

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

VEEAM SOFTWARE CORPORATION,
Petitioner,

v.

SYMANTEC CORPORATION,
Patent Owner.

Case IPR2014-00091
Patent 8,117,168 B1

Before THOMAS L. GIANNETTI, TRENTON A. WARD, and
JEREMY M. PLENZLER, *Administrative Patent Judges*.

WARD, *Administrative Patent Judge*.

DECISION
Final Written Decision
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

We have jurisdiction under 35 U.S.C. § 6(c). This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73.

With respect to the grounds asserted in this trial, we have considered the papers submitted by the parties and the evidence cited therein. For the reasons discussed below, we determine Petitioner has shown by a preponderance of the evidence that claims 1–4, 9–12, and 14 of U.S. Patent No. 8,117,168 B1 (Ex. 1001 (“the ’168 patent”)) are unpatentable.

A. *Procedural History*

Veeam Software Corporation (“Petitioner”) filed a Petition requesting an *inter partes* review of claims 1–4, 9–12, and 14 of the ’168 patent. Paper 4 (“Pet.”). Petitioner included a Declaration of Dr. Prashant Shenoy. Ex. 1002. Symantec Corporation (“Patent Owner”) filed a Preliminary Response to the Petition. Paper 8 (“Prelim. Resp.”).

In our Decision Instituting *Inter Partes* Review, we granted review as to all of the challenged claims. Paper 9 (“Dec. Inst.”). Patent Owner filed a Response to the Petition (Paper 18, “PO Resp.”), which included a Declaration of Matthew Green (Ex. 2002). Petitioner filed a Reply to the Response. Paper 21 (“Pet. Reply”). On December 18, 2014, all parties were present for an oral hearing. Paper 28 (“Tr.”).

B. *Related Matters*

Petitioner indicates that the ’168 patent is the subject of the following co-pending federal district court case: *Symantec Corp. v. Veeam Software Corp.*, No. 3:12-cv-5443-SI (N.D. Ca.) (consolidated with No. 3:12-cv-00700-SI (N.D. Ca.)). Pet. 1. In addition to this Petition, Petitioner previously filed petitions challenging the patentability of certain claims of

Patent Owner's U.S. Patent Nos. 6,931,558 B1 (IPR2013-00141, IPR2013-00142), 7,254,682 B1 (IPR2013-00144, IPR2013-00145), 7,191,299 B1 (IPR2013-00143, IPR2013-00151). In three of those proceedings we denied institution on August 7, 2013. *See* IPR2013-00144, Paper 11; IPR2013-00145, Paper 12; IPR2013-00151, Paper 7. We entered a final decision in the remaining proceedings on July 29, 2014. *See* IPR2013-00141, Paper 50; IPR2013-00142, Paper 51; IPR2013-00143, Paper 48. Additionally, on October 22, 2013, Petitioner filed petitions challenging the patentability of certain claims of Patent Owner's U.S. Patent Nos. 7,480,822 B1 (IPR2014-00088), 7,831,861 B1 (IPR2014-00089), and 7,024,527 B1 (IPR2014-00090). On March 20, 2015, we entered a final decision in IPR2015-00088 (Paper 45) and shall enter a final decision in the other cases concurrently with this decision.

C. The '168 Patent

The '168 patent is titled "Methods and Systems for Creating and Managing Backups Using Virtual Disks" and generally relates to managing multiple virtual-disk backup files at different points in time. Ex. 1001, Abstr. The patent describes a computer system that backs up data to a first virtual disk file, captures subsequent incremental changes to a second virtual disk file, and rolls the incremental changes into the first virtual disk file to create a synthetic full backup. *Id.* at 1:51–67. The patent further describes creating an empty virtual-disk file that enables a virtual machine to boot from the first virtual disk. *Id.* at 2:9–16. Figure 4 of the '168 patent, reproduced below, illustrates a timeline for the backup process of a data storage entity:

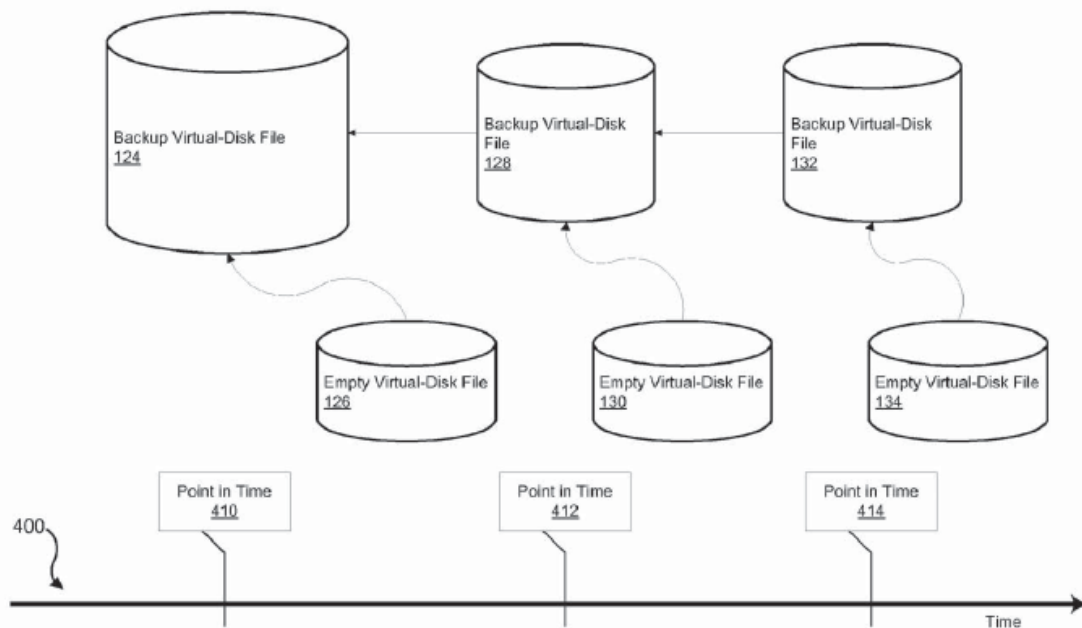


FIG. 4

As shown above in Figure 4, the '168 patent discloses, at point in time 410, creating backup virtual-disk file 124 of data-storage entity 122 (not shown). *Id.* at 7:40–43. Empty virtual-disk file 126 is created and associated with backup virtual-disk file 124 so that a virtual machine may boot from backup virtual-disk file 124 without modifying backup virtual-disk file 124. *Id.* at 7:43–49. At point in time 412, backup virtual-disk file 128 is created that contains at least one change made to data-storage entity 122 after creation of backup virtual-disk file 124. *Id.* at 7:50–52. A parent-child relationship between backup virtual-disk file 124 and backup virtual-disk file 128 is created, such that backup virtual-disk file 124 is the parent to backup virtual-disk file 128. *Id.* at 7:52–55. Empty virtual-disk file 130 is created and associated with backup virtual-disk file 128 to allow a virtual machine to boot similar to the functionality of empty virtual-disk file 126. *Id.* at 7:55–61. Any number of additional backup virtual-disk files and empty virtual-

disk files may be created as illustrated by backup virtual-disk file 132 and empty virtual-disk file 134 at point in time 414. *Id.* at 7:62–8:11.

Claim 1, reproduced below, is illustrative of the claimed subject matter:

1. A computer-implemented method for backing up data, at least a portion of the method being performed by a computing system comprising at least one processor, the method comprising:

at a first point in time, backing up at least a portion of a data-storage entity to a first virtual-disk file;

capturing, in a second virtual-disk file, at least one change made to data in the data-storage entity after the first point in time;

creating a parent-child relationship between the first virtual-disk file and the second virtual-disk file, the first virtual-disk file being a parent of the second virtual-disk file;

copying data stored in the second virtual-disk file to the first virtual-disk file so that the first virtual-disk file comprises a synthetic backup that includes the at least one change made to data in the data-storage entity after the first point in time;

storing the first virtual-disk file that comprises the synthetic backup in a manner that enables at least one virtual machine to boot from the stored first virtual-disk file;

creating a first empty virtual-disk file;

creating a parent-child relationship between the first virtual-disk file and the first empty virtual-disk file, the

first virtual-disk file being a parent of the first empty virtual-disk file;

retargeting the first empty virtual-disk file to provide a first retargeted virtual-disk file and to enable the at least one virtual machine to boot from the first virtual-disk file.

