

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

AVAGO TECHNOLOGIES INTERNATIONAL SALES PTE. LIMITED,
Patent Owner.

Inter Partes Review No. IPR2020-01582

U.S. Patent No. 8,572,138 B2

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND	2
	A. Procedural History.....	2
	B. The '138 Patent	2
III.	CLAIM CONSTRUCTION	5
IV.	LEGAL STANDARDS	6
	A. <i>Inter Partes</i> Review.....	6
	B. Anticipation	6
	C. Obviousness.....	7
V.	PETITIONER'S ALLEGATIONS FAIL FOR LACK OF SUPPORT	9
	A. The Petition Does Not Properly Define Applicable Legal Standards for Anticipation and/or Obviousness.....	9
	B. No Combination of the Cited References Discloses or Suggests Every Limitation of Any Challenged Claim.....	10
	1. Challenged Claims	10
	2. Ground 1: Khandekar.....	16
	a) Background on Khandekar	16
	b) Advantages of the Challenged Claims over Khandekar.....	19
	c) Khandekar Fails to Disclose a Number the Elements of the Challenged Claims.....	21
	3. Ground 2: Khandekar and Le.....	29
	4. Ground 3: Khandekar, Le, and Sidebottom	31
	5. Ground 4: Khandekar, Le, and Stone	31
VI.	CONCLUSION.....	31
VII.	CERTIFICATION PURSUANT TO 37 C.F.R. § 42.24(D)	32

LIST OF EXHIBITS

1001	U.S. Patent No. 8,572,138 ("the '138 Patent")
1002	Prosecution File History for the '138 Patent
1003	Declaration of Prashant Shenoy
1004	U.S. Patent No. 7,577,722 to Khandekar et al. ("Khandekar")
1005	U.S. Patent No. 8,209,680 to Le et al. ("Le")
1006	U.S. Patent No. 7,962,633 to Sidebottom et al. ("Sidebottom")
1007	U.S. Patent Application Publication No. 2003/0036886 to Stone ("Stone")
1008	Khanna et al., <i>Application Performance Management in Virtualized Server Environments</i> , NOMS 2006 ("Khanna")
1009	Urgaonkar et al., <i>Dynamic Provisioning of Multi-tier Internet Applications</i> , ICAC '05 ("Urgaonkar")
1010	Excerpt of S. Russell, P. Norvig, <i>Artificial Intelligence - A Modern Approach</i> (2d. ed. Prentice-Hall 2003) ("Russell")
1011	Barham et al., <i>Xen and the Art of Virtualization</i> , SOSP '03: ACM Symposium on Operating Systems Principles, 164-177 (Oct. 2003)
1012	U.S. Patent No. 6,496,847 to Bugnion et al. ("Bugnion")
1013	U.S. Patent Application Publication No. 2004/0019672 to Das et al. ("Das")
1014	U.S. Patent Application Publication No. 2007/0220246 to Powell et al. ("Powell")
1015	JP Patent Application Publication No. 2001-318797 to French
1016	English translation of JP Patent Application Publication No. 2001-318797 to French ("French")
1017	Declaration of Sylvia D. Hall-Ellis

Patent Owner Avago Technologies International Sale Pte. Limited ("Avago" or "Patent Owner") hereby respectfully submits this Preliminary Response to the Petition for *Inter Partes* Review Case No. IPR2020-01582 (the "Petition"). This filing is timely under 35 U.S.C. § 313 and 37 C.F.R. § 42.107(b), as it is being filed within three months of the September 21, 2020 mailing date of the Notice of Filing Date Accorded to Petition and Time for Filing Patent Owner Preliminary Response (Paper 4).

The Patent Trial and Appeal Board (the "Board") should decline to institute an *inter partes* review in this matter because none of the references or combinations of references relied upon by Netflix, Inc. ("Petitioner") establish a reasonable likelihood that Petitioner will prevail with respect to any challenged claim of U.S. Patent No. 8,572,138 (the "'138 Patent").

I. INTRODUCTION

The Petition challenges the validity of claims 1-5, 9, 10, 17, 19-21, and 26 of the '138 Patent (the "Challenged Claims"). "The Director may not authorize an *inter partes* review to be instituted unless the Director determines that the information presented in the petition filed under section 311 . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged" 35 U.S.C. § 314(a). As explained more fully below, *inter partes* review should not be instituted because Petitioner fails to demonstrate that there is a

reasonable likelihood that at least one of the Challenged Claims of the '138 Patent is unpatentable. First, the Petition does not properly define and apply the applicable legal standards for obviousness, as required by 37 C.F.R. § 42.22(a), and, thus, is procedurally deficient. Moreover, the Grounds proposed in the Petition fail to establish that the references teach each and every feature arranged as recited by the respective claims of the '138 Patent and, thus, do not meet the Petitioner's burden to establish a *prima facie* case that any challenged claim is unpatentable.

