PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1984,
edited by ICOT.

A RELATIONAL DATAFLOW DATABASE MACHINE BASED ON HIERARCHICAL RING NETWORK

J.I.KIM*, S.R.MAENG, J.W.CHO

KOREA ADVANCED INSTITUTE OF SCIENCE & TECHNOLOGY

SEOUL, KOREA

ABSTRACT

A multiprocessor-based relational database machine is proposed in this paper. The proposed system is regarded as a database server, which can process multiple queries in parallel, in a computer network. And this machine can mitigate the resource contention in the system by partitioning whole database into a finite set of functionally independent databases. The total processing time of queries is reduced by applying dataflow concept to the processing of queries. The functional behavior of the proposed system is simulated, and its performance is evaluated.

1 INTRODUCTION

There have been two important approaches in the development of the database machines, which is defined as any special purpose hardware/software combination that is specially intended to make the database system go faster[1]. One is called the associative disk system, and the other is called the back-end system. RAP[2], CASSM[3], and DBC[4] are the examples of the former, and IDMS, MADMAN, and KSU are the examples of the latter [5].

Many multiprocessor-based database machines have been proposed in order to increase the processing speed as the advancement of VLSI technologies. Direct[6-8], which processes the multiple queries under the dataflow concept, is one of the typical examples in this approach. A ring network-based relational dataflow database machine was proposed by Bic and Herendeen[9]. Tanaka[10] has proposed a database machine that processes relations in the form of streams of an attribute and uses special purpose hardware for database operations.

And the database machine may also serve the information retrieval in computer networks. The database machine, which serves queries from multiple hosts in a computer network, is called a database server[16]. IDM[11] is the first commerical product under this concept. And it provides the efficient usage of a back-end system without restricting the user interface to a particular data manipulation language.

In this paper, a new multiprocessor-based databases machine is proposed. The concept of dataflow computer[12-14] is employed for parallel processing of queries. It is basically a distributed database system which treats relations in the form of streams. Each of the streams consists of a finite number of instances of an attribute in order to avoid the congestion at the interconnection network of processors. The instruction set called base language instruction for the proposed system is designed.

It is called Hierarchical Ring-based Relational Dataflow Database Machine(HRDM). Before introducing HRDM, some assumptions are made as follows;

(1) HRDM serves database operations on a large database which is shared by multiple hosts through a back-end controller.
(2) The Database in HRDM is partitioned into a finite set of smaller databases, which are functionally independent each other.
(3) The data needed by a query is usually stored in the same database. This property is defined as locality of reference.

Additionally, it would not address such issues as the aggregate functions, updates, concurrency control, recovery, etc.; such functions belong to the domain of conventional database management systems. The hosts are responsible for parsing and optimizing queries, and they must produce query packets which can be understood by HRDM without any language translation process.

2 QUERY PROCESSING OF HRDM

Dataflow architectures are well suited for parallel processing of a large scale numerical computation. The order of tuples in a relation does not have significance in the relational data model. So the dataflow concept would increase the parallelism on processing many relational database operations.

---

* He is now in residence at Gold Star Tele-electric Research Institute, Anyang, Korea.

