]. As shown in FIG. 2, “[f]unctionally, the server system (0) 100 can be roughly divided into a physical layer, a hypervisor layer, and a virtual machine layer.” EX1005, [0090]. The “hypervisor layer is implemented by the hypervisor 103...” EX1005, [0092]. Similarly, “storage system 200 can be roughly divided into a physical layer, a hypervisor layer, and a virtual storage layer.” EX1005, [0104]. “The hypervisor layer is implemented by the storage hypervisors 214 and 224...” EX1005, [0106].

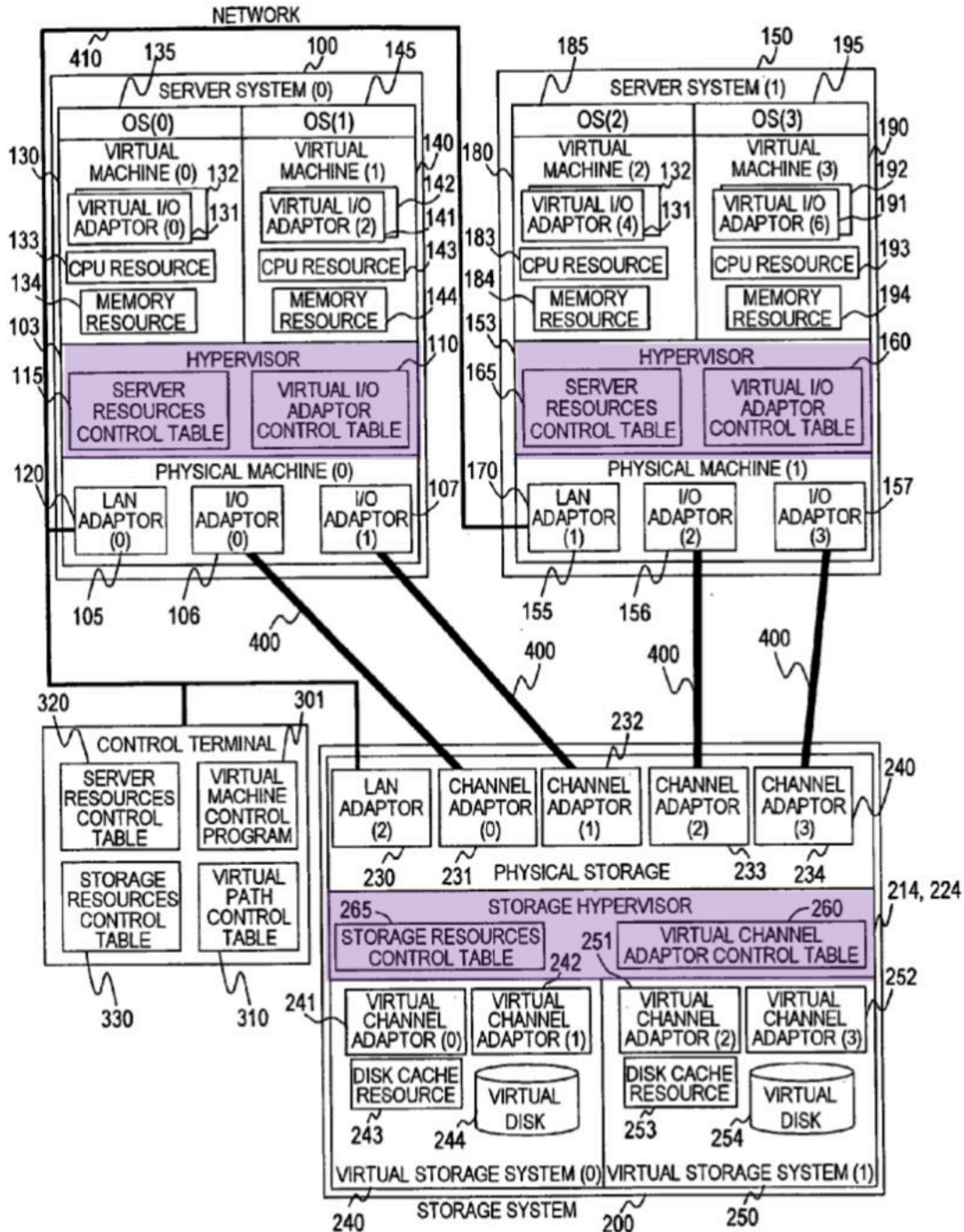


FIG. 2

EX1005, FIG. 2 (Annotated)

The hypervisors 103 and 153 perform management functions including, for example, creating the virtual machines (EX1005, [0179]-[0180]), routes I/O

requests generated by the virtual machines (EX1005, [0223]-[0224]), detecting and fixing faults in the I/O channels of the virtual machines (EX1005, [0236]-[0246]), and rebooting the virtual machines (EX1005, [0291]-[0296]). EX1003, [113]. The storage hypervisors 214 and 224 perform similar management functions including, for example, creating the virtual storage systems (EX1005, [0184]-[0185]), routes I/O requests generated by the virtual machines (EX1005, [0206], [0226]-[0227]), and detecting and fixing faults in the I/O channels of the virtual machines (EX1005, [0247]-[0259]). EX1003, [113].

Accordingly, in Hashimoto-Lubbers, *the virtual assets* (e.g., virtual machines 130, 140, 180, and 190 and virtual storage systems 240 and 250) *are managed by virtualization layer instances* (e.g., the hypervisor layers implemented by hypervisors 103 and 153 and storage hypervisors 214 and 224) *running on each of the physical resources*. EX1003, [114].

[1.1.c]

As discussed for [1.1.a], *supra*, storage system 200 has been partitioned into two virtual storage systems (240 and 250). EX1005, [0113]-[0114]; EX1003, [115].

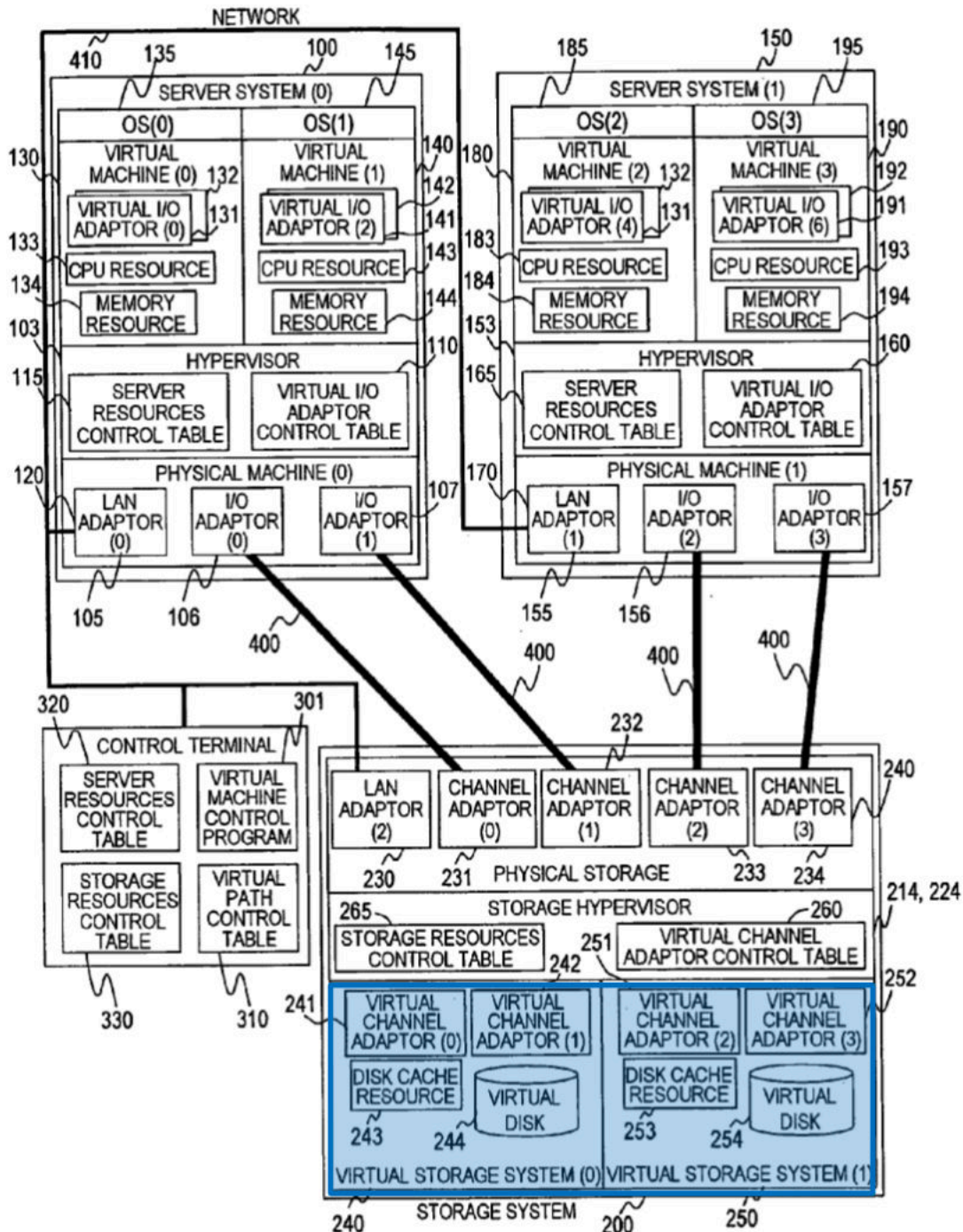


FIG. 2

EX1005, FIG. 2 (Annotated)

Accordingly, in Hashimoto-Lubbers, *at least two of the virtual assets on one or more of the physical resources* (e.g., storage system 200) *include virtual*

storage assets (e.g., virtual storage systems 240 and 250). EX1003, [116].

[1.2]

FIG. 14 of Hashimoto “is a flowchart of a disk I/O command transmission process,” and more particularly “an example in which the virtual machine (0) 130 accesses the virtual storage system (0) 240 through the I/O adaptor (0) 106 and the channel adaptor (0) 231.” EX1005, [0222]. “When a disk I/O command is issued, the OS (0) 135 sends the disk I/O command to the virtual machine (0) 130 (the virtual I/O adaptor (0) 131),” which is “received by the hypervisor 103.” EX1005, [0222]. The I/O command “specif[ies] the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. The storage system 200 receives the I/O command via channel adaptor (0) 231. EX1005, [0226]-[0227]; EX1003, [117].

As part of fulfilling the I/O command, “the storage hypervisor 214 translates the virtual block address of the accessed virtual dis 244 to a physical block address of the physical disk drive 236.” EX1005, [0230]. Lubbers similarly describes “translation of a request expressed in terms of a logical block address (LBA) from host 301 into a read/write command addressed to a particular portion of a physical disk 405 having the desired information.” EX1006, [0054]. As discussed in §III.A.3, *supra*, Lubbers provides, for example, the translation methods utilized in Hashimoto-Lubbers. EX1003, [118].

Accordingly, Hashimoto-Lubbers *obtain[s] a set of data to be stored* (e.g., a virtual storage system obtains data included in an I/O write command that is sent from a virtual machine, and the virtual storage system “writes data to... the corresponding virtual disk”) *the virtual storage assets on any of the physical resources* (e.g., a virtual disk of a virtual storage system run on storage system 200). EX1003, [119].

