AD-A146 786 FUTURE DATABASE MACHINE ARCHITECTURES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA D K HSIAO SEP 84 NPS52-84-014 1/1

UNCLASSIFIED F/G 9/2 NL

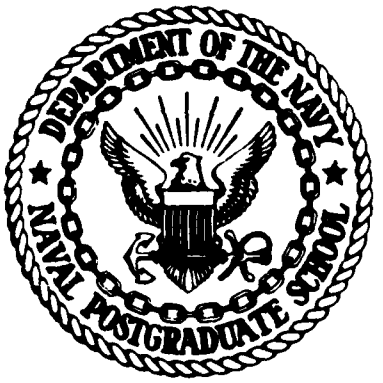
END

1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8

NPS52-84-014

NAVAL POSTGRADUATE SCHOOL

Monterey, California



FUTURE DATABASE MACHINE ARCHITECTURES

David K. Hsiao

September 1984

Prepared for:

Chief of Naval Research
Arlington, VA 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, CAlifornia

Commodore R. H. Shumaker
Superintendent

D. A. Schrady
Provost

The work reported herein was supported by Contract N00014-84-WR-24058 from the Office of Naval Research

This report was prepared by:

David K. Hsiao

DAVID K. HSIAO
Professor and Chairman
of Computer Science

Reviewed by:

David K. Hsiao

DAVID K. HSIAO, Chairman
Department of Computer Science

Kneale T. Marshall

KNEALE T. MARSHALL
Dean of Information and
Policy Sciences

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-84-014	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Future Database Machine Architecture		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David K. Hsiao		8. CONTRACT OR GRANT NUMBER(s) N00014-84-WR-24058
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, Virginia 22217		12. REPORT DATE September 1984 13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

imp 2

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Some of the fundamental architecture issues in the design of high-performance and great-capacity database machines are addressed. Solutions towards resolving these design issues are articulated. This article is written for the New York University's Sumposium in New Directions of Database Systems on the basis of a lecture given at Microelectronic and Computer Technology, Inc. (MCC).

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FUTURE DATABASE MACHINE ARCHITECTURES

David K. Hsiao
Department of Computer Science
Naval Postgraduate School
Monterey, California

There are many software database management systems available on many general-purpose computers ranging from micros to super-mainframes, with many distinct functionalities such as relational vs. hierarchical and text-retrieval vs. formatted-data-retrieval-and-update. Do we really need a few special-purpose machines for database management? In particular, there is the Grosh's law which says as follows:

to next page

"Whenever the capacity of a mainframe computer is saturated with the present work load, there is always another more powerful mainframe which can support the work load cost-effectively (with spare capacity)."

For example, if a computer system such as IBM 3033 is saturated with the database management tasks and the database on the IBM 2314 disks has grown to its capacity, we can replace IBM 3033 CPU with IBM 3081 CPU and IBM 2314 disks with IBM 3340 or 3380 disks. To this example, the Grosh's law may apply. However, the Grosh's law does not always work. Consider the next example. The presence of communications frontend computer can offload the

communications work from the mainframe cost-effectively so that, instead of replacing the present mainframe with a more powerful model due to heavy communications, we can retain the mainframe longer. As it has turned out, the communications frontend provides, in addition to cost-effectiveness, improved performance and new functionality (e.g., serving as gateways to networks). In other words, the Grosh's law does not work, only if the special-purpose computer, which offloads certain types of work from the mainframe, can provide lower cost, higher performance, and newer functionality.

→ Database machines as backend computers can offload the database management work from the mainframe so that we can retain the same mainframe longer. However, the database backend must also demonstrate lower cost, higher performance, and newer functionality. → & 1473

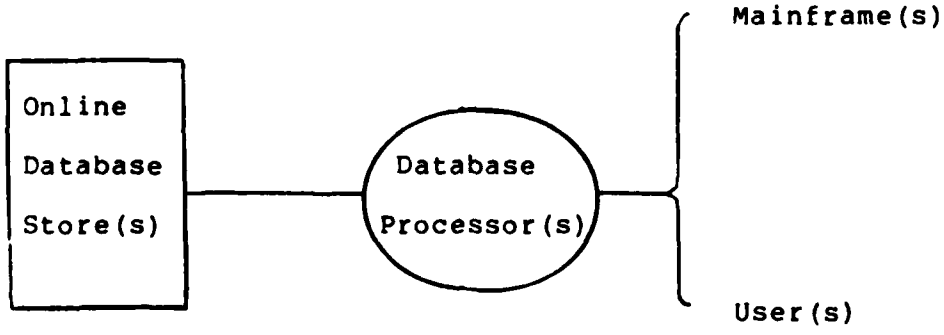
How to Keep the Cost Low?