II. BACKGROUND

A. Procedural History

Broadcom Corporation ("Broadcom") and Avago filed a lawsuit against Petitioner for infringement of the '138 Patent and eleven other patents. *See Broadcom Corp. et al. v. Netflix, Inc.*, No. 8:20-cv-00529 (C.D. Cal. 2020). The case was subsequently transferred to the N.D. Cal. on July 14, 2020, Case No. 3:20-cv-04677 (N.D. Cal. 2020). Petitioner filed the Petition on September 10, 2020. The '138 Patent has not been previously asserted in any related lawsuits.

B. The '138 Patent

In order to better understand the inventions set forth in the Challenged Claims, the following is an overview of some of the basic components of these various inventions.

VM: The inventions of the '138 Patent generally relate to deployment of virtual machines ("VMs") on nodes in a distributed computing system. *See* Ex. 1001 ('138 Patent) at Abs. A VM is a virtual environment that emulates a physical computer platform. *See id.* at 32:67-33:2. A distributed computing system refers to a collection of independent networked computers (*i.e.*, nodes) that are configured to function (or to appear to users) as a single coherent computing system. *See id.* at 1:18-25. Each networked computer or node may be a physical machine with physical hardware and ports. *See id.* at 32:65-33:24. The nodes are the target locations for deploying VMs. *See, e.g., id.*

VMM: A virtual machine manager ("VMM") is a computer software, firmware, or hardware that is used to manage and, as set forth in the Challenged Claims, to provide the VMs. *See id.* at 32:59-67; *see also id.* at 35:6-9.

Software Applications: A VM without any executable software applications on it is merely a non-functional shell or unconfigured machine. *See id.* at 32:65-33:24. To provide a working machine, software applications are deployed or installed on the VMs. *See, e.g., id.* For example, to emulate a home computer, software applications including, but not limited to, Microsoft Windows 10 operating system ("OS"), Microsoft Office Suite of applications (*e.g.*, WORD, EXCEL, OUTLOOK, etc.), an Internet browser, etc., may be deployed on the VM. *See id.*

Image Instances: In some cases, VMs operate by instantiating a software image. *See id.* Generally, an image of a software application is a file that contains a snapshot or "image" of an installed and configured software application. *See id.* at 35:1-24. The file may also contain established or selected settings and/or preferences. *See id.* at 33:25-67. An instance of such an image is a copy of the image deployed onto a particular VM. *See id.* at 32:65-33:24. In this manner, deployment of the image instance of the software application can avoid the time-consuming and tedious task of installing each software application on each VM, and thereafter selecting the desired settings and/or preferences for each installed software application. *See id.* at 33:25-36:20.

In a non-limiting example, a first user, who has already installed and selected optimal settings and/or preferences for the operating system and tailored software applications for performing at least one set of desired functions, can take a "snapshot" of the system to create an image of the operating system with the installed collection of software applications and then send an instance of that image to a second user. *See id.* at 33:25-67, *see also id.* 35:1-24. Upon deploying (and in some cases, executing) the image instance of the collection of software applications, the second user would immediately be provided with a collection of software applications with the same functionalities, settings, and preferences as the software applications of the first user. *See id.* at 35:1-24. Accordingly, the second user does

not need to install the collection of software applications or to manually set the settings and preferences to match those that were set by the first user. *See id.*

The Challenged Claims of the '138 Patent contain these various components and claim specific aspects related to the deployment of VMs on nodes within a distributed computing system. *See, e.g., id.* at 36:26-49. As discussed in detail below, these claimed inventions of the '138 Patent are fundamentally different than – and provide significant advantages over – the prior art relied on in the Petition.

III. CLAIM CONSTRUCTION

The Petitioner seeks a proposed construction only for the term "a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines." *See* Pet. at 13. The Patent Owner does not agree with Petitioner's specific construction, but instead submits that this term should be construed in accordance with its plain and ordinary meaning. Because, however, the construction of this term is not material as to the reasons why the Board should deny institution of *inter partes* review as set forth in this Preliminary Response, Patent Owner will not address Petitioner's claim construction as part of this Preliminary Response.

IV. LEGAL STANDARDS

A. *Inter Partes* Review

To institute an *inter partes* review, the Board must find a "reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition." 35 U.S.C. § 314(a). Petitioner therefore carries the burden to "demonstrate that there is a reasonable likelihood that at least one of the claims challenged in the petition is unpatentable." 37 C.F.R. § 42.108(c).

The Petition must include "[a] full statement of the reasons for the relief requested, including a detailed explanation of the significance of the evidence including material facts, and the governing law, rules, and precedent." 37 C.F.R. § 42.22(a)(2); *see also* 37 C.F.R. § 42.104.

B. Anticipation

"To establish anticipation, each and every element in a claim, arranged as is recited in the claim, must be found in a single prior art reference." *ZTE Corp. v. ContentGuard Holdings Inc.*, IPR 2013-00134, Paper 12 at 24 (PTAB June 19, 2013). "[To anticipate,] [t]here must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention." *Scripps Clinic & Res. Found. v. Genentech, Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991). Thus, the Board "must analyze prior art references as a

skilled artisan would." *Microsoft Corp. v. Proxyconn, Inc.*, IPR2012-00026, Paper 73 at 33 (PTAB Feb. 19, 2014).

