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.
13/290,848	11/07/2011	Ivan Schreter	34874-821F01US	1015

64280 7590 06/04/2020
Mintz Levin/SAP
Mintz Levin Cohn Ferris Glovsky and Popeo, P.C.
One Financial Center
Boston, MA 02111

EXAMINER	
AYASH, MARWAN	
ART UNIT	PAPER NUMBER
2133	
NOTIFICATION DATE	DELIVERY MODE
06/04/2020	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):

IPDocketingBOS@mintz.com
IPFileroombos@mintz.com
mintzdocketing@cpaglobal.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte IVAN SCHRETER and DANIEL BOOSS

Appeal 2019-005889
Application 13/290,848
Technology Center 2100

BEFORE JOHNNY A. KUMAR, BETH Z. SHAW, and JASON M. REPKO, *Administrative Patent Judges*.

KUMAR, *Administrative Patent Judge*.

DECISION ON APPEAL

Pursuant to 35 U.S.C. § 134(a), Appellant[1] appeals the Final Rejection of claims 13, 19, 24, 27, 28, 31–34, and 36. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

[1] We use the word "Appellant" to refer to "applicant" as defined in 37 C.F.R. § 1.42. According to Appellant, SAP SE is the real party in interest. Appeal Br. 2.

INVENTION

The claimed invention relates to process-based memory allocation to dynamically allocate portions of memory to processes and free portions of memory when they are no longer needed. Spec. ¶¶ 1–2. Claim 13 is illustrative of the invention and is reproduced below:

> 13. A method comprising:
>
> assigning, by a memory allocator, a temporary memory limit to each of a plurality of processes requiring memory in a shared memory, the assigning of the temporary memory limit being based on *a vector in the shared memory, the vector comprising a plurality of slots, each slot storing, for a corresponding one of the plurality of processes, a unique identifier for the corresponding one of the plurality of processes, a quantity of bytes allocated to the corresponding one of the plurality of processes, and the temporary memory limit for memory consumption by the corresponding one of the plurality of processes*, wherein a sum of each of the temporary memory limits stored in the vector cannot exceed a global memory limit representative of a maximum amount of shared memory to be consumed by the plurality of processes;
>
> receiving a memory request of a first process of the plurality of processes;
>
> determining, based on the vector, that the first process has exceeded a first temporary memory limit and/or that increasing the first temporary memory limit would exceed the global memory limit;
>
> signaling, in response to the determining, a second process of the plurality of processes to reduce a second temporary memory limit of the second process;
>
> reducing, during execution of the second process of the plurality of processes and based on the determining, the second temporary memory limit corresponding to the second process being executed, *the second process releasing an amount of memory to the*

> *shared memory in response to the reducing and storing the released memory in a process-local cache, the amount of memory released proportional to the overall usage of the second temporary memory limit of the second process*; and
>
> signaling, in response to reducing the second temporary memory limit of the second process, the first process to allocate memory from the shared memory.

Appeal Br. 23–24 (Claims App.) (emphases added to indicate limitations in dispute).

REJECTION

Claims 13, 19, 24, 27, 28, 31–34, and 36 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Knowles et al. (US 2011/0138147 A1; published June 9, 2011) ("Knowles"), Pliss et al. (US 7,814,290 B1; issued Oct. 12, 2010) ("Pliss"), and Chekuri et al. (US 6,779,183 B1; issued Aug. 17, 2004) ("Chekuri"). Final Act. 4–8.

ANALYSIS

We have only considered those arguments that Appellant actually raised in the Briefs.[2] Arguments Appellant could have made, but chose not to make, in the Briefs have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(iv) (2018).

[2] Claims 19 and 32–34 are not argued separately from claim 13 in either of Appellant's briefs (Appeal Br. 13–21; Reply Br. 2–14), and will not be separately addressed. Claim 27 is not argued separately from claim 24 in either of Appellant's briefs (Appeal Br. 13–21; Reply Br. 2–14), and will not be separately addressed. Claim 31 is not argued separately from claim 28 in either of Appellant's briefs (Appeal Br. 13–21; Reply Br. 2–14), and will not be separately addressed.