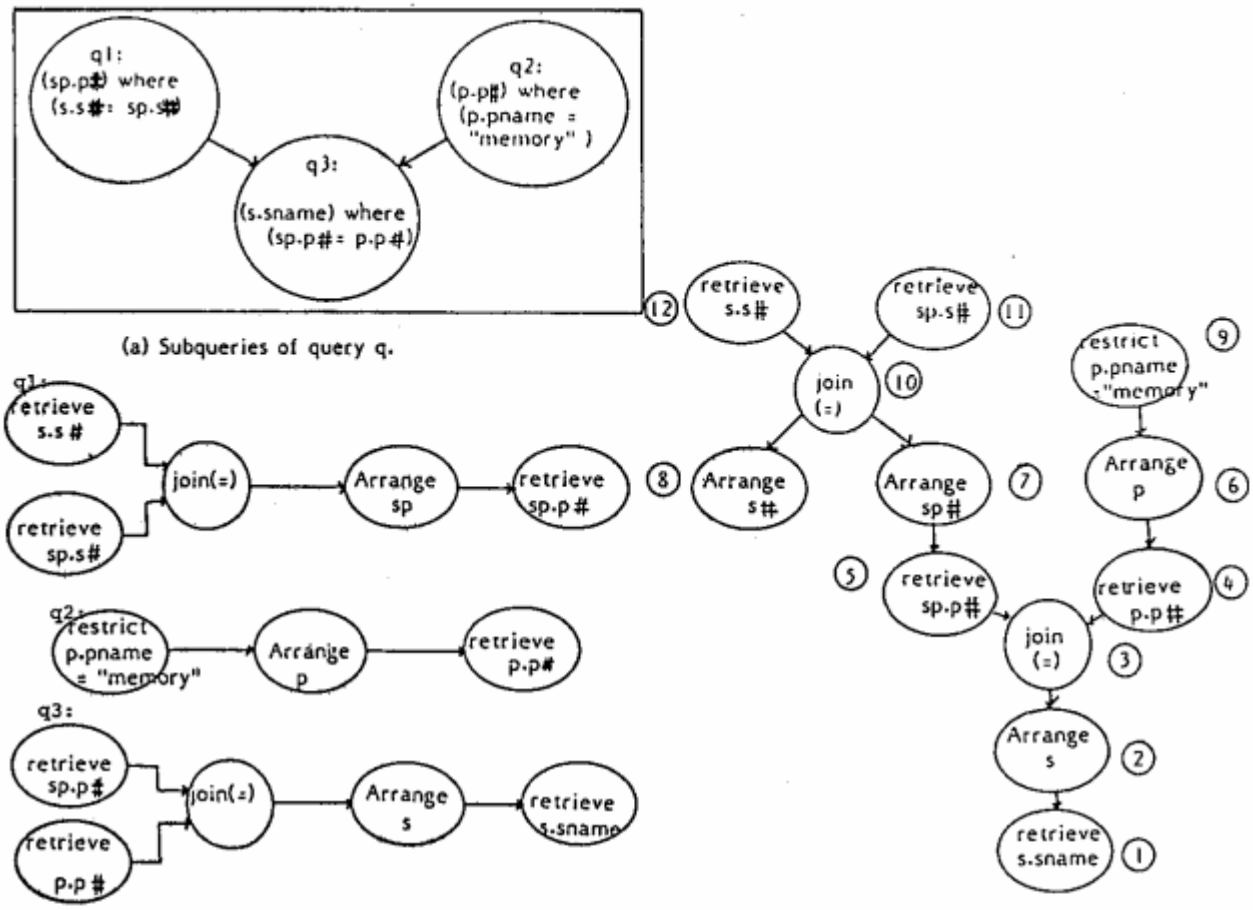


Fig.1. Detatchment process of query q

2.1 Query Processing Strategy

Assume that there are three relations SUPPLIER(s#, sname, status, city), PARTS(p#, pname, color, weight, qty), and SHIPMENT(s#, p#, qty) in HRDM. Consider a query q, which is to get a list of suppliers who supply a part named "memory". It can be expressed Quel-like[17-18] data language as followings;

```
q: range of (s,p,sp) is (SUPPLIER, PARTS,
                         SHIPMENT)
   retrieve (s.sname)
   where p.pname = "memory"
         and s.s# = sp.s#
         and sp.p# = p.p#
```

At the host, detachment process of Quel optimization in Ingres[19] is applied to q. The result is presented in the form of a directed graph as shown in Fig.1(a). Next step in the query processing, shown Fig.1(b), is a translation of each of the subqueries into a set of basic database operations defined in HRDM. This process can easily be implemented as follows;

(1) Calssify a subquery into one- or two-variable type. One-variable type is defined as a subquery which has only one variable in its where clause. Another type has two variables in its perdicate.
(2) A one-variable type subquery is translated into sequential operations of Restriction and Retrieve as q_2 in Fig.1(b).
(3) A two-variable type subquery is interpreted as Join operation.

