GAMMA - A High Performance Dataflow Database Machine

David J. DeWitt
Robert H. Gerber
Goetz Graefe
Michael L. Heytens
Krishna B. Kumar
M. Muralikrishna

Computer Sciences Department
University of Wisconsin

This research was partially supported by the Department of Energy under contract #DE-AC02-81ER10920, by the National Science Foundation under grants DCR-8512862, MCS82-01870, and MCS81-05904, and by a Digital Equipment Corporation External Research Grant.

ABSTRACT

In this paper, we present the design, implementation techniques, and initial performance evaluation of Gamma. Gamma is a new relational database machine that exploits dataflow query processing techniques. Gamma is a fully operational prototype consisting of 20 VAX 11/750 computers. The design of Gamma is based on what we learned from building our earlier multiprocessor database machine prototype (DIRECT) and several years of subsequent research on the problems raised by the DIRECT prototype.

In addition to demonstrating that parallelism can really be made to work in a database machine context, the Gamma prototype shows how parallelism can be controlled with minimal control overhead through a combination of the use of algorithms based on hashing and the pipelining of data between processes. Except for 2 messages to initiate each operator of a query tree and 1 message when the operator terminates, the execution of a query is entirely self-scheduling.

1. Introduction

While the database machine field has been a very active area of research for the last 10 years, only a handful of research prototypes [OZKA75, LEIL78, DEWI79a, STON79 HELL81, SU82, GARD83, FISH84, KAKU85, DEMU86] and three commercial products [TERA83, UBEL85, IDM85] have ever been built. None have demonstrated that a highly parallel relational database machine can actually be constructed. Of the commercial products the most successful one (the IDM500) does not exploit parallelism in any form. Why is this so? First, it is obviously much easier to develop a new database machine on paper than it is to turn the idea into a working prototype that can be measured and evaluated. Second, most academic researchers simply do not have sufficient funding to develop their ideas into something that works.[1] Third, since IBM has not endorsed the concept of a database machine there has been limited interest on the part of the major computer vendors to develop such a product.

Two recent events may, however, have radically changed the commercial outlook for database machines. First, a research project to develop a database machine has begun at the IBM Almaden Research Center. Second, the Japanese 5th generation project [MURA83] which is based on the establishment of a highly parallel database machine spurred the development of an intelligent database machine project at MCC. Since most major computer vendors (except IBM) are members of the database program at MCC, one can expect to see a number of new machines emerge in the next 5-10 years. One member company has already begun the design of a highly parallel database machine.

In this paper, we present the design of Gamma, a new relational database machine that exploits dataflow query processing techniques. Gamma is a fully operational prototype whose design is based on what we learned from building our earlier multiprocessor database machine prototype (DIRECT) and several years of subsequent research on the problems raised by the DIRECT prototype. Our evaluation of DIRECT [BITT83] showed a number of major flaws in its design. First, for certain types of queries, DIRECT's performance was severely constrained by its limited I/O bandwidth. This problem was exaggerated by the fact that DIRECT attempted to use parallelism as a substitute for indexing. When one looks at indices from the viewpoint of I/O bandwidth and CPU resources, what an index provides is a mechanism to avoid searching a large piece of the database to answer certain types of queries. With I/O bandwidth a critical resource in any database machine [BORA83], the approach used by DIRECT, while

---

[1] Rumor has it that Teradata has spent almost 40 million dollars developing their machine.

conceptually appealing, leads to disastrous performance [BITT83]. The other major problem with DIRECT was that the number of control actions (messages) required to control the execution of the parallel algorithms used for complex relational operations (e.g. join) was proportional to the product of the sizes of the two input relations. Even with message passing implemented via shared memory, the time spent passing and handling messages dominated the processing and I/O time for this type of query.

We felt that implementing a prototype of Gamma would achieve a number of important objectives. First, it would demonstrate that parallelism can be made to work in a database machine context. While Teradata claims to have already accomplished this, they have not published any performance data and have refused our repeated requests to benchmark their machine. The only numbers published on the performance of DELTA [KAKU85] are those for its parallel sort engine [KAMI85]. These numbers are disappointing as the sort engine is slower than a commercial sorting package on a super-minicomputer. Finally, while the MBDS database machine shows promising speedup factors for selection operations [DEMU86], no results are available for complex operations.