A. Whether the combination of Knowles, Pliss, and Chekuri teaches a vector comprising a plurality of slots, each slot storing, for a corresponding one of the plurality of processes, a unique identifier for the corresponding one of the plurality of processes, a quantity of bytes allocated to the corresponding one of the plurality of processes, and the temporary memory limit for memory consumption by the corresponding one of the plurality of processes

Regarding the claimed vector comprising a plurality of slots, each slot storing, for a corresponding one of the plurality of processes, a unique identifier for the corresponding one of the processes (hereinafter "unique identifier" limitation), a quantity of bytes allocated to the corresponding one of the processes (hereinafter "quantity of bytes" limitation), and the temporary memory limit for memory consumption by the corresponding one of the processes (hereinafter "temporary memory limit" limitation), the Examiner finds that Knowles teaches, *inter alia*:

> a vector data structure, used interchangeably with a table/tabular data structure, may store, within slots/entries, data pertinent to the tracking and accounting of memory resources associated with a process/VM, including the consumer/user of the resources (VM/process identifier), the "current usage" or an "actual memory usage value" (Knowles abstract) also known as the quantity of bytes allocated to the process/VM, along with a temporary memory limit for memory consumption by the process/VM, disclosed "a dynamic maximum" or "target memory amount" in fig 3b of Knowles[].

Ans. 6 (emphasis omitted). The Examiner determines, in particular, that a broadest reasonable interpretation of "a vector . . . comprising a plurality of slots," is a data or table that has entries, slots, rows, or columns for storing a process or virtual machine. Ans. 4; *see also* Final Act. 7. The Examiner further finds that Chekuri teaches a vector. Final Act. 7 (citing Chekuri 3:43–47).

Appellant contends that Knowles does not teach a single "vector" storing the temporary limits for the executed processes sharing memory, let alone the unique identifier, quantity of bytes, and temporary memory limit limitations about each process stored in the slots of the vector. Appeal Br. 15–16; *see also* Reply Br. 2, 7–10. Appellant argues that the vector is not an obvious interchangeable data structure with the tabular data structure of Knowles. Reply Br. 3. Appellant further argues that the "general discussion of a vector in Chekuri" (emphasis omitted) does not teach a vector having the unique identifier, quantity of bytes, and temporary memory limit limitations (Appeal Br. 15; Reply Br. 9), nor does Pliss teach the claimed vector having those features. Appeal Br. 17–18.

"In the patentability context, claims are to be given their broadest reasonable interpretations . . . [and] limitations are not to be read into the claims from the specification." *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (citations omitted). Any special meaning assigned to a term "must be sufficiently clear in the specification that any departure from common usage would be so understood by a person of experience in the field of the invention." *Multiform Desiccants, Inc. v. Medzam, Ltd.*, 133 F.3d 1473, 1477 (Fed. Cir. 1998); *see also Helmsderfer v. Bobrick Washroom Equip., Inc.*, 527 F.3d 1379, 1381 (Fed. Cir. 2008) ("A patentee may act as its own lexicographer and assign to a term a unique definition that is different from its ordinary and customary meaning; however, a patentee must clearly express that intent in the written description."). Absent an express "intent to impart a novel meaning to the claim terms, the words are presumed to take on the ordinary and customary meanings attributed to them by those of ordinary skill in the art." *Brookhill-Wilk 1, LLC. v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1298 (Fed. Cir. 2003) (citation omitted).

We find unavailing Appellant's argument that the table or database having entry slots of Knowles (*see* Knowles ¶ 87) is an unreasonably broad construction of "vector" (Appeal Br. 14–15; Reply Br. 3, 9), as the term is interpreted in light of Appellant's Specification. Although Appellant argues that the term "vector" is sufficiently disclosed in the Specification, the sufficiency of disclosure alone does not determine a broadest reasonable interpretation of the term.