After these processes, their results are merged while considering the data dependancies among subqueries.

Consider a relation S as an example, which is to be modified twice by subqueries q_1 first and then q_3. Initially, S is not explicity addressed to be arranged by q_1. So additional Arrange operation, which selects the tuples of a relation that satisfy the given condition by certain database operations, e.g., Join and Restriction, is inserted to the graph of q_1 to support consistant database operations. Arrange operation will be explained later in detail.

Such a relation that calls for Arrange operation is easily detected. If a relation is used as an operand of Join but not arranged in the subquery, which calls for Join operation, it becomes a candidate. When one of the candidates is referenced by another subquery, it is necessary to insert an Arrange operation node below the Join node. The node 8 in Fig.1(c) is one of such nodes.

A query is parsed and decomposed into a set of subqueries. Each subqueries are translated and merged with the data dependency among subqueries. The resulting form of query processing at the host is a base language instruction query graph, and its logical representation is shown in Fig.1(c).

In node 11 and 5 of the query graph shown in Fig.1(c), relation SP is referenced. But occurrences of the relation SP at node 11 and those of at node 5 must be different from each other because of Join operation in node 10. So Arrange operation in node 7 is applied to the relation SP to maintain consistant database operation. Arrange operation in node 7 accepts the information called Arrange token from Join in node 10, which contains a bit-stream indicating tuples which satisfy the Join predicate, and selects the satisfied tuples.

For example, assume that the instances of s.s# are (111,112,113,114,115) and those of sp.s# are (121,111,245,114,115,315) at the initial state. Two arrange tokens from Join of node 10 have bit-streams(10011) and (010110), respectively. The former will be consumed by Arrange operation in node 8, and the latter by node 7.

Each data stream and the basic database operation have their own tags for distributed control mechanism. The tag of data stream consists of three fields and they are represented as ⟨Q,I,P⟩. Q is the number of the query directed to the data stream, and I represents the node which absorbs this stream as its operand. Most database operations defined in HRDM have one input port, but some operations like Join require two input ports. P is to denote a port number of the operation which will consume the data stream. The operations in the node I must have the information to identify its operands and the next operation node in the form of ⟨Q,I,N⟩. N denotes the list of the next operation nodes which will consume the partial results from the operation in the node I.

2.2 Cascaded Processing of Join

HRDM performs the Join operation in linear time, under a cascaded mechanism of processing. In other words, the processing time of Join is linearly increased as the number of data is increased. First, a Join operation is unfolded into a number of elementary Join operations as shown in Fig.2.

