



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

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
13/491,444 06/07/2012 Lawrence Andrew Spracklen A740 2834

36378 7590 12/01/2016
VMWARE, INC.
DARRYL SMITH
3401 Hillview Ave.
PALO ALTO, CA 94304

Table with 1 column: EXAMINER

PATEL, KAMINI B

Table with 2 columns: ART UNIT, PAPER NUMBER

2114

Table with 2 columns: NOTIFICATION DATE, DELIVERY MODE

12/01/2016

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):

ipteam@vmware.com
ipadmin@vmware.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte LAWRENCE ANDREW SPRACKLEN

Appeal 2016-002331¹
Application 13/491,444
Technology Center 2100

Before JEAN R. HOMERE, JOHN F. HORVATH, and
SHARON FENICK, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellant seeks our review under 35 U.S.C. § 134(a) of the Examiner's Final Rejection of claims 1–10 and 12–19, which constitute all of the claims pending in this appeal. App. Br. 1. Claim 11 has been objected to as being dependent upon a rejected base claim, but would otherwise be allowable if rewritten to overcome the non-statutory subject matter rejection, to include the limitations of the base claim and any intervening claims. Final Act. 2. We have jurisdiction under 35 U.S.C. § 6(b).

¹ Appellant identifies the real party in interest as VMWARE, Inc. App. Br. 1.

We affirm-in-part.

Appellant's Invention

Appellant's invention is directed to a computer system (300) having a virtualization layer (304) for balancing the assignment of virtual machine processors (310) partitioned among hardware threads in a hardware layer (302) within the computer system to simultaneously process multi-threaded instructions therein. Spec. ¶¶ 4, 26, Fig. 3. In particular, the virtualization layer (304) includes a virtual machine monitor ("VMM")(318) that virtualizes physical processors to create virtual processors upon which the virtual machines execute. *Id.* ¶ 26. Each processor includes a set of performance monitoring registers (1304, 1305) for storing detected miss events and evicted events attributable to a subset of hardware threads due to high demands of hardware resources. *Id.* ¶¶ 49, 54, Figs. 13E-F. Each processor also includes a set of performance-imbalance-monitoring registers (1320, 1322) for storing accumulated indications of potential imbalances that adversely impact a hardware thread due to a resource exhaustion event associated therewith. *Id.* Upon receiving an indication from the performance-imbalance-monitoring registers (1320, 1322) of a potential performance imbalance between the processing of the hardware threads, the virtual machines are reassigned to hardware threads to thereby optimize the computational throughput and performance of the virtual machines. *Id.* ¶¶ 52, 60–61.

Representative Claims

Independent claims 1 and 9 are representative, and read as follows:

1. A performance-imbalance-monitoring register associated with a component or resource within a processor, the performance-imbalance-monitoring register comprising:

a register that stores accumulated indications of potential performance imbalances that adversely impact a hardware thread due to a resource-exhaustion event associated with the component or resource; and

processor logic that detects a potential performance imbalance when handling the resource-exhaustion event and accordingly updates the register.

9. A virtualization layer within a computer system that assigns each of two or more virtual machines to execute according to an execution schedule within a different one of two or more hardware threads within each of multiple processors, the multiple processors each including performance-monitoring and performance-imbalance-monitoring registers associated with components within the processor that are flexibly partitioned among, or shared by, the hardware threads, the virtualization layer comprising:

a virtual-machine-monitor that virtualizes physical processors to create virtual processors on which each of the virtual machines executes;

a kernel that manages memory, communications, and data-storage machine resources on behalf of executing virtual machines; and

a scheduling component that intermittently uses values in the performance-monitoring and performance-imbalance-monitoring registers to reassign virtual machines to hardware threads and reschedule execution of virtual machines in order to optimize computational throughput and performance of the virtual machines.

Rejections on Appeal

Claims 9–17 and 19 stand rejected under 35 U.S.C. § 101 as being directed to nonstatutory subject matter.