Our second objective is that, although not as flexible as a model, a prototype would provide much more reliable information about the performance bottlenecks of our design. Finally, we felt that a prototype of Gamma would provide a powerful research vehicle for exploring a variety of future research directions such as parallel algorithms for processing queries involving recursion.

The remainder of this paper is organized as follows. The architecture of Gamma and the rationale behind this design is presented in Section 2. In Section 3, we describe the process structure of the Gamma software and discuss how these processes cooperate to execute queries. In particular, we describe our mechanism for processing complex relational queries in a dataflow manner. The mechanism we have designed and implemented requires only three control messages per processor for each operator in the query tree: two to initiate the operator and one for the operator to indicate its completion to the controlling scheduler process. Except for these synchronization messages, the execution of a query is entirely self-scheduling. In Section 4 we describe the algorithms and techniques used to implement each of the relational algebra operations. In Section 5, we present the results of our preliminary performance evaluation of Gamma. Our conclusions and future research directions are described in Section 6.

2. Conclusions and Future Research Directions

In this paper we have presented the design of a new relational database machine, Gamma. Gamma's hardware design is quite simple. Associated with each disk drive is a processor and the processors are interconnected via an interconnection network. The initial prototype consists of 20 VAX 11/750 processors interconnected with an 80 megabit/second token ring. Eight of the processors have a 160 megabyte disk drive. This design, while quite simple, provides high disk bandwidth without requiring the use of unconventional mass storage systems such as parallel read-out disk drives. A second advantage is that the design permits the I/O bandwidth to be expanded incrementally. To utilize the I/O bandwidth available in such a design, all relations in Gamma are horizontally partitioned across all disk drives.

In order to minimize the overhead associated with controlling intraquery parallelism, Gamma exploits dataflow query processing techniques. Each operator in a relational query tree is executed by one or more processes. These processes are placed by the scheduler on a combination of processors with and without disk drives. Except for 3 control messages, 2 at the beginning of the operator and 1 when the operator terminates execution, data flows between between the processes executing the query without any centralized control.

The preliminary performance evaluation of Gamma is very encouraging. The design provides almost linear speedup for both selection and join operations as the number of processors used to execute an operation is increased. Furthermore, the results obtained for a single processor configuration were demonstrated to be very competitive with a commercially available database machine. While we have not yet evaluated our update operations, we have no reason not to expect similar results. Once we have completed the prototype (we have not yet implemented aggregate operations or aggregate functions), we plan on conducting a thorough evaluation of the single and multiuser performance of the system. This evaluation will include both more complex queries and non-uniform distributions of attribute values.

Using the prototype as a research vehicle we intend to explore a number of issues. Some of these issues include the use of adjustable join parallelism as a technique for load balancing and low priority queries, the effectiveness of alternative techniques for implementing bit filtering, index balancing algorithms and the effect of duplicating the root node at multiple sites, evaluation of alternative techniques for handling bucket overflows, and different strategies for processing complex queries.

3. References

[AGRA85] Agrawal, R., and D.J. DeWitt, "Recovery Architectures for Multiprocessor Database Machines," Proceedings of the 1985 SIGMOD Conference, Austin, TX, May, 1985.

[ASTR76] Astrahan, M. M., et. al., "System R: A Relational Approach to Database Management," ACM Transactions on Database Systems, Vol. 1, No. 2, June, 1976.

[BABB79] Babb, E., "Implementing a Relational Database by Means of Specialized Hardware", ACM Transactions on Database Systems, Vol. 4, No. 1, March, 1979.

[BARU84] Baru, C. K. and S.W. Su, "Performance Evaluation of the Statistical Aggregation by Categorization in the SM3 System," Proceedings of the 1984 SIGMOD Conference, Boston, MA, June, 1984.

[BELL73] Bell, J.R., "Threaded Code," Communications of the ACM, Vol. 16, No. 6, (June 1973), pp. 370-372.

[BITT83] Bitton D., D.J. DeWitt, and C. Turbyfill, "Benchmarking Database Systems - A Systematic Approach," Proceedings of the 1983 Very Large Database Conference, October, 1983.

[BLAS79] Blasgen, M. W., Gray, J., Mitoma, M., and T. Price, "The Convoy Phenomenon," Operating System Review, Vol. 13, No. 2, April, 1979.

[BORA82] Boral, H. and D. J. DeWitt, "Applying Data-Flow Techniques to Database Machines," Computer, Vol. 15, No. 8, August, 1982.