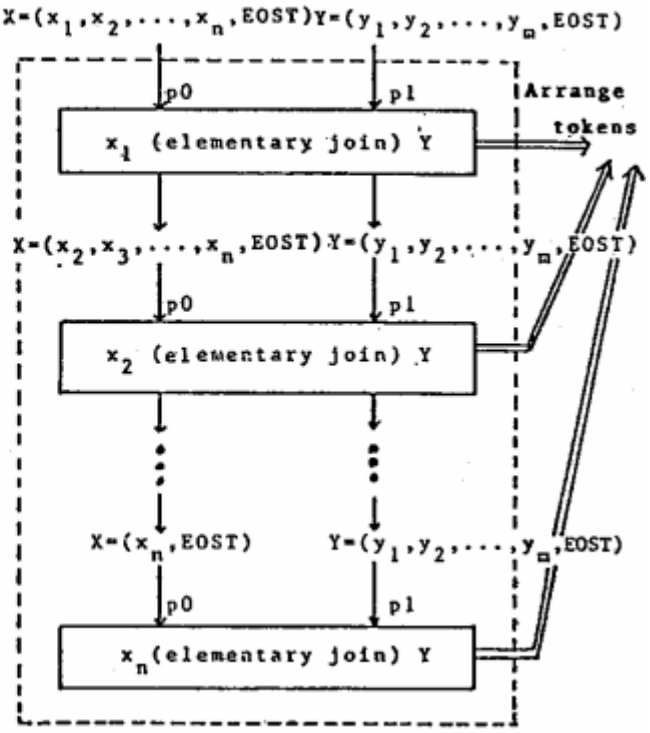


Fig.2. Unfolding a Join operation

Each elementary Join operation has two input ports, p0 and p1. The p0 is used for the reception of the first operand, which decides the actual number of the elementary Join operations. And the p1 is used to receive the counterpart operand, which enables the elementary Join operations. When the data stream X is accepted through p0, the first data element of as CAR x in LISP[15] representation, is captured by an elementary Join operation as long as it has not been allocated by the other data element yet. After this allocation process, the rest of data elements of X i.e., CDR x[15], are sent to the next elementary Join operations. The counterpart operant Y enables the allocated elementary Join operation as being copied. Then, Y is sent to the next elementary Join operation by the enabled elementary Join operation. The enabled elementary Join operation processes the allocated element of X with all the elements of Y, and generates Arrange token until End-Of-Stream-Token(EOST) of the stream Y is detected. All these procedures are summarized in Fig.4.

Each EDM has a special purpose hardware called J-ring for Join operation. If it can process n elementary join operations in cascaded manner with n processing elements, its processing time of Join would be decreased 1/n times than that using a single processor. Implementation and operating mechanism of a J-ring will be presented in the next section.

3 THE SYSTEM ARCHITECTURE OF HRDM

HRDM consists of 3 levels of a hierarchy. Overview of HRDM is shown in Fig.3.

The Back End Controller(BEC) is an interface between HRDM and the host which would issue queries to HRDM. It accepts packets of queries, which may consist of an arbitrary number of queries requesting the data in a certain Elementary Database Machine (EDM) or the data stored in several EDM's.

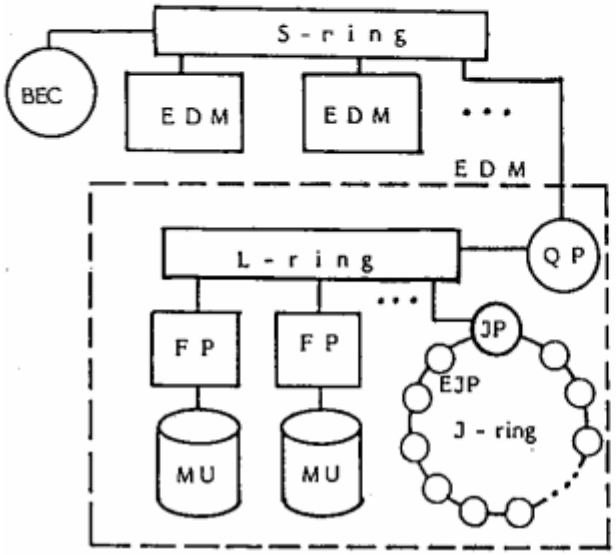


Fig.3. The overview of HRDM

The S-ring is a main interconnection path in HRDM. On S-ring, there are usually some packets of queries from BEC, data streams produced by EDM's, or partial results produced by EDM's and to be processed by other EDM's in the S-ring. In each EDM, there is a Query Processor(QP) that poses an interface to S-ring. QP is a bidirectional interface that absorbs the packets of queries and outputs results produced by EDM. QP absorbs the packets on the S-ring to test for suitability for processing in the EDM. QP sends the selected packets to File Processor (FP)'s or J-ring through L-ring. The results from the EDM are sent to other EDM's or the hosts via S-ring through QP.