C. Obviousness

In addition to being novel, a valid claim must not have been obvious to a person with ordinary skill in the art at the time of the invention. *See, e.g., Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561, 1565-68 (Fed. Cir. 1987). In determining whether a claim is obvious, the Board considers (1) the level of ordinary skill in the pertinent art; (2) the scope and content of the prior art; (3) the differences between the claimed invention and the prior art; and (4) objective secondary considerations of non-obviousness, if any. *See, e.g., SIBIA Neurosciences, Inc. v. Cadus Pharm. Corp.*, 225 F.3d 1349, 1355 (Fed. Cir. 2000); *In re Gartside*, 203 F.3d 1305 (Fed. Cir. 2000). The person of ordinary skill in the art ("POSA") is a hypothetical person who is presumed to know the relevant prior art. *See Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.*, 807 F.2d 955, 962 (Fed. Cir. 1986).

Importantly, "[a] patent composed of several elements is not proved obvious merely by demonstrating that each element was, independently, known in the prior art." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 401 (2007). Rather, to establish *prima facie* obviousness, the cited references must be shown to disclose or suggest each claimed element *and* the Petition must establish that it would have been obvious to combine the teachings in the references together to arrive at the claimed invention.

See In re Lee, 277 F.3d 1338, 1343 (Fed. Cir. 2002). The Petition must demonstrate an apparent reason to modify a reference, or to combine the cited references to create the specific invention.

The rationale asserted by the Petition must follow from the nature of the problem to be solved, it must be "clear and particular, and it must be supported by actual evidence." *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1334 (Fed. Cir. 2002). The combined art must narrow the scope of the inventor's task to the point where it would have been obvious to try the particular invention claimed. If the particular invention lies hidden in a multitude of other options suggested by the art, the invention is not obvious. *See, e.g., Leo Pharm. Prods. Ltd. v. Rea*, 726 F.3d 1346, 1356 (Fed. Cir. 2013); *In re Kubin*, 561 F.3d 1351, 1359-60 (Fed. Cir. 2009).

Hindsight is forbidden in an obviousness analysis. *See In re Dembiczak*, 175 F.3d 994, 998-999 (Fed. Cir. 1999), *abrogated on other grounds by In re Gartside*, 203 F.3d 1305 (Fed. Cir. 2000); *see also Kinetic Concepts, Inc. v. Smith & Nephew, Inc.*, 688 F.3d 1342, 1368 (Fed. Cir. 2012). Therefore, the reasons for combining references or modifying the teachings of a reference must be apparent at the time of the invention, and they must be apparent without the use of hindsight. A telltale sign of impermissible hindsight analysis is the "use [of] the invention to define the problem that the invention solves." *Mintz v. Dietz & Watson, Inc.*, 679 F.3d 1372, 1376-78 (Fed. Cir. 2012) (reversing a district court that employed hindsight analysis

because the district court's obviousness finding was improperly based on art directed to the solution as opposed to art directed to the problem).

V. PETITIONER'S ALLEGATIONS FAIL FOR LACK OF SUPPORT

A. The Petition Does Not Properly Define Applicable Legal Standards for Anticipation and/or Obviousness

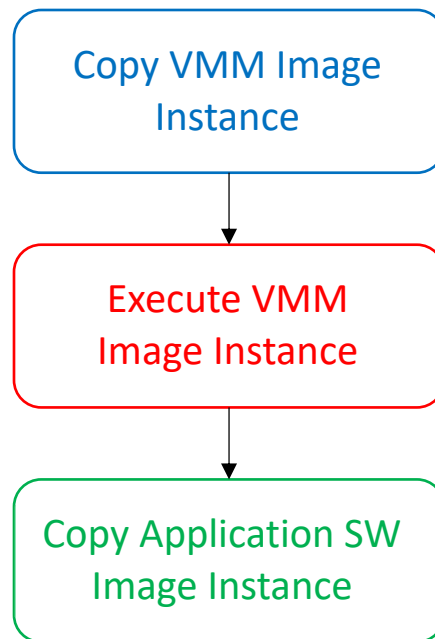
Petitioner alleges throughout the Petition that all the claims are obvious over various references. Notably absent from the Petition, however, is any statement of the applicable legal standards under which these allegations should be judged. While Petitioner's expert, Dr. Prashant Shenoy, recounts his general understanding of patent law in his declaration, he does not identify the specific governing law, rules, and precedent supporting his opinions. *See* Ex. 1003, App. 4 at 212-224. Nor is he a lawyer or agent admitted to practice before the USPTO or the Board.

Petitioner's failure to identify any applicable legal standards renders the Petition insufficient under 37 C.F.R. § 42.22(a). Dr. Shenoy's declaration cannot remedy the defects in the Petition and cannot provide competent testimony as to the governing law, rules, and precedent. Moreover, the Petition itself does not even refer to, much less adopt, Dr. Shenoy's understanding of any legal principles, and it therefore is devoid of any analysis of the legal standards for anticipation or obviousness. Because of this omission, the Petition should be denied as to all Challenged Claims.