Appellant's Specification discloses that, "[t]he memory allocator can use a vector in shared memory to store memory limits, current memory consumption and control data. Each process can have an associated slot in the vector." Spec. ¶ 6. Figure 2 illustrates the memory allocator and the shared memory:

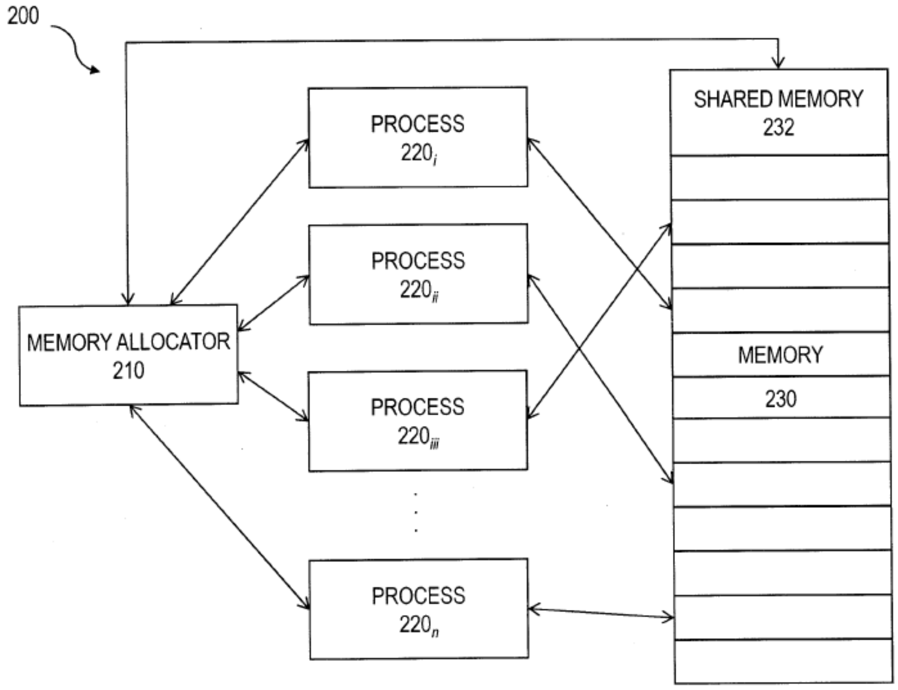

FIG. 2

Figure 2, above, is a diagram illustrating, a memory allocator 210 allocating certain amounts or portions of memory 230. Spec. ¶ 15. The Specification discloses that, "a slot is stored in a vector in shared memory 232. Each slot contains process specific information about the process $220_{i \ldots n}$ and its consumption of memory 230." *Id.*¶ 16. Based on the above-portions of Appellant's Specification, the claimed vector is disclosed as being a portion of a shared memory in which a slot is stored, the slot containing information. *Id.* at ¶¶ 6, 16. We, therefore, determine that a table or database falls within the broadest reasonable interpretation of a vector as it is described in Appellant's Specification (*Id.*, Spec. Fig. 2; *see also* Ans. 3–4), because a table or database has slots that contain information.

Furthermore, the Examiner proposes modifying Chekuri's vector with the features of Pliss and Knowles. *See* Final Act. 4–7; Ans. 5–7. Therefore, Appellant's arguments attacking Knowles, Pliss, or Chekuri singly for individual shortcomings (Appeal Br. 13–16; Reply Br. 7–8, 10, 11), without considering the combination of the references, are not effective arguments against obviousness. *See In re Keller*, 642 F.2d 413 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091 (Fed. Cir. 1986).

Here, Knowles teaches that a memory profile 305 for each of a plurality of virtual machines 232 is stored in the database or table, and that the memory profile may be *a profile* stored in a control virtual machine 232A. *See* Knowles Fig. 3A, ¶ 87 (emphasis added). The Examiner maps the virtual machine profile of Knowles to the unique identifier limitation. *See* Ans. 4–5, 6. Knowles further teaches that the memory profile includes, among other things, *a target memory amount 326*. Knowles ¶ 86 (emphasis added). A memory manager 350 compares actual memory allocated or used by each virtual machine, and

> [W]hen the memory manager **350** determines that *the actual memory allocated to or used by a particular virtual machine is greater than the target memory amount* **326** *for that virtual machine*, the memory manager **350** can reclaim additional memory from that virtual machine. . . .[T]he memory manager **350** can identify at least one virtual machine **232** that has *an actual memory usage value less than the target memory value or amount **326** assigned to that virtual machine*.