There are three kinds of tokens on L-ring. The first type of token is a token of the base language instruction, which is the mnemonic representation of basic database operation in the relational algebra[20] supported by HRDM. The list of base language instructions, their formats, and functions are given in Table 1. The second type of token is a data stream of attributes. And the third type of token is an Arrange token, it is transmitted by FP or J-ring through the L-ring.

When a query issued by the host needs more than one databases, the processing units in an EDM must process the data streams or Arrange tokens generated by other EDM's. Both final results and partial results of queries in an EDM are sent to the S-ring through QP. The final results of queries are sent to the host by BEC, and the partial results are absorbed by the other EDM's in order to get the final results.

FP's have two classes of memory units. One of them is a local memory for working data and the other is a secondary memory device for stored database. The secondary memory devices, denoted by Memory Units(MU) in Fig.3 are used to store the relations of database in the EDM. MU may be made of CCD or Magnetic Bubble memory for fast retrieval of data. The local memory, which is called Working Area, is for managing temporary relations while processing basic operations in a query. Working Area is RAM in FP.

Four major data operations performed by an FP are filtering of data in relations, a simple retrieval of data streams, an Arrange operation on relations, and managing of temporary relations. Filtering operation selects data which satisfy the given condition in the input data stream. RST in the base language instruction of HRDM is an example. A simple retrieval of data streams is denoted as RET in HRDM. Arrange operation implemented as ARR constructs a new version of specified relation in Working Area using the Arrange tokens from the previous Join or filtering operations. The last operation of an FP is to provide fresh data for the current database operation, which relocates all the occurrences of required attributes of a relation required by the current query from MU to Working Area. In the base language instructions, MARK is used for this operation.

J-ring is a special purpose hardware for the Join operations. Join Processor(JP) is an interface between L-ring and J-ring. The base language instruction tokens of JOIN and its operands are captured by JP. If one of them is received by JP, JP examines whether its counterparts reside in the queue. When they are in the queue, that is two data streams and JOIN instruction token with tags ⟨q,i,p0⟩,⟨q,i,p1⟩, and⟨q,i,n⟩, respectively, are identical, Allocation is performed. Then JP unfolds the JOIN instruction into a set of elementary Join operations each of which is allocated to an Elementary Join Processor(EJP) by Allocation. Each elementary Join operation is performed by an EJP in J-ring. The allocated elementary Join operations are executed in cascaded fashion as the counterpart operand passes by EJP's. The generated Arrange tokens are gathered, refined, and issued by JP. The cooperative processing mechanism of JP and EJP's is briefly described in Fig.4.