D. Instituted Grounds and Prior Art

We instituted an *inter partes* review on the following grounds under 35 U.S.C. § 103:

References	Claims Challenged
Colbert ¹ and Lim ²	1–4, 9–12, and 14
Liu ³ and HVR ⁴	1–4, 9–12, and 14

Dec. Inst. 24.

II. ANALYSIS

A. Claim Construction

Consistent with the statute and the legislative history of the Leahy-Smith America Invents Act,⁵ the Board will interpret claims of an unexpired patent using the broadest reasonable construction in light of the specification of the patent. *See* 37 C.F.R. § 42.100(b); *In re Cuozzo Speed Techs., LLC*, 778 F.3d 1271, 1279–83 (Fed. Cir. 2015). There is a “heavy presumption” that a claim term carries its ordinary and customary meaning. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002).

¹ U.S. Patent No. 8,239,646 B2 (Ex. 1004 (“Colbert”).

² U.S. Patent No. 6,795,966 B1 (Ex. 1005 (“Lim”).

³ WO Publication No. 2007/002398 A2 (Ex. 1006 (“Liu”).

⁴ U.S. Patent No. 7,093,086 B1 (Ex. 1007 (“HVR”).

⁵ Pub. L. No. 112–29 (2011).

“creating a parent-child relationship”

Petitioner alleges that the term “creating a parent-child relationship” means “logically connecting two virtual-disk files so that changes intended for the parent virtual-disk file are instead stored in the child virtual-disk file.” Pet. 8. Patent Owner does not expressly provide an alternate construction, but notes that Petitioner’s proposed constructions were not adopted by the Northern District of California in the related District Court matter, *Symantec Corp. v. Veeam Software Corp.*, No. 3:12-cv-5443-SI (N.D. Ca.). Prelim. Resp. 13 n.3 (citing Ex. 2001, 6–7).

We note that independent claims 1, 12, and 14 require “creating a parent-child relationship between the first virtual-disk file and the second virtual-disk file, the first virtual-disk file being a parent of the second virtual-disk file.” The ’168 Specification discloses, “[a] parent-child relationship between two virtual-disk files may indicate that the child virtual-disk file holds incremental changes made to data since the parent virtual-disk file was created.” Ex. 1001, 5:67–6:3.

In the related District Court matter between the parties, the Court adopted a similar construction as advanced by Petitioner with respect to the first limitation, albeit modifying Petitioner’s “instead stored” to “first stored.” Ex. 2001, 7 (construing the first limitation as “logically connecting two virtual-disk files so that changes intended for the parent virtual-disk file are first stored in the child virtual-disk file”). Although the District Court only construed the first limitation, we agree with the District Court’s underlying rationale that the purpose of both parent-child relationships of the ’168 patent is initially to direct changes intended for a parent file to a child file. *See id.* at 6–7. Therefore, we agree with the District Court’s

construction and construe “creating a parent-child relationship” as “logically connecting two virtual-disk files so that changes intended for the parent virtual-disk file are first stored in the child virtual-disk file.”

B. Asserted Obviousness Ground Based on Colbert and Lim

1. Overview of Colbert

Colbert is titled “Online Virtual Machine Disk Migration” and describes migrating a first virtual machine disk to a second virtual machine disk while the virtual machine is running. Ex. 1004, 1:60–62. The migration allows virtual machine disks to be relocated to different physical storage devices. *Id.* at 2:44–46. Figure 5A of Colbert illustrates one embodiment of the migration process and is reproduced below:

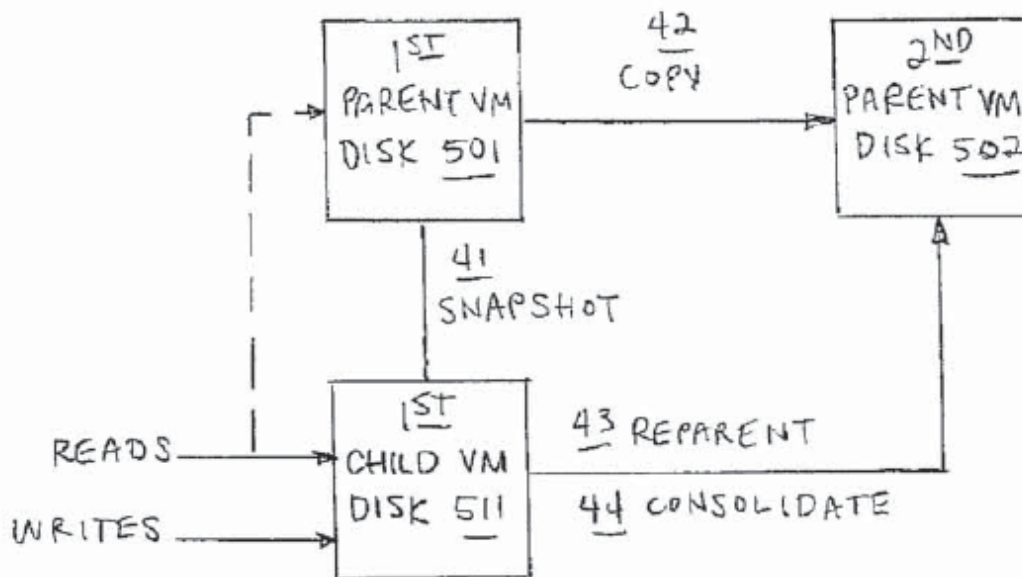


FIG. 5A

As shown above in Figure 5A, Colbert discloses migrating first parent virtual machine (VM) disk 501 to second parent VM disk 502 by first

creating, at snapshot step 41, child VM disk 511 of first parent VM disk 501. *Id.* at 4:17–18. After child VM disk 511 is created, first parent VM disk 501 is made read-only such that all writes intended for parent VM disk 501 are redirected to child VM disk 511, thus allowing the system to begin copy step 42 of first parent VM disk 501 to second parent VM disk 502. *Id.* at 4:35–39; 5:16–18. After copy step 42 is complete, child VM disk 511 is reparented in step 43 from first parent VM disk 501 to second parent VM disk 502. *Id.* at 5:29–32. Child VM disk 511 is next consolidated in step 44 with second parent VM disk 502 such that changes written into child VM disk 511 are incorporated into parent VM disk 502. *Id.* at 5:45–48. Colbert Figure 5B discloses a multi-child VM disk migration embodiment and is reproduced below:

