UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/642,656	03/09/2015	Mukund Gunti	C429	6624

139360 7590 10/24/2019

VMWARE C/O WAGNER BLECHER LLP
123 WESTRIDGE DRIVE
WATSONVILLE, CA 95076

EXAMINER	
CHEN, ZHAN	

ART UNIT	PAPER NUMBER
2194	

NOTIFICATION DATE	DELIVERY MODE
10/24/2019	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ipadmin@vmware.com
patents@wagnerblecher.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte MUKUND GUNTI, VISHNU SEKHAR, and BERNHARD POESS

Appeal 2018-003413
Application 14/642,656
Technology Center 2100

Before JUSTIN BUSCH, STEVEN J. AMUNDSON, and JASON M. REPKO, *Administrative Patent Judges.*

BUSCH, *Administrative Patent Judge.*

DECISION ON APPEAL

Pursuant to 35 U.S.C. § 134(a), Appellant[1] appeals from the Examiner's decision to reject claims 1–26, which constitute all the claims pending in this application. We have jurisdiction under 35 U.S.C. § 6(b).

We REVERSE.

---

[1] We use the word Appellant to refer to "applicant" as defined in 37 C.F.R. § 1.42(a). Appellant identifies the real parties in interest as Dell Technologies and VMware, Inc., a wholly owned subsidiary of Dell Technologies. Appeal Br. 3.

CLAIMED SUBJECT MATTER

Appellant's disclosure generally relates to "hot-swapping operating systems." Spec. ¶ 14, Abstract. The claimed invention relates to an original operating system instance generating and storing update launch code, quiescing resources assigned to a target partition where a new instance of the operating system (OS) will be run, and launching the new OS on the target system. *See* Spec. ¶¶ 19–21, 24–28. Then the original and new OSes move applications from the original OS to the new OS. Spec. ¶ 29. Claims 1, 9, and 18 are the independent claims. Claim 1 is reproduced below:

> 1. A process comprising:
>
> generating and storing, by an original operating-system (OS) instance controlling hardware devices of a computer, update launch code including,
>
> computer-device identifiers identifying a computer set of devices included in the computer,
>
> partition-device identifiers identifying a target subset of the computer set, the target subset including devices to be initially included in a target logical partition of the computer, and
>
> executable launch code for launching an update OS instance;
>
> quiescing, by the original OS instance, the devices of the target subset so that the original OS instance is executed on a source logical partition of the computer, the source logical partition including a source subset of the computer set;
>
> launching, by the original OS instance and using the executable launch code, the updated OS instance on the target logical partition so that each of the original OS instance and the updated OS instance run on the hardware previously controlled by the original OS instance; and
>
> moving, by the original OS instance and the updated OS instance, an application that had been executing on the original OS instance from the source logical partition to the target logical partition so that the application executes on the updated OS instance.

REJECTIONS

Claims 1–4, 8–12, and 16–26 stand rejected under 35 U.S.C. § 103 as obvious in view of Barrat (US 2015/0169329 A1; June 8, 2015), Hartung (US 2006/0294337 A1; Dec. 28, 2006), and Ringseth (US 2009/0288087 A1; Nov. 19, 2009). Final Act. 3–15.

Claims 5 and 13 stand rejected under 35 U.S.C. § 103 as obvious in view of Barrat, Hartung, Ringseth, and Santos (US 2012/0017029 A1; Jan. 19, 2012). Final Act. 15–16.

Claims 6 and 14 stand rejected under 35 U.S.C. § 103 as obvious in view of Barrat, Hartung, Ringseth, and Vega (US 2008/0320194 A1; Dec. 25, 2008). Final Act. 16–17.

Claims 7 and 15 stand rejected under 35 U.S.C. § 103 as obvious in view of Barrat, Hartung, Ringseth, Vega, and Nath (US 2011/0153872 A1; June 23, 2011). Final Act. 17–18.

ANALYSIS

The Examiner finds a combination of Barrat, Hartung, and Ringseth teaches or suggests every limitation recited in independent claims 1, 9, and 18. Final Act. 3–8, 11, 13. Of particular relevance to the dispositive issue with respect to this Appeal, the Examiner finds Barrat teaches or suggests "generating and storing, by an original operating-system (OS) instance . . ., update launch code." *See* Final Act. 4 (citing Barrat ¶¶ 22, 65); Ans. 18–19 (citing Barrat ¶¶ 22, 65, 79). In particular, the Examiner finds Barrat discloses creating a second logical partition (LPAR), cloning and updating a first LPAR's root volume group (rootvg), and using the cloned and updated rootvg to boot the second LPAR. Final Act. 4 (quoting Barrat ¶ 22). We understand the Examiner to find this disclosure in Barrat teaches or suggests

generating and storing updated launch code. The Examiner further finds Barrat discloses its "live update engine," which clones and updates the first LPAR's rootvg and boots the second LPAR using the updated rootvg, may "be split across the LPARs 310 and 340 such that a live update command may be initiated in the *original LPAR* 310 to begin the live update operation." Final Act. 4 (quoting Barrat ¶ 65, emphasis added); *see* Ans. 19. The Examiner finds Barrat discloses the live update engine "has to reside in the original partition." Ans. 18 (citing Barrat ¶¶ 22, 79). The Examiner concludes the *original OS* has to generate and store the update launch code (i.e., clone and update the first LPAR's rootvg) in situations where the second LPAR is created at update time because only the first LPAR exists at that time. Ans. 18–19 (citing Barrat ¶¶ 22, 79).

