Exhibit 20

US005893120C1

(12) **EX PARTE REEXAMINATION CERTIFICATE** (8148th)

United States Patent

Nemes

(10) **Number:** **US 5,893,120 C1**

(45) **Certificate Issued:** **Apr. 12, 2011**

(54) **METHODS AND APPARATUS FOR INFORMATION STORAGE AND RETRIEVAL USING A HASHING TECHNIQUE WITH EXTERNAL CHAINING AND ON-THE-FLY REMOVAL OF EXPIRED DATA**

(76) Inventor: **Richard Michael Nemes**, Brooklyn, NY (US)

Reexamination Request:
No. 90/010,856, Feb. 9, 2010

Reexamination Certificate for:

Patent No.:	**5,893,120**
Issued:	**Apr. 6, 1999**
Appl. No.:	**08/775,864**
Filed:	**Jan. 2, 1997**

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/999.001**; 707/999.1; 707/999.101; 707/999.202; 707/999.206; 707/E17.036

(58) **Field of Classification Search** .... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,695,949	A	9/1987	Thatte et al.
4,989,132	A	1/1991	Mellender et al.
4,996,663	A	2/1991	Nemes
5,043,885	A	8/1991	Robinson
5,701,432	A	12/1997	Wong et al.
5,724,538	A	3/1998	Morris et al.
5,765,174	A	6/1998	Bishop
5,778,430	A	7/1998	Ish et al.
5,881,241	A	3/1999	Corbin
5,991,775	A	11/1999	Beardsley et al.
6,119,214	A	9/2000	Dirks
6,243,667	B1	6/2001	Kerr et al.

OTHER PUBLICATIONS

John A. Morrison, Larry A. Shepp, and Christopher J. Van Wyk, A Queing Analysis of Hashing with Lazy Deletion, 16 Siam J. Comput. 6, 1155–1164, Dec. 1997.

Johnson, T. et al.; A Distributed, Replicated, Data–Balanced Search Structure; pp. 1–27.

Appel, A. W., et al.; Hash–consing Garbage Collection, pp. 1–18; Feb. 1993.

Bays, C.; Some Techniques for Structuring Chained Hash Tables; The Computer Journal; vol. 16.; No. 2; pp. 126–131; Apr. 1972.

Bastani, F. B. et al.; Concurrent Maintenance of Data Structures in a Distributed Environment; The Computer Journal; vol. 31; No. 2; 1988; pp. 165–174.

Kruse, R. L., et al.; CD–ROM Prepared by Mailhot, P.A.; Prentice Hall—Data Structures and Program Design in C++; 2000; 717 Pgs.

Kirsch, A., et al; Hash–Based Techniques for High–Speed Packet Processing; pp. 1–40.

Seligmann, J., et al.; Incremental Mature Garbage Collection Using the Train Algorithm; 1995; pp. 235–252.

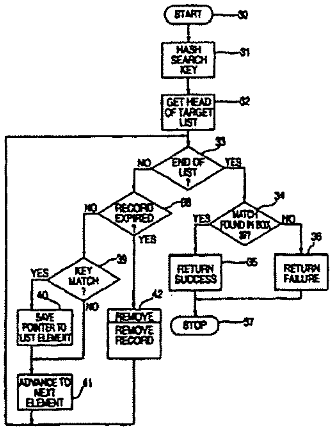
Bohannon, P., et al.; The Architecture of the Dali Main–Memory Storage Manager; 1997; pp. 115–151.

(Continued)

Primary Examiner—Alexander J Kosowski

(57) **ABSTRACT**

A method and apparatus for performing storage and retrieval in an information storage system is disclosed that uses the hashing technique with the external chaining method for collision resolution. In order to prevent performance deterioration due to the presence of automatically expiring data items, a garbage collection technique is used that removes all expired records stored in the system in the external chain targeted by a probe into the data storage system. More particularly, each insertion, retrieval, or deletion of a record is an occasion to search an entire linked-list chain of records for expired items and then remove them. Because an expired data item will not remain in the system long term if the system is frequently probed, it is useful for large information storage systems that are heavily used, require the fast access provided by the hashing, and cannot be taken off-line for removal of expired data.



Defendants' Exhibit

Exhibit No. 147H

Case No. 6:09-cv-00269-LED

OTHER PUBLICATIONS

Van Wyk, C. J., et al.; Algorithmica—The Complexity of Hashing with Lazy Deletion; 1986; pp. 17–29.

Szymanski, T. G., et al.; Special Feature: Goalie: A Space Efficient System for VLSI Artwork Analysis; 1985; pp. 64–72.

van Rijsbergen, C. J.; Department of Computing Science—University of Glasgow; Information Retrieval; pp. 1–147.

Jannick, J.; Sigmond Record, vol. 24, No. 1; Implementing Deletion in B+–Trees; Mar. 1995; pp. 33–38.

Rusling; D. A.; The Linux Kernel; Version 0.8–3; pp. 1–176.

Chandranmenon, G. P.; et al.; IEEE/ACM Transactions on Networking, vol. 4, No. 2; Trading Packet Headers for Packet Processing; pp. 141–152; 1996.

Ramakrishna, M. V.; Hashing in Practice, Analysis of Hashing and Universal Hashing; 1988; pp. 191–199.

Varghese, G., et al.; Hashed and Hierarchical Timing Wheels: Data Structures for the Efficient Implementation of a Timer Facility; 1987; pp. 25–38.

Waldvogel, M., et al.; Scalable High Speed IP Routing Lookups; 1997; pp. 25–36.

Fagin, R., et al.; ACM Transactions on Database Systems, vol. 4, No. 3; Extendible Hashing—A Fast Access Method for Dynamic Files; Sep. 1979; pp. 315–344.