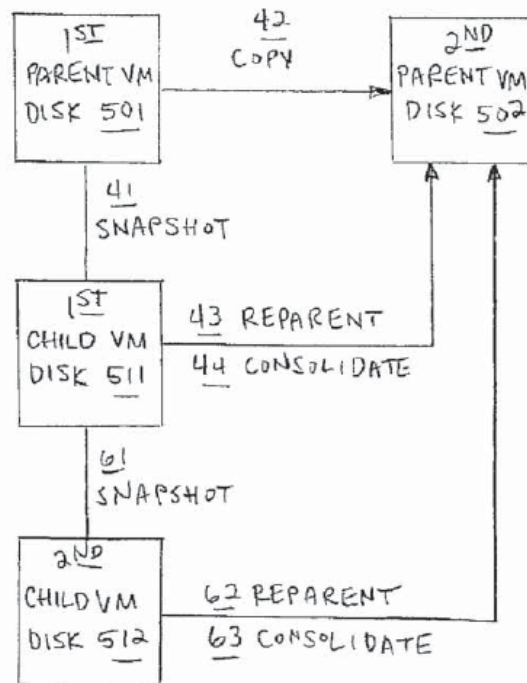
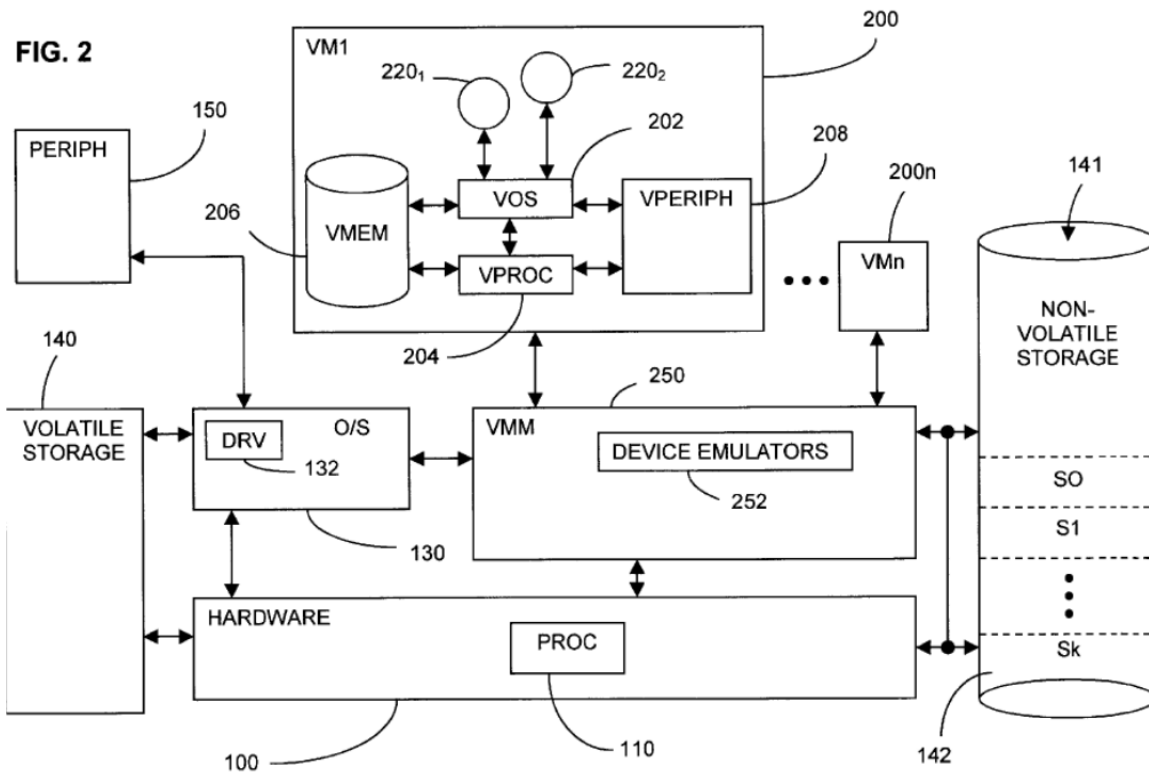


FIG. 5B

As shown above in Figure 5B, Colbert discloses, similar to Figure 5A, creating first child VM disk 511 of first parent VM disk 501 at snapshot step 41. *Id.* at 4:17–18. Figure 5B further discloses second snapshot step 61, which creates second child VM disk 512 of first child VM disk 511. *Id.* at 6:11–18. First child VM disk 511 is made read-only, and writes directed to first parent VM disk 501 are redirected to second child VM disk 512. *Id.* at 6:27–31. First child VM disk 511 is then reparented in step 43 and consolidated in step 44 with second parent VM disk 502. *Id.* at 6:37–42. After reparenting step 43 and consolidation step 44, second child VM disk 512 also is reparented in step 62 and consolidated in step 63 with second parent VM disk 502 to incorporate any changes written to second child VM disk 512. *Id.* at 6:44–51.

2. *Overview of Lim*

Lim is titled “Mechanism for Restoring, Porting, Replicating and Checkpointing Computer Systems Using State Extraction” and discloses a virtual machine monitor on which multiple virtual computer systems are installed whose states can be checkpointed under control of the virtual machine monitor. Ex. 1005, Abstr. The checkpointed virtual machine may be restored into the system at a later time. *Id.* Figure 2 of Lim is reproduced below:



As shown above in Figure 2, Lim discloses virtual machine (“VM1”) 200, including virtual operation system (“VOS”) 202, virtual processor (“VPROC”) 204, virtual disk that is virtual memory (“VMEM”) 206, and virtual peripheral devices 208, “all of which are implemented in software to emulate the corresponding components of an actual computer.” *Id.* at 14:27–32. Additionally, as shown in Figure 2, Lim discloses virtual machine monitor (“VMM”) 250 between virtual machines 200 to 200n and the system hardware. *Id.* at 15:26–31. To enable computer system restoration, Lim discloses capturing the “total machine state” of a computer system, which is “the entire collection of all information that is necessary and sufficient to uniquely determine the status of all hardware and software components at the completion of any given processor instruction.” *Id.* at

10:26–30. Therefore, if the processor’s execution is interrupted, Lim discloses that:

the total machine state is then the set of data that, when loaded at any time into the appropriate memory positions (both internal and external to the processor), will cause the processor, and all connected hardware and software components, to continue executing in exactly the same way as if there had been no interruption at all.

Id. at 10:32–38.

3. Analysis of Asserted Ground of Obviousness In View of Colbert and Lim

Petitioner argues that claims 1–4, 9–12, and 14 would have been obvious in view of Colbert and Lim. Pet. 11–32. Petitioner argues Colbert discloses many of the limitations of claim 1, including that Colbert’s second parent VM disk 502 corresponds to the “first virtual disk file,” Colbert’s first child VM disk 511 corresponds to the “second virtual disk file,” and Colbert’s second child VM disk 512 corresponds to the “first empty virtual disk file.” *Id.* at 11–21. Petitioner argues that Colbert discloses retargeting second child VM disk 512 (“empty virtual-disk file”) to receive writes intended for parent VM disk 502 (“first virtual-disk file”), thus allowing Virtual Machine 200 to continue executing, including restarting or booting, by reading information stored on the second parent VM disk 502 or second child VM disk 512. *Id.* at 21 (citing Ex. 1002 ¶ 55). Furthermore, Petitioner argues that, to the extent that Colbert does not disclose allowing a virtual machine to boot, Lim discloses that booting is an alternative to running from a virtual machine’s snapshot, and thus Lim discloses enabling the “virtual machine to boot from the first virtual-disk file,” as required by claim 1. *Id.*

(citing Ex. 1005, 28:15-20). Patent Owner argues that Petitioner’s proposed challenge does not render the challenged claims obvious.

a. Migration Versus Backup

Patent Owner argues that Colbert’s data migration may not be characterized as a “backup,” because Colbert discloses that certain VM disks can be deleted and such deletion is inconsistent with the notion of a backup. PO Resp. 17 (citing Ex. 1004, 5:16–18, 6:51–53). In response, Petitioner argues that Colbert discloses that the original “*can* be deleted” not that it *must*. Pet. Reply 1 (citing Ex. 1004, 5:55–56, 6:51–53). Additionally, Petitioner’s declarant Dr. Shenoy states that one of ordinary skill in the art would understand that parent VM disk 501 and child VM disk 511 disclosed in Colbert need not be deleted and may have use in the template deployment described in Colbert. Ex. 1002 ¶ 40 (citing Ex. 1004, 8:37–47). Based on this record, we are not persuaded by Patent Owner’s arguments that Colbert’s teaching of a possible deletion of the VM disks is inconsistent with “backing up data,” as recited in the claims.

b. Capturing a Change Made to the Data

Patent Owner argues that Colbert does not teach or suggest the claim limitation “captur[ing], in a second virtual-disk file, at least one change made to data in the data-storage entity,” as recited in each of the challenged independent claims 1, 12, and 14. PO Resp. 20. Specifically, Patent Owner argues that Petitioner alleges that Colbert’s VM disk 501 corresponds to the claimed “data storage entity” being backed up, but Colbert teaches that VM disk 501 remains static after it is copied. *Id.* at 20–21 (citing Ex. 1002 ¶ 33; Ex. 2004, 54:17–55:25; Ex. 1004, 4:35–43, 7:14–21). Thus, Patent Owner

argues that Colbert does not teach capturing “a change made to the data-storage entity.” *Id.* at 21 (citing Ex. 2002 ¶ 56).