[BORA83] Boral H. and D. J. DeWitt, "Database Machines: An Idea Whose Time has Passed," in **Database Machines**, edited by H. Leilich and M. Missikoff, Springer-Verlag, Proceedings of the 1983 International Workshop on Database Machines, Munich, 1983.

[BRAT84] Bratbergsengen, Kjell, "Hashing Methods and Relational Algebra Operations", Proceedings of the 1984 Very Large Database Conference, August, 1984.

[BROW85] Browne, J. C., Dale, A. G., Leung, C. and R. Jenevein, "A Parallel Multi-Stage I/O Architecture with Self-Managing Disk Cache for Database Management Applications," in **Database Machines: Proceedings of the 4th International Workshop**, Springer Verlag, edited by D. J. DeWitt and H. Boral, March, 1985.

[CHOU85] Chou, H-T, DeWitt, D. J., Katz, R., and T. Klug, "Design and Implementation of the Wisconsin Storage System (WiSS)", Software Practices and Experience, Vol. 15, No. 10, October, 1985.

[DEMU86] Demurjian, S. A., Hsiao, D. K., and J. Menon, "A Multi-Backend Database System for Performance Gains, Capacity Growth, and Hardware Upgrade," Proceedings of Second International Conference on Data Engineering, Feb. 1986.

[DEWA75] Dewar, R.B.K., "Indirect Threaded Code," Communications of the ACM, Vol. 18, No. 6, (June 1975), pp. 330-331.

[DEWI79a] DeWitt, D.J., "DIRECT - A Multiprocessor Organization for Supporting Relational Database Management Systems," IEEE Transactions on Computers, June, 1979.

[DEWI79b] DeWitt, D. J., "Query Execution in DIRECT," Proceedings of the 1979 SIGMOD International Conference on Management of Data, May 1979, Boston, Mass.

[DEWI84a] DeWitt, D. J., Katz, R., Olken, F., Shapiro, D., Stonebraker, M. and D. Wood, "Implementation Techniques for Main Memory Database Systems", Proceedings of the 1984 SIGMOD Conference, Boston, MA, June, 1984.

[DEWI84b] DeWitt, D. J., Finkel, R., and Solomon, M., "The Crystal Multicomputer: Design and Implementation Experience," to appear, IEEE Transactions on Software Engineering. Also University of Wisconsin–Madison Computer Sciences Department Technical Report, September, 1984.

[DEWI85] DeWitt, D., and R. Gerber, "Multiprocessor Hash-Based Join Algorithms," Proceedings of the 1985

VLDB Conference, Stockholm, Sweden, August, 1985.

[ENSC85] "Enscribe Programming Manual," Tandem Part# 82583-A00, Tandem Computers Inc., March 1985.

[FISH84] Fishman, D.H., Lai, M.Y., and K. Wilkinson, "Overview of the Jasmin Database Machine," Proceedings of the 1984 SIGMOD Conference, Boston, MA, June, 1984.

[GARD83] Gardarin, G., et. al., "Design of a Multiprocessor Relational Database System," Proceedings of the 1983 IFIP Conference, Paris, 1983.

[GOOD81] Goodman, J. R., "An Investigation of Multiprocessor Structures and Algorithms for Database Management", University of California at Berkeley, Technical Report UCB/ERL, M81/33, May, 1981.

[HELL81] Hell, W. "RDBM - A Relational Database Machine," Proceedings of the 6th Workshop on Computer Architecture for Non-Numeric Processing, June, 1981.

[HEYT85a] Heytens, M., "The Gamma Query Manager," Gamma internal design documentation, December, 1985.

[HEYT85b] Heytens, M., "The Gamma Catalog Manager," Gamma internal design documentation, December, 1985.

[IDM85] The IDM 310 Database Server, Britton-Lee Inc., 1985.

[JARK84] Jarke, M. and J. Koch, "Query Optimization in Database System," ACM Computing Surveys, Vol. 16, No. 2, June, 1984.

[JOHN82] Johnson, R. R. and W. C. Thompson, "A Database Machine Architecture for Performing Aggregations," Tech. Report UCRL - 87419, June 1982.

[KAKU85] Kakuta, T., Miyazaki, N., Shibayama, S., Yokota, H., and K. Murakami, "The Design and Implementation of the Relational Database Machine Delta," in **Database Machines: Proceedings of the 4th International Workshop**, Springer Verlag, edited by D. DeWitt and H. Boral, March, 1985.

