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.
13/160,963	06/15/2011	Daniel Vasquez Lopez	332826.01	4409

69316 7590 01/28/2019
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052

EXAMINER	
MAHMOOD, REZWANUL	

ART UNIT	PAPER NUMBER
2164	

NOTIFICATION DATE	DELIVERY MODE
01/28/2019	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):

usdocket@microsoft.com
chriochs@microsoft.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte DANIEL VASQUEZ LOPEZ and WON YOO

Appeal 2018-004232
Application 13/160,963
Technology Center 2100

Before JOHN A. JEFFERY, JOHNNY A. KUMAR, and CARL L. SILVERMAN, *Administrative Patent Judges*.

JEFFERY, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants[1] appeal under 35 U.S.C. § 134(a) from the Examiner's decision to reject claims 1–12 and 14–21. We have jurisdiction under 35 U.S.C. § 6(b). We affirm-in-part.

STATEMENT OF THE CASE

Appellants' invention evicts entries from a file handle cache. To this end, entries in the file handle cache are analyzed and marked for eviction while a shared lock is maintained on the cache. The entries are then

[1] Appellants identify the real party in interest as Microsoft Technology Licensing, LLC. App. Br. 1.

removed while an exclusive lock is maintained on the cache. *See generally* Abstract. Claims 1, 12, and 16 are illustrative:

> 1. A method for evicting entries from a file handle cache, each entry in the file handle cache storing a file handle associated with an open file in a file system, the method comprising:
>
> analyzing each entry in the file handle cache to identify entries to be marked for eviction and marking the identified entries for eviction while maintaining a shared lock on the entire file handle cache, the shared lock allowing the file handle cache to be concurrently accessed by one or more processes to obtain file handles for use in servicing requests for content from the file system; and
>
> removing the entries marked for eviction from the file handle cache while maintaining an exclusive lock on the entire file handle cache, the exclusive lock preventing the file handle cache from being concurrently accessed by the one or more processes.
>
> 12. A system, comprising:
>
> at least one processor;
>
> memory connected to the at least one processor;
>
> a file system stored at least in part in the memory and executed by the at least one processor; and
>
> a content serving system stored at least in part in the memory and executed by the at least one processor, the content serving system being configured to service requests for content by accessing files on the file system, the content serving system including:
>
> a file handle cache having a plurality of entries, each entry in the file handle cache storing a file handle associated with an open file in the file system, the file handle cache being accessible to obtain file handles for accessing open files on the file system; and
>
> file handle cache management logic that is configured to analyze a time to live counter assigned to each entry in the file handle cache and to mark an entry for eviction if the time to live counter assigned thereto has expired, the value of the time to live counter that is assigned to each entry in the file handle cache being adaptively determined based on a measure of utilization of a system memory that stores contents of each open file associated with each file handle stored in the file handle cache, the system memory being a different entity than the file handle cache.

16. A system, comprising:

at least one processor;

memory connected to the at least one processor;

a file system stored at least in part in the memory and executed by the at least one processor; and

a content serving system stored at least in part in the memory and executed by the at least one processor, the content serving system being configured to service requests for content by accessing files on the file system, the content serving system including:

a file handle cache having a plurality of entries, each entry in the file handle cache storing a file handle associated with an open file in the file system, the file handle cache being accessible to obtain file handles for accessing open files on the file system; and

file handle cache management logic that is configured to analyze a hit count associated with each entry in the file handle cache and to mark an entry for eviction if the hit count is less than a statistical threshold, the hit count associated with each entry representing a number of times that the entry was accessed to service a request for content from the file system, the hit count associated with each entry being compared to the same statistical threshold, the statistical threshold being determined based on a combination of the hit counts associated with all of the entries in the file handle cache.

THE REJECTIONS

The Examiner rejected claim 1 under 35 U.S.C. § 103 as unpatentable over Wintergerst (US 2006/0064549 A1; published Mar. 23, 2006), Pawar (US 8,204,871 B1; issued June 19, 2012), and Ezra (US 7,143,393 B1; issued Nov. 28, 2006). Final Act. 13–15.[2]

[2] Throughout this opinion, we refer to (1) the Final Rejection mailed June 22, 2017 ("Final Act."); (2) the Appeal Brief filed August 31, 2017 ("App. Br."); (3) the Examiner's Answer mailed January 11, 2018 ("Ans."); and (4) the Reply Brief filed March 12, 2018 ("Reply Br.").

The Examiner rejected claims 2 and 3 under 35 U.S.C. § 103 as unpatentable over Wintergerst, Pawar, Ezra, and Coulson (US 2007/0233947 A1; published Oct. 4. 2007). Final Act. 16–17.

The Examiner rejected claims 4–11 under 35 U.S.C. § 103 as unpatentable over Wintergerst, Pawar, Ezra, and Petev (US 2006/0248285 A1; published Nov. 2, 2006). Final Act. 18–22.

The Examiner rejected claims 12, 14, 15, and 21 under 35 U.S.C. § 103 as unpatentable over Wintergerst, Pawar, Petev, and Steely (US 2005/0154805 A1; published July 14, 2005). Final Act. 22–27.

The Examiner rejected claims 16–20 under 35 U.S.C. § 103 as unpatentable over Wintergerst, Pawar, Petev, and Matthews (US 2007/0005889 A1; published Jan. 4, 2007). Final Act. 27–32.

THE REJECTION OVER WINTERGERST, PAWAR, AND EZRA

Regarding independent claim 1, the Examiner finds that Wintergerst discloses, among other things, a method for evicting entries from a cache, where each cache entry is analyzed to identify entries to be marked for eviction, and so marking the identified entries. Final Act. 13–14. Although the Examiner acknowledges that Wintergerst lacks (1) a file handle cache storing a file handle associated with an open file, and (2) shared and exclusive locks on the file handle cache, the Examiner cites Pawar for teaching these features. *Id.* 14–15. The Examiner also acknowledges that the Wintergerst/Pawar system does not lock an entire cache, but cites Ezra to cure that deficiency. *Id.* 15. Based on these collective teachings, the Examiner concludes that the claim would have been obvious. *Id.*

Appellants argue that the cited prior art does not teach or suggest analyzing each entry in a file handle cache to identify entries to be marked, and marking the entries for eviction while maintaining a shared lock on the entire file handle cache, as claimed. App. Br. 8–13; Reply Br. 3–5.[3] According to Appellants, not only does the Examiner improperly dissect the disputed limitation and address various portions of this limitation "in a vacuum," the Examiner's reliance on the cited prior art references is said to be misplaced in any event. *See* App. Br. 8–11. Although Appellants acknowledge that Wintergerst's system evicts cache entries that are so marked, Appellants emphasize that Wintergerst does mark these entries while a shared lock is maintained on any portion of a cache, let alone an entire cache as claimed. App. Br. 9. Appellants add that although Pawar's locks are applied to certain blocks of a specified file stored in a cache, and Ezra maintains an exclusive lock on an entire cache, the references do not maintain a shared lock on an entire file handle cache, let alone mark cache entries for deletion while maintaining the shared lock, as claimed. App. Br. 10–11. Lastly, Appellants argue that there is no apparent reason to combine the references as the Examiner proposes apart from improper hindsight, and that the Examiner merely provides conclusory statements that it would have been obvious to combine the teachings of the cited references. App. Br. 12–13; Reply Br. 4–5.

[3] Although the Reply Brief is unpaginated (unlike the Appeal Brief), we nonetheless refer to the Reply Brief's pages in the order that they appear in the record.

ISSUE

Under § 103, has the Examiner erred in rejecting claim 1 by finding that Wintergerst, Pawar, and Ezra collectively would have taught or suggested analyzing each entry in a file handle cache to identify entries to be marked for eviction, and so marking the entries while maintaining a shared lock on the entire file handle cache?

ANALYSIS

We begin by noting a key temporal aspect of claim 1, namely that the identified entries are marked for eviction *while* maintaining a shared lock on the entire file handle cache. Our emphasis on the term "while" underscores that this limitation specifies *when* entries are marked as Appellants indicate (App. Br. 9), namely simultaneously with maintaining the shared lock.

Given this temporal requirement, the Examiner's findings in this regard are problematic on this record. In the rejection, the Examiner finds that Wintergerst's system (1) analyzes each entry in a cache to be marked for eviction, and (2) marks the identified entries for eviction "*while* maintaining a shared . . . on the . . . cache . . . ." Final Act. 13 (emphasis added). These omissions indicated by the ellipses are telling, for it is unclear from these findings what exactly Wintergerst's system maintains while marking occurs.

On page 5 of the Answer, the Examiner cites Wintergerst's paragraphs 12, 13, 15, 68, 75, and 101, and equates Wintergerst's (1) marking entries in a shared cache for eviction, and (2) maintaining a shared cache with marking identified entries for eviction *while* maintaining shared *access* on the cache. Despite the claim reciting that marking occurs while maintaining a shared *lock*, the Examiner nevertheless cites Wintergerst's paragraphs 16, 65, 68, 75, and 81, and finds that Wintergerst's shared cache allows read-only access to cache entries or locks them out entirely. *See* Ans. 6. Although Wintergerst's paragraph 81 describes a hash map that can indicate whether an item is locked, we cannot say that marking identified entries *while* maintaining a shared lock on an entire file handle cache would have been obvious from the references' cited passages.

The Examiner's reliance on the other references fares no better in this regard. Although Pawar teaches placing shared range locks on a file specified by a file handle in column 8, lines 22 to 57 and step 131 of Figure 5, and Ezra teaches locking an entire cache data structure in column 21, lines 47 to 58, the cited prior art collectively still does not teach or suggest marking cache entries for eviction *while* maintaining a shared lock on an entire file handle cache, as claimed.

So even if the cited references were combinable as the Examiner proposes, they still do not teach or suggest the disputed temporal limitation. Accordingly, on the record before us, we are persuaded that the Examiner erred in rejecting claim 1.

THE REJECTIONS OF CLAIMS 2–11

Because the Examiner has not shown that the additional cited prior art cures the foregoing deficiencies regarding the rejection of independent claim 1, we do not sustain the obviousness rejections of dependent claims 2–11 (Final Act. 16–22) for similar reasons.

THE REJECTION OVER WINTERGERST, PAWAR, PETEV, AND STEELY

We also do not sustain the Examiner's obviousness rejection of independent claim 12 over Wintergerst, Pawar, Petev, and Steely. Final Act. 22–27. Claim 12 recites, in pertinent part, file handle cache management logic configured to analyze a time-to-live[4] counter assigned to each file handle cache entry and to mark an entry for eviction if the assigned counter has expired, *where the assigned counter's value is adaptively determined based on a measure of utilization of a system memory* that stores contents of each open file associated with each file handle stored in the file handle cache.

Our emphasis underscores a key aspect of the recited counter, namely that its value is *adaptively determined* based on a particular measure of system memory utilization, where the system memory differs from the file handle cache. As Appellants explain, by adapting time-to-live values based on system memory utilization, file handles can be evicted from the file handle cache faster when system memory utilization is high, thus causing files to be closed and reducing system memory utilization. App. Br. 20–21.

In the rejection, the Examiner finds that Wintergerst's cache management logic "is configured to analyze . . . each entry in the . . . cache and to mark an entry for eviction . . . *adaptively determined* based on a measure of utilization of a system memory." Final Act. 23–24 (citing Wintergerst ¶¶ 12, 13, 15; emphasis added). Our emphasis underscores that the Examiner relies on Wintergerst for the recited adaptive determination,

[4] Although the term "time to live" is not hyphenated in the claim, we nonetheless hyphenate it for clarity and grammatical form.

yet acknowledges that Wintergerst lacks a time-to-live counter. *See* Final Act. 23–25. That is, the Examiner apparently finds that Wintergerst adaptively determines marking cache entries for eviction—*not* adaptively determining a counter value as Appellants indicate. App. Br. 16.