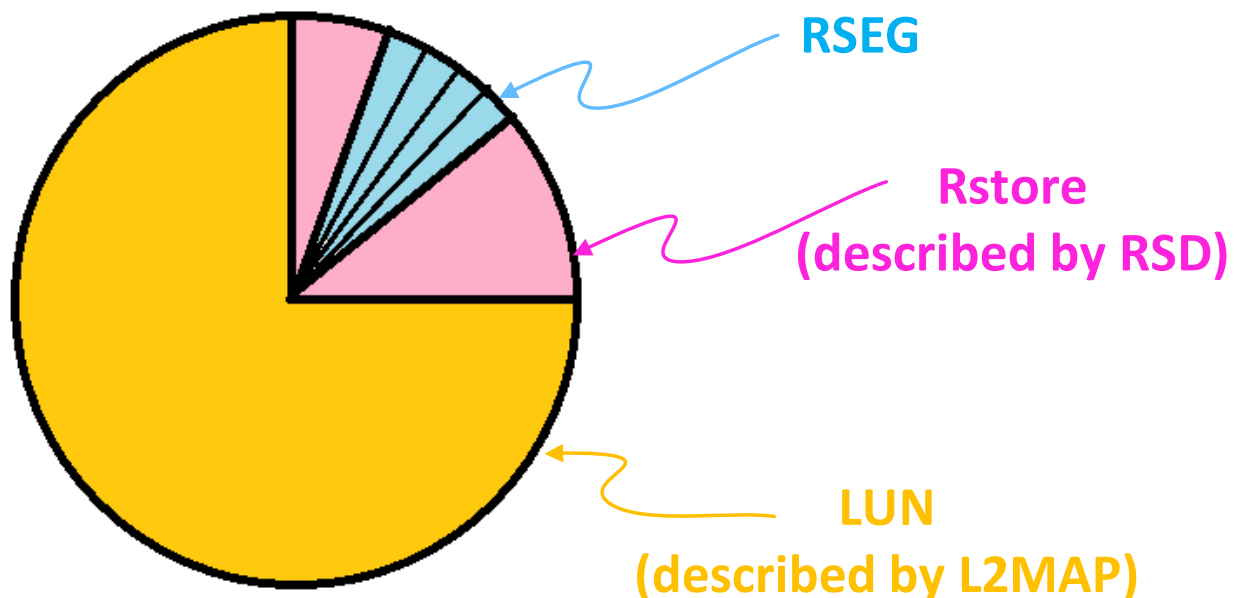
[1.3]

As discussed in §III.A.3, *supra*, Hashimoto-Lubbers would implement the various data structures and control functionality described by Lubber to virtualize Hashimoto’s storage system 200 and “enable[] translation of a request expressed in terms of a logical block address (LBA) from host 301 into a read/write command addressed to a particular portion of a physical disk 405 having the desired information,” as taught by Hashimoto. EX1006, [0030]-[0033], [0046], [0054], [0127]-[0128]; EX1003, [120].

As discussed in §III.A.2, *supra*, Lubbers describes that logical disks (also called logical units or LUNs) 102 are allocated from a storage pool provided by a set of physical storage devices. EX1006, [0034]-[0035]. “A plurality of RStores make up a logical disk that is presented to a user.” EX1006, [0019]. “An RSEG is an atomic logical unit that is roughly analogous to a PSEG in the physical domain—akin to a logical disk partition of an RStore.” EX1006, [0060]. In other

words, a logical disk is made up of logical units called RStores, which themselves are partitioned into logical units called RSEGs. EX1006, [0060]-[0061], [0065]-[0070]; EX1003, [121].

Lubbers describes that “[a]n RStore is analogous to a RAID disk set, differing in that it comprises PSEGs rather than physical disks.” EX1006, [0068]. “An RStore is smaller than conventional RAID storage volumes, and so a given LUN 102 will comprise multiple RStores as opposed to a single RAID storage volume in conventional systems.” EX1006, [0068]. Below is a diagram of Lubber’s virtual storage structure utilized in Lubbers. EX1003, [122].



Virtual Storage

Diagram of Lubber’s Virtual Storage Structure

As shown below in Fig. 5, Lubbers uses a set of data mapping tables to

locate data within the logical and physical storage. EX1003, [123]. For example, “an **L2MAP 501**, an **LMAP 503**, and a **redundancy set descriptor (RSD) 505** as the primary structures for mapping a logical disk address to physical storage location(s) represented by that address.” EX1006, [0057]. “A single L2MAP handles the entire LUN 102.” EX1006, [0057]. Lubbers describes that “[t]he logical disk address specified in a request [] selects a particular entry within **LMAP 503** corresponding to a particular RSEG that in turn corresponds to 1 Mbyte address space allocated to the particular RSEG#.” EX1006, [0060]. “Each LMAP entry also includes state information about the particular RSEG, and an RSD pointer.” EX1006, [0060]. “The RSD pointer points to a specific **RSD 505** that contains metadata describing the RStore in which the corresponding RSEG exists.” EX1006, [0062].

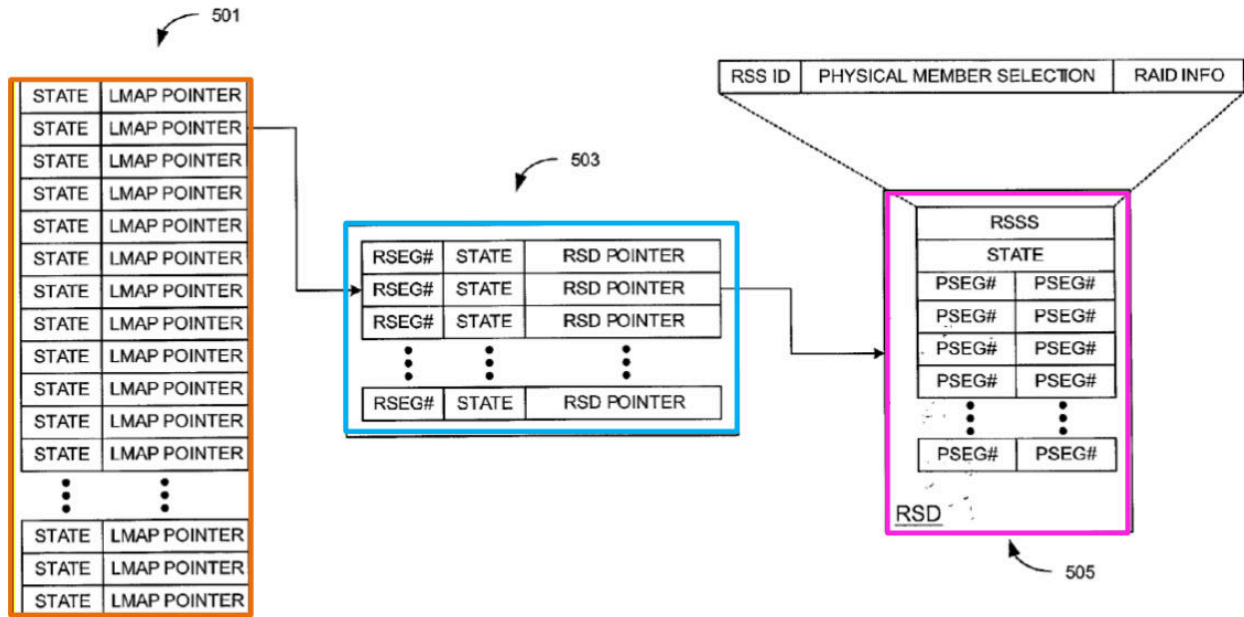


FIG. 5

EX1006, Fig. 5 (Annotated)

“[E]ach request for storage access specifies a LUN 102, and an address.”

EX1006, [0063]. “The [logical disk address] LDA specified by the request is used to index into **L2MAP 501**, which in turn points to a specific one of the **LMAPs**.”

EX1006, [0063]. “The address specified in the request is used to determine an offset into the specified **LMAP** such that a specific RSEG that corresponds to the request-specified address is returned.” EX1006, [0063]. “Once the RSEG# is known, the corresponding **RSD** is examined to identify specific PSEGs [physical storage segments] that are members of the redundancy segment, and metadata that enables a NSC 401 to generate drive specific commands to access the requested data.” EX1006, [0063]; EX1003, [124].

Lubber describes that “[t]he relationship of particular [physical] disks can be modified to provide various levels of protection and allow for construction of virtual disks with variable levels of data protection.” EX1006, [0128]. “[M]ultiple types of RAID data protection may be implemented within a single LUN 102 such that a first range of logical disk addresses (LDAs) correspond to unprotected data, and a second set of LDAs within the same LUN 102 implement RAID 5 protection.” EX1006, [0055]. “RAID information, or more generically data protection information, describes the type of data protection, if any, that is implemented in [a] particular RStore.” “Hence, the data structures implementing the memory representation must be flexible to handle this variety, yet efficient such that LUNs 102 do not require excessive data structures.” EX1006, [0055]. “The RAID information, or more generically data protection information, describes the type of data protection, if any, that is implemented in the particular RStore.” EX1006, [0062].

The NSC stores rules used to select the level of data protection. EX1006, [0065]. A POSITA would have understood and found obvious that these rules would have been based on a number of factors, including, e.g., the type of application and its priority level, to better “satisfy business objectives.” EX1006, [0055]; EX1015, [0027], [0028], [0031], [0050], [0051], [0053], [0054]; EX1003, [126]. In other words, it would have been obvious that the rules implemented by

Lubbers to select a data protection level would have been based on application-specific rules, including, e.g., the priority of the application, as indicated by its requirements for the speed and format of data access and/or for recoverability of the data in the event of an outage. *Id.*

In Lubbers, “RStores preferably implement a selected [level] of data protection.” EX1006, [0019]. The membership of an RStore “is selected in a round-robin fashion such that the first RStore will have a PSEG selected from disk 0 as a member 0, a PSEG selected from Disk 1 as member 1, and so forth,” according to a given data protection scheme. EX1006, [0079]. “Any number of RStores can be defined in this manner, and PSEGs are selected for any given RStore to meet the requirement of a given data protection scheme.” EX1006, [0079]. Accordingly, a logical address is associated with a given RStore, which in turn is structured and associated with a set of PSEGs based on the data protection requirements or rules of a given application. EX1003, [127].

Accordingly, Hashimoto-Lubbers *select[s] one or more of the virtual assets* (e.g., the storage hypervisor of the storage system selects the RStore within a virtual disk associated with the logical disk address specified in an I/O command) *for storage of the set of data* (e.g., when the I/O command is a write command, the I/O command results in storing data in the storage system) *based on an allocation of data on the virtual storage assets* (e.g., RStores are associated with PSEGs and

selected for inclusion within a LUN based on application-specific rules related to data protection level, and as a result the rules dictate how the data must be allocated on the virtual storage assets). EX1003, [128].

[1.4]

Lubber's "mapping structure that is implemented in memory of a NSC 401 [] enables translation of a request expressed in terms of a logical block address (LBA) from host 301 into a read/write command addressed to a particular portion of a physical disk 405 having the desired information." EX1006, [0054]. As discussed for [1.3], *supra*, "each request for storage access specifies a LUN 102, and an address." EX1006, [0063]. A NSC 401 of the storage system uses the data mapping tables (*e.g.*, L2MAP 501, LMAP 503, and RSD 505) "to generate drive specific commands to access the requested data." EX1006, [0063].