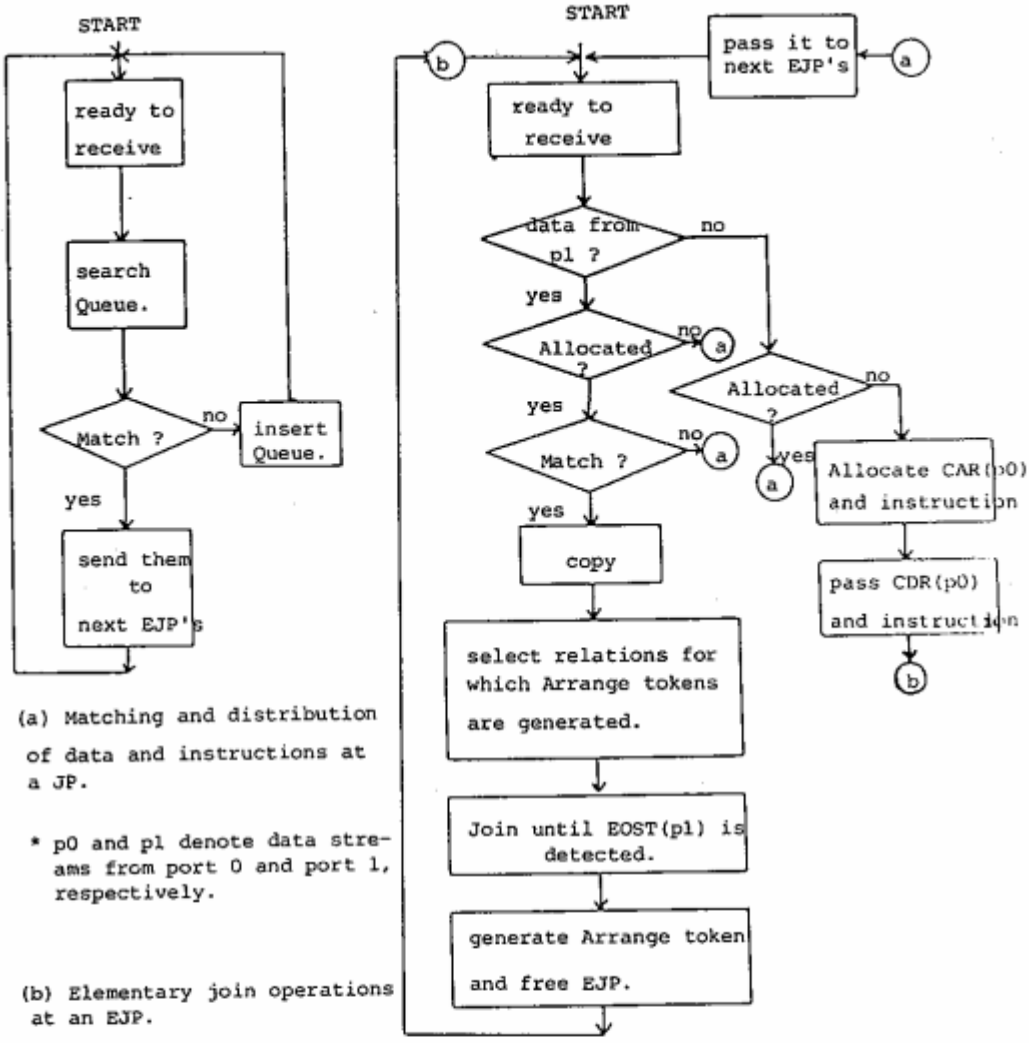


Fig.4. The cooperative processing mechanism of JP and EJP

4 EXPERIMENTS

The functional behavior of HRDM was simulated. The simulation has been performed in VAX 11/780 running UNIX 4.1 BSD, and the program was written in C. And the performance of HRDM is measured by the operational analysis [21].

4.1 The Simulation of HRDM

The simulation model of HRDM consists of two EDM's,each of which consists of a QP, fourteen FP's, and a J-ring. In this section, two sample queries written in Quel like data language will be shown. The base language instruction program for these queries and their graphical representations are also presented. A sample database for these queries consists of three relations, CLS(subj, c#, prof), ST (st#, name, class), and LEC(c#, st#).

(1) Query 1 : Retrieve the names of students who are Ph.D candidates.

```
range of s is ST
retrieve (s.name)
where s.class = "Ph.D"
```

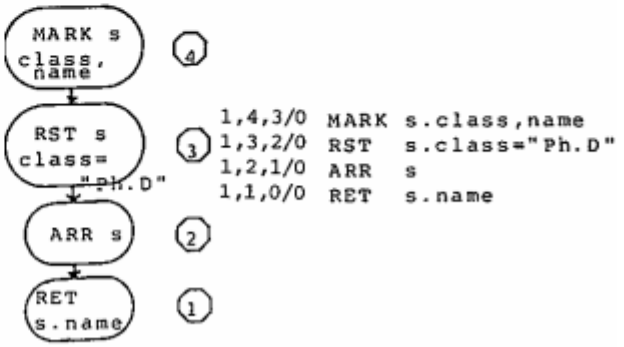


Fig.5. Query graph of Query 1

First, the Restriction operation should be forced on s.class. And the Arrange token for s.name is generated by the Restriction operation. With the Arrange token, the Arrange operation on s.name is performed.