Knowles ¶ 101 (emphases added). The Examiner maps the Knowles target memory amount to the temporary memory limit limitation and the actual memory allocated or used of Knowles to the quantity of bytes limitation. *See* Ans. 4–5, 6. We agree with the Examiner that Knowles teaches the memory profile storing the unique identifier and the temporary memory limit, and therefore each slot storing, for a corresponding one of the virtual machines, i.e., processes (*see* claim 13), the unique identifier and temporary limit limitations. *See* Ans. 4–5, 6.

Notably, the Examiner further cites Chekuri for teaching that, "[a] d-dimensional vector can be used to represent each multi-dimensional task. The components of the d-dimensional vector represent the amount of resources required by the task from each of the d system resources and can be derived or estimated from system parameters." Final Act. 7; *see also* Chekuri 3:43–47 (emphasis omitted). In other words, Chekuri teaches a vector that stores, for a corresponding one of a plurality of processes, an actual amount of resources used by the process, i.e., a quantity of bytes allocated. *See* Chekuri 3:43–47; claim 13.

In view of the above teachings of Knowles and Chekuri, we agree with the Examiner that the claimed vector comprising slots, each slot storing for one a plurality of processes, the unique identifier, quantity of bytes, and temporary memory limit limitations, is taught by the combination of Knowles, Pliss, and Chekuri. Final Act. 7; Ans. 4–5, 6.

We are further guided that "[r]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some 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) (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).

Here, the Examiner additionally concludes that it would have been obvious to a skilled artisan at the time of the invention to combine the teachings of Knowles and Chekuri, "because a vector data structure lends itself well to array-matrix operations and optimizations which could be advantageously implemented to schedule multidimensional tasks/VMs."

Final Act. 7, Ans. 7 (citing Chekuri 1:8–21, 2:39–67) (emphasis omitted). As such, we disagree with Appellant's contentions that the Examiner's findings and conclusion of obviousness amount to common sense used to fill a gap in the prior art, bald conclusions required to make a creative leap to bridge a gap, or relying on inherency. Appeal Br. 13–16; Reply Br. 9, 12–13. Instead, the Examiner has proffered an articulated reasoning with rational underpinnings for the combination, namely an improvement to scheduling multi-dimensional tasks or virtual machine processes. Final Act. 7; Ans. 7.

Appellant argues that Knowles and Chekuri fail to teach the claimed "determining" step and "signaling, in response to the determining" steps. Appeal Br. 16–17; Reply Br. 10–11; *see also* claim 13. We have considered these arguments, which do not point to any specific alleged errors by the Examiner, and find them unpersuasive in view of the Examiner's findings with respect to Knowles. Final Act. 5. We agree with the Examiner's findings, conclusions and underlying reasoning and adopt them as our own.

B. *Whether the combination of Knowles, Pliss, and Chekuri teaches the second process releasing an amount of memory to the shared memory in response to the reducing and storing the released memory in a process-local cache, the amount of memory released proportional to the overall usage of the second temporary memory limit of the second process*

The Examiner concludes that Knowles and Pliss teach the disputed limitations (hereinafter "releasing, reducing, and storing" limitations) above, because Knowles teaches requesting memory from each virtual machine, and Pliss teaches a remedial garbage collection requiring free cached data associated with a task. Final Act. 6 (citing Knowles Fig. 4; Pliss 2:18–23); Ans. 8–9 (citing Knowles Abstract, ¶ 98).

Appellant disagrees, arguing that "the inflation and deflation of balloon drivers of Knowles or the 'garbage collection' of Pliss" do not teach the limitation. Appeal Br. 17; Reply Br. 11 (both citing Spec. ¶ 19; emphases omitted). Appellant further argues that Knowles "requires" each virtual machine to free memory by "forc[ing]" the virtual machines to release memory using balloon drivers, whereas, "[a]s the published application clearly states, the claimed memory allocator uses a technique to signal" release of an amount of memory by one of the processes proportional to usage, or that the memory allocator "can request . . . release" of memory allocated to the process. Appeal Br. 19–20; Reply Br. 3, 13 (both citing Spec. ¶ 18).