In Hashimoto-Lubbers, this process of address translation would have been implemented within the address translation described in Hashimoto. EX1003, [130]. Specifically, Hashimoto describes that "[w]hen the virtual storage system (0) 240 receives a disk I/O command from the virtual machine (0) 130, the virtual storage system (0) 240 obtains the virtual disk number that is the target of the access by the disk I/O command to specify the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244." EX1005, [0229]. "In this process, the storage hypervisor 214 translates the virtual block address of

the accessed virtual disk 244 to a physical block address of the physical disk drive

236.” EX1005, [0229]. This translation happens according to the method described in Lubbers, discussed above. EX1003, [130]. “Then, the storage hypervisor 214 accesses the physical disk drive 236 and reads/writes data.”

EX1005, [0229]. Thus, a POSITA would read from the broad disclosure that Lubbers is indicating that it would store to any of the physical resources, and that a POSITA would nevertheless find that obvious from the above disclosures.

EX1003, [130]. EX1003, [130].

Accordingly, Hashimoto-Lubbers *caus[es] storage of the set of data at the selected virtual storage assets on any of the physical resources* (e.g., the storage hypervisor utilizes Lubbers partitioning and address translation techniques to determine how to store data being written as part of an I/O command on the addressed virtual asset and any corresponding physical storage devices). EX1003, [131].

5. Claim 2

As discussed for [1.1.b], *supra*, “storage system 200 can be roughly divided into a physical layer, a hypervisor layer, and a virtual storage layer.” EX1005, [0104]; EX1003, [132]. “The hypervisor layer is implemented by the storage hypervisors 214 and 224...” EX1005, [0106].

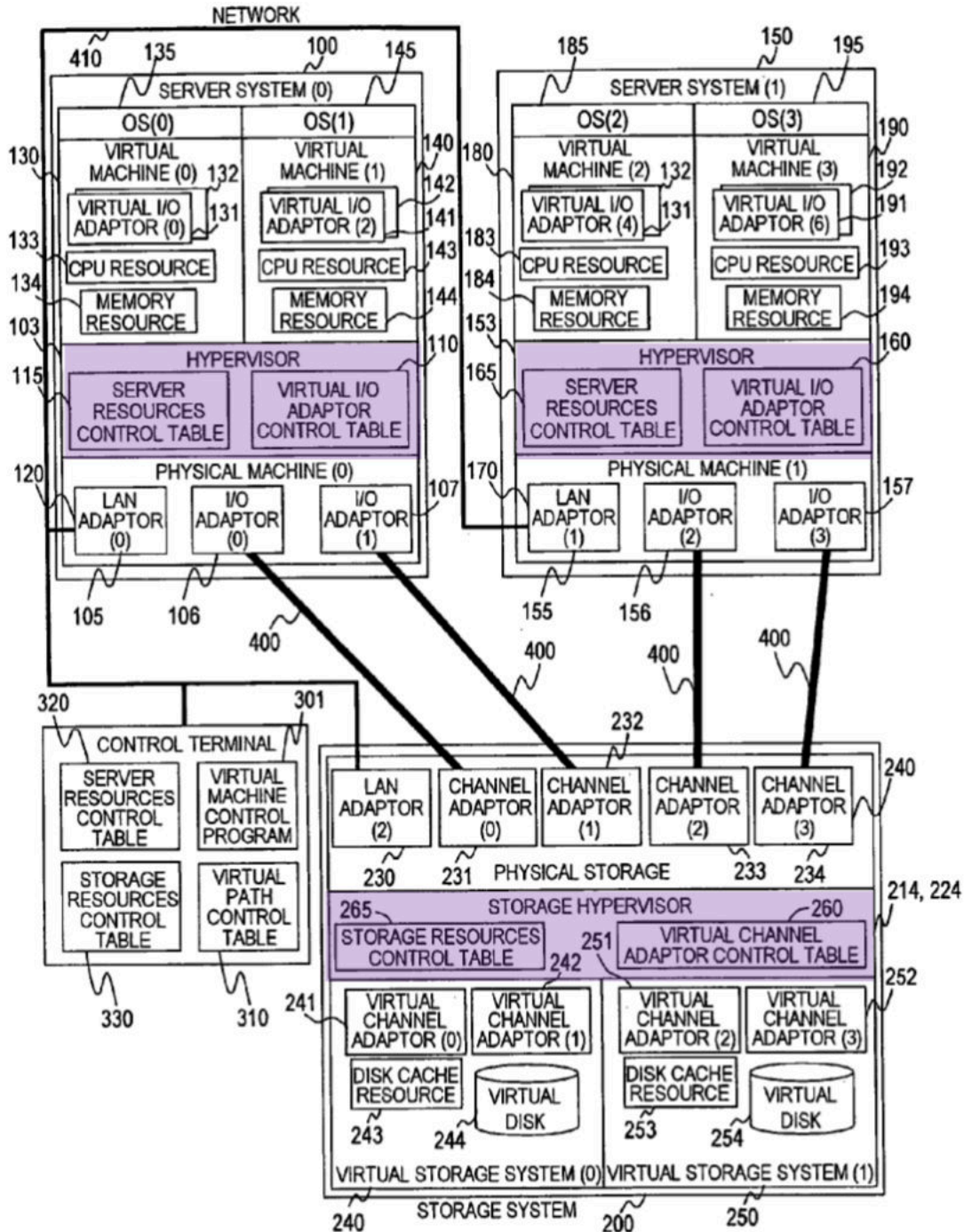


FIG. 2

EX1005, FIG. 2 (Annotated)

The storage hypervisors 214 and 224 perform similar management functions including, e.g., creating the virtual storage systems (EX1005, [0184]-[0185]),

routes I/O requests generated by the virtual machines (EX1005, [0206], [0226]-[0227]), and detecting and fixing faults in the I/O channels of the virtual machines (EX1005, [0247]-[0259]). EX1003, [133].

Accordingly, *the at least one virtual storage asset* (e.g., virtual storage systems 240 and 250) *is virtualized via storage virtualization layers running on each of the physical resources* (e.g., storage hypervisors 214 and 224 running on storage system 200). EX1003, [134].

6. Claim 3

As discussed for [1.1.a], *supra*, “[t]he hypervisor 103, implemented by the execution of processing by the CPU (0) 101, is a means that enables logical partitioning of physical resources of the server system (0) 100.” EX1006, [0059]. The “hypervisor 103 is a control program that creates logical partitions in the server system (0) 100 to create virtual machines that operate independently.” EX1006, [0060]. These virtual machines are software instantiations of computer systems. EX1005, [0004], [0062]; EX1003, [135].

Accordingly, *at least two virtual assets* (e.g., virtual machines 130 and 140) *each comprise software instantiations of computer systems.* EX1003, [136].

7. Claim 4

Hashimoto describes that the “virtual machine (0) 130 includes two virtual I/O adaptors (0) 131 and (1) 132.” EX1005, [0164]. Similarly, the “virtual storage

system (0) 240 has two virtual channel adaptors (0) 241 and (1) 242.” EX1005, [0164]. “The virtual I/O adaptor (0) 131 is connected with the virtual channel adaptor (0) 241 and the virtual I/O adaptor (1) 132 is connected with the virtual channel adaptor (1) 242.” EX1005, [0164]. “Virtual I/O adaptors are created by the hypervisor 103, 153, and virtual channel adaptors are created by the storage hypervisor 214, 224.” EX1005, [0189].

The “virtual I/O adaptor (0) of the virtual machine (0) is implemented by server resources of the physical I/O adaptor (0).” EX1005, [0143]. In other words a virtual I/O adapter is an allocation of a physical I/O adaptor to a virtual machine. EX1003, [138]. “The I/O adaptors 106 and 107 send data input/output requests and output data to the storage system 200 and receive data stored in the storage system 200.” EX1005, [0057].

It was common for this functionality to at least in part be implemented in a type of software called middleware. EX1003, [139]. For example, “applications typically use a software component that operatively connects the application to [a] database allowing the application to store and retrieve the data it needs.” EX1013, 1:20-23. “Since this software component is coupled between the application and the database, it is commonly referred to as middleware.” EX1013, 1:23-25. In fact, Hashimoto describes that “application programs [running on the virtual machines] provide database services.” EX1005, [0102]. This type of middleware

was also present in the context of virtual machines. EX1014, [0002], [0008], [0036]; EX1003, [139]. Accordingly, it would have at least been obvious that, to implement the virtual I/O adapter, Hashimoto's virtual machines would run middleware to act as a bridge between a database (e.g., stored in a virtual disk) and applications running on the virtual machine. EX1003, [139].

Accordingly, *at least one of the virtual assets* (e.g., virtual machines 130, 140, 180, and 190) *comprise middleware applications running on each of the physical resources* (e.g., at least a portion of the virtual I/O adapter functionality, which would be implemented as middleware) *to access the virtual storage assets via the storage virtualization layer* (e.g., the virtual I/O adapter accesses a corresponding virtual storage system via the storage hypervisor and channel adapter of the virtual storage system). EX1003, [140].

8. Claim 5

As discussed for [1.1.a], *supra*, Hashimoto describes that “[a] virtual machine control program 301 runs in the CPU of the control terminal 300.” EX1005, [0083]. “The virtual machine control program 301 controls virtual configuration of the computer system by using the virtual path control table 310, the server resources control table 320, and the storage resources control table 330.” EX1005, [0084]. For example, “the control terminal 300 includes a control unit (not shown) having connection information (the virtual path control table 310) that

defines connection paths between virtual machines and virtual storage systems.”

EX1005, [0084].

Further, when Hashimoto’s system creates a virtual machine, “the administrator operates the control terminal 300 to set allocation of server resources of the server system (0) 100 and gives an instruction to create a virtual machine (S111).” EX1005, [0178]. Setting the allocation is a configuration of the virtual machine in relationship to the underlying physical resources. EX1005, [0010], [0149]; EX1003, [142]. “The control terminal 300 sends the resource allocation to the server system (0) 100.” EX1005, [0178]. “The server system (0) 100 receives the virtual machine creation instruction from the control terminal 300 and then the hypervisor 103 creates a virtual machine by allocating server resources of the server system (0) 100, other than I/O adaptors, according to the received virtual machine creation instruction (S112).” EX1005, [0179].

Accordingly, *the control center application is configured to perform configuration management or forecasting services via the virtualization layer instances* (e.g., the control terminal 300 and its virtual machine control program 301 perform configuration management via the hypervisors 103 and 153 and the storage hypervisors 214 and 224). EX1003, [143].

9. Claim 7

Hashimoto’s “storage system 200 selects a single virtual disk or a plurality

of virtual disks from the virtual disks 244 and 254 and provides the virtual disk or disks to the virtual machines 100 and 150 as storage areas.” EX1005, [0115].

“The virtual disks thus selected are called logical units (LU).” EX1005, [0115].

These virtual disks are akin to the logical disks (also called logical units or LUNs) 102 described by Lubbers. EX1006, [0034]; EX1003, [144].

As discussed for [1.3], *supra*, the LUNs 102 described by Lubbers are made up of a plurality of RStores, which is itself a logical unit. EX1006, [0019], [0060]-[0061], [0065]-[0070]; EX1003, [145]. Specifically, Lubbers describes that “[a]n RStore is analogous to a RAID disk set, differing in that it comprises PSEGs rather than physical disks.” EX1006, [0068]. “An RStore is smaller than conventional RAID storage volumes, and so a given LUN 102 will comprise multiple RStores as opposed to a single RAID storage volume in conventional systems.” EX1006, [0068]. As discussed for [1.3], *supra*, Lubbers method for virtualizing a storage system selects RStores for inclusion in a LUN based on the data protection level required for a given application, and as a result the data protection level dictates how the data must be allocated on the virtual storage assets. EX1006, [0019], [0055], [0062], [0065], [0079]; EX1003, [145].