B. No Combination of the Cited References Discloses or Suggests Every Limitation of Any Challenged Claim

1. Challenged Claims

The Challenged Claims cover specific aspects of establishing VMs that are a novel and non-obvious improvement over the prior art. The '138 Patent discloses and claims inventions in which VMs are established from a software image repository that stores both (i) a plurality of **image instances** of **VMMs** and (ii) a plurality of **images instances** of a plurality of **software applications**. *See supra* at 3-4 (*referencing* explanations of bolded terms). A control node containing an automation infrastructure causes both the autonomic deployment of individual VMs by using image instances of a VMM as well as the autonomic deployment of image instances of software applications to those VMs. Specifically, the Challenged Claims cover inventions where the process of installing a VM and the additional application software is divided into three steps: (1) copying a VMM image onto a target node; (2) executing the VMM image to create a working VM environment; and (3) copying a software application image into the VM environment. The following is flow chart summarizing these three steps, which is color coded to match relevant claim language below:



In this regard, Claim 1 is instructive:

1. A distributed computing system comprising:
 - a software image repository comprising non-transitory, computer-readable media operable to store: (i) a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes, wherein **when executed on the applications nodes, the image instances of the virtual machine manager provide a plurality of virtual machines**, each of the plurality of virtual machine operable to provide an environment that emulates a computer platform, and (ii) a plurality of image

instances of a plurality of software applications that are executable on the plurality of virtual machines; and

a control node that comprises an automation infrastructure to provide autonomic deployment of the plurality of image instances of the virtual machine manager on the application nodes by causing the plurality of image instances of the virtual machine manager to be copied from the software image repository to the application nodes and to provide autonomic deployment of the plurality of image instances of the software applications on the virtual machines by causing the plurality of image instances of the software applications to be copied from the software image repository to the application nodes.

Thus, in addition to other features, Claim 1 requires the control node to perform three fundamental operations, corresponding to the illustration and claim limitations above:

1. causing the plurality of image instances of the virtual machine manager to be copied from the software image repository to the application nodes;

2. execut[ing], on the applications nodes, the image instances of the virtual machine manager [to] provide a plurality of virtual machines; and
3. provid[ing] autonomic deployment of the plurality of image instances of the software applications on the virtual machines by causing the plurality of image instances of the software applications to be copied from the software image repository to the application nodes.

It is important to note that step 3 requires the image instances of the software applications to be deployed on the VMs. The VMs themselves are only created (provided) when the VMM image instances are executed. Thus, the VMM image instances must be executed (step 2) before the image instances of the software applications can be deployed. As such, Claim 1 recites a novel technique in which a first image instance (the VMM image instance) is deployed to a machine and executed to provide a VM that acts as a host for a second image instance (the software application image instance), which can then be deployed to the VM.

Claims 19 and 26 require the same general process. These Claims, in pertinent part, are reproduced below:

19. A method comprising:

storing a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes in a distributed computing system, wherein the image instances of the virtual machine manager provide a plurality of virtual machines, each virtual machine operable to provide an environment that emulates a computer platform;

storing a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines;

executing a rules engine to perform operations to autonomically deploy the image instances of the virtual machine manager on the application nodes by causing the image instances of the virtual machine manager to be copied to the application nodes; and

executing the rules engine to perform operations to autonomically deploy the image instances of the software applications on the virtual machines by causing the image instances of the software applications to be copied to the application nodes.

26. A non-transitory, computer-readable medium comprising instructions stored thereon, that when executed by a programmable processor:

store a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes in a distributed computing system, wherein the image instances of the virtual machine manager provide a plurality of virtual machines, each virtual machine operable to provide an environment that emulates a computer platform;

store a plurality of image instances of a plurality of software applications that are executable on the plurality of virtual machines;

perform operations to autonomically deploy the image instances of the virtual machine manager on the application nodes by causing the image instances of the virtual machine manager to be copied to the application nodes; and

perform operations to autonomically deploy the image instances of the software applications on the virtual machines by causing the image instances of the software applications to be copied to the application nodes.

Thus, each of these independent Challenged Claims requires that VMM image instances be deployed, and then that software application image instances be deployed onto VMs created by the VMM image instances. But the claims also

specify that the VMM image instances provide the VMs by executing on the application nodes. In other words, the VMM image instance must be deployed and executed before the software application image instance can be deployed onto the VM.

2. Ground 1: Khandekar

Petitioner identifies Khandekar as the primary reference for each of its four Grounds. *See* Pet. at 9. With respect to Ground 1, Petitioner alleges that Khandekar renders independent Claims 1, 19, and 26, and dependent Claims 2-5, 9, 20, and 21, obvious under 35 U.S.C. § 103. *See id.* That allegation is incorrect, and it evidences a fundamental misunderstanding of the claim limitations of the '138 Patent as well the disclosure in Khandekar.

a) Background on Khandekar

In general, Khandekar discloses "a method and system implementation for creating a virtualized computer system based on input information identifying a desired computer configuration." *See* Ex. 1004 (Khandekar) at 4:30-32. While there are many differences between the implementation of the virtualized computer systems disclosed in Khandekar and the '138 Patent, one of the clearest examples is with respect to the deployment of the VMM image instances.