In response, Petitioner argues that the broadest reasonable interpretation of the capturing step in claim 1 encompasses capturing changes to the data using a copy-on-write method as taught in Colbert and the '168 patent. Pet. Reply 2. First, Petitioner argues that the plain meaning of capturing includes catching and holding. *Id.* (citing Ex. 1009; Ex. 1010; Ex. 1008 ¶¶ 8–9; Ex. 2002 ¶ 62). Second, Petitioner argues that the Specification of the '168 patent discloses a copy-on-write embodiment, which describes “capturing” by catching and holding changes intended for the data storage entity. *Id.* at 2–3 (citing Ex. 1004, 5:47–51, 1:54–55, 4:33–35). The Specification states that “[i]n at least one embodiment, backup module 112 may monitor data-storage entity 122 and use a *copy-on-write method* to capture the at least one change by capturing every write made to data-storage entity 122.” Ex. 1004, 5:47–51 (emphasis added). Petitioner argues that in accordance with copy-on-write techniques, a person of ordinary skill in the art would understand the phrase “capturing every write made to data-storage entity 122” to refer to writes that are sent to the data storage entity, but caught and held using the copy-on-write method. PO Resp. 3. Dr. Shenoy states that the claim 1 language reciting “capturing . . . at least one change made to data-storage entity after the first point in time” does not require that the change *must be* stored in the data-storage entity, but rather that the change is made only to the data and captured. Ex. 1008 ¶ 9.

We are not persuaded by Patent Owner’s argument and determine that the '168 patent Specification discloses embodiments that do not change the

data in the original data-storage entity. For example, the '168 patent discloses that “[i]n some embodiments, data-storage entity 122 may comprise . . . an optical disk (e.g., DVD-ROM [Read Only Memory], CD-ROM [Read Only Memory], etc.).” Ex. 1004, 4:59–63. Thus, if “data-storage entity 122” is a read only DVD-ROM or CD-ROM, then data in the “data-storage entity 122” cannot be changed. Accordingly, we determine that the broadest reasonable interpretation of the “capturing” step recited in claim 1 encompasses the copy-on-write method disclosed in the '168 patent Specification and in Colbert.

c. First Empty Virtual-Disk File

Patent Owner argues that the proposed combination does not teach “creat[ing] a first empty virtual-disk file” or “creating a parent-child relationship between the first virtual-disk file and the first empty virtual-disk file,” as recited in each of challenged independent claims 1, 12, and 14. PO Resp. 22. Petitioner alleges that Colbert’s second child VM disk 512 corresponds to the claimed “first empty virtual-disk file,” and Petitioner quotes Colbert’s disclosure that “second child VM disk 512 initially starts out empty.” Pet. 19 (quoting Ex. 1004, 6:17–19).

Patent Owner argues that Colbert’s VM disk 512 fails to meet the claimed “first empty virtual-disk file” because VM disk 512 does not remain empty before it is reparented to VM disk 502 but captures changes that ordinarily would have been directed to first child VM disk 511 or first parent VM disk 501. PO Resp. 22–23. Petitioner responds by arguing that the claims and the '168 patent Specification use the term “empty” in “first empty virtual-disk file” as nomenclature rather than an unalterable condition of the file. Pet. Reply 5. More particularly, Petitioner argues that the

'168 patent Specification refers to the “empty virtual-disk file” as an empty virtual-disk file even after it has received writes. Pet. Reply 5 (citing Ex. 1001 6:66–7:1, 7:24–27, 7:32–34). For example, the '168 patent Specification states that “[t]he virtual machine may *redirect future writes to empty virtual-disk file 126*, allowing backup virtual-disk file 124 to remain unchanged.” Ex. 1001, 6:66–71. Furthermore, the '168 patent Specification states that “backup module 112 may create empty virtual-disk file 126, which may be an incremental virtual-disk file,” and that “[a]n incremental virtual-disk file may include any file that stores incremental changes.” *Id.* at 4:29–31. Dr. Shenoy testifies that claim 5 in the '168 patent recites creating a second empty virtual-disk file that is reparented to a first virtual disk file after it has received data. Ex. 1008 ¶ 24 (“reparenting the second empty disk file to the first virtual disk file because the second empty disk includes data.”).

In view of the disclosures in the '168 patent, we are not persuaded that Patent Owner’s proposal that the independent claims require that the “first empty virtual-disk file” must remain empty before creating a parent-child relationship. Further, in view of this evidence, we are persuaded that Petitioner sufficiently establishes that Colbert teaches or suggests the claimed “first empty virtual-disk file.”

d. Retargeting the First Empty Virtual-Disk File

Patent Owner argues that the proposed combination does not teach that the first empty virtual-disk file be “retarget[ed]” such that the virtual machine can boot from the virtual-disk file, as recited in independent claims 1, 12, and 14. PO Resp. 25. Patent Owner argues that the claimed “retargeting” may involve “replacing, reconfiguring, and/or installing one

or more of a Hardware Abstraction Layer (‘HAL’), kernel, mass storage driver, and/or any other device drivers.’” PO Resp. 25 (quoting Ex. 1001, 6:58–63).

Petitioner responds that the ’168 patent broadly defines the retargeting of empty virtual-disk files. Pet. Reply 5–6 (citing Ex. 1004, 6:58–40). The ’168 patent Specification states in the sentence immediately preceding the examples of retargeting (i.e., replacing, reconfiguring, etc.) cited by Patent Owner that “[r]etargeting empty virtual-disk file 126 may include *any action* that enables a virtual machine to boot from backup virtual-disk file 124.” Ex. 1001, 6:58–60 (emphasis added). Any special definition for a claim term must be set forth with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). We determine Patent Owner’s statement in the Specification to provide clarity, deliberateness, and precision about the meaning of “retargeting an empty virtual-disk file.” Accordingly, we determine that the claim limitation “retargeting the first empty virtual-disk file” must include any action that enables a virtual machine to boot from a virtual-disk file.

Additionally, Patent Owner argues that Petitioner’s challenge fails because it points to the same disclosure in Colbert to demonstrate two separate limitations, namely, “creat[ing] a parent-child relationship” and “retargeting.” PO Resp. 26. Petitioner responds that unless the claim requires otherwise, separately recited elements may be mapped to a single component or step. Pet. Reply 6 (citing *Powell v. Home Depot USA Inc.*, 663 F.3d 1221, 1231–32 (Fed. Cir. 2011)). Furthermore, Petitioner argues that the cited Colbert combination teaches or suggests creating a parent-child relationship and retargeting. Pet. Reply 8. First, Petitioner argues that

Colbert's disclosure that the "second child VM disk 512 is re-parented to second parent VM disk 502" corresponds to the claimed "creating a parent-child relationship between the first virtual-disk file and the first empty virtual-disk file." Pet. 20 (quoting Ex. 1004, 6:45–46). Second, Patent Owner argues that second child VM disk 512 ("first empty virtual-disk file") is retargeted to receive writes intended for parent VM disk 502 ("first virtual-disk file"); thus, allowing Virtual Machine 200 to continue executing, including being able to shutdown, restart, or boot by reading information stored on second parent VM disk 502 or second child VM disk 512. Pet. 21 (citing Ex. 1002 ¶ 55). Furthermore, Petitioner argues that Lim teaches "enabling at least one virtual machine to boot," and, therefore, Colbert in view of Lim discloses the claimed "retargeting the first empty virtual-disk file to provide a first retargeted virtual-disk file and to enable the at least one virtual machine to boot from the first virtual-disk file." Pet. 21 (citing Ex. 1005, 28:15–20). As the claimed "retargeting the first empty virtual-disk file" may "include *any action* that enables a virtual machine to boot from backup virtual-disk file" (Ex. 1001, 6:58–60 (emphasis added)), we are persuaded that Petitioner sufficiently establishes by a preponderance of the evidence that this claim element would have been obvious in view of Colbert's disclosure regarding the retargeting of VM disk 512 in combination with enabling a virtual machine to boot taught in Lim. Therefore, we are persuaded that Petitioner sufficiently establishes that Colbert in view of Lim teaches or suggests the claimed retargeting of the first empty virtual-disk file.