Lubbers describes that each of the LUNs has “a unique identification.” EX1006, [0035]. “Each LUN 102 is essential a contiguous range of logical addresses that can be addressed by host devices 105, 106, 107 and 109 by mapping

requests from the connection protocol used by the hosts to the uniquely identified LUN 102.” EX1006, [0035].

RStores are similarly uniquely identified within a LUN. EX1006, [0037], [0042], [0057], [0062], [0065]-[0068]; EX1003, [147]. Specifically, “[a] logical disk mapping layer maps a LDA specified in a request to a specific RStore as well as an offset within the RStore.” EX1006, [0057]. An “RSD pointer points to a specific RSD 505 that contains metadata describing the RStore in which” corresponding RSEGs exist. EX1006, [0062]. In other words, each RSD uniquely identifies a corresponding RStore. EX1006, [0057], [0062]; EX1003, [147].

Accordingly, *the at least one virtual storage asset* (e.g., the RStores—which are “logical units” and “analogous to a RAID disk set”—that are selected for inclusion in the one or more LUNs that are provided to a virtual machine) *is a logical unit with a logical unit number (LUN)* (e.g., a “uniquely identified LUN” is associated with one or more RSDs that uniquely identify the RStores that make up the LUN). EX1003, [148].

10. Claim 8

[8.1]

As discussed for [1.2], *supra*, FIG. 14 of Hashimoto “is a flowchart of a disk I/O command transmission process.” EX1005, [0222]. As part of this process, the OS of a virtual machine issues an I/O command, which is ultimately

communicated to a corresponding virtual storage system via a hypervisor and a storage hypervisor. EX1006, [0222]-[0227]. The I/O command “specif[ies] the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229].

“When the virtual storage system (0) 240 receives a disk I/O command from the virtual machine (0) 130, the virtual storage system (0) 240 obtains the virtual disk number that is the target of the access by the disk I/O command to specify the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. “In this process, the storage hypervisor 214 translates the virtual block address of the accessed virtual disk 244 to a physical block address of the physical disk drive 236.” EX1005, [0229]. That is, the I/O commands are able to access (*e.g.*, read) any data within the virtual storage system to which the issuing virtual machine has access, which would include data previously written by the virtual machine to the virtual storage system. EX1005, [0183], [0116], [0168], [0196], [0205]; EX1003[150].

Accordingly, Hashimoto-Lubbers *obtain[s] a request from a user device to obtain at least a portion of the set of stored data* (*e.g.*, an I/O request that reads data from a corresponding virtual disk). EX1003, [151].

[8.2]

As discussed for [8.1], *supra*, the OS of a virtual machine issues an I/O

command, which is ultimately communicated to a corresponding virtual storage system via a hypervisor and a storage hypervisor. EX1006, [0222]-[0227]; EX1003, [152]. The I/O command “specif[ies] the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. “In this process, the storage hypervisor 214 translates the virtual block address of the accessed virtual disk 244 to a physical block address of the physical disk drive 236.” EX1005, [0229] “Then, the storage hypervisor 214 accesses the physical disk drive 236 and reads/writes data.” EX1005, [0229].

Accordingly, Hashimoto-Lubbers *retriev[es] at least the portion of the set of stored data from the virtual storage asset* (e.g., the storage hypervisor 214 accesses the physical disk drive 236 and reads data). EX1003, [153].

[8.3]

Hashimoto describes that “[w]hen completing the data read from...the physical disk, the virtual storage system (0) 240 issues a disk I/O completion notification.” EX1005, [0231]. “The storage hypervisor 214 then adds information to the disk I/O completion notification (see FIG. 12) and transfers the disk I/O completion notification to the physical channel adaptor (0) 231 (S142).” EX1005, [0232]. FIG. 12 shows that the disk I/O layer information includes data. EX1005, FIG. 12, [0212]-[0213]. Indeed, in the case of a read I/O command, the disk I/O completion notification would naturally include the data being read.

EX1005, [0212], [0229]; EX1003, [154].

“The channel adaptor (0) 231 [of storage system 200] sends the disk I/O completion notification to the connected I/O adaptor (0) 106 (the virtual machine (0) 130) (S143).” EX1005, [0233]. “The I/O channels 400 are formed of a transmission medium that can make communication according to a protocol suitable for data transfer, e.g., the Fibre Channel protocol.” EX1005, [0086]. “In the first embodiment, the server systems 100 and 150 and the storage system 200 are connected in a one-to-one manner, but they may be connected through a network (SAN)...” EX1005, [0086].

Accordingly, Hashimoto-Lubbers *transmit[s] at least the portion of the set of stored data to the user device* (e.g., the read data is contained in a disk I/O completion notification transmitted from storage system 200 to a corresponding virtual machine on server system 100, 150) *via a network* (e.g., via a SAN interconnecting server systems 100 and 150 and the storage system 200). EX1003, [156].

B. GROUND 2

1. Hirschfeld

Hirschfeld is directed to a “virtualized logical server cloud” in which users access logical servers through a portal-like user interface without needing to know the access information of the underlying physical servers. EX1007, [0008],

[0031], [0066]; EX1003, [78].

Hirschfeld identifies a desire “to completely separate logical servers from physical servers so that there is no permanent tie between physical and logical resources,” enabling physical servers to function as a resource pool that supports logical servers and allowing a logical server to be reallocated to a different physical server “without users experiencing any change in access approach.” EX1007, [0008]; EX1003, [79].

Consistent with this objective, Hirschfeld describes a server cloud manager (“SCM”) that provides a “web page portal or other access mechanism” for remote terminals to communicate with the logical server cloud, such that “a user of any remote terminal” may “login to and access any of the logical servers” via the user interface and SCM. EX1007, [0030]. A user may employ a “local browser to access a web page” associated with the user interface, and the GUI provides a “portal or other interface” through which the user may access one or more logical servers based on permission (e.g., by “point and click” selection of a logical server graphic). *Id.*, [0031]; EX1003, [80]-[81].

2. The Hashimoto-Lubbers-Hirschfeld Combination

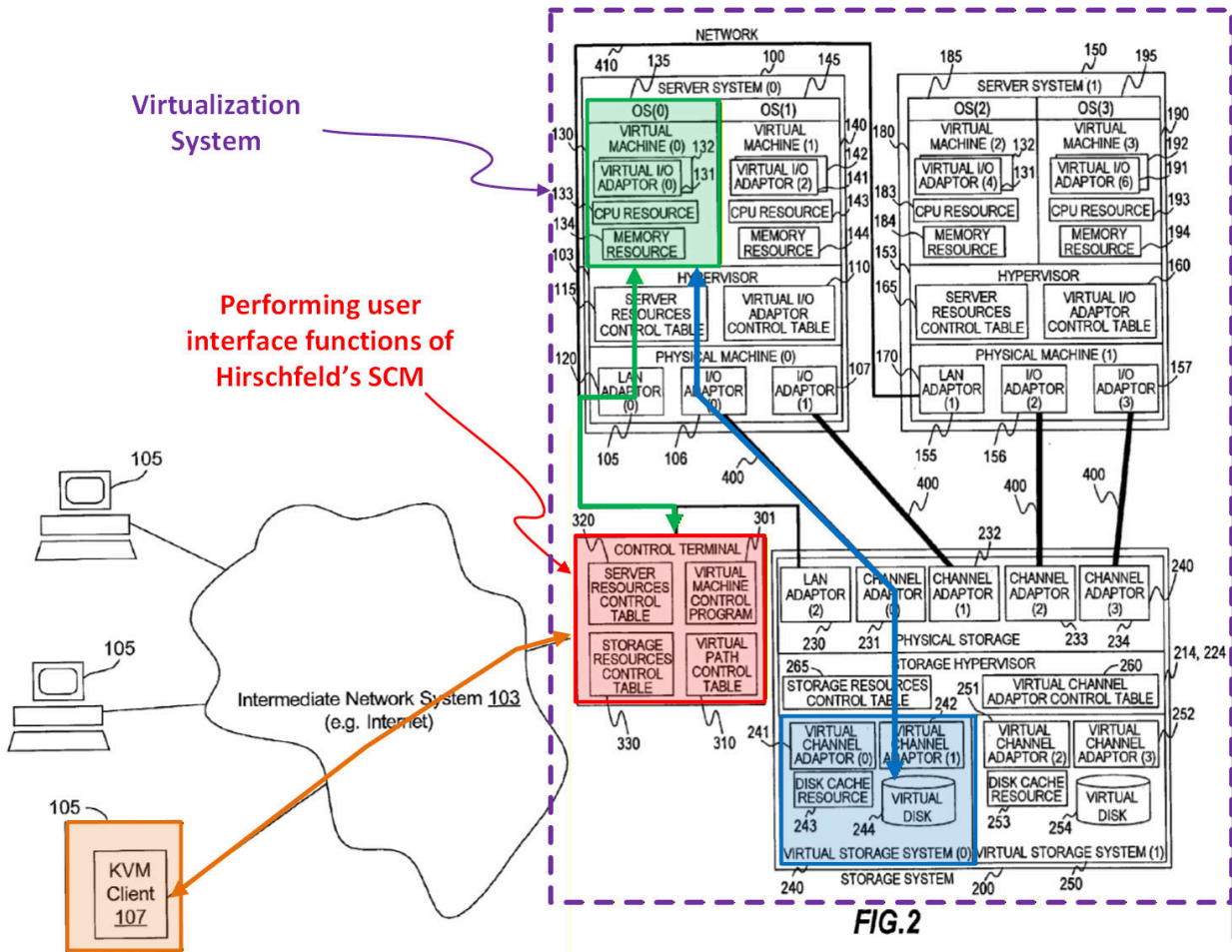
Like Hirschfeld, Hashimoto describes the partitioning of physical resources into virtual machines. EX1005, [0059]-[0060], [0069]-[0070]; EX1007, [0028], [0039]; EX1003, [96]. And like Hirschfeld’s SCM, Hashimoto describes a distinct

control terminal 300 that controls the overall virtualization system. EX1005, [0052]; EX1007, [0040]; EX1003, [96]. However, unlike Hirschfeld, Hashimoto does not explicitly describe a mechanism by which an end user would interface with its system to utilize the virtual machines and virtual storage. EX1003, [96]. Rather, Hashimoto simply notes that the virtual machines execute “application programs provide database services, Web services, etc. to clients (not shown) connected to the server system (0) 100.” EX1005, [0102], [0294], [0301]. How user machines (*i.e.*, “clients”) connect to the server system is left to the POSITA. EX1003, [96].

Hirschfeld provides one obvious solution for how user’s would have interfaced with the Hashimoto-Lubbers virtualization system. EX1003, [97]. Hirschfeld expressly identifies that, in conventional virtualization deployments, the user “had to be aware of and have specific information about the underlying physical server” (*e.g.*, IP address/server name and port number) to access logical servers, and that this creates usability and operational issues (*e.g.*, users needing distinct access information for each logical server, and needing updated information when servers are moved). EX1007, [0008]. Hirschfeld further describes that it is “desired to completely separate logical servers from physical servers” to enable reallocation of logical servers to different physical servers “without users experiencing any change in access approach.” *Id.*, [0008]; EX1003,

[97].