Specifically, Khandekar provides little disclosure of how the VMM image instances are to be deployed under its disclosure. Indeed, the totality of this disclosure occurs in following sections of Khandekar.

In the following description of the invention, merely for the sake of simplicity, only one VM/VMM pair is discussed. The discussion applies equally, however, to all such VM/VMM pairs that may be included in any given implementation of the invention. Moreover, **it will be assumed below that when a VM is installed on a host, then a corresponding VMM is either already installed on the host, or is included and installed together with the VM.**

See id. at 8:36-43 (emphasis added).¹

As is mentioned above, a VMM is typically included for each VM in order to act as its software interface with the underlying host system. **In the preferred embodiment of the invention, a VMM is already pre-installed on each host 1000, 1001 so as to ensure compatibility and proper configuration.** Several products are available from VMware, Inc. of Palo Alto, Calif., that configure

¹ For clarification, it should be noted that Khandekar refers to the software image instances themselves as VM disk files, or, more briefly, VMs. This usage is somewhat unclear, in that Khandekar thus uses the term "VM" to refer both to the software image instance and to the virtualized computing platform that can be created from that software application image instance. The '138 Patent uses the term "VM," as noted above, to refer to a virtualized computing platform into which a software application image instance can be deployed.

conventional computers, both stand-alone desktop computers and multi-user servers, to load and run multiple VMs. **It would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known. An advantage of pre-installing a VMM, without a VM, on a plurality of hosts, such as hosts 1000, 1001, 1003, is that it will then be possible to deploy a given VM on any arbitrary one (or more) of these hosts. Moreover, as long as the VMM on each host exports a known hardware interface to the VM deployed on it, then VM deployment will be substantially independent of the actual physical hardware configuration.**

See id. at 13:17-35 (emphasis added).

Therefore, to summarize, Khandekar discloses two potential ways to deploy the VMM image instances. **First**, Khandekar discloses that a VMM is "pre-installed" on the host computer. *See id.* Khandekar prefers this method of pre-installing a VMM in order to ensure that the VMM will correctly match with the actual physical hardware configuration of the host. *See id.* This is important because, as Khandekar explains, the VMM is the "interface" that "runs directly on top of the host and virtualizes all, or at least some, of the resources of the machine." *Id.* at 7:5-9. **Second**, Khandekar mentions that "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the

characteristics of the host on which the VM is to be installed are known." *Id.* at 13:25-28. However, because the characteristics of the host are often not known when attempting to establish the VM, Khandekar states that this is not the preferred method of deploying the VMM. *Id.* at 13:25-35. In both cases, however, it is undisputed that Khandekar provides no disclosure as to where the VMM image instance is stored prior to its installation.

Finally, while Petitioner spends significant time discussing various embodiments of the Khandekar disclosure for creating different types of VMs such as Pre-Built VMs, Staged VMs, Custom-Built VMs, and Instantly-Deployable VMs (*see, e.g.*, Pet. at 41-47), it is important to note that these variations have nothing to do with the VMM, which in each case is either pre-installed (*i.e.*, the preferred method) or installed with the VM itself, as discussed above. *See* Ex. 1004 (Khandekar) at 8:36-44.

b) Advantages of the Challenged Claims over Khandekar

Before discussing the individual limitations of the Challenged Claims that Khandekar fails to disclose, it is worth noting that the techniques recited by the Challenged Claims provide significant advantages over the prior art, including Khandekar. Essentially, the methodology of the '138 Patent provides for separate remote deployment of a VMM image instance and a software application image instance. Under the system of the '138 Patent, a VMM image instance can be

provided to correspond to each known physical or virtual configuration of the application nodes. *See* Ex. 1001 ('138 Patent), Fig. 25; *see also id.* at 34:48-67 (describing a process for capturing a VMM "golden image" for a particular node configuration). A software application image instance can be provided to correspond to each type of VM environment needed (with different operating systems, applications, etc.). *See id.* at 34:30-49 (describing different operating systems); *see also id.* at Fig. 26 and 35:1-24 (describing a process for capturing "a golden image" of an operating system and applications). Then, the appropriate VMM image instance for the application node can be selected, and the appropriate software application for the VM can be selected. *See id.*, Figs. 27A-27B; *see also id.* at 35:25-67 (describing a process of selecting and deploying VMM image instances and software application image instances). The selected VMM image instance can be copied to the application node and executed to provide the VM, and then the selected software application image instance can be copied onto the VM. *See id.*

This technology provides significant scalability enhancements over any of the techniques taught by Khandekar. For example, consider the alternative embodiment of Khandekar in which the VMM is included with the VM. In an environment with, for example, 100 different physical hardware configurations of host nodes and 100 different software configurations of VMs, a system in which a VMM and VM were

combined would require storing 10,000 images to cover every possible configuration (*i.e.*, 100 image instances of each of the VMM configurations to correspond to each of the 100 VM configurations, multiplied by the 100 VMM configurations).