e. Combinability of Colbert with Lim

Patent Owner argues that the combination of Colbert and Lim would frustrate Colbert's purpose of migrating a virtual machine in a manner that avoids interruption and allows it to continue executing, because Lim's checkpointing process requires interrupting the virtual machine in order to capture and extract its state vector information. PO Resp. 31 (citing, Ex. 1004, 1:32–62, 3:55–62, 7:49–54; Ex. 1005, 29:53–58, 6:45–52, 8:15–19, 18:2–10; Ex. 2002 ¶ 84). More particularly, Patent Owner argues that using Lim's checkpointing techniques for capturing and copying the contents of the original VM disk in Colbert would contradict Colbert's intended goal, and render superfluous Colbert's children VM disks. *Id.* at 32.

Nonobviousness cannot be established by attacking the references individually when a challenge is predicated upon a combination of prior art disclosures. *See In re Merck & Co., Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986). Patent Owner's arguments attack Colbert and Lim individually, but fail to account for the combination as proposed by the Petitioner. The Petitioner's challenge does not use "Lim's checkpointing technique for capturing and copying the contents of the original VM disk in Colbert," as argued by Patent Owner (PO Resp. 32), but rather Petitioner's proposed combination simply provides that "to the extent that Colbert does not explicitly teach 'enabling at least one virtual machine to boot,' Lim discloses this limitation" (Pet. 19). Therefore, Patent Owner's arguments about Lim's checkpointing process frustrating the teachings of Colbert are unpersuasive with respect to challenged independent claims.

We have reviewed the combination of Colbert and Lim proposed by the Petitioner and the supporting evidence relied upon by Petitioner. Pet. 11–13 (citing Ex. 1002 ¶¶ 28–29). With respect to this combination, we are persuaded that Petitioner sufficiently establishes an “articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007).

f. Dependent Claim 3

Dependent claim 3 requires that the “first virtual-disk file comprises a first incremental backup of the data-storage entity” and the “second virtual-disk file comprises a second incremental backup of the data-storage entity.” Petitioner alleges that Colbert’s child VM disk 511 corresponds to the claimed incremental backup first virtual-disk file and child VM disk 512 corresponds to the claimed incremental backup second virtual-disk file. Pet. 23–24 (citing Ex. 1004, 4:17–18, 20–22, 6:1–2). Additionally, Petitioner argues that it would have been obvious to one of ordinary skill in the art to combine the disclosure in Lim regarding rolling incremental backups into another incremental backup. Pet. 24 (citing Ex. 1005, 24:14–16; Ex. 1002 ¶ 58). Dr. Shenoy states that a person of ordinary skill would have found it obvious to copy the changes of one child VM disk into another child VM disk to reduce the amount of disk space used. Ex. 1002 ¶ 58.

Patent Owner argues that Petitioner’s challenge is deficient because Petitioner fails to show that a portion of the data storage entity is backed up to a first virtual-disk file as required by the challenged independent claims. PO Resp. 35. Contrary to Patent Owner’s arguments, however, Petitioner identifies that Colbert’s child VM disk 511 serves as incremental backup to

the data storage entity. *See* Pet. 23–24 (citing Ex. 1004, 4:17–18, 20–22, 6:1–2).

Patent Owner argues that “child VM disk 511 is made read-only during consolidation with VM disk 502”; thus, the contents of child VM disk 512 cannot be copied to child VM disk 511. PO Resp. 37 (emphasis added). Patent Owner’s argument overlooks the specifics of Petitioner’s challenge. With respect to claim 1, Petitioner argues that VM disk 502 corresponds to the “first virtual-disk file,” but for claim 3, Petitioner argues that Colbert’s child VM disk 511 corresponds to the claimed “first virtual-disk file [that] comprises a first incremental backup of the data-storage entity.” Pet. 23–24 (citing Ex. 1004, 4:17–18, 20–22, 6:1–2). Accordingly, Patent Owner’s arguments regarding Colbert’s VM disk 502 are not relevant to the challenge against dependent claim 3. Based on the record before us, we are persuaded that Petitioner sufficiently establishes that Colbert in view of Lim teaches or suggests the elements of claim 3.

g. Conclusion With Respect the Asserted Ground of Obviousness In View of Colbert and Lim

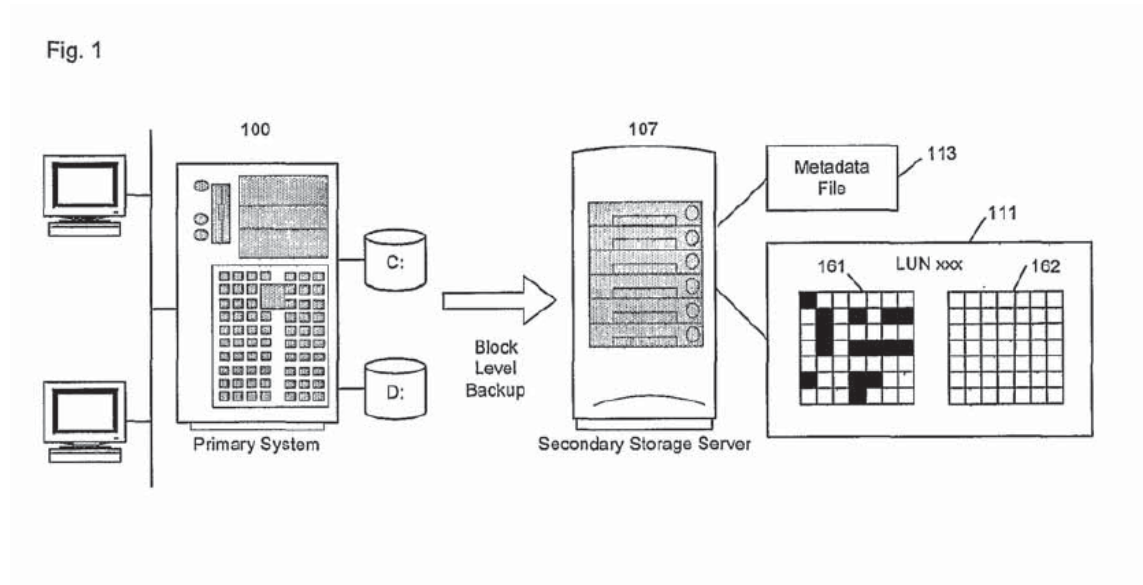
With respect to the Petitioner’s challenge that claims 1–4, 9–12, and 14 of the ’168 patent would have been obvious in view of Colbert and Lim, we have considered the papers submitted by the parties and the evidence cited therein. For the foregoing reasons, we determine that Petitioner has demonstrated by a preponderance of the evidence that claims 1–4, 9–12, and 14 of the ’168 patent would have been obvious in view Colbert and Lim.