(2) Query 2 : Retrieve the names of the students who are Ph.D candidates and credited by professor "J.W. Cho".

```
range of (s,l,c) is (ST,LEC,CLS)
retrieve (s.name)
where s.class = "Ph.D"
   and s.st# = l.st#
   and c.c# = l.c#
   and c.prof = "J.W.Cho"
```

In relation ST, the s.st# is restricted by Arrange token from the occrurrences of s.class whose value is Ph.D. Between the relations CLS and LEC, a Join operation is performed to retrieve the id.number of students who were taught by professor "J.W.Cho". Since these two operations are functionally indepandant each other, they are processed in parallel.

4.2 The Performance of HRDM

Using the parameters from the simulation, the logical time of processing queries in HRDM is measured using Buzen's algorithm[21].

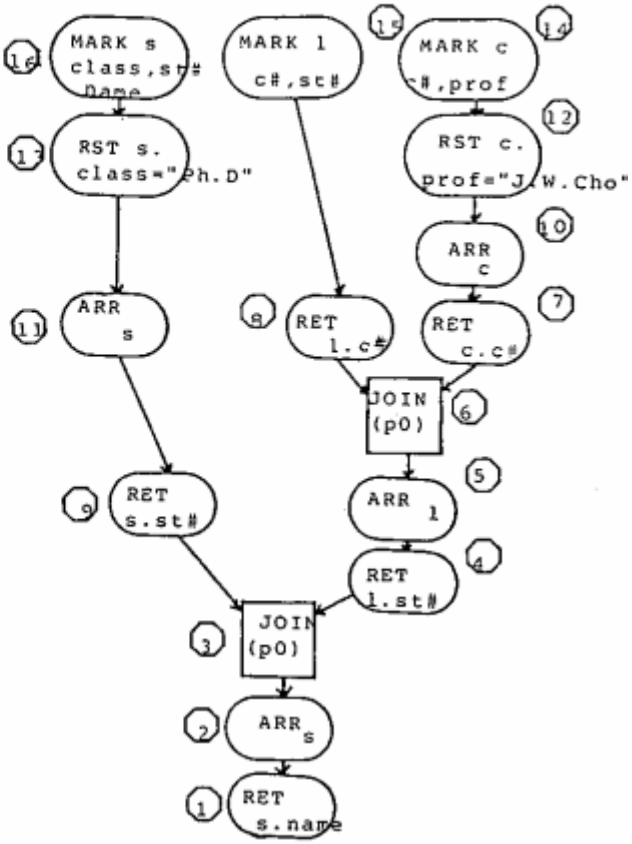
Consider an L-ring which is a main interconnection network in an EDM. There are three types of processing units in an L-ring, QP, FP's, and J-ring. QP sends instruction tokens to J-ring and all FP's. FP generates data streams to J-ring and QP, and makes Arrange tokens to be consumed by itself. Fig.7 contains such a relationship among QP, FP, and J-ring. The dotted line implies the movement within an FP. Regarding the volume of data which are moving on the interconnection network, the visit ratio[21] is calculated.

It is assumed that the speeds of all processing elements on L-ring are equal. The calculated visit ratio and the assumed processing speed of processing elements become the input parameters of Buzen's alrorithm[21].

The experiment showed followings; as the number of queries issued from hosts are increased, the processing time increases. And the system is saturated when the number of different queries are over 20. The performance of HRDM is denoted as the solid line of Fig.8 which indicates the processing time per a query.

Additionally, the representation of relations, that should be transferred among the processing elements in HRDM, is considered.