By contrast, the systems claimed in the '138 Patent, by abstracting the software application image instances from the VMM image instances, would require storage of only 200 images. This is because the VMM image instance can be selected independent of, and then paired with, the software application image instance. This advancement is not possible with any of the embodiments described by Khandekar. As noted above, Khandekar acknowledges that it is better to pre-install a VMM on each host (which is highly inefficient) than attempt to deploy a VMM image instance with a software application image instance, because, according to Khandekar, the latter technique cannot accommodate differing configurations of the underlying nodes.

c) Khandekar Fails to Disclose a Number the Elements of the Challenged Claims

Because of these substantial differences between systems claimed in the '138 Patent and Khandekar, it is not surprising that Khandekar fails to satisfy many of the elements of Challenged Claims.

(1) Khandekar fails to disclose a "software image repository" that comprises "a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes"

First and foremost, Khandekar does not even remotely suggest a "software image repository" that comprises "a plurality of image instances of a virtual machine manager that is executable on a plurality of application nodes" as set forth in Claim 1. *See* Ex. 1001 ('138 Patent) at 36:27-30. Rather, Khandekar states that its preferred method is to have the VMM "pre-installed on each host." *See* Ex. 1004 (Khandekar) at 13:17-35. Petitioner did not rely on this preferred method of deploying the VMM to satisfy this element of Claim 1 because, in this embodiment, the VMM is preinstalled on the host, not deployed from a plurality of VMM image instances stored in a software image repository.

Instead, Petitioner relies on the single sentence in Khandekar that, "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known" (*see id.* at 13:25-28), to claim that "a POSITA would have found it obvious to likewise store a plurality of VMM images in the provisioning

server so they may be deployed together to hosts, as taught by Khandekar." *See* Pet. at 27.²

Petitioner supports its position with the following paragraph from its expert's declaration that states:

A POSITA would have found it obvious for software image repository to be operable to store a plurality of image instances of a virtual machine manager (e.g., VMM) so that VMM images may be deployed with the VM images stored in the repository, as taught by Khandekar. *See id.* A POSITA would have found it obvious to follow these teachings from Khandekar, with VMM images being stored in the provisioning server and provided to hosts together with VM images, because doing so would have allowed the hosts to be remotely initialized with VMM software and configured to support virtual machines, thereby facilitating one of Khandekar's goals: automated, flexible deployment of virtual machines to physical host platforms.

See Ex. 1003 at 51-52, ¶ 83 (quoting Ex. 1004 (Khandekar) at 8:36-44, 13:17-35) (quotations omitted).

This analysis is fatally deficient for two reasons. First, the proposition espoused by Petitioner's expert, that "A POSITA would have found it obvious to

² This Preliminary Response, like the Petition, uses the term "POSITA" to refer to "a person of ordinary skill in the art."

follow these teachings from Khandekar, with VMM images being stored in the provisioning server and provided to hosts together with VM images," *id.*, finds no support whatsoever in the actual disclosure of Khandekar. The isolated, single sentence of disclosure in Khandekar, that "[i]t would also be possible, however, for a corresponding VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known," Ex. 1004 (Khandekar) at 13:25-28, provides no teaching or suggestion about how a VMM might be stored, let alone teaching "VMM images being stored in the provisioning server." Ex. 1003 at 51-52, ¶ 83. This analysis is unfounded speculation by Petitioner's expert, rather than any teaching from Khandekar.

Second, this analysis is nothing but plain hindsight, using the claims of the '138 Patent themselves as a roadmap to build the alleged rationale for the modification of Khandekar from whole cloth. As the International Trade Commission has explained,

The point of examining whether a motivation to combine exists is to prevent an improper hindsight analysis. "A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning." Thus the motivation to combine cannot come from the actual patent that [the challenger] is attempting to invalidate; it must come from a time prior to the [patent at issue]. Otherwise

[the patent at issue] would be used as a roadmap to combine obviousness references.

In the Matter of Certain Self-Cleaning Litter Boxes & Components Thereof, USITC Inv. No. 337-TA-625, Initial Determination on Violation of Section 337 & Recommended Determination on Remedy & Bond at 51 (Dec. 1, 2008) (quoting *KSR v. Teleflex*, 550 U.S. at 421; citing *Ruiz v. A.B. Chance Co.*, 357 F.3d 1270, 1275 (Fed. Cir. 2004)) (citations omitted).