C. Asserted Obviousness Ground Based on Liu and HVR

1. Overview of Liu

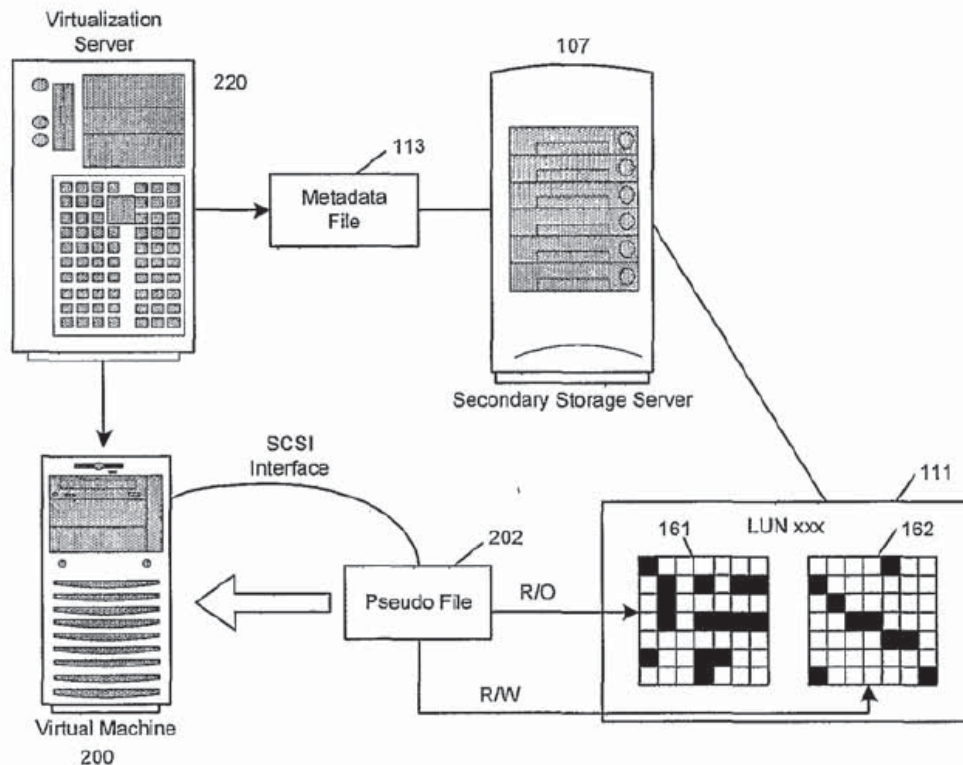
Liu is titled “System and Method for Virtualizing Backup Images” and discloses recreating a computer at a point in time from a backup image.

Ex. 1006, 2:20–23. The backup mechanism of Liu includes creating a first backup image followed by capturing subsequent changes separately. *Id.* at 4:26–30. Figure 1 of Liu, illustrating the backup mechanism, is reproduced below:



As shown above in Figure 1, Liu discloses backing up primary system 100 to read/write logical unit number (LUN) 111 on secondary storage server 107. *Id.* at 5:6–8. LUN 111 includes a read-only backup image 161 and additional space 162 for changed data. *Id.* Figure 2 of Liu, illustrating the virtualization process using the backup image, is reproduced below:

Fig. 2



As shown above in Figure 2, Liu describes configuring virtual machine 200 to run from LUN 111 by virtualization server 220. *Id.* at 6:4–12. Virtual machine 200 operates from LUN 111 using pseudo file 202, which transforms backup image 161 into a LUN and further maintains the integrity of backup image 161 by writing changes to separate space 162. *Id.* at 6:10–16. Virtual machine 200 may then be launched using boot records appropriate for the desired operating system. *Id.* at 7:1–3, 13–14.

2. Overview of HVR

HVR is titled “Disaster Recovery and Backup Using Virtual Machines” and describes backing up a virtual machine by capturing a state of the virtual machine at a point in time and copying the state to a storage device. Ex. 1007, Abstr. Figure 1 of HVR is reproduced below:

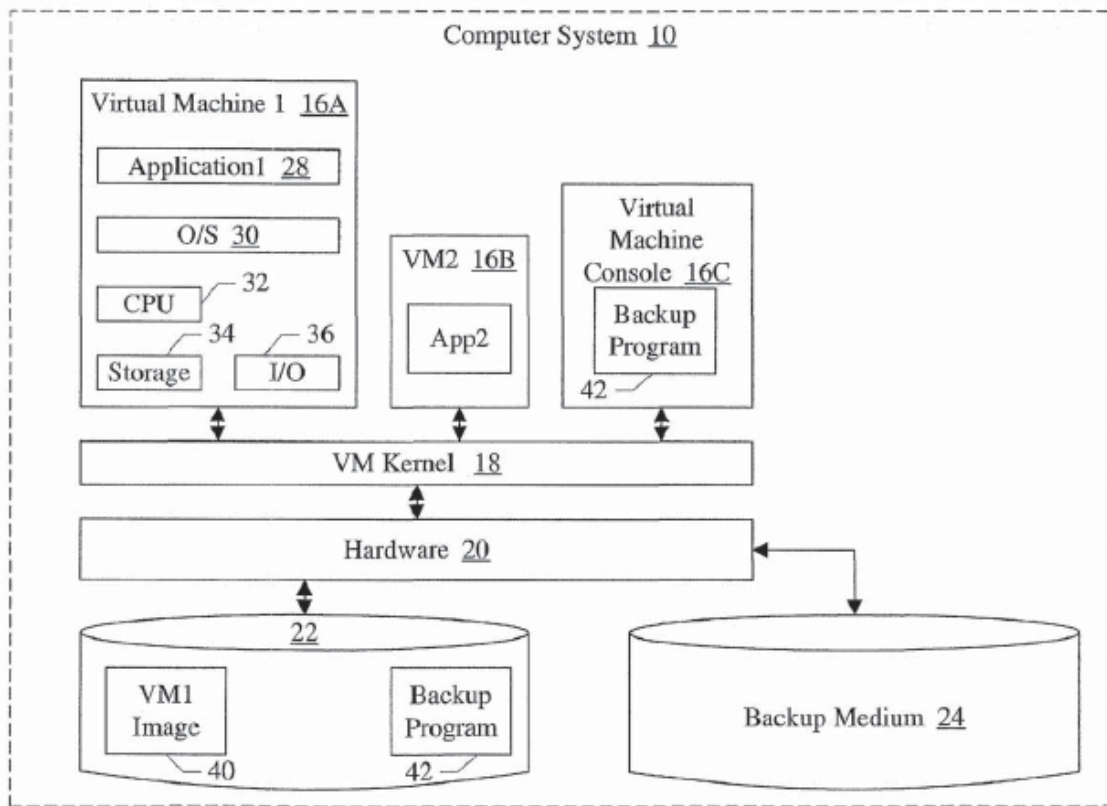
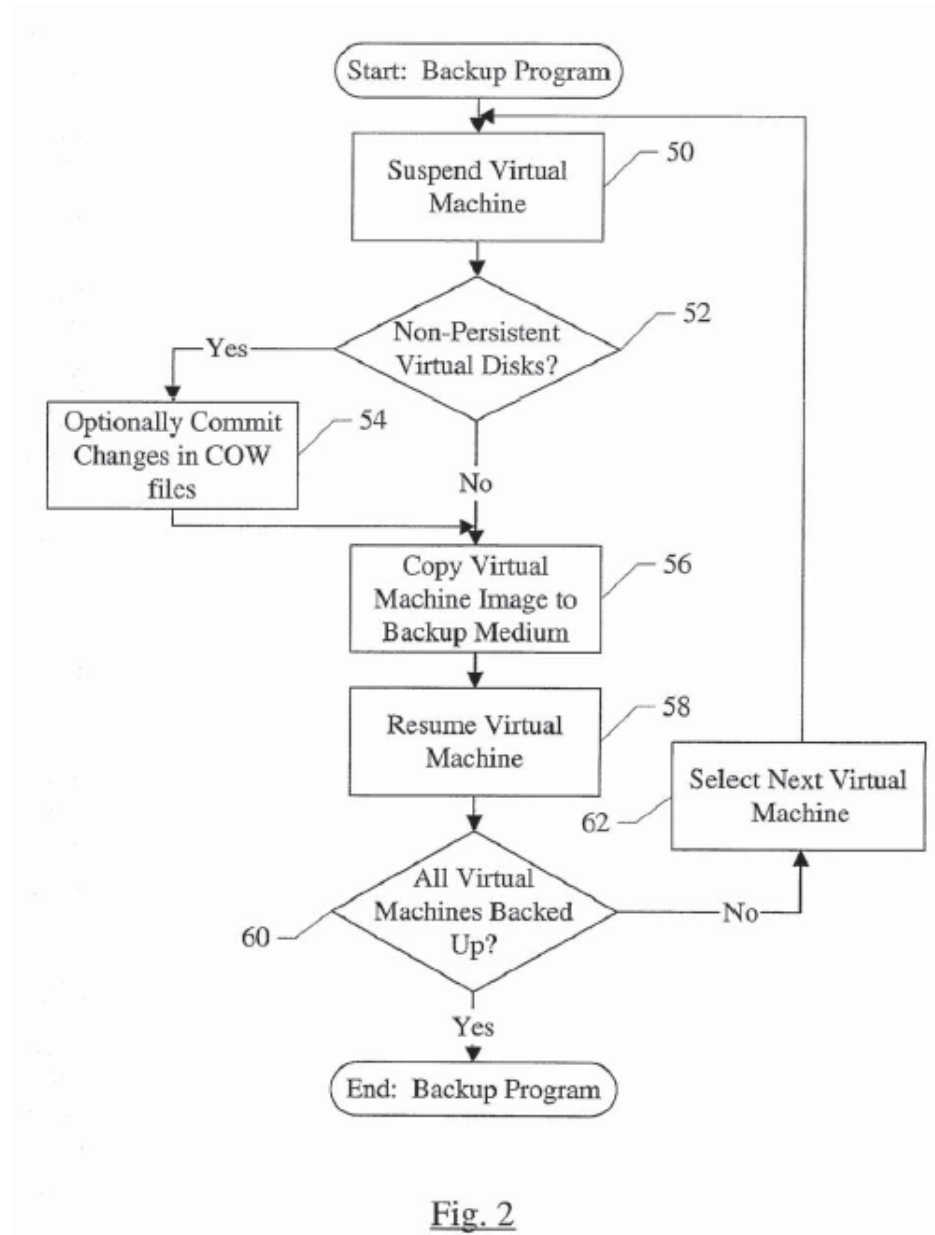