In seeking techniques that address user-access and mapping issues in a virtualized environment like Hashimoto, a POSITA would have found Hirschfeld, which provides a “web page portal or other access mechanism” through an SCM and user interface that enables remote terminals to login to and access logical servers in a logical-server cloud via a browser-based portal. EX1007, [003]-[0031]. Hirschfeld describes that the SCM maintains a user database incorporating a “mapping of users, the logical servers and the physical servers,” including link information mapping each logical server to a physical server, such that the SCM “automatically accesses the appropriate physical server corresponding to the selected logical server,” and the user does not need to supply (or even be aware of) physical-server access information. *Id.*, [0032]; EX1003, [98].



Amalgamation of EX1005 Fig. 2 and EX1007 Fig. 1

A POSITA would have viewed Hirschfeld's SCM/portal architecture as a compatible enhancement to Hashimoto-Lubbers's control console 300 because both include network-accessible control-plane interfaces for interacting with virtualized server instances. *Compare* EX1005, [0052] ("The control terminal 300 manages information for controlling the computer system and controls operations of the entire computer system.") *with* EX1007, [0040] ("the SCM 102 is implemented on a separate physical server or administrator computer," and

“includes SCM software to manage the logical servers 106 and the physical servers 108”); EX1003, [99].

Accordingly, a POSITA would have been motivated to leverage Hirschfeld’s SCM-managed mapping and portal-based access mechanisms in Hashimoto-Lubber’s virtualization system to provide a user-facing experience in which users interact through a browser-based console/portal while the system maintains and uses the mapping between logical server instances (*e.g.*, provisioned VMs/logical servers) and the underlying physical servers. EX1007, [0031]-[0033] ; EX1003, [100]. By implementing the user interface functions of Hirschfeld’s SCM with Hashimoto’s control terminal (a “SCM/control terminal”), Hashimoto-Lubbers-Hirschfeld advances Hirschfeld’s stated objective of removing the requirement that users know physical-server-specific access information, while remaining consistent with Hashimoto-Lubber’s architecture in which a control terminal 300 provides an interface through which to “manage the computer system,” and is connected to the server systems 100 and 150 and storage system 200 through a network 410. EX1005, [0083]-[0087], [0310]-[0311], [0321]-[0322]; MS-1006, [0030]-[0033], [0066]; EX1003, [100]. A benefit of this combination would have been a simpler and more efficient user experience with the virtualization system. *Id.*

Finally, a POSITA would have reasonably expected to succeed in combining the teachings of Hashimoto, Lubbers, and Hirschfeld in the manner discussed

above. All references address virtualized computing environments in which centralized control-plane components coordinate provisioning and/or access to logical server instances, and describe interfaces for system control. EX1005, [0083]-[0087], [0310]-[0311], [0321]-[0322]; EX1006, [0040]; EX1007, [0030]-[0033]; EX1003, [101]. The combination involves applying Hirschfeld's SCM portal access framework to Hashimoto-Lubbers's control terminal, which already provides a similar system control function as the SCM. EX1007, [0029]-[0030]; EX1005, [0052], [0083]-[0087]; EX1003, [101]. A POSITA would have understood this as combining known elements according to known methods to yield predictable results. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 415-21 (2007); EX1003, [101].

3. Claim 9

[9.pre]

See [1.pre], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [157].

[9.1]

As discussed for [1.1.a], *supra*, Hashimoto describes a computer system that includes server systems 100 and 150, at least one storage system 200, and a control terminal 300, all of which are connected via a network 410. EX1005, [0052], [0371]-[0375]. "A virtual machine control program 301 runs in the CPU of the control terminal 300." EX1005, [0083]. "The virtual machine control program

301 controls virtual configuration of the computer system by using the virtual path control table 310, the server resources control table 320, and the storage resources control table 330.” EX1005, [0084]. “The control terminal 300 manages information for controlling the computer system and controls operations of the entire computer system.” EX1005, [0052]. This virtual machine control program 301 running on control terminal 300 performs many of the same control function as Hirschfeld’s SCM. EX1007, [0030]; EX1003, [158]. Thus, in Hashimoto-Lubbers-Hirschfeld, an SCM/control terminal would perform the described control functions. EX1003, [158].

Accordingly, Hashimoto-Lubbers-Hirschfeld includes *a control computer running a control center application* (e.g., SCM/control terminal running a virtual machine control program). EX1003, [159].

[9.2.a]

See [1.1.a], *supra*, which is unaffected by the addition of Hirschfeld.
EX1003, [160].

[9.2.b]

See [1.1.b], *supra*, which is unaffected by the addition of Hirschfeld.
EX1003, [161].

[9.2.c]

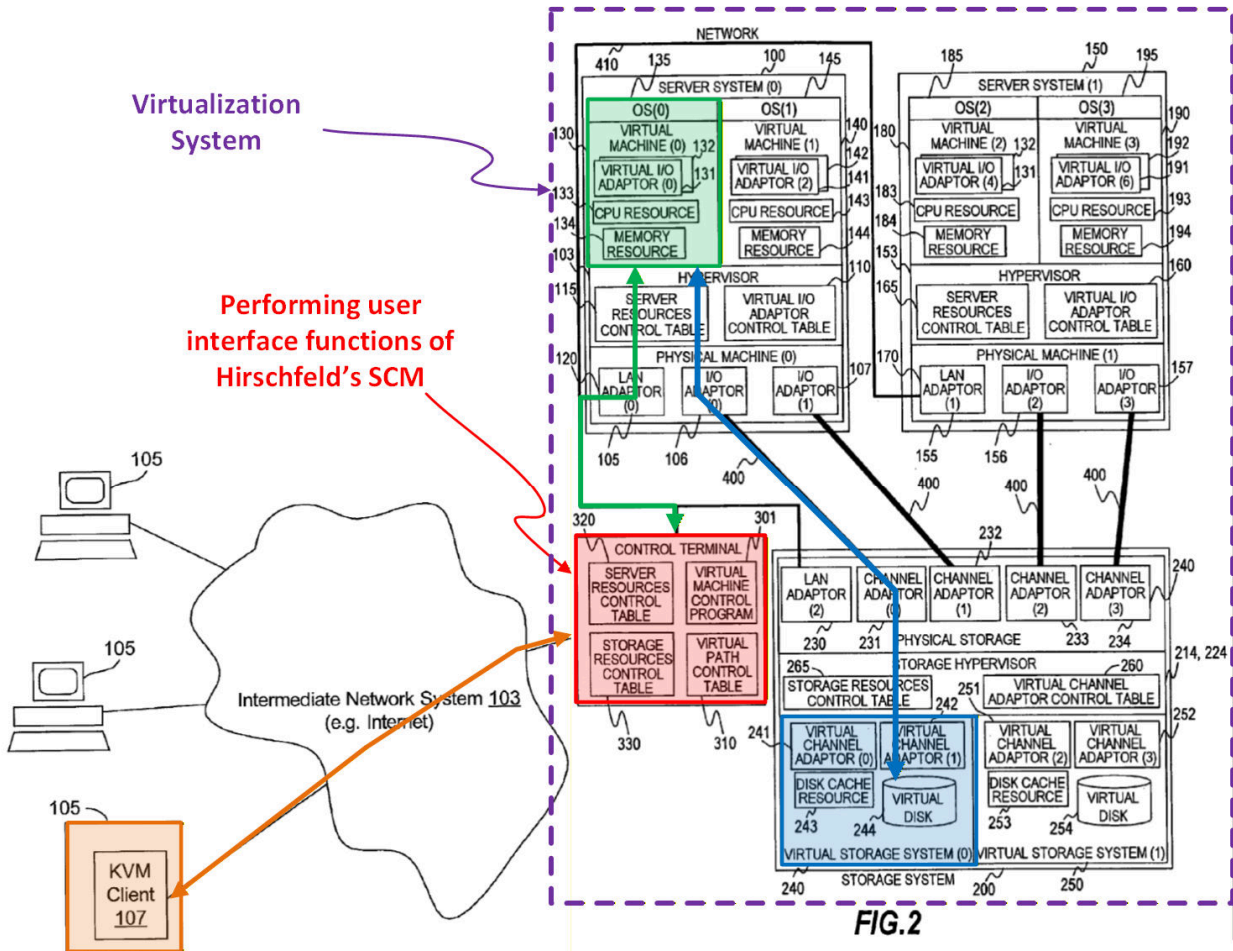
See [1.1.c], *supra*, which is unaffected by the addition of Hirschfeld.
EX1003, [162].

[9.3]

As discussed for [8.1], *supra*, FIG. 14 of Hashimoto “is a flowchart of a disk I/O command transmission process.” EX1005, [0222]. As part of this process, the OS of a virtual machine issues an I/O command, which is ultimately communicated to a corresponding virtual storage system via a hypervisor and a storage hypervisor. EX1006, [0222]-[0227]. The I/O command “specif[ies] the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229].

As discussed for [9.1], *supra*, in Hashimoto-Lubbers-Hirschfeld, the virtual machine is controlled via the SCM/control terminal. EX1005, [0030], [0052], [0083]-[0084], [0222]-[0227]; EX1007, [0030]-[0032]; EX1003, [164]. For example, Hirschfeld describes that “the SCM 102 that interfaces and controls each of the logical servers 106 and the bank of physical servers 108.” EX1007, [0030]. Additionally, “the SCM 102 includes a user interface (I/F) 104 that provides a web page portal or other access mechanism to facilitate communication with the remote terminals 105 via the network system 103.” EX1007, [0030]. “In this manner, a user of any remote terminal 105 may login to and access any of the logical servers 106 of the virtualized logical server cloud 101 via the user interface 104 and SCM 102 as though logging into a server directly or via a local area network (LAN).” EX1007, [0030]. “The graphical user interface (GUI) of the user interface 104

provides the mechanism (portal or other interface) through which the user may access one or more logical servers 106 based on permission,” and “control a selected logical server.” EX1007, [0031].



Amalgamation of EX1005 Fig. 2 and EX1007 Fig. 1

Thus, as shown in the above diagram of Hashimoto-Lubbers-Hirschfeld, a user would control one or more of the virtual machines 130, 140, 180, and 190 via a remote terminal in communication with the combination's SCM/control terminal, which in turn passes the user's commands onto the server system(s) hosting the

virtual machine(s) to which the user has access. EX1007, [0029]-[0032]; EX1003, [165]. For example, Hashimoto describes that the virtual machines run “application programs [that] provide...services...to clients (not shown) connected to the server system (0) 100.” EX1005, [0102]. In Hashimoto-Lubbers-Hirschfeld, the user would interact with and control these application programs via the SCM/control terminal. EX1007, [0029]-[0032]; EX1003, [165]. Hashimoto describes that the “application programs provide database services, Web services, etc.” EX1005, [0102]. “[F]rom web servers, database servers, spreadsheets and word-processing programs, to games and browsers,” there are many well-known application programs run on virtual machines. EX1012, 11:50-53. In the case of a spreadsheet or word-processing program, it was well-known that a user could open a spreadsheet or document previously stored in the storage system. EX1003, [165]. Similarly, in the case of a game, it was well-known that starting the game requires loading the components of the game stored in an associated storage system. EX1003, [165].