Such hindsight reasoning is strictly forbidden in an obviousness analysis. *See Kinetic Concepts, Inc. v. Smith & Nephew, Inc.*, 688 F.3d 1342, 1368 (Fed. Cir. 2012). First, while Khandekar states that it would be "possible" for the "VMM to be included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known" (*see* Ex. 1004 (Khandekar) at 13:25-28), Khandekar immediately teaches away from this possibility in the very next sentence by stating, "**[a]n advantage of pre-installing a VMM, without a VM**, on a plurality of hosts, such as hosts 1000, 1001, 1003, is that it will then be possible to deploy a given VM on any arbitrary one (or more) of these hosts." *See id.* at 13:28-32. This teaching away from the alternative embodiment negates any argument that a POSITA would find reason to modify Khandekar as suggested by Petitioner.

Neither Petitioner nor its expert address how a POSITA would decide to store a plurality of image instances of a VMM on a software repository **based on any art**

preceding the teachings of the '138 Patent. This shortcoming is particularly troublesome in light of the fact Khandekar (the only prior art relied on by the Petitioner for this particular aspect of the claim) never discloses that a plurality of image instances of a VMM could be stored on a software repository and expressly teaches away from deploying the VMM image instances with the VM. To make matters worse, Petitioner's expert essentially mirrors the claim language from the Challenged Claims to support his position. *See* Ex. 1003 at 51, ¶ 83 (arguing that "[a] POSITA would have found it obvious for the software image repository to be operable to store a plurality of image instances of a virtual machine manager"). This is a classic indicator of hindsight reasoning. *See Mintz v. Dietz & Watson, Inc.*, 679 F.3d 1372, 1377 (Fed. Cir. 2012) (holding that a telltale sign of impermissible hindsight is the "use [of] the invention to define the problem that the invention solves"). Indeed, rather than relying on anything in Khandekar for this reasoning, Petitioner's expert appears to be reading from the disclosure of the '138 Patent, which explains that, by deploying VMM image instances and software application image instances separately,

distributed computing system 10 may autonomically utilize application nodes, e.g., node 400, to flexibly support a wide variety of operating systems and applications, and provide virtualized execution environment for those operating systems and applications.

For instance, node 400 may autonomically configured to Support a Macintosh Operating System designed for a PowerPC processor, and a Microsoft Windows Operating System designed for an x86 processor based on the particular goals of the autonomic system. Thus, control node 12 may not need to distinguish between hardware processor architectures when deploying operating systems and applications.

See Ex. 1001 ('138 Patent) at 34:30-41.

Rather than any teaching from Khandekar itself (or, for that matter, any other prior art), Petitioner and its expert rely on the claims and disclosure of the '138 Patent to propose extensive modifications to an alternative embodiment of Khandekar. Further, Khandekar discloses this alternative embodiment in a single sentence and then immediately teaches away from it. Such hindsight reasoning cannot support a *prima facie* argument that a POSITA would find obvious from Khandekar the storage of multiple VMM image instances in a software repository for deployment to application nodes.

- (2) Khandekar fails to disclose (1) copying a VMM image onto a target node; (2) executing the VMM to create a working VM environment; and (3) copying a software application image into the VM environment**

Even if the Board were to accept Petitioner's blatant hindsight analysis with respect to the storage of the VMM, Khandekar still fails to disclose the process of

deploying the VMM as set forth in the Challenged Claims. As noted above, the Challenged Claims require a particularized deployment process that includes (1) copying a VMM image onto a target node; (2) executing the VMM to create a working VM environment; and (3) copying a software application image into the VM environment. *See supra* Section V.B.1.

Khandekar fails utterly to teach or suggest this sequence. Indeed, neither method of deploying the VMM disclosed in Khandekar (*i.e.*, either pre-installing the VMM or installing the VMM with the VM) discloses or suggests that a VMM image instance might be deployed and executed before the deployment of the VM itself. Indeed, as explained in detail above, the Challenged Claims recite the deployment of the plurality of image instances of the VMMs first (such that the VMM image instances can be executed to create the VMs) and then the deployment of software applications onto the VMs. *See id.* Khandekar's preferred method presumes that a VMM has been pre-installed before the deployment process even begins, which clearly fails to disclose any of these steps.

Moreover, Khandekar's bare statement about a VMM being "included along with each staged VM, as long as the characteristics of the host on which the VM is to be installed are known" (*see* Ex. 1004 (Khandekar) at 13:26-28) also does not disclose these claims elements because (1) this disclosure requires that the VMM is installed with the software application image instance (not installed prior to the

software application image instance),³ (2) the disclosure provides no suggestion that the VMM image instance is first executed to create the VM (as that term is used in the claims of the '138 Patent) prior to the installation of the software applications image instances, and (3) the disclosure fails to suggest that the software application image instances are deployed onto the VM provided by the VMM image instance, rather than just copied to the host.

Thus, Petitioner has not established that Khandekar discloses or suggests each limitation of any of independent Claims 1, 19, and 26. Ground 1, therefore, must fail with respect to those claims. The remaining claims ultimately depend from either Claim 1 or Claim 19 and are valid over Khandekar at least by virtue of their dependence. Accordingly, the Board should decline to institute on Ground 1.