From the technological viewpoint, the database machine should not be built with distant or expensive technologies such as very large associative arrays. The database machine should utilize existing and improved technologies such as VLSI and parallel transfer disks. From the architectural viewpoint, the database machine should not insist on a pure architecture such as the cellular machine architecture. As a special-purpose machine for database management, we should first characterize its database manage-

ment functions and then realize these functions in the hardware directly. Instead of applying a single architectural principle to the entire machine such as applying pipelining principle to come up with a pipelining machine, we should apply architectural principles such as pipeling, concurrency and parallelism to various design levels of the database machine. From the viewpoint of transforming the software techniques for database management into hardware database machine components, we should not use unproven or seldom-practiced software techniques such as data pools [1]. Instead, we should transform proven software techniques such as indexing and clustering into the hardware.

How to Keep the Performance High?

Let us take a look at the following gross aggregates of a database management systems (DBMS):



First Hypothesis (observation): No matter how great the amount of data is to be processed by the database processors, we can always process the data at the rate that they are being received.

Second Hypothesis: No matter how complex the DBMS software is, there is always a hardware architecture which can cause the execution of the DBMS to be I/O-bound.

The aforementioned two hypotheses, i.e., my observations, have important consequences. In order to articulate their importance, these consequences are expressed as corollaries.

Corollary One: Building very fast and massive database processors is not a difficult task.

Corollary Two: Supporting communications interfaces, preprocessing database transactions and executing DBMS software do not impact upon the performance of the machine.

Corollary Three: The performance of the machine is proportional to the rate that the data can be moved in and out of the database stores.

What these three corollaries are saying is that the performance of a database machine hinges on its 'I/O bandwidth' between the database stores and the database processors. The machine performance has little to do with the amount of processing that the database processors must perform, since we know how to build fast processors.

How can we achieve a very high rate of data movement between the database stores and the database processors?

Solution One: At the device level, we may, for example, use

parallel read-out-and-write-in disks and simultaneous read-out-and-write-in drives. In other words, we may have parallel data streams coming out and going in an individual disk. In addition, we may have many such disks moving parallel data streams in and out of the drives simultaneously.

Solution Two: At system level, we process the incoming or outgoing data streams separately and parallelly at the speed of data movement with minimal communications among the processors.

In other words, we do not merge data streams before processing, since the merged stream would have to move faster and to be processed sooner. Also we do not attempt to increase the traffic in inter-processor communications and to rely on complex communications networks. With these solutions, we have arrived at some important consequences which have impacts on the machine performance and architecture. They are listed below.

Consequence One: The effective rate of data movement is increased by the degree of parallelism and simultaneity of the database stores' read-out and write-in capabilities.

Consequence Two: The processing power and processing speed of the individual database processors are only required to keep up with the rate of data movement of a single data stream and do not need to keep up with the effective rate of data movement.

Consequence Three: Due to the previous consequences, it follows that multiple use of cheaper processors and local memories are

possible for sustaining high performance, that engineering changes of the disk's I/O bus structures and triggering mechanisms are required (but no change to the read/write heads) and that a redesign of the disk controller by incorporating multiple processors and their local memories for the multiple data buses is required.

Third Hypothesis: The processing of meta data such as catalogs, directories and schemas and the processing of raw data such as records, attributes and values are different in nature, scope and sequence.

Corollary Four: The meta data and raw data should have their separate stores and processors. Furthermore, their processing should be made concurrent with the processing of the raw data.

Consequence Four: The design of meta data stores and processors and the design of raw data stores and processors may be different and specially tailored for achieving concurrent processing of both types of data.