Appellant argues the proposed combination fails to teach or suggest, among other things, generating and storing launch code *by an original OS instance*. Appeal Br. 12–16; Reply Br. 5–8. Appellant argues Barrat does not explicitly state which elements clone and update the original LPAR's rootvg. Appeal Br. 13 (citing Barrat ¶ 22). Appellant contends—and the Examiner appears to agree—Barrat suggests its live update engine clones and updates the rootvg. Appeal Br. 13 (citing Barrat ¶¶ 64–65); *see* Final Act. 4 (citing Barrat ¶ 65). Appellant acknowledges that Barrat's live update engine may reside *in the original LPAR*, but asserts Barrat does not teach the live update engine resides in the *original OS*. Appeal Br. 15. Appellant argues the Examiner's findings that the live update engine generates update launch code and the live update engine may reside, in part or in whole, in the original LPAR do not support the Examiner's conclusion that Barrat discloses an *original OS generating* update launch code. Appeal Br. 15–16; *see also* Reply Br. 6 ("the teaching that the live update engine may be split

across LPARs does not imply that the live update engine somehow is *included* in the original and new operating systems"). Appellant further argues that, even if "the live update engine resides in the original" LPAR, it need not reside in the original OS because it "could be part of the software that constitutes the LPAR or else be a separate object hosted by the original partition." Reply Br. 6.

We agree with Appellant. Barrat teaches live OS update mechanisms and methods. *See* Barrat, Abstract. Barrat's Figure 3 depicts the primary aspects of the live update mechanism and is reproduced below.

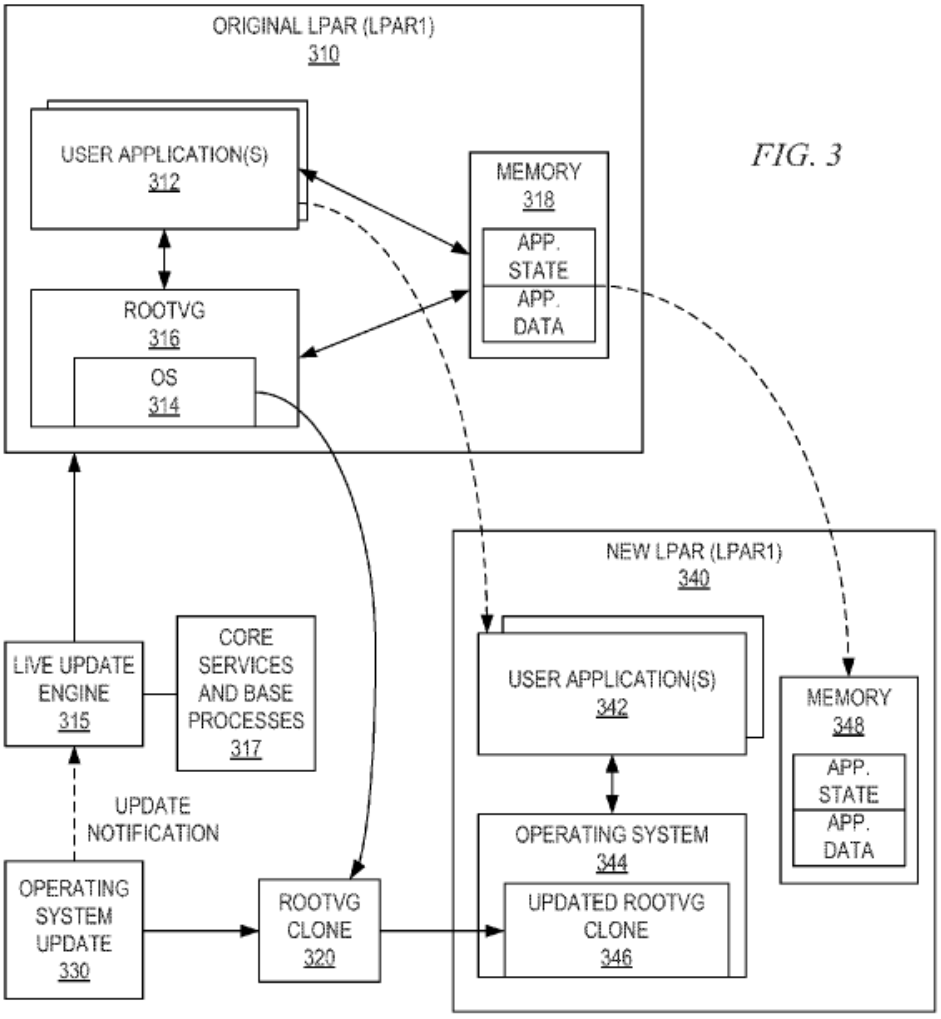

FIG. 3

Figure 3 depicts "an update/patch operation for performing a live update of an operating system." Barrat ¶ 12.