Accordingly, *the control center application* (e.g., the control application run by the SCM/control terminal of the combination implementing the user interface functionality of Hirschfeld’s SCM) *is configured to obtain a request from a user device for data stored at any of the virtual storage assets* (e.g., a user provides commands—including, e.g., “open” commands that retrieve previously stored

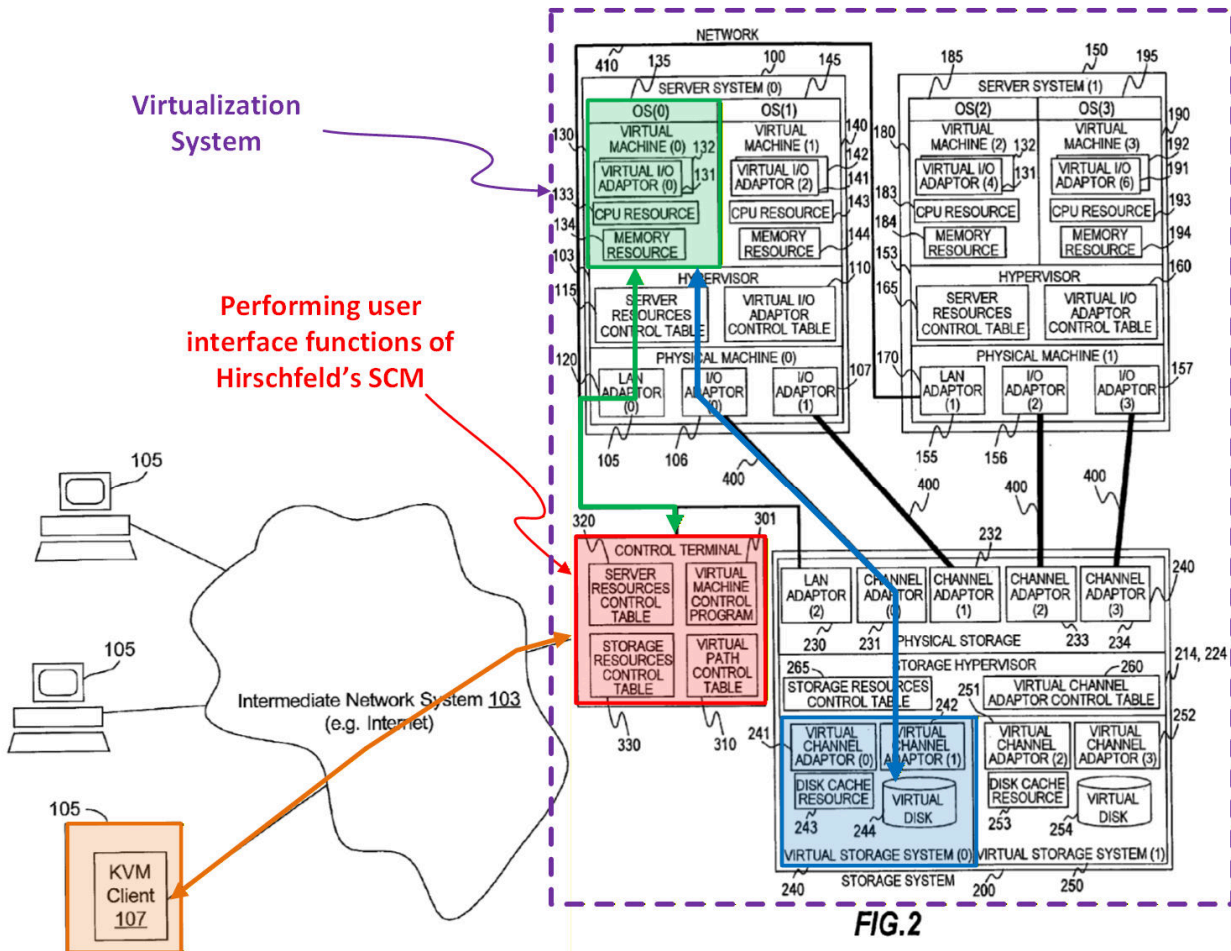
documents from a virtual storage system—through their remote terminal that are transmitted to the SCM/control terminal, which then pass these commands to the virtual machine with which the user is interfacing). EX1003, [166].

[9.4]

As discussed for [9.3], *supra*, Hirschfeld describes that “a user of any remote terminal 105 may login to and access any of the logical servers 106 of the virtualized logical server cloud 101 via the user interface 104 and SCM 102 as though logging into a server directly or via a local area network (LAN).” EX1007, [0030]. “[A] terminal 105 executes a keyboard/video/mouse (KVM) client 107 that transmits keyboard and mouse commands to a selected logical server 106 and that receives video and audio information from the selected logical server 106.” EX1007, [0031]. In other words, the output associated with commands issued by the user are provided via the SCM 102. EX1003, [167].

As discussed for [8.2], *supra*, Hashimoto describes that the OS of a virtual machine issues an I/O command, which is ultimately communicated to a corresponding virtual storage system via a hypervisor and a storage hypervisor. EX1006, [0222]-[0227]; EX1003, [168]. The I/O command “specif[ies] the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. “In this process, the storage hypervisor 214 translates the virtual block address of the accessed virtual disk 244 to a physical block address of

the physical disk drive 236.” EX1005, [0229] “Then, the storage hypervisor 214 accesses the physical disk drive 236 and reads/writes data.” EX1005, [0229].



Amalgamation of EX1005 Fig. 2 and EX1007 Fig. 1

Thus, as shown in the above diagram of Hashimoto-Lubbers-Hirschfeld, a user issues a command from their remote terminal 105, which **transmits the command to the combination's SCM/control terminal**. EX1007, [0030]-[0031]; EX1003, [169]. As discussed for [9.3], *supra*, this command may be with respect to an application program being run by the virtual machine with which the

user is interacting. EX1005, [0105]; EX1007, [0029]-[0032]; EX1003, [169]. The **SCM/control terminal passes the user's command to a virtual machine 130, 140, 180, or 190** via network 410. EX1007, [0030]-[0031]; EX1005, [0052], [0083]-[0087]; EX1003, [169]. When the user's command requires the application program to access data stored in memory (e.g., a request to "open" a spreadsheet or document), the application program interfaces with the virtual machines OS to access the stored data. EX1005, [0101]-[0102]; EX1003, [169].

To access stored data, **the OS of the virtual machine 130 transmits the I/O command to a corresponding virtual storage system 240 or 250**, consistent with the teachings of Hashimoto. EX1005, Fig. 14, [0222]-[0227]; EX1003, [170]. As discussed for [8.3], *supra*, in the case of a read I/O command, a disk I/O completion notification that would naturally include the data being read (e.g., the spreadsheet or document being opened by an application program) would be returned to the virtual machine 130 that issued the I/O command. EX1005, Fig. 15, [0212]-[0213], [0228]-[0235]; EX1003, [170].

Accordingly, ***the control center application*** (e.g., the control application run by the SCM/control terminal of the combination) ***is configured to retrieve the data from one or more virtual storage assets that are storing the data*** (e.g., the SCM/control terminal of the combination passes a user's application program commands to the virtual machine with which the user is interfacing, which in turn

transmits corresponding I/O commands to a corresponding virtual storage system, and display of the result of the I/O commands is sent back to the SCM/control terminal by the virtual machine on behalf of the application program). EX1003, [171].

[9.5]

As discussed for [9.4], *supra*, Hirschfeld describes that “a user of any remote terminal 105 may login to and access any of the logical servers 106 of the virtualized logical server cloud 101 via the user interface 104 and SCM 102 as though logging into a server directly or via a local area network (LAN).” EX1007, [0030]. “[A] terminal 105 executes a keyboard/video/mouse (KVM) client 107 that transmits keyboard and mouse commands to a selected logical server 106 and that receives video and audio information from the selected logical server 106.” EX1007, [0031]. In other words, the output associated with the commands issued by the user to the application program running on a virtual machine is sent back to the user’s terminal 105 via SCM 102. EX1003, [172].

Accordingly, in Hashimoto-Lubbers-Hirschfeld, *the control center application is configured to transmit the retrieved data to the user device.* EX1003, [173].

4. Claim 10

See [2], *supra*, which is unaffected by the addition of Hirschfeld. EX1003,

[174].

5. Claim 11

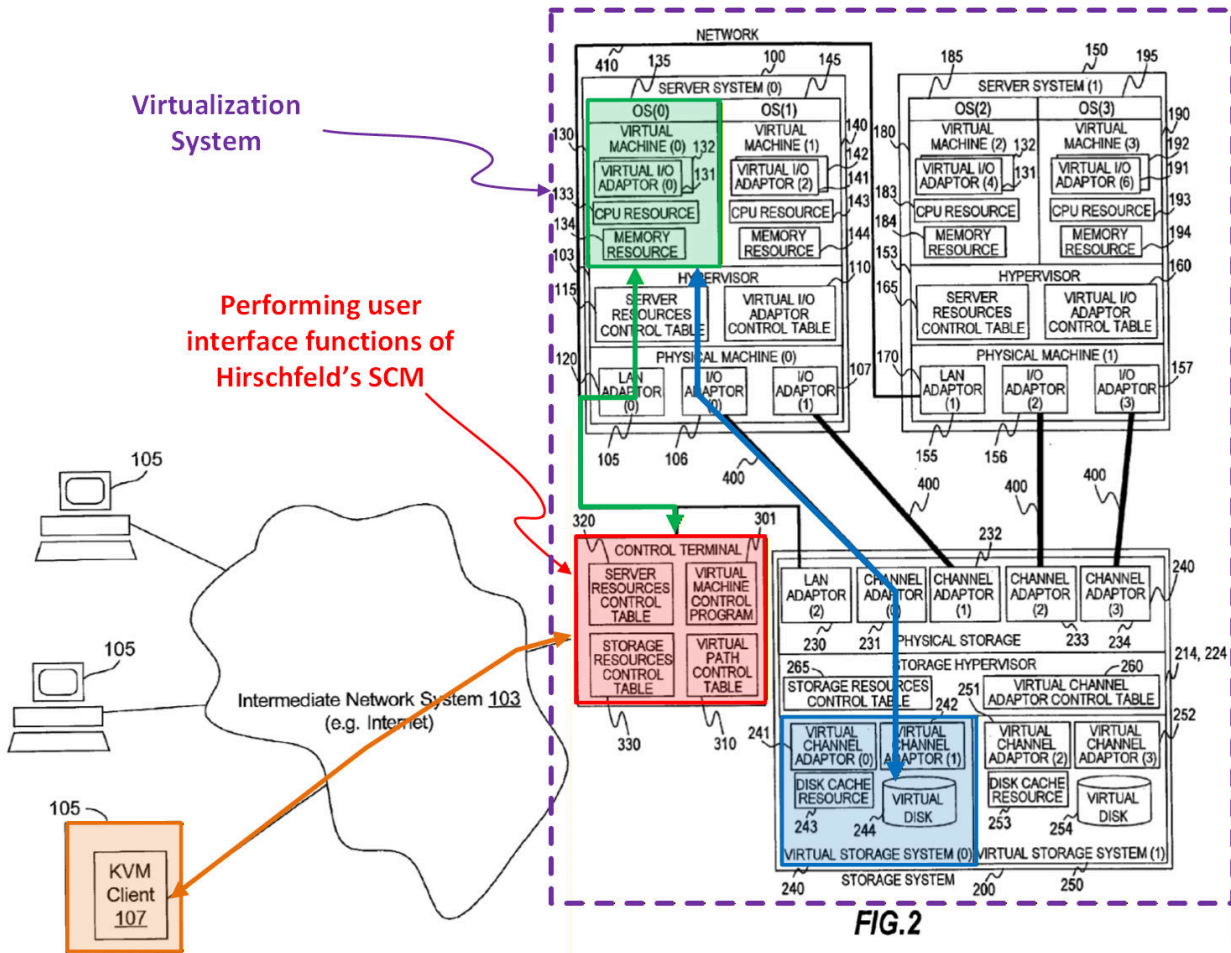
[11.1]

Initially, the term “the data to be stored” does not have antecedent basis in claim 9. EX1001, 24:55-25:3; EX1003, [175]. Claim 9 recites “a request...for data stored” and “retriev[ing] the data from one or more virtual storage assets.” EX1001, 24:66-7:2. This is akin to a read I/O command. EX1005, [0229]-[0231]. On the other hand, “obtain[ing]...data to be store” and “stor[ing] the data across any of the virtual storage assets” is akin to a write I/O command. EX1001, 25:7-12; EX1005, [0229]-[0231]; EX1003, [175]. For purposes of this IPR only, where §112 grounds are unavailable, the Petition assumes that [11.1] relates to a separate request to “store...data” and not the “request from a user device for data stored” recited in [9.3]. EX1003, [175].