Claims 1–8 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Hankins et al. (US 2012/0017221 A1, published Jan. 19, 2012).

Claims 9 and 12–19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Hankins and De et al. (US 2010/0185823 A1, published July 22, 2010).

Claim 10 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Hankins, De, and Arimilli et al. (2010/0153542 A1, published June 17, 2010).

ANALYSIS

We consider Appellant’s arguments *seriatim*, as they are presented in the Appeal Brief, pages 7–32, and the Reply Brief, pages 2–14.²

35 U.S.C. § 101 Rejection

Claims 9–17 and 19

Appellant argues that the Examiner erred in concluding that claims 9–17 are directed to software per se, and are thereby directed to nonstatutory subject matter. App. Br. 8. According to Appellant, because the virtualization layer recited in claim 9 pertains to computer instructions executed by physical processors that carry out a variety of different operations when a computer system is powered on and while the computer system continues to operate, the claim is directed to a combination of hardware and software components. *Id.* at 9–12 (citing Spec. ¶¶ 22, 29). This argument is persuasive.

² Rather than reiterate the arguments of Appellant and the Examiner, we refer to the Appeal Brief (filed May 21, 2015), the Reply Brief (filed December 15, 2015) and the Answer (mailed October 15, 2015) for their respective details.

We do not agree with the Examiner that the virtualization layer recited in claim 9 is directed to software per se. Although the Examiner correctly finds that the virtual machine monitor, the kernel, and the scheduling component comprised in the virtualization layer are software components, the Examiner overlooks the claim requirement that such instructions perform the functions associated therewith upon being executed by processors included in the computer system. Ans. 4–5; Spec. ¶ 26. In particular, the claim recites that the virtualization layer is implemented within a computer system having a plurality of processors to which two or more assigned hardware threads are executed. Accordingly, we agree with Appellant that because the virtualization layer pertains to software modules executed by hardware components, it is not directed to software per se. We therefore do not sustain the nonstatutory subject matter rejection of claims 9–17 and 19.

35 U.S.C. § 102(e) Rejection

Claims 1-8

Appellant argues that Hankins' disclosure of performance monitoring registers does not describe performance-*imbalance*-monitoring registers as recited in claim 1 because the former cannot detect unbalanced or unfair distributions of resource exhaustion-caused events between hardware threads executing on a processor as does the latter. App. Br. 17–19 (quoting Spec. ¶¶ 52–54). According to Appellant, Hankin's disclosed performance monitoring registers are limited to accounting for the total amount of page faults collectively produced by all hardware threads operating within the processor during a particular time interval. *Id.* at 19–20. In contrast, the claimed performance imbalance monitoring registers account for individual page faults each hardware thread is responsible for producing during the

time interval, and can be used as a way to ascertain and rectify performance imbalance between threads. *Id.* 20–27 (quoting Hankins ¶¶ 31, 51, 55; Spec. ¶ 29). These arguments are not persuasive.

At the outset, we note Appellant’s arguments are not commensurate with the scope of claim 1. While the very essence of the present application, as detailed in Appellant’s Specification, is concerned with providing an architected hardware support including performance-imbalance monitoring registers to monitor performance imbalance among threads simultaneously executing on a single processor, such language is not recited in the claim. Spec. ¶¶ 4, 21. The Specification discusses in great details the shortcomings of traditional performance monitoring registers as being limited to monitoring the overall performance of simultaneously executing threads collectively without accounting for individual degradations for each thread. Spec. ¶ 2. Consequently, the Specification offers performance-imbalance-monitoring registers as a supplement to the latter kind of registers for additionally monitoring the performance of individual simultaneously executing threads to thereby resolve possible imbalances. Spec. ¶¶ 43, 49. However, as correctly noted by the Examiner, claim 1 is not so limiting. The claim merely recites "a register that stores accumulated indications of potential performance imbalances that adversely impact a hardware thread . . . associated with a component" Nothing in the claim limits the register to one that stores the performance imbalance between two threads associated with the same component. Thus, the performance imbalance can be between the recited “hardware thread . . . associated with the component,” and a second hardware thread associated with a second component. Because Appellant’s Specification does not provide an express definition for the