Fig. 1

As shown above in Figure 1, HVR describes computer system 10 including virtual machines 16A-16C controlled by VM kernel 18, backup program 42, and backup medium 24. *Id.* at 3:30–33, 42–61. Virtual machines 16A-16C mimic hardware used during execution of an application and are “any combination of software, one or more data structures in memory, and/or one or more files stored on a storage device (such as the storage device 22).” *Id.* at 3:45–49. Backup program 42 of computer system 10 temporarily

suspends one of virtual machines 16A-16C to create and store backup image 40 of the suspended virtual machine on backup medium 24. *Id.* at 4:53–61. Figure 2 of HVR describes one embodiment of the backup process and is reproduced below:



As shown above in Figure 2, HVR first suspends virtual machine at step 50 and determines if the suspended virtual machine includes a non-persistent virtual disk at step 52. *Id.* at 6:45–51, 7:10–11. HVR describes a non-

persistent virtual disk as including a virtual disk file and a separate copy-on-write (“COW”) file, of which

[t]he virtual disk file may be organized as a set of disk blocks, in a fashion similar to physical disk drives. The COW file stores updated copies of disk blocks in a log form. Thus, the virtual disk file may contain the disk blocks prior to any uncommitted updates being made.

Id. at 6:63–7:1. To commit changes from the COW file in step 54 of Figure 2,

the blocks from the COW file may be written to the corresponding block locations in the virtual disk file, beginning at the start of the COW file and proceeding, in order, to the end. Both the virtual disk files and the corresponding COW files may be included in the virtual machine image 40.

Id. at 7:3–14.

3. Analysis of Asserted Obviousness Ground Over Liu and HVR

Petitioner argues that claims 1–4, 9–12, and 14 would have been obvious in view of Liu and HVR. Pet. 32–56. With respect to claim 1, Petitioner argues Liu’s backup image 111 discloses a first virtual-disk file. *Id.* at 37. Petitioner argues that Liu discloses capturing subsequent changes to the image separately, but to the extent that Liu does not disclose a separate virtual file, Petitioner argues that HVR discloses virtual COW files may be used to capture changes, and thus Liu in view of HVR teaches a second virtual-disk file. *Id.* at 37–38. Petitioner argues that Liu’s storage backup 162 teaches an empty file that captures changes and allows a virtual machine to boot, but to the extent that Liu does not disclose explicitly that it is a virtual file, Petitioner argues that HVR discloses that virtual COW files are used to capture and apply changes. *Id.* at 41–45.

a. Data Storage Techniques of Liu Versus HVR

Patent Owner argues that Petitioner fails to reconcile the contrasting data storage techniques between Liu, which Patent Owner argues is directed to physical machines, and HVR, which Patent Owner argues is directed to virtual machines. PO Resp. 44–46. More particularly, Patent Owner argues that it would not have been obvious to implement HVR’s COW file structure, which is used in non-persistent virtual disks, in a physical machine such as Liu’s where disk blocks change as data is written. PO Resp. (citing Ex. 1006, 5:26–27, 12:21–23, Ex. 2002 ¶¶ 106, 108, 110). Petitioner responds that it does not propose physically incorporating HVR’s features into Liu, but rather what the combined features would have suggested to a person of ordinary skill in the art. Pet Reply 10–11.

An obviousness analysis “need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ.” *KSR*, 550 U.S. at 418; *see also In re Translogic Tech*, 504 F.3d 1249, 1259 (Fed. Cir. 2007). A prima facie case of obviousness is established when the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art. *In re Rinehart*, 531 F.2d 1048, 1051 (CCPA 1976). Petitioner argues that implementing HVR’s COW file in Liu would have been obvious to a person of ordinary skill in the art. Pet. 37–38. Additionally, Petitioner argues that Patent Owner concedes that HVR’s “COW files are well-known structure” used to capture changes for a virtual-disk file. Pet Reply 11 (quoting PO Resp. 45, 49). Furthermore, Dr. Shenoy states that both Liu and HVR relate to backup and disaster recovery using virtual machines, and that although

Liu relates to physical machines, Liu discloses creating a virtual machine backup from physical machines. Ex. 1002 ¶¶ 69–70. Furthermore, Dr. Shenoy states that substituting the COW file of HVR, which HVR describes using for incremental backups, with Liu’s incremental backup procedure would be merely a simple substitution for a person or ordinary skill in the art. *Id.* at ¶ 71. We are persuaded that Petitioner sufficiently provides an articulated reasoning with a rational underpinning to combine the teachings of Liu and HVR.

b. Creating a Parent-Child Relationship

Patent Owner argues that Petitioner fails to point to any disclosure in Liu or HVR that satisfies the claim limitation regarding “creating a parent-child relationship between the first virtual-disk file and the second virtual-disk file.” PO Resp. 50–51. As discussed above, we construe “creating a parent-child relationship” as “logically connecting two virtual-disk files so that changes intended for the parent virtual-disk file are first stored in the child virtual-disk file.” Patent Owner argues that Petitioner merely points to an association, not a “relationship,” between an already captured set of changes for a volume and the proper backup image to which these changes ought to be applied subsequently. PO Resp. 51 (citing Pet. 39; Ex. 1002 ¶ 84). Contrary to Patent Owner’s argument, Petitioner alleges that Liu discloses that “[a]fter the first backup, only changes since the last backups are captured and applied to the existing secondary image.” Pet. 39 (citing Ex. 1006, 4:29–30). Furthermore, Petitioner alleges that because each volume has its own associated stored backup image, the system must determine which volume a particular set of changes is to be applied and create a relationship between the backup image of that volume (first virtual-

disk file) and its change file (second virtual-disk file). *Id.* (citing Ex. 1002 ¶ 84 (quoting Ex. 1006, 4:32–33 (“[a] separate image exists for each volume of the backed up machine”))). Dr. Shenoy states that a parent-child relationship exists because Liu’s changed data stores changes intended for its associated backup image. Ex. 1002 ¶ 84. In view of this evidence, we are persuaded that Petitioner sufficiently establishes that Liu teaches or suggests the claimed “creating a parent-child relationship.”