As discussed for [9.4], *supra*, Hirschfeld describes that “a user of any remote terminal 105 may login to and access any of the logical servers 106 of the virtualized logical server cloud 101 via the user interface 104 and SCM 102 as though logging into a server directly or via a local area network (LAN).” EX1007, [0030]. “[A] terminal 105 executes a keyboard/video/mouse (KVM) client 107 that transmits keyboard and mouse commands to a selected logical server 106 and that receives video and audio information from the selected logical server 106.”

EX1007, [0031]. In other words, the output associated with the commands issued by the user to the application program running on a virtual machine is sent back to the user's terminal 105 via SCM 102. EX1003, [176].

As discussed for [8.2], *supra*, Hashimoto describes that the OS of a virtual machine issues an I/O command, which is ultimately communicated to a corresponding virtual storage system via a hypervisor and a storage hypervisor. EX1006, [0222]-[0227]; EX1003, [177]. The I/O command “specif[ies] the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. “In this process, the storage hypervisor 214 translates the virtual block address of the accessed virtual disk 244 to a physical block address of the physical disk drive 236.” EX1005, [0229] “Then, the storage hypervisor 214 accesses the physical disk drive 236 and reads/writes data.” EX1005, [0229].



Amalgamation of EX1005 Fig. 2 and EX1007 Fig. 1

Thus, as shown in the above diagram of Hashimoto-Lubbers-Hirschfeld, a user would control one or more of the virtual machines 130, 140, 180, and 190 via **a remote terminal in communication with the combination's SCM/control terminal**, which in turn **passes the user's commands onto the server system(s) hosting the virtual machine(s) to which the user has access**. EX1007, [0029]-[0032]; EX1003, [178]. As discussed for [9.3], *supra*, Hashimoto describes that the virtual machines run “application programs [that] provide...services...to

clients (not shown) connected to the server system (0) 100.” EX1005, [0102]. In Hashimoto-Lubbers-Hirschfeld, the user would interact with and control these application programs via the SCM/control terminal. EX1007, [0029]-[0032]; EX1003, [178]. In the case of a spreadsheet or word-processing program, it was well-known that a user could enter data into a document or spreadsheet file and “save” the file, which would then be stored on an associated storage system. EX1003, [178].

The SCM/control terminal passes the user’s command to a virtual machine 130 via network 410. EX1007, [0030]-[0031]; EX1005, [0052], [0083]-[0087]; EX1003, [179]. When the user’s command requires the application program to store data in memory (*e.g.*, a request to “save” a spreadsheet or document file created through user input), the application program interfaces with the virtual machines OS to store the data. EX1005, [0101]-[0102]; EX1003, [179]. Specifically, **the OS of the virtual machine 130 transmits a write I/O command to a corresponding virtual storage system 240 or 250**, consistent with the teachings of Hashimoto. EX1005, Fig. 14, [0222]-[0227]; EX1003, [179].

Accordingly, Hashimoto-Lubbers-Hirschfeld *obtain[s] the data to be stored at any of the virtual storage assets* (*e.g.*, the SCM/control terminal receives data input by a user with the KVM client 107 of remote terminal 105). EX1003, [180].

[11.2]

As discussed for [9.4], *supra*, Hashimoto describes that “[w]hen the virtual storage system (0) 240 receives a disk I/O command from the virtual machine (0) 130, the virtual storage system (0) 240 obtains the virtual disk number that is the target of the access by the disk I/O command to specify the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. “In this process, the storage hypervisor 214 translates the virtual block address of the accessed virtual disk 244 to a physical block address of the physical disk drive 236.” EX1005, [0229].

To translate the address, Lubber teaches that a “mapping structure...enables translation of a request expressed in terms of a logical block address (LBA) from host 301 into a read/write command addressed to a particular portion of a physical disk 405 having the desired information.” EX1006, [0054]. As discussed for [1.3], *supra*, “each request for storage access specifies a LUN 102, and an address.” EX1006, [0063]. In Hashimoto-Lubbers-Hirschfeld, the hypervisor 214 of the corresponding virtual storage system uses storage mapping tables (e.g., L2MAP 501, LMAP 503, and RSD 505) “to generate drive specific commands to access the requested data.” EX1006, [0063]; EX1003, [182].

Thus, in the example discussed for [11.1], *supra*, the word processing application and OS being run by the virtual machine would have issued an I/O

command to write the document input by the user into the virtual storage system.

EX1003, [183].

Accordingly, Hashimoto-Lubbers-Hirschfeld *store[s] the data across any of the virtual storage assets running on the physical resources* (e.g., the SCM/control terminal receives an input associated with a save command from a user via the KVM client 107 of remote terminal 105, and forwards this input to the virtual machine, which in turn issues a write I/O command to the virtual storage system). EX1003, [184].

6. Claim 12

See [3], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [185].

7. Claim 13

See [4], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [186].

8. Claim 14

See [5], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [187].

9. Claim 15

Hashimoto describes that “[a]pplication programs run on the OSs 135 and 145” of virtual machines 130 and 140. EX1005, [0101]-[0102]. “These application programs provide database services, Web services, etc. to clients (not

shown) connected to the server system (0) 100.” EX1005, [0102]. Hashimoto teaches that “when a virtual machine is created for the first time in the server system (0) 100, it is necessary to create a virtual storage system that the virtual machine accesses.” EX1005, [0183]. In other words, the virtual storage system is the resource a virtual machine uses for storage, and thus the virtual storage system is an obvious place that the virtual machine would store one or more databases associated with the “database services” application program run by the virtual machine. EX1005, [0101]-[0102], [0183]; EX1003, [188].

Accordingly, in Hashimoto-Lubbers-Hirschfeld, *the virtual storage assets comprise databases*. EX1003, [189].

10. Claim 16

See [7], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [190].

11. Claim 17

Hashimoto describes that “[t]he administrator can manage the computer system by operating the control terminal 300.” EX1003, [0310], [0321], [0052] (“The control terminal 300 manages information for controlling the computer system and controls operations of the entire computer system”). For example, “[t]he administrator specifies allocation of the storage resources by means of the control terminal 300 and the storage resources control table 265 is generated as per

the administrator's setting." EX1005, [0160]. These allocation instructions are communicated to the server and storage systems via network 410. EX1005, [0184], [0085]; EX1003, [191].

Accordingly, *the virtualization system is further configured to receive, over a network by a user, one or more instructions to configure the virtual storage assets* (e.g., the storage system 200 receives control instructions to configure virtual storage systems 240 and 250 over network 410 from control terminal 300 as a result of input from an administrator—a type of user). EX1003, [192].

12. Claim 18

[18.pre]

See [1.pre], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [193].

[18.1]

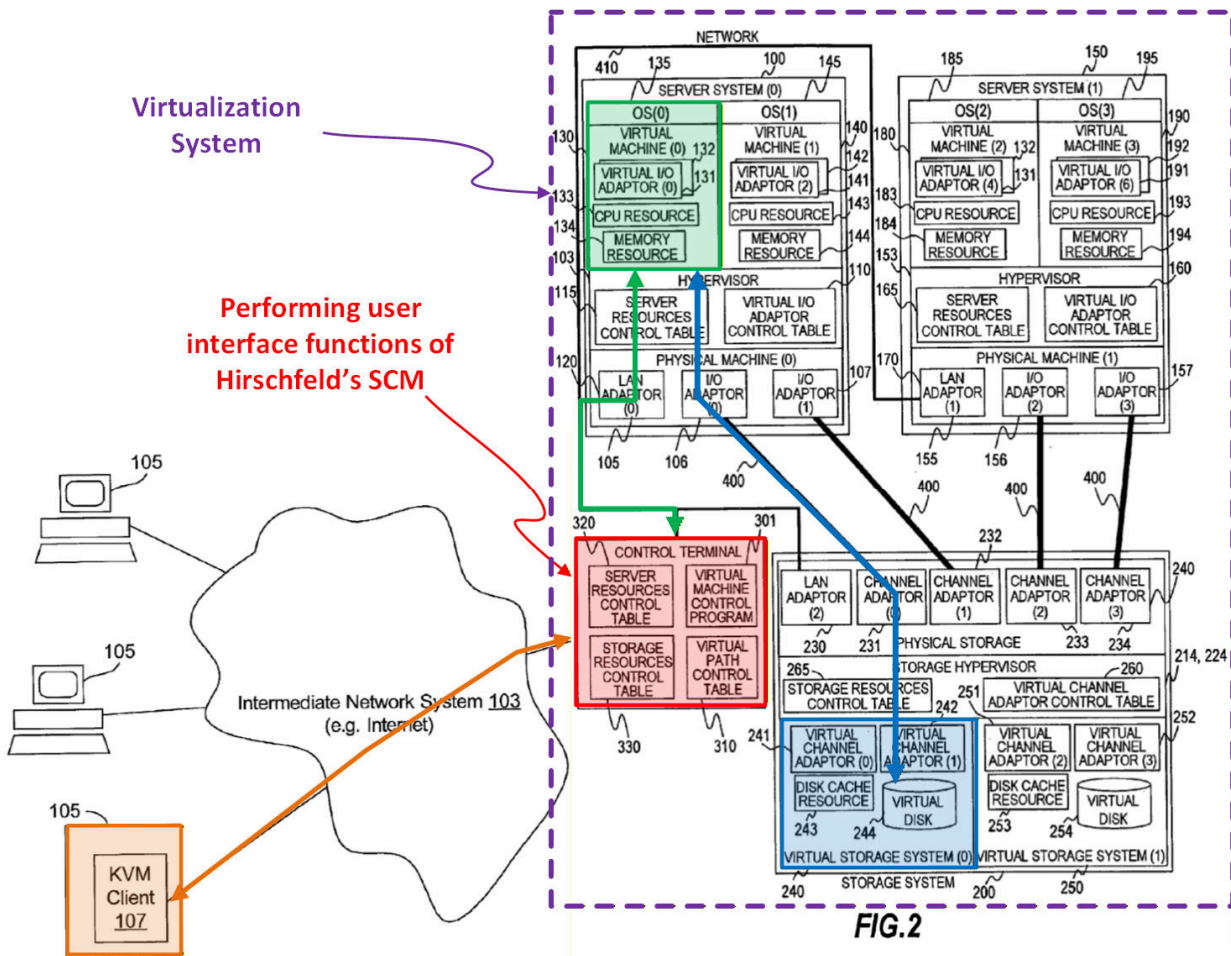
Hashimoto-Lubbers-Hirschfeld includes *a virtualization system that includes the control center application, the one or more physical resources, and one or more virtual assets on each of the physical resources* for substantially the same reasons discussed for [1.1.a], *supra*, which are unaffected by the addition of Hirschfeld. EX1003, [194].