Database practitioners do appreciate the differences between meta data and raw data of a database. They also appreciate the different processing and storage requirements of these two types of data. They should be pleased that the architect of the future database machines takes these differences into design consideration.

How to provide newer functionality?

Fourth Hypothesis: Presently, every DBMS is model-specific which implies language-specific and in turn it implies application-specific.

By model-specific, we mean that the DBMS is based on a single data model. For example, the IBM IMS database system is based on hierarchical model. Consequently, the DL/1 is a hierarchical language and all the applications programs written for the IMS are in the DL/1 language.

Solution Three: The new functionality of a future database machine lies in its capability in supporting multiple data models (therefore, data languages and applications).

Corollary Five: The future database machine looks like, for example, a relational machine to the relational database user, a hierarchical machine to the hierarchical database user, a Codasyl machine to the Codasyl database user, and a new machine to the new database user.

Corollary Six: A single machine can support various model-specific databases and languages; or, there are many machines each of which can support a model-specific database and language.

How do we go about designing and implementing a database machine which will support many models?

Solution Four: Come up with a database kernel (or kernel machine) which takes care of all the access and update operations of the raw data and the meta data, which allows 'natural' mappings of

model-specific languages to the machine language of the kernel, and which provides a model-general database structure for various model-specific database organizations.

Fifth Hypothesis: It is possible to come up with a low-cost and high-performance database kernel machine which takes care of all the data-intensive operations [2,3,4].

Sixth Hypothesis: It is also possible to discover natural mappings of model-specific languages to the machine language of the kernel. These mappings are computation-intensive and are not data-intensive [5,6,7,8].

Corollary Seven: The mapping software (i.e., the model-specific software interface) can be quickly executed by the database processors.

Consequence Five: The support of multiple model-specific languages has little impact on the performance and cost of the database machine.

Conclusions: On the basis of these hypotheses, corollaries, consequencies and solutions, the future of database machines is bright, since these are sound hypotheses, reasonable corollaries and good solutions. We believe that the future database machine can be cost-effective with high performance. It can also have new functionality.

Acknowledgement: The author would like to thank MCC for inviting him to deliver a talk on the same subject matter. Much material in this paper is based on the talk given at MCC. That talk was also given at NYU and NPS. The author would like to thank the participants for their critiques and enthusiasms. The work is supported by Contract N00014-84-WR-24058 from the Office of Naval Research.

References:

[1] Hsiao, D. K. (Editor), Advanced Database Machine Architecture, Prentice-Hall, 1983.

[2] Hsiao, D. K., "Cost-effective Ways of Improving Database Computer Performance," AFIPS-Conference Proceedings, V-52 (1983), pp. 293-298.

[3] Banerjee, J., Hsiao, D. K., and Kannan, K., "DBC - A Database Computer for Very Large Databases," IEEE Transactions on Computers, C-28 (1979), pp. 414-429.

[4] Banerjee, J., Baum, R. I., and Hsiao, D. K., "Concepts and Capabilities of a Database Computer," ACM Transactions on Database Systems, V-3, No. 4 (1978), pp. 347-384.

[5] Banerjee, J., Hsiao, D. K., and Ng, F. K., "Database Transformation, Query Translation, and Performance Analysis of a New Database Computer in Supporting Hierarchical Database Management," IEEE Transactions on Software Engineering, SE-6, No. 2

(1980), pp. 91-109.

[6] Banerjee, J., and Hsiao, D. K., "The Use of a Database Machine for Supporting Relational Databases," Proceedings of 4th Workshop on Computer Architecture for Non-numerical Processing, (1978).

[7] _____, "Performance Evaluation of a Database Computer in Supporting Relational Databases," Proceedings of Fourth International Conference on Very Large Data Bases, (1978).

[8] ____, "A Methodology for Supporting Existing Codasyl Databses with New Database Machines," Proceedings of ACM '78 Conference, (1978).

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library Code 0142 Naval Postgraduate School Monterey, CA 93943	2
Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93943	1
Chairman, Code 52Hq Computer Science Department Naval Postgraduate School Monterey, CA 93943	100
Chief of Naval Research Arlington, VA 22217	1

END

FILMED

11-84

DTIC