[KAMI85] Kamiya, S., et. al., "A Hardware Pipeline Algorithm for Relational Database Operations and Its Implementation Using Dedicated Hardware," Proceedings of the 1985 SIGARCH Conference, Boston, MA, June, 1985.

[KIM85] Kim, M. Y, "Parallel Operation of Magnetic Disk Storage Devices," in **Database Machines: Proceedings of the 4th International Workshop**, Springer Verlag, edited by D. DeWitt and H. Boral, March, 1985.

[KITS83a] Kitsuregawa, M., Tanaka, H., and T. Moto-oka, "Application of Hash to Data Base Machine and Its Architecture", New Generation Computing, Vol. 1, No. 1, 1983.

[KITS83b] Kitsuregawa, M., Tanaka, H., and T. Moto-oka, "Architecture and Performance of Relational Algebra Machine Grace", University of Tokyo, Technical Report, 1983.

[LEIL78] Leilich, H.O., G. Stiege, and H.Ch. Zeidler, "A Search Processor for Database Management Systems," Proceedings of the 4th VLDB International Conference, 1978.

[LIVN85] Livny, M., Khoshafian, S., and H. Boral, "Multi-Disk Management Algorithms," Proceedings of the International Workshop on High Performance Transaction Systems, Pacific Grove, CA, September 1985.

[MURA83] Murakami, K., et. al., "A Relational Data Base Machine: First Step to Knowledge Base Machine," Proceedings of the 10th Symposium on Computer Architecture, Stockholm, Sweden, June 1983.

[OZKA75] Ozkarahan E.A., S.A. Schuster, and K.C. Smith, "RAP - An Associative Processor for Data Base Management," Proc. 1975 NCC, Vol. 45, AFIPS Press, Montvale N.J.

[PROT85] Proteon Associates, Operation and Maintenance Manual for the ProNet Model p8000, Waltham, Mass, 1985.

[RIES78] Ries, D. and R. Epstein, "Evaluation of Distribution Criteria for Distributed Database Systems," UCB/ERL Technical Report M78/22, UC Berkeley, May, 1978.

[SALE84] Salem, K., and H. Garcia-Molina, "Disk Striping", Technical Report No. 332, EECS Department, Princeton University, December 1984.

[SELI79] Selinger,P. G., et. al., "Access Path Selection in a Relational Database Management System," Proceedings of the 1979 SIGMOD Conference, Boston, MA., May 1979.

[STON76] Stonebraker, Michael, Eugene Wong, and Peter Kreps, "The Design and Implementation of INGRES", ACM Transactions on Database Systems, Vol. 1, No. 3, September, 1976.

[STON79] Stonebraker, M. R., "MUFFIN: A Distributed Database Machine," Proceedings of the 1st International Conference on Distributed Computing, Huntsville, Alabama, Oct. 1979, pp. 459-469.

[STON83] Stonebraker, M., et. al., "Performance Enhancements to a Relational Database System," ACM Transactions on Data Systems, Vol. 8, No. 2, (June 1983), pp. 167-185.

[SU82] Su, S.Y.W and K.P. Mikkilineni, "Parallel Algorithms and their Implementation in MICRONET", Proceedings of the 8th VLDB Conference, Mexico City, September, 1982.

[TAND85] 4120-V8 Disk Storage Facility, Tandem Computers Inc., 1985.

[TANE81] Tanenbaum, A. S., **Computer Networks**, Prentice-Hall, 1981.

[TERA83] Teradata: DBC/1012 Data Base Computer Concepts & Facilities, Teradata Corp. Document No. C02-0001-00, 1983.

[UBEL85] Ubell, M., "The Intelligent Database Machine (IDM)," in **Query Processing in Database Systems**, edited by Kim, W., Reiner, D., and D. Batory, Springer-Verlag, 1985.

[VALD84] Valduriez, P., and G. Gardarin, "Join and Semi-Join Algorithms for a Multiprocessor Database Machine", ACM Transactions on Database Systems, Vol. 9, No. 1, March, 1984.

[WAGN73] Wagner, R.E., "Indexing Design Considerations," IBM System Journal, Vol. 12, No. 4, Dec. 1973, pp. 351-367.

[WATS81] Watson, R. W., "Timer-based mechanisms in reliable transport protocol connection management", *Computer Networks 5*, pp. 47-56, 1981.