Schimdt, J .P., et al.; On Aspects of Universality and Performance for Closed Hashing; 1989; pp. 355–366.

Larson, P. A; Algorithms and Data Structures; Dynamic Hash Tables, vol. 31, No. 4; Apr. 1988; pp. 446–457.

Chiang, K. W., et al.; Time Efficient VLSI Artwork Analysis Algorithms in GOALIE2; 1988; pp. 471–476.

Maurer, W. D. et al; Computing Surveys, vol, 7, No. 1; Hash Table Methods; Mar. 1975; pp. 5–19.

Deutsch, L. P., et al.; Communications of the ACM, vol. 19, No. 9; An Efficient, Incremental, Automatic Garbage Collector; Sep. 1976; pp. 522–525.

Graham, S. L., et al.; Communications of the ACM, vol. 21, No. 7; Pseudochaining in Hash Tables; Jul. 1978; pp. 554–557.

Yao, A.; Journal of the Association for Computing Machinery, vol. 28, No. 3; Should Tables be Sorted!; Jul. 1981; pp. 615–628.

Karger, D., et al; Consistent Hasing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web; 1997; pp. 654–663.

Celis, P., et al; Information Sciences 62, The Analysis of Hashing with Lazy Deletions; 1992; pp.13–36.

Albers, S., et al.; Self–Organizing Data Structures; pp. 1–39.

Seltzer, M.; A New Hashing Package for UNIX; Winter '91; pp. 173–184.

Tamassia, R.; ACM Computing Surveys, vol. 28, No. 1; Data Structures; Mar. 1996; pp. 23–26

Morrison, J. A., et al., et al; SIAM J. Comp. vol. 16, No. 6—A Queuing Analysis of Hashing with Lazy Deletion; Dec. 1987; pp. 1155–1164.

Dietzfelbinger, M., et al.; Dynamic Perfect Hashing: Upper and Lower Bounds, Revised Version; Jan. 7, 1990; pp. 1–33.

Kenyon–Mathieu, C.M., et al.; General Methods for the Analysis of the Maximum Size of Dynamic Data Structures; pp. 473–487.

Aldous, D., et al.; SIAM J Comp., vol. 21, No. 4, Maximum Size of a Dynamic Data Structure: Hashing with Lazy Deletion Revisited*; Aug. 1992; pp. 713–732.

Printout—Ethernet Address Resolution Protocol; 7 Pgs.

Printout—Copyright (c) 1989, 1993—The Regents of University of California; Copyright (c) 1995—Paul Henning Kamp; 6 Pgs.

Printout—Copyright (c) 1994 by Florian La Roche; pp. 1–20.

Brown, E. W.; Execution Performance Issues in Full–Text Information Retrieval—Technical Report 95–81; Oct. 1995; 179 Pgs.

Costello, A. M. et al.; Redesigning the BSD Callout and Timer Facilities; Nov. 2, 1995; 12 Pgs.

Foster, J.M.—Senior Research Fellow, Dept. of Engineering, Aberdeen University; List Processing; 1967; 54 Pgs.

Printout—Wavelan Pomcia driver, Jean II–HPLB '96; pp. 1–78.

Van Wyk, C. J.; Data Structures & C Programs; 1988; 30 Pgs.

Sessions, R., Reusable Data Structures For C; 1989; 33 Pgs.

Frakes, W. B., et al.; Information Retrieval Data Structures & Algorithms, Chapter 13; 1992; pp. 293–305.

Xu, J.; IEEE Transactions on Computers, vo. 51, No. 9; Cost–Effective Flow Table Designs for High–Speed Routers: Architecture and Performance Evaluation; Sep. 2002; pp. 1089–1099.

Keshav, S.; On the Efficient Implementation of Fair Queuing; Pgs. 1–20.

Kruse, R. L.; Data Structures and Program Design 2^{nd} Edition, Chapter 4; 1987, 1984; pp. 104–155.

Vanghese, G., et al.; Hashed and Hierarchical Timing and Wheels: Data Structures for the Efficient Implementation of a Time Facility; 1987; pp. 25–38.

Xu, J.; IEEE Transactions on Computers, vo. 51, No. 9; Cost–Effective Flow Table Designs for High–Speed Routers: Architecture and Performance Evaluation; Sep. 2002; pp. 1089–1099.

EX PARTE REEXAMINATION CERTIFICATE ISSUED UNDER 35 U.S.C. 307

THE PATENT IS HEREBY AMENDED AS INDICATED BELOW.

Matter enclosed in heavy brackets [ ] appeared in the patent, but has been deleted and is no longer a part of the patent; matter printed in italics indicates additions made to the patent.

AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

The patentability of claims **1**, **2**, **5** and **6** is confirmed.

Claims **3** and **7** are determined to be patentable as amended.

Claims **4** and **8**, dependent on an amended claim, are determined to be patentable.

New claims **9**-**12** are added and determined to be patentable.

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:

accessing the linked list of records *to search for a target record*,

identifying at least some of the automatically expired ones of the records *while searching for the target record*, and

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

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:

accessing a linked list of records having same hash address *to search for a target record*,

identifying at least some of the automatically expired ones of the records *while searching for the target record*,

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

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

9. The method of claim 3, further comprising:

deallocating memory of the at least some of the automatically expired records.

10. The method of claim 9, further comprising:

inserting the target record into the linked list if the target record was not found during the searching.

11. The method of claim 7, further comprising:

deallocating memory of the at least some of the automatically expired records.

12. The method of claim 11, wherein the inserting, retrieving or deleting includes inserting the target record into the linked list if the target record was not found during the searching.

* * * * *