performance-imbalance monitoring register, and we cannot import the details from the Specification into the claim, we agree with Examiner that the disputed claim limitation must be interpreted under the broadest reasonable interpretation consistent with the Specification.³ Fin. Act 6. Therefore, we concur with the Examiner’s finding that the performance-imbalance monitoring register, as broadly claimed, reads on the traditional performance monitoring register described in the “Background” section of the present application. *Id*, Spec. ¶ 2.⁴ Accordingly, we agree with the Examiner that the performance monitoring register disclosed in Hankins describes the performance-imbalance-monitoring register as recited in claim 1. Fin. Act. 6– 7 (citing Hankins ¶¶ 28, 55). Consequently, we are not persuaded of error in the Examiner’s rejection of claim 1, as well as claims 2–8, which are not argued separately.

35 U.S.C. § 103(a) Rejections

Claims 9 and 12–19

Regarding claims 9 and 12–19, Appellant argues that the combination of Hankins with De does not teach or suggest performance-imbalance

³ *In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997). *See also In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989) (stating that “claims must be interpreted as broadly as their terms reasonably allow.”).

⁴ “In many modem processors, performance monitoring registers accumulate counts of the number of unstalled processor cycles and the number of pipeline-executed instructions successfully retired to allow performance-monitoring programs to compute general performance metrics, including the number of instructions executed per hardware cycle (“IPC”) and/or the number of processor cycles per successfully retired instruction (“CPI”), with IPC and CPI inversely related.”

monitoring registers that monitor event occurrences assigned to particular hardware threads. App. Br. 29. This argument is not persuasive. Although claim 9 requires that both the performance monitoring registers and performance-imbalance monitoring registers be included in each of the plurality of processors partitioned between the hardware threads, the claim does not delineate a difference between the two kinds of registers. Instead, the claim merely recites using values in the registers to reassign virtual machines to hardware threads, and to reschedule execution of virtual machines. As discussed above, such monitoring of performance-impacting events are described in conjunction with the functions of the performance monitoring registers in the “Background” section of Appellant’s Specification. Spec. ¶ 2.⁵ Consequently, we are not persuaded of error in the Examiner’s rejection of claim 9 and 11– 17, which are not argued separately.

Claim 10

Regarding the rejection of claim 10, Appellant argues that the combination of Hankins, De, and Arimilli does not teach or suggest the performance-imbalance monitoring registers are used for monitoring

⁵ Generally, when the IPC falls below a first threshold value and/or the CPI rises above a second threshold value, performance monitoring components of virtual-machine monitors (“VMMs”), operating systems (“OSs”), and various higher-level performance-analysis systems may invoke a variety of different performance-monitoring analyses to attempt to determine likely causes of the observed decreases in computational throughput to take or suggest various types of ameliorative procedures, *such as alternatively scheduling execution of virtual machines (“VMs”) or tasks, flagging particular tasks for redesign and reimplementation for performance optimization, reallocating computational and data-storage resources among VMs and tasks, and other types of ameliorative procedures.* (Emphasis added).

imbalances between simultaneously executing processes on a single processor. App. Br. 29–30. This argument is persuasive. Because the performance monitoring registers described in the prior art of record does not monitor performance imbalance between simultaneously executing threads, we agree with Appellant that the proposed combination of references does not teach the performance-imbalance monitoring registers as recited in the claim 10. Accordingly, we do not sustain the obviousness rejection of claim 10.

DECISION

We reverse the Examiner’s nonstatutory subject matter rejection rejections of claims 9-17 and 19, as well as the obviousness rejection of claim 10. However, we affirm the anticipation rejection of claims 1–8 as well as the obviousness rejection of claims 9 and 12–19 as set forth above.

AFFIRMED-IN-PART