3. Ground 2: Khandekar and Le

Petitioner further alleges that the combination of Khandekar and Le renders independent Claims 1, 19, and 26, and dependent Claims 2-5, 9, 20, and 21, obvious under 35 U.S.C. § 103. *See* Pet. at 9. As noted above, Khandekar fails to disclose or suggest each limitation of any independent claim of the '138 Patent. Le fails to remedy these deficiencies of Khandekar, and the Challenged Claims are equally valid over the combination of Khandekar and Le.

³ Again, to be clear, Khandekar refers to the software application image instances themselves as VM disk files, or, more briefly, VMs. *See supra* n.1.

With regard to the installation of a VMM image instance, Le teaches only that, "[t]o power on and run the cloned virtual machine, all that the host computer needs is to have the appropriate VMM software installed." *See* Ex. 1005 (Le) at 14:28-30. Thus, Le, like Khandekar, presumes that a VMM has already been installed when discussing deployment, and provides no enabling disclosure of how the pre-existing installation of a VMM might have occurred. Le also appears to teach that the "cloned virtual machine," which would correspond to the software application image instance, is copied to the host before the VMM image instance is executed. Certainly, the Petitioner fails to identify, and a careful review of Le fails to reveal, any teaching or suggestion of the three-step deployment process required by the independent claims of the '138 Patent, in which the VMM image instance is copied to the application node, the VMM image instance is executed to provide a VM, and a software application image instance is deployed onto the VM.

Accordingly, Le fails to supplement Khandekar in any material way with regard to independent Claims 1, 19, and 26, and those claims are valid over the combination of Khandekar and Le. The remaining claims challenged by Ground 2 are valid at least by virtue of their dependence from valid base claims. Accordingly, the Board should refuse to institute on Ground 2 as well.

4. Ground 3: Khandekar, Le, and Sidebottom

Petitioners further allege that Claims 9 and 10 are obvious under 35 U.S.C. § 103, over the combination of Khandekar, Le, and Sidebottom. *See* Pet. at 9. Petitioner does not allege that Sidebottom provides any material disclosure with regard to Claims 1, 19, and 26. Those claims, therefore, are valid over the combination of Khandekar, Le, and Sidebottom for the reasons discussed *supra* with regard to Grounds 1 and 2. Claims 9 and 10 are valid over the same combination at least by virtue of their dependence from Claim 1.

5. Ground 4: Khandekar, Le, and Stone

Petitioners further allege that Claim 17 is obvious under 35 U.S.C. § 103, over the combination of Khandekar, Le, and Stone. *See id.* Petitioner does not allege that Stone provides any material disclosure with regard to Claims 1, 19, and 26. Those claims, therefore, are valid over the combination of Khandekar, Le, and Stone for the reasons discussed *supra* with regard to Grounds 1 and 2. Claim 17 is valid over the same combination at least by virtue of its dependence from Claim 1.

VI. CONCLUSION

Avago respectfully submits that this Petition should be dismissed and/or denied for at least the foregoing reasons. Further, as this is Avago's Preliminary Response, it is not intended to be a comprehensive rebuttal to all contentions and arguments raised by the Petition. As such, if a trial is nonetheless instituted, Avago

reserves the right to contest the Petition on all grounds permitted under the applicable rules, including, without limitation, raising claim construction issues, challenging Petitioner's proffered expert testimony and submitting rebuttal expert testimony, challenging Petitioner's assertions that the claims of the '138 Patent are anticipated and/or obvious in light of its cited references, and any other permitted bases to rebut Petitioner's proffered evidence and arguments. Moreover, nothing herein is or should be construed as a concession or admission by Avago as to any fact or argument proffered in the Petition.

VII. CERTIFICATION PURSUANT TO 37 C.F.R. § 42.24(d)

Pursuant to 37 C.F.R. § 42.24(d), Patent Owner certifies that based on the "word count" feature of Microsoft Word, this Preliminary Response includes 7071 words.

Respectfully submitted,

Dated: December 21, 2020

By: /s/Daniel S. Young/

Daniel S. Young, Reg. No. 48,277
 Chad E. King, Reg. No. 44,187
 ADSERO IP LLC D/B/A/
 SWANSON & BRATSCHELL LLC
 8210 Southpark Terrace
 Littleton, CO 80120
 (303) 268-0066 (telephone)
 (303) 268-0065 (facsimile)

*Counsel for Patent Owner Avago Technologies
 International Sales Pte. Limited ("Avago")*

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6, the undersigned hereby certifies that a copy of the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** was filed through the Patent Trial and Appeal Board End to End (PTAB E2E) electronic filing system on December 21, 2020 with a confirmation copy served via electronic mail, on the following counsel of record for Petitioner:

Harper Batts
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: HBatts@sheppardmullin.com

Jeffrey Liang
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: JLiang@sheppardmullin.com

Chris Ponder
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: CPonder@sheppardmullin.com

SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
email: Legal-Netflix-Broadcom-IPRs@sheppardmullin.com

Dated: December 21, 2020

By: /s/Daniel S. Young/
Daniel S. Young
Registration No. 48,277