Although claim terms are interpreted in light of the specification, we do not read limitations from the specification into the claims. *See Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571–72 (Fed. Cir. 1988), *cert. denied*, 488 U.S. 892 (1988) (various limitations on which appellant relied were not stated in the claims; the specification did not provide evidence indicating these limitations must be read into the claims to give meaning to the disputed terms); *see also In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004). In addition, proper claim construction requires a broadest reasonable interpretation consistent with the specification. *In re Bond*, 910 F.2d 831, 833 (Fed. Cir. 1990); *see also Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005).

Per the guidance above, we find Appellant's argument that Knowles requires or forces the virtual machines to release memory (Appeal Br. 19–20; Reply Br. 3, 13) to be not commensurate with the scope of claim 13. In particular, "a technique" to signal and "can request" release, are not recited in the claim. *See* claim 13. Furthermore, the language of the claim 13 does not preclude the alleged requiring and forcing release of memory of Knowles. *See id.*

To the extent that Appellant's argues that Knowles does not teach the claimed "signaling" or amount of memory release proportional to the overall usage (Appeal Br. 19–20; Reply Br. 3, 13), we disagree. Notably, Knowles teaches that "the memory manager **350** *can then instruct a balloon driver **310***" to inflate to reclaim "a predetermined number of memory pages (e.g.[,] amount of memory)," and "[t]he memory manager **350** *can then instruct the balloon driver*" to deflate to allocate the reclaimed memory pages to other virtual machines. Knowles ¶ 98. In other words, Knowles teaches that the memory manager sends a signal to the balloon driver, instructing it to either inflate or deflate. *Id.* Furthermore, Knowles teaches in paragraph 101, replicated *supra* at 6, a virtual machine releasing an amount of memory proportional with actual memory allocated or used, i.e., overall usage, by the machine. Knowles ¶ 101.

Regarding Appellant's contention that the releasing and storing of memory in the process-local cache "can allow for quick reuse of memory" (Appeal Br. 17; Reply Br. 11), the Examiner additionally finds, and we agree, that a process-local cache, as recited in the claim:

> is not defined as any special-purpose memory "reservoir" and as such, a broadest reasonable interpretation of the limitation "storing the released memory in a process-local cache" is understood to correspond to *the allocation of memory resources to a process/VM and the subsequent "storage" of that allocated memory within any available memory reservoir which is available to the VM*. In other words, a process-local cache is understood to be any memory/buffer/cache which is used by the process/VM such that this would be the primary reservoir into

> which new/additional memory would be added after (re)allocation.

Ans. 8 (original emphasis omitted; emphasis added). We agree and adopt the Examiner's reasoning here. Although Appellant's Specification discloses "a process-local cache or quick reuse" (Spec. ¶ 19), consistent with the guidance above, we do not read "quick reuse" into the claims. The Examiner's broadest reasonable construction of a process-local cache is consistent with Appellant's Specification (*see* Ans. 8), and Appellant does not provide arguments to rebut the Examiner's broadest reasonable interpretation. *See* Appeal Br. 17; Reply Br. 11.

Accordingly, we find no error with the Examiner's conclusions in this regard. In addition, by a preponderance of the evidence, we agree with the Examiner's conclusion of obviousness of the releasing, reducing, and storing limitations, finding that Knowles teaches reclaiming and reducing memory for a virtual machine in an amount proportional to the overall usage of the machine, and allocating the memory to an available memory reservoir of another virtual machine. Final Act. 6; Ans. 8–10.

For the foregoing reasons, we sustain the Examiner's obviousness rejection of claim 13. Arguments directed to claims 24, 28, and 36 refer to arguments presented for claim 13. Appeal Br. 21; Reply Br. 14. Therefore, we sustain the Examiner's obviousness rejection of claims 13, 19, 24, 27, 28, 31–34, and 36.

CONCLUSION

The Examiner did not err in rejecting claims 13, 19, 24, 27, 28, 31–34, and 36 as being obvious under 35 U.S.C. § 103, over the cited combination of references.

DECISION

In summary:

Claim(s) Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Affirmed**	**Reversed**
13, 19, 24, 27, 28, 31–34, 36	103(a)	Knowles, Pliss, Chekuri	13, 19, 24, 27, 28, 31–34, 36	

TIME PERIOD FOR RESPONSE

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv). *See* 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED