EXHIBIT 1A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Inventor: Nemes Original Application No.: 775,864 Patent No.: 5,893,120 Filed: January 2, 1997 Title: Methods and Apparatus for Information Storage and Retrieval Using a Hashing Technique with External Chaining and On-the-Fly Removal of Expired Data	REQUEST FOR REEXAMINATION UNDER 35 U.S.C. §§ 302-307 AND 37 C.F.R. § 1.510

Mail Stop *Ex Parte* Reexamination
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT NO. 5,893,120

TABLE OF CONTENTS

TABLE OF EXHIBITS

The exhibits to the present request are arranged in four groups: prior art ("PA"), relevant patent prosecution file history, patents, and claim dependency relationships ("PAT"), claim charts ("CC"), and other ("OTH").

Prior Art (PA)

PA-A	John A. Morrison, Larry A. Shepp, and Christopher J. Van Wyk, *A Queueing Analysis of Hashing with Lazy Deletion*, 16 Siam J. Comput. 6, 1155-1164 (December 1997) ("Morrison")
PA-B	U.S. Patent No. 4,695,949 to Thatte *et al.* ("Thatte")
PA-C	U.S. Patent No. 6,119,214 to Dirks ("Dirks")
PA-D	U.S. Patent No. 5,724,538 to Morris *et al.* ("Morris")
PA-SB08A	USPTO Form SB/08A

Relevant Patent Materials (PAT)

PAT-A	U.S. Patent No. 5,893,120 ("the '120 patent")

Claim Charts (CC)

CC-A	Claim Chart comparing Morrison to the claims of the '120 patent
CC-B	Claim Chart comparing Morrison in view of Dirks to the claims of the '120 patent
CC-C	Claim Chart comparing Thatte to the claims of the '120 patent
CC-D	Claim Chart comparing Thatte in view of Dirks to the claims of the '120 patent
CC-E	Claim Chart comparing Morrison in view of Thatte to the claims of the '120 patent
CC-F	Claim Chart comparing Dirks in view of Morris to the claims of the '120 patent

Other Documents (OTH)

OTH-A	Complaint; *Bedrock Computer Technologies LLC v. Softlayer Technologies, Inc. et al.*, Case No. 6:09-cv-269, filed June 16, 2009 (E.D. TX).

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Inventor: Nemes Original Application No.: 775,864 Patent No.: 5,893,120 Filed: January 2, 1997 Title: Methods and Apparatus for Information Storage and Retrieval Using a Hashing Technique with External Chaining and On-the-Fly Removal of Expired Data	REQUEST FOR REEXAMINATION UNDER 35 U.S.C. §§ 302-307 AND 37 C.F.R. § 1.510

Mail Stop *Ex Parte* Reexamination
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT NO. 5,893,120

Dear Sir or Madam:

The undersigned hereby respectfully requests *ex parte* reexamination pursuant to 35 U.S.C. §§ 302-307, 37 C.F.R. § 1.510, and MPEP §§ 2240(II) and 2295, of claims 1-8 of U.S. Patent No. 5,893,120 ("the '120 patent") filed January 2, 1997 and issued April 6, 1999 to Richard Michael Nemes (Exhibit PAT-A).

Reexamination is requested in view of the substantial new questions of patentability ("SNQP") presented below. Requester reserves all rights and defenses available including, without limitation, defenses as to invalidity and unenforceability. By simply filing this Request in compliance with the Patent Rules, Requester does not represent, agree or concur that the '120 patent is enforceable, and by asserting the SNQPs herein, Requester specifically asserts that Claims 1-8 of the '120 patent are in fact not patentable. As such, the U.S. Patent and Trademark Office (the

"Office") should reexamine and find Claims 1-8 unpatentable and cancel Claims 1-8 of the '120 patent, rendering Claims 1-8 of the '120 patent null, void, and otherwise unenforceable.

Reexamination is requested in view of the teachings of John A. Morrison, Larry A. Shepp, and Christopher J. Van Wyk, *A Queueing Analysis of Hashing with Lazy Deletion* ("Morrison"); U.S. 4,695,949 ("Thatte"); U.S. 6,119,214 ("Dirks"); U.S. 5,724,538 ("Morris"); and U.S. 4,996,663 ("Nemes"). The SNQPs established by these references teach the elements recited by the claims of the '120 patent and, importantly, teach the apparent reasons for the allowance of the claims. Based on the Applicant's arguments in prosecution, it appears that the '120 patent was allowed most directly because it recites on-the-fly deletion of records in linked lists based on automatic expiration of data.

The SNQPs presented below evidence that the '120 patent does not meet the novelty and non-obvious requirements of the United States Code and, as such, should not have been found patentable. Specifically, many of the claims of the '120 patent recite a method and apparatus for on-the-fly deletion of records in linked lists based on automatic expiration of data as claimed. As described below and throughout this Request, such claims do no recite new subject matter. In fact, many of the claims relate to the use of external chaining, linked lists, and hashing techniques, which the Applicant disclosed was well known in the art at the time of the alleged invention. *See* PAT-A at p. 1-3.

The Requester submits that reexamination should be granted and Claims 1-8 found unpatentable by issuance of a Certificate of Reexamination canceling all claims in view of the fact that: 1) the art discussed herein, especially Morrison and Thatte, anticipate many of the claims, and when combined with supporting references, render obvious the balance of the claims; and 2) the arguments made by the Applicant to support the patentability of many of the claims do not result in novel or non-obvious claims.

I. REQUIREMENTS FOR *EX PARTE* REEXAMINATION UNDER 37 C.F.R. § 1.510

Requester satisfies each requirement for *ex parte* reexamination of the '120 patent.

A. 37 C.F.R. § 1.510(B)(1) AND (B)(2): STATEMENT POINTING OUT EACH SUBSTANTIAL NEW QUESTION OF PATENTABILITY

A statement pointing out each substantial new question of patentability ("SNQP") based on the cited patents, and a detailed explanation of the pertinence and manner of applying the patents to Claims 1-8 of the '120 patent, is presented below in accordance with 37 C.F.R. § 1.510 (b)(1) and (b)(2).

The SNQPs raised herein are based on art that was either not discussed during the prosecution of the '120 patent, or was not cited altogether. The references, alone or in combination, are not cumulative to the prior art discussed during the original prosecution. Thus, they are appropriate for use in supporting the SNQPs raised herein.

B. 37 C.F.R. § 1.510(B)(3): COPY OF EVERY PATENT OR PRINTED PUBLICATION RELIED UPON TO PRESENT AN SNQP

A copy of every patent or printed publication relied upon to present an SNQP is submitted herein, pursuant to 37 C.F.R. §1.510(b)(3), as Exhibits PA-A through Exhibits PA-D, citation of which may be found on the accompanying Form PTO-SB/08 at Exhibit PTO-SB/08. Each of these cited prior art publications constitutes effective prior art as to the claims of the '120 patent under 35 U.S.C. § 102 and 35 U.S.C. § 103.

C. 37 C.F.R. § 1.510(B)(4): COPY OF THE ENTIRE PATENT FOR WHICH REEXAMINATION IS REQUESTED

A full copy of the '120 patent is submitted herein as Exhibit PAT-A and its corresponding file history is submitted as EXHIBIT PAT-B in accordance with 37 C.F.R. § 1.510(b)(4).

D. 37 C.F.R. § 1.510(B)(5): CERTIFICATION THAT A COPY OF THE REQUEST HAS BEEN SERVED IN ITS ENTIRETY ON THE PATENT OWNER

A copy of this request has been served in its entirety on the Patent Owner of record in accordance with 37 C.F.R. § 1.510(b)(5) at the following address:

RICHARD MICHAEL NEMES
1432 EAST 35TH STREET
BROOKLYN, NY 11234-2604

Additionally, a copy of this request has been served on counsel of the purported Patent Owner at the following address:

DOUGLAS A CAWLEY
MCKOOL SMITH, P.C.
300 CRESCENT COURT, SUITE 1500
DALLAS, TX 75201

E. 37 C.F.R. (A): FEE FOR REQUESTING REEXAMINATION

In accordance with 37 C.F.R. § 1.510(a), a credit card authorization to cover the fee for reexamination of $2,520.00 is attached. If this authorization is missing or defective please charge the Fee to the Novak Druce and Quigg Deposit Account No. 14-1437.

F. RELATED CO-PENDING LITIGATION REQUIRES TREATMENT WITH SPECIAL DISPATCH AND PRIORITY OVER ALL OTHER CASES.

The '120 patent is presently the subject of *Bedrock Computer Technologies, LLC v. Softlayer Technologies, Inc. et al.*, No. 609-CV-269 (E.D. TX filed June 16, 2009). *See* OTH-A, Complaint filed by Bedrock in *Bedrock Computer Technologies, LLC v. Softlayer Technologies, Inc. et al.*

Pursuant to 35 U.S.C. § 305, Requester respectfully urges that this Request be granted and reexamination conducted not only with **"special dispatch,"** but also with **"priority over all other cases"** in accordance with 37 C.F.R. 1.550(a) due to the ongoing nature of the underlying litigation.

Furthermore, pursuant to the recently announced policy of the Office concerning revised reexamination procedures to provide for a scheduling-type order of expected substantive action dates in Requests ordered after the Office's 2005 fiscal year, Requester respectfully seeks such a scheduling order upon the granting of this request.

II. OVERVIEW OF THE '120 PATENT AND PROSECUTION HISTORY

A. SUMMARY OF CLAIMED EMBODIMENTS OF THE '120 PATENT

The '120 patent claims a method and apparatus for information storage and retrieval using a hashing technique with external chaining and on-the-fly removal of expired records. The

patent relates to techniques, admittedly well known in the art at the time, for storing records through the use of a hashing technique with external chaining. The problem that the claimed invention purported to solve was that of efficiently removing expired records from a hash table with external chaining. The solution the patent describes involves "on-the-fly" removal of expired records that occurs while the hash table is otherwise being accessed to perform an insertion, retrieval, or deletion of records.

As the applicant admits in the patent, at the time of the patent, a hashing technique was a "well known and much faster way of storing and retrieving information from computer storage." '120 patent at Col. 1, lines 34-35. Hashing techniques store records based on keys. A hashing function is applied to each key to determine the location, or "bucket," in which a record will be stored. Because typically multiple keys can "hash" to the same bucket, causing a "collision," some form of collision resolution is needed. The '120 patent uses an external chaining collision resolution strategy, which, as the applicant admits, was also well known in the art at the time. '120 patent at Col. 1, line 47- Col. 2, line 6. External chaining involves the use of linked lists to resolve collisions, where all records that hash to the same bucket are stored in the same linked list, and a pointer to that linked list is stored in the hash table. Insertions, retrievals, or deletions from the hash table involve a traversal of the appropriate linked list, followed by the insertion (if the record is not found), or a retrieval or deletion (if the record is found).

Because these basic hashing techniques were well known in the art at the time, the claims of the '120 patent were directed towards the addition of a technique for "on-the-fly" removal of expired records. '120 patent at claims 1, 3, 5, 7. This "on-the-fly" method takes advantage of the fact that, during a traversal of a linked list for an insertion, deletion, or retrieval, each record in the list has to be examined for a match with the record being inserted/retrieved/deleted. '120 patent at Col. 5, line 53- Col. 8, line 59; Figs. 3-7. The claimed invention simply adds another check to see if the record being examined has expired. If so, the record is removed. '120 patent at Cols. 5-8; Figs. 3-7.

The '120 patent also claims this method of on-the-fly removal of expired elements where the maximum number of expired records to remove can be dynamically determined when the

linked list is accessed. '120 patent at claims 2, 4, 6, 8. This can be accomplished by a dynamic runtime decision to remove all expired records, some but not all of the expired records, or none of the expired records. '120 patent at Col. 7, lines 1-10.

B. SUMMARY OF THE '120 PATENT'S PROSECUTION HISTORY

1. PAPER NUMBER 1: U.S. PATENT APPLICATION NO. 775,864- JANUARY 2, 1997

The '120 patent was filed on January 2, 1997, as U.S. Patent Application No. 08/775,864 ("the '864 application"), naming Richard M. Nemes ("the Applicant") as inventor. The '864 application contained four independent and four dependent claims.

In the background section of the application, the Applicant admitted that many of the claimed limitations were known in the prior art. '864 application at p. 1-3. For example, the Applicant disclosed that it was well known to use external chaining and linked lists with hashing techniques to resolve collisions in hash tables. *Id.*

2. PAPER NUMBER 3: OFFICE ACTION- APRIL 20, 1998

On March 20, 1998, the USPTO rejected Claims 1 – 8 of the '864 application.

The Examiner rejected Claims 1 – 4 of the '864 application for double patenting based on the '495 patent also issued to Nemes. The Examiner noted that "the subject matter claimed in the instant application is fully disclosed in the ['495] patent and is covered by the ['495] patent since the ['495] patent and the application are claiming common subject matter." Paper No. 3 at p. 3.

The Examiner also rejected claims 5 – 8 of the '864 application for double patenting based on the '499 patent also issued to Nemes.

The Examiner further rejected Claims 1 – 8 of the '864 application under 35 U.S.C. §103 as being unpatentable over the '499 patent and "Schackelford". *Id.* at p. 9. The Examiner noted:

> 10. With respect to claims 1 – 8, Nemes ['499 patent] teaches everything that is claims (col. 2, line 60-64, col.6, line 49-51; '499 reference) except that it does not explicitly indicate the determination of threshold as being the maximum number of records.

> Schackelford teaches the maximum number of pointers to records wherein the records are linked in a bidirectional linked list (col. 3, line 61 through col. 4, line 2).
>
> It would have been obvious to a person of ordinary skill in the art at the time the invention was made to determine the maximum number of records to make the best use of the memory space available to a user and thus to reduce the system overhead (col. 4, lines 1-2, Schackelford).
>
> 11. The above reasoning does not discuss the linked list and the step of removing the records, it rather addresses the issue of determining the maximum number of records. For the limitations directed to the linked lists and the step of removing, the Applicant is requested to refer to the Obvious Double Patenting rejection set forth hereinabove where they have been discussed in details. *Id.* at pp. 6-7.

3. PAPER NUMBER 5: APPLICANT'S RESPONSE TO THE OFFICE ACTION- APRIL 10, 1998

In the Applicant's response the Applicant argued that the '495 patent was only applicable to the linear probing technique of hashing. Paper No. 5 at pp. 1-3. Thus, according to the Applicant, "since the ['864 application] claims linked lists, not suggested by linear probing of the cited art, the subject matter claimed is not disclosed in [the '495 patent]." *Id.* at p. 2.

The Applicant next attempted to distinguish the '499 patent by arguing that "[the '499 patent] does not teach or suggest on-the-fly deletion of a least some records based on automatic expiration of data. . . ." *Id.* at p. 3. The Applicant wrote, "the '499 patent … does not teach means or methods for identifying and removing 'at least some expired ones of the records' from the linked list 'when the linked list is accessed' . . . which is taught by the instant application and is integral to claims 5 and 7. . . ." *Id.* at p. 4. Additionally, the Applicant clarified that the claim term "maximum number of records" in the '864 application "is a single quantity that serves as an upper limit on the number of records removed from the linked list whenever the linked list is accessed. . . ." *Id.*

Finally, regarding the Examiner's §103 rejections, the Applicant stated that "Claims 1 – 8 of the ['864] application address on-the-fly deletion of at least some record from a linked list based on automatic expiration of data. . . ." *Id.* at p. 5. The Applicant again argued that the '499 patent does not "teach deletion from the system, nor does it teach regarding automatically

expiring data." Next the Applicant wrote that "Shackelford does not address an application with automatically expiring data, nor does he address how many items to delete." *Id.* at p. 6.

4. PAPER NUMBER 6: NOTICE OF ALLOWABILITY- SEPTEMBER 28, 1998

On September 28, 1998, the Examiner issued a Notice of Allowability for Claims 1-8 of the '864 Application. The Applicant successfully achieved allowance without filing any terminal disclaimer to overcome the double patenting rejections. The Examiner's reasoning is as follows:

> The prior art does not teach or fairly suggest a method and apparatus for on-the-fly deletion of records in linked lists based on automatic expiration of data as claimed. In other words, the prior art of record does not teach or fairly suggest the means (or an equivalent step in the method claim) of "means for [. . .] accessing a linked list at the same time, removing [. . .] some of the expired ones of the records in the linked list," as recited in lines 7 – 8 of claim 1. Although the prior art of record (Nemes, '495 reference) teaches the use of chains of records and the deletion of records, the Applicant, in the Response dated August 11, 1998, Paper No. 5, provided arguments as to why the chain of records as taught in the '495 reference is not the same as the linked list as claimed. The Applicant also distinguishes the claimed invention over the teachings of the '499 references. Paper No. 6 at ¶ 1.

C. ASSIGNMENT HISTORY

Bedrock purports to own the '120 patent by assignment. *See* OTH-A at ¶ 15. At the time of the filing of this Request, the purported assignment has not been recorded with the USPTO. Additionally, the Requestor is not aware of any assignment of the '120 patent. The Requestor expects that the ownership of the '120 patent may be challenged in the related litigation.

D. CLAIM CONSTRUCTION

Requester notes that for the purposes of this Request, the claim terms are presented by the Requester in accordance with 37 C.F.R § 1.555(b) and MPEP § 2111. Specifically, each term of the claims is to be given its "broadest reasonable construction" consistent with the specification. MPEP § 2111; *In re Trans Texas Holding Corp.*, No. 2006-1599, -1600, p.14 (Fed. Cir. August 22, 2007) (citing *In re Yamamoto*, 670 F.2d 1569, 1571 (Fed. Cir. 1984)). As the Federal Circuit noted in *Trans Texas*, the Office has traditionally applied this standard during reexamination and

does not interpret claims as a court would interpret claims. MPEP § 2111. The Office is not bound by any prior district court claim construction. *Trans Texas*, No.2006-1599 at 14, 19. Rather:

> the PTO applies to verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in applicant's specification.

In re Morris, 127 F.3d 1048, 1054-55, 44 U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 1997). The rationale underlying the "broadest reasonable construction" standard is that it reduces the possibility that a claim, after issue or certificate of reexamination, will be interpreted more broadly than is justified. 37 C.F.R § 1.555(b), MPEP § 2111.

Because the standards of claim interpretation used in the courts in patent litigation are different from the claim interpretation standards used in the Office in claim examination proceedings (including reexamination), any claim interpretations submitted herein for the purpose of demonstrating an SNQP are neither binding upon litigants in any litigation related to the '895 patent; nor do such claim interpretations correspond to the construction of claims under the legal standards that are mandated to be used by the Courts in litigation. *See* 35 U.S.C. § 314; *see also* MPEP § 2286 II (determination of an SNQP is made independently of a court's decision on validity because of different standards of proof and claim interpretation employed by the District Courts and the Office); *see also In re Trans Texas Holding Corp.*, No. 2006-1599, -1600, p.14 (Fed. Cir. August 22, 2007); *In re Zletz,* 893 F.2d 319, 322, 13 USPQ2d 1320, 1322 (Fed. Cir. 1989).

The interpretation and/or construction of the claims in the '120 patent presented either implicitly or explicitly herein should not be viewed as constituting, in whole or in part, Requester's own interpretation and/or construction of such claims. In fact, Requester expressly reserves the right to present its own interpretation of such claims at a later time, which interpretation may differ, in whole or in part, from that presented herein.

III. SUMMARY OF THE PRIOR ART

A. MORRISON

Morrison was published in 1987 and constitutes effective prior art under 35 U.S.C. § 102 and 103.

Morrison discloses an information storage and retrieval system in a dynamic dictionary using a hashing technique to maintain certain data structures within the storage system. The algorithm disclosed in Morrison is known as "hashing with lazy deletion," and the "lazy" aspect of the lazy deletion algorithm is exactly what the '120 patent describes as "on-the-fly" deletion--expired records are removed from linked lists in the hash table when those linked lists are traversed during an insertion operation.

The authors provide a simple definition for this technique: "[h]ashing with lazy deletion means keeping the items hashed by search key in a table of linked lists (separate chains); each time an item is added to a list, any items on that list that the new item shows to be expired are deleted from that list." Morrison at 1155, ¶ 2. Importantly, "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.* Morrison utilizes a data structure called a separate-chaining hash table, which is another name for the external-chaining table utilized in the '120 patent, wherein each hash table location is a pointer to the head of a linked list of records that contains the key-value pairs that hash to the same location. Morrison also provides a search key in order to access the table of linked lists and discloses that the search key may be hashed.

Additionally, Morrison was not before the Examiner during prosecution of the '120 patent. As such, it is clear that an examiner would consider Morrison important in determining the patentability of the claims of the '120 patent. Therefore, Morrison presents a substantial new question of patentability.

B. U.S. PATENT NO. 4,659,949 TO THATTE ET AL.

Thatte was filed on July 19, 1984 and constitutes effective prior art under 35 U.S.C. § 102 and 103.

Thatte discloses a method and apparatus for managing a block-oriented memory of the type in which each memory block has an associated reference count representing the number of pointers to it from other memory blocks and itself. Thatte at Abstract. The problem addressed in Thatte is similar to that discussed in the '120 patent: to provide a memory management technique that is efficient, cost-effective, alleviates the need for frequent garbage collection, reduces the reference counting overhead, and allows reference counting to be implemented in practice. Thatte at Col. 5, lines 23-28. Thatte describes the use of a structure known as a "reference count filter" that keeps track of blocks of memory whose reference counts equal zero. Thatte explains that the preferred implementation of the reference count filter is "a hash table that can efficiently support the insert, delete, and reconcile operations." Thatte at Col. 8, lines 39-41. The hash table uses external chaining to resolve collisions. Thatte at Col. 8, lines 49-59; Fig. 7.

Moreover, Thatte describes on-the-fly removal of expired records from the reference count filter. When the memory manager attempts to insert a record into the reference count filter when the filter is full, a reconciliation operation takes place. This reconciliation operation removes expired records from the reference count filter on the fly. Thatte explains that "[i]f the reference count filter is determined to be full, box 60, the MMU suspends the insertion operation and performs a reconciliation operation, box 62, on the reference count filter . . . to create room in the reference count filter so the suspended insertion operation can be completed." Thatte at Col. 7, lines 21-26. The reconciliation procedure identifies expired records, referred to as garbage, and then removes those garbage records from the reference count filter into a "garbage buffer" for later reclamation. Thatte at Col. 7, line 40- Col. 8, line 10. Once that reconciliation is complete, the insert operation resumes. Thatte at Col. 8, lines 10-14.

Additionally, Thatte was not before the Examiner during prosecution of the '120 patent. As such, it is clear that an examiner would consider Thatte important in determining the

patentability of the claims of the '120 patent. Therefore, Thatte presents a substantial new question of patentability.

C. U.S. PATENT NO. 6,119,214 TO DIRKS

Dirks was filed on April 25, 1994 and constitutes effective prior art under 35 U.S.C. § 103.

Dirks discloses a memory manager for a virtual memory system wherein each time a new address is allocated to a program, a limited number of entries in the page table are examined to determine whether the addresses associated with the entries are no longer in use, and the entries can be removed from the page table. Dirks at Abstract. Dirks also "provides an efficient approach to the allocation of large amounts of address space, without requiring time-consuming, and therefore annoying, interruptions in processing." Dirks at Col. 8, lines 57-67. The page table disclosed in Dirks maps physical addresses to virtual addresses and uses a hash function to identify entries. Dirks at Col. 5, lines 4-13. Entries from the page table expire when the portions of the memory associated with that entry are no longer accessed by the CPU, thus becoming inactive. When the page table becomes full, these expired entries must be removed before any further assignments of addresses can occur. Traditionally, this required a time consuming scan of the entire page table, temporarily halting a program's operation. However, Dirks describes a method of scanning portions of the page table on-the-fly every time a new virtual segment identifier (VSID) is assigned to a new application or thread. This "guarantees that a predetermined number of VSIDs are always available in the free list without requiring a time-consuming scan of the complete page table at once." Dirks at Col. 7, lines 11-14.

Dirks teaches the dynamic determination of the number of entries in the page table to scan whenever a new VSID is assigned. In one embodiment of the invention, the number could be calculated as the total number of page table entries divided by the maximum number of active threads. Dirks at col. 8:30-31. However, Dirks also teaches that "[a]ny other suitable approach can be employed to determine the number of entries to be examined during each step of the sweeping process" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46.

Additionally, Dirks was not before the Examiner during prosecution of the '120 patent. As such, it is clear that an examiner would consider Dirks important in determining the patentability of the claims of the '120 patent. Therefore, Dirks presents a substantial new question of patentability.

D. U.S. PATENT NO. 5,724,538 TO MORRIS

Morris was filed on February 27, 1996 and constitutes effective prior art under 35 U.S.C. § 103.

Morris discloses a memory management system implementing a page table technique incorporating virtual memory. The system provides for applying a hash function on a virtual address to form an index into a page table, which gives a pointer to a linked list of entries, which can be searched. Morris further discloses external chaining by providing for a method of searching page table entries (in order to find the physical address, given the virtual address in a reduced tag) wherein it sequentially searches through a linked list of page table entries by following the pointer in each page table entry to the next page table entry in the linked list. Morris at Col. 4, lines 1-15; Col. 5, lines 45-56.

IV. STATEMENT OF SUBSTANTIAL NEW QUESTIONS OF PATENTABILITY

1. MORRISON RAISES A SUBSTANTIAL NEW QUESTION WITH RESPECT TO CLAIMS 1-8 OF THE '120 PATENT .

A substantial new question of patentability ("SNQP") as to claims 1-8 is raised by Morrison, either alone or in combination, as set forth in detail below. Morrison, which was not cited or discussed in the prosecution history of the '120 patent, presents an SNQP because it discloses the technical teaching of a hashing technique to be used with certain data structures to maintain a storage system. The technical teachings of Morris disclose that

> "[h]ashing with lazy deletion means keeping the items hashed by search key in a table of linked lists (separate chains); each time an item is added to a list, any items on that list that the new item shows to be expired are deleted from that list." Morrison at 1155.

In the reasons for allowance, as noted above, the examiner indicated that

> "The prior art does not teach or fairly suggest a method and apparatus for on-the-fly deletion of records in linked lists based on automatic expiration of data as claimed. In other words, the prior art of record does not teach or fairly suggest the means (or an equivalent step in the method claim) of "means for [. . .] accessing **a linked list** at the same time, removing [. . .] some of the expired ones of the records in the linked list," as recited in lines 7 – 8 of claim 1. Although the prior art of record (Nemes, '495 reference) teaches the use of chains of records and the deletion of records, the Applicant, in the Response dated August 11, 1998, Paper No. 5, provided arguments as to why the chain of records as taught in the '495 reference is not the same as **the linked list** as claimed. The Applicant also distinguishes the claimed invention over the teachings of the '499 references. . . .

Paper No. 6 at ¶ 1. Accordingly, a reasonable examiner would consider the technical teachings of Morrison important in deciding the patentability of claims 1-8. Additionally, since the examiner indicated that the technical teaching of Morrison was not present in the prior art or the prosecution history of '120 patent, the teachings of Morrison are substantially new and non-cumulative to the prior art of record.

2. MORRISON IN COMBINATION WITH DIRKS RAISES A SUBSTANTIAL NEW QUESTION WITH RESPECT TO CLAIMS 1-8 OF THE '120 PATENT.

A substantial new question of patentability ("SNQP") as to claims 1-8 is raised by Morrison in combination with Dirks as provided in detail in the manner of applying the cited prior art, as set forth below. Morrison provides a SNQP as stated above with respect to claims 1-8. Furthermore, the combination of Morrison and Dirks was not cited or discussed in the prosecution history of the '120 patent.

Dirks provides the technical teaching that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the

records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, lines 2-14; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. According to Dirks, the number of entries *k* to examine can be calculated as the total number of page table entries divided by the maximum number of active threads. Dirks at Col. 7, lines 15-37. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Accordingly, Dirks provides the technical teaching of "**dynamically determining the maximum number for the record search means to remove.**"

In view of the above, a reasonable examiner would consider the teachings of Morrison in combination with Dirks to be important in deciding patentability, since the combination includes all the elements of the claims and such combination was not before the examiner during the prosecution of the '120 patent. Additionally, the technical teachings of the combination of Morrison and Dirks were not discussed or cited in the prosecution of the '120, thus making this combination non-cumulative to the prosecution history of the '120 patent.

3. THATTE RAISES A SUBSTANTIAL NEW QUESTION WITH RESPECT TO CLAIMS 1-8 OF THE '120 PATENT.

A substantial new question of patentability ("SNQP") as to claims 1-8 is raised by Thatte, either alone or in combination as set forth in detail below. Thatte was not cited or discussed in the prosecution history of the '120 patent. Thatte presents a SNQP because it discloses a "reference count filter" that utilizes the technique of hashing with external chaining and on-the-fly removal of expired records. The on-the-fly removal occurs through a reconciliation operation that takes place when the reference count filter is full. Thatte explains that "[i]f the reference count filter is determined to be full, box 60, the MMU suspends the insertion operation and performs a reconciliation operation, box 62, on the reference count filter . . . to create room in the reference count filter so the suspended insertion operation can be completed." Thatte at Col. 7, lines 21-26. The reconciliation procedure identifies expired records, referred to as garbage, and then removes those garbage records from the reference count filter into a "garbage buffer" for

later reclamation. Thatte at Col. 7, lines 40-8:10. Once that reconciliation is complete, the insertion operation resumes. Thatte at Col. 8, lines 10-14.

In the reasons for allowance, as noted above, the examiner indicated that

> "The prior art does not teach or fairly suggest a method and apparatus for on-the-fly deletion of records in linked lists based on automatic expiration of data as claimed. In other words, the prior art of record does not teach or fairly suggest the means (or an equivalent step in the method claim) of "means for [. . .] accessing **a linked list** at the same time, removing [. . .] some of the expired ones of the records in the linked list," as recited in lines 7 – 8 of claim 1. Although the prior art of record (Nemes, '495 reference) teaches the use of chains of records and the deletion of records, the Applicant, in the Response dated August 11, 1998, Paper No. 5, provided arguments as to why the chain of records as taught in the '495 reference is not the same as **the linked list** as claimed. The Applicant also distinguishes the claimed invention over the teachings of the '499 references. . . .

Paper No. 6 at ¶ 1. However, the reconciliation operation is the type of on-the-fly deletion of records in linked lists claimed by the '120 patent. Moreover, Thatte teaches the dynamic determination of the maximum number of records to remove. Because the method disclosed in Thatte only performs the reconciliation when the reference count filter is full, that method is dynamically determining the maximum number to remove--all of the expired records if the filter is full, or none of the expired records if the filter is not full. Accordingly, a reasonable examiner would consider the technical teachings of Thatte important in deciding the patentability of claims 1-8. Additionally, since the examiner indicated that the technical teaching of Thatte was not present in the prior art or the prosecution history of '120 patent, the teachings of Thatte are substantially new and non-cumulative to the prior art of record.

4. THATTE IN COMBINATION WITH DIRKS RAISES A SUBSTANTIAL NEW QUESTION WITH RESPECT TO CLAIMS 1-8 OF THE '120 PATENT.

A substantial new question of patentability ("SNQP") as to claims 1-8 is raised by Thatte in combination with Dirks as provided in detail in the manner of applying the cited prior art, set forth below. Thatte provides a SNQP as stated above with respect to claims 1-8. Furthermore,

the combination of Thatte and Dirks was not cited or discussed in the prosecution history of the '120 patent.

Dirks teaches that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, lines 2-14; Col. 7, line 66-Col. 8, lines 46; Col. 7, lines 14-37. According to Dirks, the number of entries *k* to examine can be calculated as the total number of page table entries divided by the maximum number of active threads. Dirks at Col. 7, lines 15-37. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Accordingly, Dirks provides the technical teaching of "**dynamically determining the maximum number for the record search means to remove.**"

In view of the above, a reasonable examiner would consider the teachings of Thatte in combination with Dirks to be important in deciding patentability, since the combination includes all the elements of the claims and such combination was not before the examiner during the prosecution of the '120 patent. Additionally, the technical teachings of the combination of Thatte and Dirks were not discussed or cited in the prosecution of the '120, thus making this combination non-cumulative to the prosecution history of the '120 patent.

5. MORRISON IN COMBINATION WITH THATTE RAISES A SUBSTANTIAL NEW QUESTION WITH RESPECT TO CLAIMS 1-8 OF THE '120 PATENT.

A substantial new question of patentability ("SNQP") as to claims 1-8 is raised by Morrison in combination with Thatte, as set forth in detail below, in the manner of applying the cited prior art. Morrison provides a SNQP as stated above with respect to claims 1-8. Thatte provides a SNQP with respect to claims 1-8. Furthermore, the combination of Morrison and Thatte was not cited or discussed in the prosecution history of the '120 patent. Thatte teaches the dynamic determination of the maximum number of records to remove. Because the method disclosed in Thatte only performs the reconciliation when the reference count filter is full, that method is dynamically determining the maximum number to remove-- all of the expired records

if the filter is full, or none of the expired records if the filter is not full. Thatte at Col. 7, line 40-Col. 8, line 18.

In view of the above, a reasonable examiner would consider the technical teachings of Morrison in combination with Thatte to be important in deciding patentability since the combination includes all the elements of the claim and such combination was not before the examiner during the prosecution of the '120 patent. Additionally, the technical teachings of the combination of Morrison and Thatte were not discussed or cited in the prosecution of the '120 patent, thus making this combination non-cumulative to the prosecution history of the '120 patent.

6. DIRKS IN COMBINATION WITH MORRIS RAISES A SUBSTANTIAL NEW QUESTION WITH RESPECT TO CLAIMS 1-8 OF THE '120 PATENT.

A substantial new question of patentability ("SNQP") as to claims 1-8 is raised by Dirks in combination with Morris as provided in detail in the manner of applying the cited prior art, set forth below. The combination of Dirks and Morris was not cited or discussed in the prosecution history of the '120 patent.

Dirks discloses the technical teachings of a memory manager for a virtual memory system wherein each time a new address is allocated to a program, a limited number of entries in the page table are examined to determine whether the addresses associated with the entries are no longer in use, and the entries can be removed from the page table. Dirks describes the use of a hash function to identify entries in the page table. Dirks at Col. 5, lines 4-13. Dirks also provides for on-the-fly removal of inactive entries in the page table, in parallel with the assignment of virtual memory addresses for use by threads and applications. Dirks at Col. 6, lines 5-15.

Further, Dirks teaches that during an insertion operation in the page table, the memory manager examines a number of record entries k in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, lines 2-14, Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. According to Dirks, the number of entries k

to examine can be calculated as the total number of page table entries divided by the maximum number of active threads. Dirks at Col. 7, lines 15-37. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Accordingly, Dirks provides the technical teaching of "**dynamically determining the maximum number for the record search means to remove.**"

Morris discloses a virtual memory management system that includes a page table implemented using hashing with external chaining. As Morris explains, "a hash function 301 is performed on the virtual page number 203 to form an index," which is then used to access an entry in the page table. Morris at Col. 3, line 66- Col. 4, line 1. Where multiple virtual page numbers hash to the same page table entry, the collision is resolved using a chain of entries, each with a pointer to the next entry. Morris at Col. 4, lines 10-20.

In the reasons for allowance, as noted above, the examiner indicated that

> "The prior art does not teach or fairly suggest a method and apparatus for on-the-fly deletion of records in linked lists based on automatic expiration of data as claimed. In other words, the prior art of record does not teach or fairly suggest the means (or an equivalent step in the method claim) of "means for [. . .] accessing **a linked list** at the same time, removing [. . .] some of the expired ones of the records in the linked list," as recited in lines 7 – 8 of claim 1. Although the prior art of record (Nemes, '495 reference) teaches the use of chains of records and the deletion of records, the Applicant, in the Response dated August 11, 1998, Paper No. 5, provided arguments as to why the chain of records as taught in the '495 reference is not the same as **the linked list** as claimed. The Applicant also distinguishes the claimed invention over the teachings of the '499 references. . . .

Paper No. 6 at ¶ 1. Accordingly, a reasonable examiner would consider the technical teachings of Dirks in combination with Morris important in deciding the patentability of claims 1-8. Additionally, since the examiner indicated that the technical teaching of Morris was not present in the prior art or the prosecution history of '120 patent, the teachings of the combination of Dirks and Morris are substantially new and non-cumulative to the prior art of record.

V. DETAILED EXPLANATION UNDER 37 CFR 1.510(B) OF THE PERTINENCY AND MANNER OF APPLYING THE CITED PRIOR ART TO EVERY CLAIM FOR WHICH REEXAMINATION IS REQUESTED.

Claims 1-8 of the '120 patent are fully anticipated under 35 U.S.C. § 102 by, and/or are unpatentable under 35 U.S.C. § 103 in view of, the several different prior art references cited herein, which were not previously considered by the Examiner during the examination of the '120 patent application. Claims 1-8 of the '120 patent are set forth in detail in the attached claim charts (Exhibits CC-A through CC-F) that compare the limitations of the claims of the '120 patent to the pertinent prior art references. As the claim charts demonstrate, claims 1-8 are unpatentable under 35 U.S.C. § 102 and/or 35 U.S.C. § 103 in view of the prior art references presented herein.

A. CLAIMS 1-8 ARE ANTICIPATED BY MORRISON UNDER 35 U.S.C. § 102

Please see the attached Exhibit CC-A presenting claim charts for comparison of Morrison with claims 1-8 of the '120 patent.

CLAIM 1

An information storage and retrieval system, the system comprising:

Morrison discloses an information storage and retrieval system. Morrison at p. 1155, abstract. Specifically, Morrison discloses a dynamic dictionary that is used for records storage and retrieval. *Id.*

a linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring,

Morrison discloses that records in a memory are inserted and retrieved in a separate-chaining hash table of linked lists. Morrison at p. 1155, abstract and ¶ 2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2 ("A sequence of items is given; each item includes a search key, a starting time, and an expiration time.").

a record search means utilizing a search key to access the linked list,

Morrison discloses a record search means utilizing a search key to access the linked list. Morrison at p. 1155, ¶2. In particular, Morrison discloses that each record includes a search key, a starting time, and an expiration time. *Id.* Morrison further discloses that records stored in the table of linked lists are searched by hashing the search key. *Id.*

the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed, and

Morrison discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed by hashing with lazy deletion. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses hashing with lazy deletion, wherein each record includes a starting time and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.

Morrison discloses means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

CLAIM 2

The information storage and retrieval system according to claim 1 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Morrison discloses means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records. Morrison at pp. 1155-1161. In

particular, Morrison teaches a method for calculating a probabilistic bound on the space complexity of the hashing with lazy deletion algorithm, *i.e.*, the number of expired elements that would remain in the linked lists forming the external chains of the hash table. *See id.* That bound calculation is a means for dynamically determining the maximum number to remove, since the maximum number of elements to remove should never exceed the number of expired elements that would remain in the linked lists.

CLAIM 3

A method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring, the method comprising the steps of:

Morrison discloses a method for storing and retrieving information records in a dynamic dictionary using a linked list to store and provide access to the records, at least some of the records automatically expiring. Morrison at p. 1155, abstract and ¶ 2. Morrison specifically discloses the use of a linked list that can be searched and accessed by hashing a search key. Morrison at p. 1155, ¶2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2 ("A sequence of items is given; each item includes a search key, a starting time, and an expiration time.").

accessing the linked list of records,

Morrison discloses accessing the linked list of records by hashing a search key. Morrison at p. 1155, abstract and ¶2.

identifying at least some of the automatically expired ones of the records, and

Morrison discloses identifying at least some of the automatically expired ones of the records from the linked list. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that each record includes a search key, a starting time, and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed.

Morrison discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

CLAIM 4

The method according to claim 3 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Morrison discloses dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed. Morrison at pp. 1155-1161. In particular, Morrison teaches a method for calculating a probabilistic bound on the space complexity of the hashing with lazy deletion algorithm, *i.e.*, the number of expired elements that would remain in the linked lists forming the external chains of the hash table. *See id.* That bound calculation is a means for dynamically determining the maximum number to remove, since the maximum number of elements to remove should never exceed the number of expired elements that would remain in the linked lists.

CLAIM 5

An information storage and retrieval system, the system comprising:

Morrison discloses an information storage and retrieval system. Morrison at p. 1155, abstract. Specifically, Morrison discloses a dynamic dictionary that is used for records storage and retrieval. *Id.*

a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring,

Morrison discloses hashing with lazy deletion, which provides access to records stored in a memory of the system, the dynamic dictionary, and also discloses using a separate, or external, chaining technique to store the records with the same hash address, at least some of the records automatically expiring. Morrison at p. 1155, abstract and ¶2. In particular, the records disclosed in Morrison are "hashed by search key in a table of linked lists (separate chains); each time an item is added to a list, any items of that list that the new item shows to be expired are deleted from that list." Morrison at p. 1155, ¶2.

a record search means utilizing a search key to access a linked list of records having the same hash address,

Morrison discloses a record search means utilizing a search key to access a linked list of records having the same address. Morrison at p. 1155, ¶2. In particular, Morrison discloses that each record includes a search key, a starting time, and an expiration time. *Id.* Morrison further discloses that records stored in the table of linked lists are searched by hashing the search key. *Id.* Morrison further discloses the case wherein hashing with lazy deletion uses a single bucket in the hash table, in which case the linked list of records would have the same hash address. Morrison at p. 1156, ¶3.

the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed, and

Morrison discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed by hashing with lazy deletion. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses hashing with lazy deletion, wherein a record identifies a starting time and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.

Morrison discloses means, utilizing record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some

expired ones of the records in the accessed linked list of records. Morrison at p. 1155, abstract and ¶2. More specifically, Morrison discloses using hashing with lazy deletion to insert records, seek out and therefore retrieve records, and deleting records "that no longer need to be in the data structure." Morrison at p. 1155, abstract. Morrison further discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed; any records in the linked list that the new record shows to be expired are deleted from the list. Morrison at p. 1155, abstract and ¶2. Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id*.

CLAIM 6

> **The information storage and retrieval system according to claim 5 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.**

Morrison discloses means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records. Morrison at pp. 1155-1161. In particular, Morrison teaches a method for calculating a probabilistic bound on the space complexity of the hashing with lazy deletion algorithm, *i.e.*, the number of expired elements that would remain in the linked lists forming the external chains of the hash table. *See id*. That bound calculation is a means for dynamically determining the maximum number to remove, since the maximum number of elements to remove should never exceed the number of expired elements that would remain in the linked lists.

CLAIM 7

> **A method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring, the method comprising the steps of:**

Morrison discloses a method for storing and retrieving information records in a dynamic dictionary using hashing with lazy deletion on a linked list to store and provide access to the

records, and also discloses using a separate, or external, chaining technique to store records with the same hash address. Morrison at p. 1155, abstract and ¶ 2. Morrison specifically discloses the use of a linked list that can be searched and accessed by hashing a search key. Morrison at p. 1155, ¶2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2 ("A sequence of items is given; each item includes a search key, a starting time, and an expiration time."). Additionally, Morrison discloses the case wherein hashing with lazy deletion uses a single bucket in the hash table, in which case the linked list of records would have the same hash address. Morrison at p. 1156, ¶3.

accessing a linked list of records having same hash address,

Morrison discloses accessing the linked list of records by hashing a search key. Morrison at p. 1155, abstract and ¶2. Morrison also discloses the case where hashing with lazy deletion uses only a single bucket in the hash table of linked lists, and therefore discloses a linked list of records having the same hash address. Morrison at p. 1156.

identifying at least some of the automatically expired ones of the records,

Morrison discloses identifying at least some of the automatically expired ones of the records from the linked list. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that each record includes a search key, a starting time, and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed, and

Morrison discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are

deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

inserting, retrieving or deleting one of the records from the system following the step of removing.

Morrison discloses inserting, retrieving, or deleting one of the records from the system following the step of removing. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that expired records are only deleted when they are encountered during an insertion operation, indicating that the insertion operation has not yet completed before the automatically expiring records are removed. Morrison at p. 1155, ¶2.

CLAIM 8

The method according to claim 7 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Morrison discloses dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed. Morrison at pp. 1155-1161. In particular, Morrison teaches a method for calculating a probabilistic bound on the space complexity of the hashing with lazy deletion algorithm, *i.e.*, the number of expired elements that would remain in the linked lists forming the external chains of the hash table. *See id.* That bound calculation is a means for dynamically determining the maximum number to remove, since the maximum number of elements to remove should never exceed the number of expired elements that would remain in the linked lists.

B. CLAIMS 1-8 ARE RENDERED OBVIOUS BY MORRISON IN VIEW OF DIRKS UNDER 35 U.S.C. § 103

Please see the attached Exhibit CC-B, presenting claim charts for comparison of Morrison in view of Dirks with claims 1-8 of the '120 patent.

Reasons to Combine:

One of ordinary skill in the art would have been motivated to combine the memory manager in Dirks with the dynamic dictionary in Morrison in order to provide for efficiently determining the extent to which the automatically expiring records in Morrison should be removed. Both Morrison and Dirks are directed towards navigating a list-type data structure and removing inactive or expired records as efficiently as possible. Furthermore, both Morrison and Dirks disclose the use of a hashing technique in an effort to effectively allocate space within a memory system. Additionally, one of ordinary skill in the art would recognize that the result of combining Morrison with Dirks would be nothing more than the predictable use of prior art elements according to their established functions. The result would simply be the application of Dirks' recycles operation to the linked list data structure disclosed in Morrison.

CLAIM 1

An information storage and retrieval system, the system comprising:

Morrison in view of Dirks discloses an information storage and retrieval system. Morrison at p. 1155, abstract. Specifically, Morrison discloses a dynamic dictionary that is used for records storage and retrieval. *Id.*

a linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring,

Morrison in view of Dirks discloses that records in a memory are inserted and sought in a separate-chaining hash table of linked lists. Morrison at p. 1155, abstract and ¶ 2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2 ("A sequence of items is given; each item includes a search key, a starting time, and an expiration time.").

a record search means utilizing a search key to access the linked list,

Morrison in view of Dirks discloses a record search means utilizing a search key to access the linked list. Morrison at p. 1155, ¶2. In particular, Morrison discloses that each record includes a search key, a starting time, and an expiration time. *Id.* Morrison further discloses that records stored in the table of linked lists are searched by hashing the search key. *Id.*

the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed, and

Morrison in view of Dirks discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed by hashing with lazy deletion. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses hashing with lazy deletion, wherein a record identifies a starting time and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list.

Morrison in view of Dirks discloses means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

CLAIM 2

The information storage and retrieval system according to claim 1 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Morrison in view of Dirks discloses the information storage and retrieval system according to claim 1 further including the means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Morrison at p. 1155, ¶2; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In

particular, Morrison discloses accessing a table of linked lists. Morrison at p. 1155, abstract and ¶ 2. Additionally, Dirks discloses that during an insertion operation into the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 2, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 2, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 2.

CLAIM 3

A method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring, the method comprising the steps of:

Morrison in view of Dirks discloses a method for storing and retrieving information records in a dynamic dictionary using a linked list to store and provide access to the records, at least some of the records automatically expiring. Morrison at p. 1155, abstract and ¶ 2. Morrison specifically discloses the use of a linked list that can be searched and accessed by hashing a search key. Morrison at p. 1155, ¶2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2

("A sequence of items is given; each item includes a search key, a starting time, and an expiration time.").

accessing the linked list of records,

Morrison in view of Dirks discloses accessing the linked list of records by hashing a search key. Morrison at p. 1155, abstract and ¶2.

identifying at least some of the automatically expired ones of the records, and

Morrison in view of Dirks discloses identifying at least some of the automatically expired ones of the records from the linked list. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that each record includes a search key, a starting time, and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed.

Morrison in view of Dirks discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

CLAIM 4

The method according to claim 3 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Morrison in view of Dirks discloses the method according to claim 3 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Morrison at p. 1155, ¶2; Dirks at Col. 6, line 66- Col. 7, line 14;

Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Morrison discloses accessing a table of linked lists. Morrison at p. 1155, abstract and ¶ 2. Additionally, Dirks discloses that during an insertion operation into the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 4, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 4, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 4.

CLAIM 5

An information storage and retrieval system, the system comprising:

Morrison in view of Dirks discloses an information storage and retrieval system. Morrison at p. 1155, abstract. Specifically, Morrison discloses a dynamic dictionary that is used for records storage and retrieval. *Id*.

a hashing means to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring,

Morrison in view of Dirks discloses hashing with lazy deletion, which provides access to records stored in a memory of the system, the dynamic dictionary, and also discloses using a separate, or external, chaining technique to store the records with the same hash address, at least

some of the records automatically expiring. Morrison at p. 1155, abstract and ¶2. In particular, the records disclosed in Morrison are "hashed by search key in a table of linked lists (separate chains); each time an item is added to a list, any items of that list that the new item shows to be expired are deleted from that list." Morrison at p. 1155, ¶2.

a record search means utilizing a search key to access a linked list of records having the same hash address,

Morrison in view of Dirks discloses a record search means utilizing a search key to access a linked list of records having the same has address. Morrison at p. 1155, ¶2. In particular, Morrison discloses that each record includes a search key, a starting time, and an expiration time. *Id.* Morrison further discloses that records stored in the table of linked lists are searched by hashing the search key. *Id.* Morrison further discloses the case wherein hashing with lazy deletion uses a single bucket in the hash table, in which case the linked list of records would have the same hash address. Morrison at p. 1156, ¶3.

the record search means including means for identifying and removing at least some expired ones of the records from the linked list of records when the linked list is accessed, and

Morrison in view of Dirks discloses a record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed by hashing with lazy deletion. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses hashing with lazy deletion, wherein a record identifies a starting time and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.

Morrison in view of Dirks discloses means, utilizing record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records. Morrison at p. 1155, abstract and ¶2. More specifically, Morrison discloses using hashing with lazy deletion to insert records,

seek out and therefore retrieve records, and deleting records "that no longer need to be in the data structure." Morrison at p. 1155, abstract. Morrison further discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed; any records in the linked list that the new record shows to be expired are deleted from the list. Morrison at p. 1155, abstract and ¶2. Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

CLAIM 6

The information storage and retrieval system according to claim 5 further including means for dynamically determining maximum number for the record search means to remove in the accessed linked list of records.

Morrison in view of Dirks discloses the information storage and retrieval system according to claim 5 further including the means for dynamically determining the maximum number for the record search means to remove in the accessed linked list of records. Morrison at p. 1155, ¶2; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Morrison discloses accessing a table of linked lists. Morrison at p. 1155, abstract and ¶ 2. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries k in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation k by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 6, as set forth in the detailed description of the '120 patent,

"dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66-Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 6, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 6.

CLAIM 7

A method for storing and retrieving information records using a hashing technique to provide access to the records and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring, the method comprising the steps of:

Morrison in view of Dirks discloses a method for storing and retrieving information records in a dynamic dictionary using hashing with lazy deletion on a linked list to store and provide access to the records, and also discloses using a separate, or external, chaining technique to store records with the same hash address. Morrison at p. 1155, abstract and ¶ 2. Morrison specifically discloses the use of a linked list that can be searched and accessed by hashing a search key. Morrison at p. 1155, ¶2. Morrison further discloses that at least some of the records automatically expire and are removed from the list when the starting time of a new record added to the list is later than another record's expiration time. Morrison at p. 1155, ¶2 ("A sequence of items is given; each item includes a search key, a starting time, and an expiration time."). Additionally, Morrison discloses the case wherein hashing with lazy deletion uses a single bucket in the hash table, in which case the linked list of records would have the same hash address. Morrison at p. 1156, ¶3.

accessing a linked list of records having same hash address,

Morrison in view of Dirks discloses accessing the linked list of records by hashing a search key. Morrison at p. 1155, abstract and ¶2. Morrison also discloses the case where hashing with lazy deletion uses only a single bucket in the hash table of linked lists, and therefore discloses a linked list of records having the same hash address. Morrison at p. 1156.

identifying at least some of the automatically expired ones of the records,

Morrison in view of Dirks discloses identifying at least some of the automatically expired ones of the records from the linked list. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that each record includes a search key, a starting time, and an expiration time, and each time the linked list is accessed to add a new item to the list, for example, any items on the list that the new item shows to be expired are deleted from the list. *Id.*

removing at least some of the automatically expired records from the linked list when the linked list is accessed, and

Morrison in view of Dirks discloses removing at least some of the automatically expired records from the linked list when the linked list is accessed. Morrison at p. 1155, abstract and ¶ 2. In particular, Morrison discloses that in hashing with lazy deletion, records are hashed by a search key in a table of linked lists, and each time a record is added to the linked list and the linked list is therefore accessed, any records in the linked list that the new record shows to be expired are deleted from the list. *Id.* Morrison refers to this form of information storage and retrieval as "hashing with lazy deletion" because "there is no separate operation associated with clearing expired items out of the table: expired items are only deleted when they are encountered during an insertion operation." *Id.*

inserting, retrieving or deleting one of the records from the system following the step of removing.

Morrison in view of Dirks discloses inserting, retrieving, or deleting one of the records from the system following the step of removing. Morrison at p. 1155, abstract and ¶ 2. Specifically, Morrison discloses that expired records are only deleted when they are encountered during an insertion operation, indicating that the insertion operation has not yet completed before the automatically expiring records are removed. Morrison at p. 1155, ¶2.

CLAIM 8

The method according to claim 7 further including the step of dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.

Morrison in view of Dirks discloses the method according to claim 7 further including the step of dynamically determining the maximum number of expired ones of the records to remove when the linked list is accessed. Morrison at p. 1155, ¶2; Dirks at Col. 6, line 66- Col. 7, line 14; Col. 8, lines 13-30; Col. 3, lines 25-32. In particular, Morrison discloses accessing a table of linked lists. Morrison at p. 1155, abstract and ¶ 2. Additionally, Dirks discloses that during an insertion operation in the page table, the memory manager examines a number of record entries *k* in order to determine if any of the records are inactive and need to be recycled, or removed. Dirks at Col. 6, lines 15-34; Col. 7, line 66- Col. 8, line 46; Col. 7, lines 14-37. Dirks further discloses one method of determining the maximum number of expiring records to be removed during the insertion operation *k* by dividing the total number of record entries by the maximum number of active threads. Dirks at Col. 7, line 66- Col. 8, line 46. It can also be calculated using "[a]ny other suitable approach" and "it is not necessary that the number of examined entries be fixed for each step. Rather, it might vary from one step to the next." Dirks at Col. 7, lines 38-46. Under the broadest reasonable interpretation of claim 8, as set forth in the detailed description of the '120 patent, "dynamically determining the maximum number for the search record means to remove" includes "sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them." '120 patent at Col. 6, line 66- Col. 7, line 15. As a result, under the broadest reasonable interpretation of claim 8, in disclosing a number *k* of automatically expiring records to be swept or removed wherein *k* can include all, some, or none of the records, Dirks discloses the limitations of claim 8.

C. CLAIMS 1-8 ARE ANTICIPATED BY THATTE UNDER 35 U.S.C. § 102

Please see the attached Exhibit CC-C, presenting claim charts for comparison of Thatte with claims 1-8 of the '120 patent.

CLAIM 1

An information storage and retrieval system, the system comprising:

Thatte discloses an information storage and retrieval system. Thatte at Col. 1, lines 7-13; Col. 2, lines 3- 10; Col. 5, lines 40-62. In particular, Thatte discloses a memory management

unit ("MMU") and associated reference count filter to perform memory management functions. Thatte at Col. 2, lines 3-10; Col. 6, lines 1-10; Col. 6, lines 64-67; Col. 5, line 66- Col. 6, line 1.

a linked list to store and provide access to records stored in a memory of the system, at least some of the records automatically expiring,

Thatte discloses a linked list to store and provide access to records that contain virtual addresses, at least some of the records automatically expiring. Thatte at Col. 8, lines 49-62; Col. 5, lines 40-62; Col. 7, lines 1-60. In particular, Thatte discloses marking pointers to references, and evacuating, or expiring, all unmarked pointers to records as garbage. Thatte at Col. 7, line 61- Col. 8, line 18. Thatte further discloses using a linked list to store these records. Thatte at Col. 8, lines 57-59.

a record search means utilizing a search key to access the linked list,

Thatte discloses a record search means, the "locate" operation of the MMU and reference count filter, utilizing a search key to access the linked list. Thatte at Col. 8, lines 39-62; Fig. 7. Specifically, Thatte discloses that the MMU in association with the reference count filter performs insert, delete, reconcile, and locate operations. Thatte at Col. 6, lines 64-68; Col. 8, line 55-62. As further disclosed in Thatte, a virtual address functions as a search key, as it is "applied to means for implementing a hash function," wherein the "hashed output from the hash function is applied as an address to a 'hash bucket' table, in which a corresponding entry is located" in a linked list. Thatte at Col. 8, lines 49-62; Fig. 7.