c. Storing the First Virtual-Disk File

Patent Owner argues that Petitioner’s proposed combination of Liu in view of HVR fails to teach “storing the first virtual-disk file that comprises the synthetic backup in a manner that enables at least one virtual machine to boot from the stored first virtual-disk file,” as recited in the challenged independent claims. PO Resp. 52. Specifically, Patent Owner argues that Petitioner’s proposed challenge fails because the backup image disclosed in Liu “is not made bootable until other steps are taken in the virtualization phase.” PO Resp. 53 (citing Ex. 1002 ¶ 84). “Moreover, Liu requires other actions in order to allow the virtual machine to boot.” *Id.*

Patent Owner’s arguments are not commensurate with the scope of the challenged claims, because we are not persuaded that the recited limitation regarding “storing the first virtual-disk file that comprises the synthetic backup in a manner that enables at least one virtual machine to boot from the stored first virtual-disk file” prohibits further steps to allow the virtual machine to boot. In fact, Petitioner argues that the later claim recitation regarding “retargeting the first empty virtual-disk file to provide a first retargeted virtual-disk file and to enable the at least one virtual machine to boot from the first virtual-disk file” identifies one additional step that allows

the virtual machine to boot. Pet. Reply 13. In view of this evidence, we are persuaded that Petitioner sufficiently establishes that Liu in view of HVR teaches or suggests “storing the first virtual-disk file that comprises the synthetic backup in a manner that enables at least one virtual machine to boot from the stored first virtual-disk file.”

d. Creating a First Empty Virtual-Disk File

Patent Owner argues that Petitioner fails to point to any disclosure in Liu or HVR that satisfies “creat[ing] a parent-child relationship between the first virtual-disk file and the first empty virtual-disk file.” PO Resp. 54. Patent Owner argues that Liu’s persistent storage 162 (“first empty virtual-disk file”), is not empty when the parent child relationship is created. PO Resp. 55 (citing Ex. 1006, 7:10–12). As discussed above with respect to Patent Owner’s arguments against the combination of Colbert and Lim, in view the disclosures in the ’168 patent, we are not persuaded by Patent Owner’s proposal that the independent claims require that the “first empty virtual-disk file” must remain empty before creating a parent-child relationship. Accordingly, we are persuaded that Petitioner sufficiently establishes that Liu teaches or suggests the claimed “first empty virtual-disk file.”

e. Retargeting the First Empty Virtual-Disk File

Patent Owner argues that Petitioner fails to point to any disclosure in Liu or HVR that satisfies “retargeting the first empty virtual-disk file to provide a first retargeted virtual-disk file and to enable the at least one virtual machine to boot from the first virtual-disk file,” as recited in the challenged independent claims. PO Resp. 55. Patent Owner argues that “retargeting” requires taking an action that enables the virtual machine to

boot from the virtual-disk file. *Id.* at 56. Subsequently Patent Owner argues, however, that “Liu requires a series of actions to enable booting the virtual machine.” *Id.* at 57. It is these additional actions that Petitioner relies upon as teaching the claimed retargeting step. Pet. 45; Pet. Reply 14.

Petitioner alleges that Liu discloses that virtual machines boot from the backup image 161, but any modifications to the backup image are stored in an empty disk, the persistent storage 162. Pet. 45 (citing Ex. 1006, 6:10–16). Petitioner argues that Liu describes reconfiguring the empty disk to enable a virtual machine to boot by disclosing that “[i]n order that these virtual machines may boot, the boot and system drives are impregnated with boot records appropriate for the Operating System.” *Id.* (citing Ex. 1006, 7:1–2). Accordingly, we are persuaded that Petitioner sufficiently establishes that Liu teaches or suggests the claimed “retargeting.”

f. Dependent Claim 3

Dependent claim 3 requires that the “first virtual-disk file comprises a first incremental backup of the data-storage entity” and the “second virtual-disk file comprises a second incremental backup of the data-storage entity.” Petitioner alleges that Liu’s “secondary image” corresponds to the claimed incremental backup first virtual-disk file. Pet. 47 (citing Ex. 1006, 4:29–30). Specifically, Petitioner cites to the disclosure in Liu that: “[a]fter the first backup, only changes since the last backups are captured and applied to the existing secondary image.” *Id.* Petitioner argues that because the changes constitute an incremental backup and are applied to the secondary image after they are captured, the secondary image includes the changes after they have been applied. *Id.* Petitioner further alleges that although Liu does not disclose explicitly a virtual-disk file, HVR teaches such files. *Id.* at 48.

Patent Owner argues that the secondary image 161 in Liu is a full backup, not the incremental backup required by claim 3. PO Resp. 58. Furthermore, Patent Owner argues that Petitioner relies upon the secondary image 161 in its challenge to claim 2 to meet the claimed “first virtual-disk file comprises a full backup of the data-storage entity.” *Id.* (citing Pet. 46). We agree that the secondary image 161 disclosed in Liu is a full backup, not an incremental backup.

In its reply, Petitioner states that it “did not ‘state that the secondary image 161 . . . comprises an incremental backup’” but rather stated that “‘the changes constitute an incremental backup file,’ and further relied on H[V]R to teach the *second incremental backup*.” Pet. Reply 15 (citing Pet. 47–48) (emphasis added). Contrary to Petitioner arguments, the only allegations in Petitioner’s challenge to claim 3 regarding HVR are with respect to the claimed “second virtual-disk file,” in which Petitioner argues that HVR teaches that “changes are stored in a virtual-disk file.” See Pet. 48. Petitioner fails to provide any arguments that HVR teaches the “first virtual-disk file comprises a first incremental backup of the data-storage entity,” as recited in claim 3. Accordingly, we are not persuaded that Petitioner has established sufficiently by a preponderance of the evidence that claim 3 would have been obvious in view of Liu and HVR.

g. Conclusion With Respect to the Asserted Ground of Obviousness In View of Liu and HVR against Claims 1, 2, 4, 9–12, and 14

With respect to the Petitioner’s challenge that claims 1, 2, 4, 9–12, and 14 would have been obvious in view of Liu and HVR, we have considered the papers submitted by the parties and the evidence cited therein. For the foregoing reasons, we determine that Petitioner has

demonstrated by a preponderance of the evidence that claims 1, 2, 4, 9–12, and 14 of the '168 patent would have been obvious in view of Liu and HVR.

III. CONCLUSION

Petitioner has shown by a preponderance of the evidence that claims 1–4, 9–12, and 14 of the '168 patent would have been obvious in view of Colbert and Lim; and Petitioner has shown by a preponderance of the evidence that claims 1, 2, 4, 9–12, and 14 of the '168 patent would have been obvious in view of Liu and HVR.

IV. ORDER

For the reasons given, it is

ORDERED that, based on Petitioner's showing by a preponderance of the evidence, claims 1–4, 9–12, and 14 of the '168 patent are unpatentable;

FURTHER ORDERED that this is a final written decision and that parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2014-00091
Patent 8,117,168 B1

For PETITIONER:

Lori A. Gordon
Michael Q. Lee
Jonathan M. Strang
Daniel S. Block
STERNE, KESSLER, GOLDSTEIN
& FOX PLLC
lgordon-PTAB@skgf.com
mlee-PTAB@skgf.com
jstrang-PTAB@skgf.com
dblock-PTAB@skgf.com

For PATENT OWNER:

Joseph J. Richetti
Lawrence G. Kurland
Daniel A. Crowe
BRYAN CAVE LLP
joe.richetti@bryancave.com
lgkurland@bryancave.com
dacrowe@bryancave.com