To be sure, Petev's paragraphs 95 and 124 indicate that a time-to-live counter value can be associated with a cache entry, and that an object can be evicted if it resides in cache longer than the time-to-live. But Petev says nothing about *adaptively determining* this value, let alone basing this adaptive determination on a particular measure of system memory utilization, where the system memory differs from the file handle cache as claimed. *Accord* App. Br. 18 (noting this deficiency). Nor do Pawar and Steely cure this deficiency. Although Pawar's extent cache can include file handles as suggested in column 8, lines 22 to 28, and Steely's paragraph 2 notes that processors can have their own memory cache separate from main system memory, these general teachings do not cure the foregoing deficiencies regarding adaptively determining the recited counter value as claimed. To the extent that the Examiner finds that the cited references collectively teach the recited counter value adaptive determination, we disagree.

So even if the cited references were combinable as the Examiner proposes, they still do not teach or suggest the recited counter value adaptive determination. Accordingly, we are persuaded that the Examiner erred in rejecting independent claim 12, and dependent claims 14, 15, and 21 for similar reasons.

THE REJECTION OVER WINTERGERST, PAWAR, PETEV, AND MATTHEWS

We sustain the Examiner's obviousness rejection of independent claim 16 over Wintergerst, Pawar, Petev, and Matthews. Final Act. 27–32. Claim 16 recites, in pertinent part, file handle cache management logic configured to (1) analyze a hit count associated with each cache entry, and (2) mark an entry for eviction if the hit count is less than a statistical threshold, where the hit count represents a number of times the entry was accessed, and the threshold is determined based on a combination of hit counts associated with all entries in the cache.

The Examiner cites Petev's paragraphs 113 to 115 and 119 to 123 for teaching much of the recited cache management logic functionality, including (1) analyzing hit counts associated with each cache entry, and (2) marking an entry for eviction if the hit count is less than a statistical threshold. Final Act. 29; Ans. 17–18. We see no error in these findings. Petev uses a counter to measure the number of times an object is accessed, and based on this measurement, places the object in a particular location in a queue. *See* Petev ¶¶ 113–115. Notably, the object with the lowest counter value (i.e., the object that is the least frequently used) is at the bottom of the queue and, consequently, will be evicted next from the cache, thus effectively marking that entry for eviction. *See id.*

Although the Examiner acknowledges that Petev's threshold is not determined based on a combination of hit counts associated with all entries in the cache, we see no error in the Examiner's reliance on Matthews for at least suggesting such a determination. *See* Final Act. 30–31; Ans. 18–19 (citing Matthews ¶¶ 14, 19, 23, 26–28). To be sure, Matthews' system uses the threshold length of data to determine which data requests are cached. *See* Matthews ¶ 19. But a key aspect of this determination is that the threshold length can be *changed dynamically based on hit rates* for *all* data in the cache as well as data rejected from being read from or written to the cache. *See* Matthews ¶¶ 19, 26–27.

Given Matthews' fundamental teaching of changing a threshold based on a combination of hit counts for all cached entries, we see no reason why such a determination could not be applied to a hit count threshold, such as that in Petev, as the Examiner proposes to, among other things, manage the cache efficiently by setting a threshold dynamically based on the total hit counts. *See* Final Act. 30–31; Ans. 18–20. That Matthews pertains to a cache *entry* scheme unlike Petev's cache *eviction* scheme as Appellants indicate (App. Br. 25) is of no consequence here, for both schemes ultimately determine whether data will be stored in a cache based at least partly on hit count. Therefore, the Examiner's proposed combination is supported by articulated reasoning with some rational underpinning to justify the Examiner's obviousness conclusion.

We reach the same conclusion regarding the Examiner's reliance on Wintergerst and Pawar for at least suggesting the respective features for which whey were cited, as well as the Examiner's articulated reason to combine the cited references. Final Act. 27–31; Ans. 15–20. Despite Appellants' arguments to the contrary (App. Br. 26–27; Reply Br. 8–9), the Examiner's articulated reason to combine the references is not merely conclusory as Appellants assert, but rather has at least some rational underpinning to justify the Examiner's obviousness conclusion.

Therefore, we are not persuaded that the Examiner erred in rejecting claim 16, and claims 17–20 not argued separately with particularity.

CONCLUSION

Under § 103, the Examiner erred in rejecting claims 1–12, 14, 15, and 21, but did not err in rejecting claims 16–20.

DECISION

We affirm-in-part the Examiner's decision to reject claims 1–12 and 14–21.

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).

AFFIRMED-IN-PART