As discussed for [9.3], *supra*, in Hashimoto-Lubbers-Hirschfeld, control of the virtual machine is controlled via the SCM/virtual machine. EX1005, [0030], [0052], [0083]-[0084], [0222]-[0227]; EX1007, [0030]-[0032]; EX1003, [195].

For example, Hirschfeld describes that “the SCM 102 that interfaces and controls each of the logical servers 106 and the bank of physical servers 108.” EX1007, [0030]. Additionally, “the SCM 102 includes a user interface (I/F) 104 that provides a web page portal or other access mechanism to facilitate communication with the remote terminals 105 via the network system 103.” EX1007, [0030]. “In this manner, a user of any remote terminal 105 may login to and access any of the logical servers 106 of the virtualized logical server cloud 101 via the user interface 104 and SCM 102 as though logging into a server directly or via a local area network (LAN).” EX1007, [0030]. “The graphical user interface (GUI) of the user interface 104 provides the mechanism (portal or other interface) through which the user may access one or more logical servers 106 based on permission,” and “control a selected logical server.” EX1007, [0031]. Thus, in Hashimoto-Lubbers-Hirschfeld, a user would control one or more of the virtual machines 130, 140, 180, and 190 via a remote terminal in communication with the combination’s SCM/control terminal, which in turn passes the user’s commands onto the server system(s) hosting the virtual machine(s) to which the user has access. EX1003, [195].

As discussed for [8.2], *supra*, Hashimoto describes that the OS of a virtual machine issues an I/O command, which is ultimately communicated to a corresponding virtual storage system via a hypervisor and a storage hypervisor.

EX1006, [0222]-[0227]; EX1003, [196]. The I/O command “specif[ies] the accessed virtual disk and reads/writes data to/from the corresponding virtual disk 244.” EX1005, [0229]. “In this process, the storage hypervisor 214 translates the virtual block address of the accessed virtual disk 244 to a physical block address of the physical disk drive 236.” EX1005, [0229] “Then, the storage hypervisor 214 accesses the physical disk drive 236 and reads/writes data.” EX1005, [0229].



Amalgamation of EX1005 Fig. 2 and EX1007 Fig. 1

Thus, as shown in the above diagram of Hashimoto-Lubbers-Hirschfeld, a

user issues a command from their remote terminal 105, which **transmits the command to the combination's SCM/control terminal**. EX1007, [0030]-[0031]; EX1003, [197]. As discussed for [9.3], *supra*, this command may be with respect to an application program being run by the virtual machine with which the user is interacting. EX1005, [0105]; EX1007, [0029]-[0032]; EX1003, [197]. In the case of a spreadsheet or word-processing program, it was well-known that a user could enter data into a document or spreadsheet file and “save” the file, which would then be stored on an associated storage system. EX1003, [197].

The SCM/control terminal passes the user's command to a virtual machine 130, 140, 180, or 190 via network 410. EX1007, [0030]-[0031]; EX1005, [0052], [0083]-[0087]; EX1003, [198]. When the user's command requires the application program to store data in memory (*e.g.*, a request to “save” a spreadsheet or document file created through user input), the application program interfaces with the virtual machines OS to store the data. EX1005, [0101]-[0102]; EX1003, [198]. Specifically, **the OS of the virtual machine 130 transmits a write I/O command to a corresponding virtual storage system 240 or 250**, consistent with the teachings of Hashimoto. EX1005, Fig. 14, [0222]-[0227]; EX1003, [198].

Accordingly, Hashimoto-Lubbers-Hirschfeld *receiv[es] a request to store data, at a control center application* (*e.g.*, the SCM/control terminal receives an

I/O command from the KVM client of a remote terminal) *in communication with one or more physical resources that are part of [the] virtualization system* (e.g., the SCM/control terminal is connected to one or more server systems and one or more storage systems). EX1003, [199].

[18.2]

See [1.3], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [200].

[18.3]

See [11.2], *supra*. EX1003, [201].

[18.4]

See [2], *supra*, which is unaffected by the addition of Hirschfeld. EX1003, [202].

13. Claim 19

See [1.1.a], [1.1.b], *supra*, which are unaffected by the addition of Hirschfeld. EX1003, [203].

14. Claim 20

See [3], [15], *supra*. EX1003, [204].

C. GROUND 3

1. The Hashimoto-Lubbers-Khandekar Combination

Hasimoto describes that “[a]pplication programs run in the server system (0) 100 and the server system (1) 150.” EX1005, [0052]. These “[a]pplication

programs run on the OSs 135 and 145” of the virtual machines. EX1005, [0102].

However, Hashimoto does not describe how the applications are distributed to the virtual machines or otherwise managed within the virtualization environment.

EX1003, [102].

Khandekar provides details about at least one obvious way in which Hashimoto’s virtualization system would distribute and manage the application programs of the virtual machines. EX1003, [102]. Khandekar teaches “a method and system...for creating a virtualized computer system based on input information identifying a desired computer configuration.” EX1012, 4:30-35; EX1005, FIG. 10, [0177]-[0180] (describing creation of a virtual machine); EX1003, [84]. “Of course, most computers are intended to run various applications, and VMs are usually no exception.” EX1012, 6:24-27. Khandekar teaches that “the requester specifies various components of the desired VM, including the characteristics of a desired operating system (OS), of a desired hardware platform, of desired applications, or of any combination of these components.” EX1012, 4:42-46. “Once a VM has been configured and deployed,” Khandekar “alter[s] its configuration, for example by upgrading the software and/or hardware, installing different applications, etc.” EX1012, 4:48-51.

“The deployed VMs can be monitored regularly so that any unexpected event, like the crash of the operating system in the VM, can be reported to the user

or administrator.” EX1012, 13:44-46. “Also, maintenance tasks like upgrade of software and hardware[,]. . . installation of service packs or hot fixes, detection of virus, etc., can be run by the administrative component of the provisioning server.” EX1012, 13:46-51. “Hosts can be monitored for load and if the load crosses a threshold, the administrator can be alerted and/or actions can be taken such as migrating VMS to a lightly loaded server.” EX1012, 13:51-54.

These tasks of monitoring deployed VMs are facilitated by virtual machine monitors (VMMs). EX1012, 6:62-8:44 (describing VMM functions).

Khandekar’s VMM performs a similar role to Hashimoto’s hypervisor. *Compare* EX1012, 7:7-9 *with* EX1005, [0060]; EX1003, [86].

Based on these teachings a POSITA would have found it obvious to apply the application selection, deployment, and management techniques described by Khandekar. EX1003, [103]. For example, a POSITA would have found it obvious to utilize Hashimoto’s hypervisor and control terminal to perform the application deployment and management functions described by Khandekar with respect to its VMM and provisioning server. EX1012, 7:7-27, 8:20-36, 22:14-33; EX1003, [103]. A POSITA would have been motivated to combine Khandekar with Hasimoto and Lubbers in this manner, because the deployment and management functions described by Khandekar would have created greater flexibility within the Hasimoto-Lubbers system (*e.g.*, allowing users to freely select applications for use

on the virtual machines and change them after deployment), increased resilience (e.g., detecting and recovering from faults and upgrading older software), and increased performance (e.g., real-time load balancing). EX1012, 4:42-51, 13:44-54; EX1003, [103].

2. Claim 6

As part of deploying VMs, Khandekar teaches that “the user is given some choice about how the provisioned VM's virtual hardware, system software, and applications are selected.” EX1012, 11:11-13. “If the user’s configuration preferences cannot be matched exactly, then [a] heuristics engine 880 [of the provisioning server] suggests the best available match by applying a database of heuristic rules about properties of VMs, available applications and operating systems, virtualized devices, etc.” EX1012, 11:14-19. The heuristics engine 880 also selects an appropriate host computer on which to deploy the VM and its applications “in order to balance the load on the different available hosts.” EX1012, 13:4-8.

“Once a VM has been configured and deployed, note that it will also be possible to alter its configuration, for example by upgrading the software and/or hardware, installing different applications, etc.” EX1012, 4:48-51.

“[M]aintenance tasks like upgrade of software and hardware (for example, activating a different number of processors, memory chips, etc.) installation of

service packs or hot fixes, detection of virus, etc., can be run by the administrative component of the provisioning server.” EX1012, 13:46-51. “Hosts can be monitored for load and if the load crosses a threshold, the administrator can be alerted and/or actions can be taken such as migrating VMs to a lightly loaded server.” EX1012, 13:51-54. In other words, Khandekar describes dynamic deployment or allocation of applications to virtual assets based on business policies (e.g., upgrades and hot fixes) and workload conditions (e.g., the load of the hosts/servers). EX1003, [206].

The above-described deployment and management functionality would be carried out by the control terminal 300, which “controls operations of the entire computer system.” EX1005, [0052], [0084], [0178]; EX1003, [207].

The configuration management (e.g., system configuration by the control terminal 300 and its virtual machine control program 301) ***includes application routing for virtual assets or implementation of application rules*** (e.g., dynamic deployment or allocation of applications to virtual assets based on business policies and workload conditions). EX1003, [208].

IV. FEES

Petitioner authorizes charge of fees to Deposit Account 06-1050.

V. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

Microsoft Corporation (“Microsoft”) is the real party-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

The ’667 Patent is the subject of civil action *ToutVirtual, Inc. v. Microsoft Corporation*, 2-25-cv-00943 (EDTX) filed September 8, 2025. The ’667 Patent is subject to a terminal disclaimer with respect to U.S. Patent No. 9,444,762 EX1002, 66-68. Petitioner is not aware of any reexamination certificates or other IPR petitions addressing the ’667 Patent.

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation of counsel.

Lead Counsel	Backup counsel
W. Karl Renner, Reg. No. 41,265 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR42342-0157IP1@fr.com	Usman Khan, Reg. No. 70,439 David Holt, Reg. No. 65,161 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 IPR42342-0157IP1@fr.com

D. Service Information

Please address all correspondence and service to the address listed above.

Petitioner consents to electronic service by email at IPR42342-0157IP1@fr.com.

Respectfully submitted,

Dated: March 7, 2026

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265

Usman Khan, Reg. No. 70,439

David Holt, Reg. No. 65,161

Fish & Richardson P.C.

60 South Sixth Street, Suite 3200

Minneapolis, MN 55402

T: 202-783-5070

F: 877-769-7945

(Control No. IPR2026-00297)

Counsel for Petitioner

CERTIFICATION UNDER 37 CFR § 42.24

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for Inter Partes Review totals 13,917 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated: March 7, 2026

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265

Usman Khan, Reg. No. 70,439

David Holt, Reg. No. 65,161

Fish & Richardson P.C.

60 South Sixth Street, Suite 3200

Minneapolis, MN 55402

T: 202-783-5070

F: 877-769-7945

Counsel for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) et seq. and 42.105(b), the undersigned certifies that on March 7, 2026, a complete and entire copy of this Petition for *Inter Partes* Review and all supporting exhibits were provided by Federal Express, to the Patent Owner, by serving the correspondence address of record as follows:

DLA PIPER LLP (US)
3203 Hanover Street
Suite 100
PALO ALTO, CA 94304
UNITED STATES
(650) 833-2000

/Crena Pacheco/

Crena Pacheco
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
pacheco@fr.com