Specifically, Barrat discloses receiving an update or patch to an operating system, determining which LPARs in a system are running instances of the OS that for which the update was received, cloning and updating the rootvg for those LPARs, booting new LPARs using the updated rootvg so that updated OS logic in the updated rootvg is running in the new LPARs, migrating applications from the original LPARs to the LPARs booted using the updated rootvg, and deactivating the original LPARs. *See* Barrat ¶¶ 63–74, Fig. 3. As seen in Figure 3, Barrat discloses live update engine 315 receives an update notification indicating that an OS update is available. Moreover, as Appellant argues, it is possible that Barrat's live update engine may be part of the original LPAR (e.g., LPAR1 310 in Figure 3), but not be part of the operating system that is running in that partition (e.g., OS 314).

Barrat explicitly and repeatedly discloses that the disclosed mechanisms for the invention are implemented on systems using virtual devices. *See* Barrat ¶¶ 21 ("The mechanisms of the illustrative embodiments may be implemented on any computing system utilizing virtual devices and storage."), 57 (describing computing devices used in the invention as "implement[ing] a logically partitioned environment in which the particular computing device hosts one or more logical partitions (LPARs)"), 62 ("The elements shown in FIG. 3 may be implemented within a computing device implementing virtualized environments including one or more logical partitions."), 79 ("Thus, the illustrative embodiments provide mechanisms for performing live updates/patches of operating systems executing in virtualized environments, e.g., within logical partitions of a computing device."). In other words, Barrat's invention presumes an environment that includes OSes running inside of logical partitions and, therefore, implies the

system includes a mechanism, such as a hypervisor, to manage the various LPARs within the system. This understanding is supported by Barrat's depiction of rootvg 316 as including OS 314, as well as Barrat's depiction of ORIGINAL LPAR (LPAR1) 310 including other elements inside LPAR in addition to OS 314. Barrat Fig. 3. Thus, even in embodiments in which the live update engine is within, rather than external to, Barrat's LPARs, we agree with Appellant that this does not necessarily teach or suggest that the live update engine resides *within an OS*.

Therefore, on this record, we find the Examiner's rejection problematic. Although we agree with the Examiner that Barrat discloses embodiments in which Barrat's live update engine resides within the original partition (e.g., LPAR1), the Examiner has not explained sufficiently how this disclosure demonstrates that Barrat teaches "the generating and storing of update launch code (to launch the updated OS) has to be performed by the original OS because the updated OS is not yet available." *See* Ans. 18 (citing Barrat ¶ 22). In other words, the Examiner fails to explain sufficiently why a person of ordinary skill in the art would have understood Barrat's disclosure that the live update engine resides within the original partition as teaching or suggesting that the live update engine resides within or runs inside of the original OS rather than running elsewhere within the original partition—i.e., as a process *outside* the OS, but inside the LPAR.

Accordingly, we are constrained by this record to reverse the Examiner's rejection of claim 1 as obvious in view of Barrat, Hartung, and Ringseth. For the same reasons, we reverse the rejection of independent claims 9 and 18, which recite commensurate limitations, and dependent claims 2–4, 8, 10–12, 16, 17, and 19–26, which ultimately depend from one of claims 1, 9, and 18, as obvious in view of Barrat, Hartung, and Ringseth.

The Examiner does not find Santos, Vega, or Nath, which were cited in the rejections of claims 5–7 and 13–15, cure the identified deficiency of claims 1 and 9, from which these claims ultimately depend. Therefore, we also reverse the rejections of claims 5–7 and 13–15 as obvious in view of Barrat, Hartung, and Ringseth in combination with Santos, Vega, or Vega and Nath.

CONCLUSION

The Examiner's rejections of claims 1–26 under 35 U.S.C. § 103 are reversed.

DECISION SUMMARY

In summary:

Claims Rejected	35 U.S.C. §	Basis	Affirmed	Reversed
1–4, 8–12, 16–26	103	Barrat, Hartung, Ringseth		1–4, 8–12, 16–26
5, 13	103	Barrat, Hartung, Ringseth, Santos		5, 13
6, 14	103	Barrat, Hartung, Ringseth, Vega		6, 14
7, 15	103	Barrat, Hartung, Ringseth, Vega, Nath		7, 15
Overall Outcome				1–26

REVERSED