The quantities of moving data on the interconnection netowrk have a great influence on the performance of the system. As the quantities of data on L-ring becomes larger, the processing time of queries becomed longer. The comparison of two representation methods of moving data is presented in Fig.8. It shows that the attribution-based representation method has the definite advantage in terms of the processing time over relation-based representation method.




```
2,16,13/0   MARK s.class,st#,name
2,15,8/0    MARK l.c#,st#
2,14,12/0   MARK c.c#,prof.
2,13,11/0   RST  s.class="Ph.D"
2,12,10/0   RST  c.prof="J.W.Cho"
2,11,9/0    ARR  s
2,10,7/0    ARR  c
2,9,3/0     RET  s.st#
2,8,6/0     RET  l.c#
2,7,6/1     RET  c.c#
2,6,5/0     JOIN l.c#=c.c#,p0
2,5,4/0     ARR  l
2,4,3/1     RET  l.st#
2,3,2/0     JOIN s.st#=l.st#,p0
2,2,1/0     ARR  s
2,1,0/0     RET  s.name
```

Fig.6 Query graph of Query 2

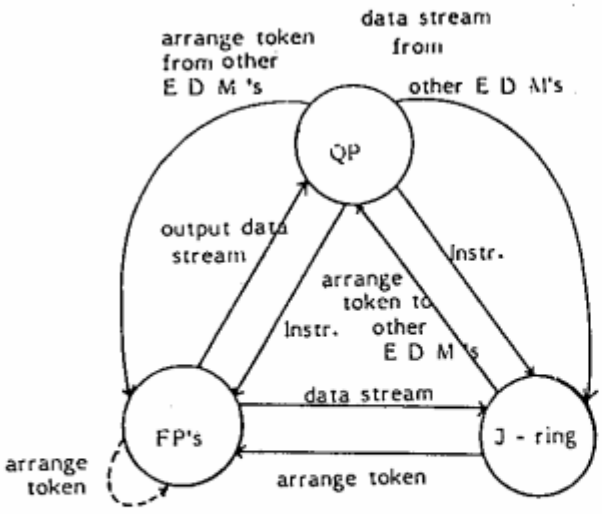


Fig.7. The flow of tokens on L-ring

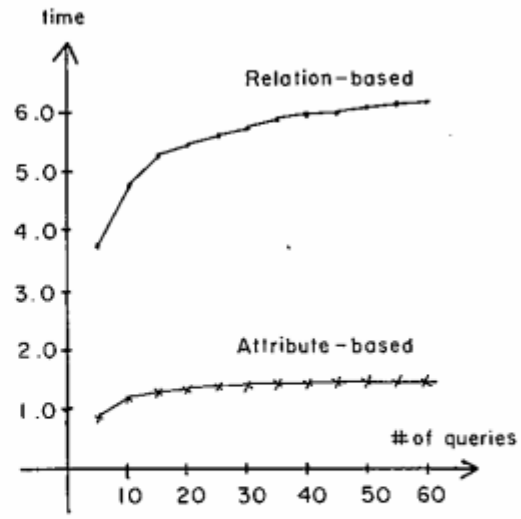


Fig.8. The performance of HRDM

Instruction format	Function
MARK r1,a1,a2,..,ai	Relocates the values of the attributes r1.a1, a2, ..., ai from MU to Working Area in FP.
RET r1,a1	Retrieves the data stream of an attribute r1.a1 and sends it to the next operation.
RST r1,a1,pred	Restricts r1.a1 and generates the Arrange tokens without a data stream.
ARR r1	Arranges the occurences in r1 stored in Working Area as described in the bit-streams of Arrange tokens which have already captured by FP's.
JOIN p0,p1,pred,prt#	Joins r1.a1 and r2.a2 received from p0 and p1, respectively under the predicate and generates the Arrange tokens to the next operations. The objective relations of the Arrange tokens are determined by the value of prt#. They are r1, r2, and both when the value of prt# are 0, 1, and 2, respectively.

Table 1 . The base language instructions of **HRDM**

5 CONCLUSIONS

HRDM(Hierarchical Ring based Relational Dataflow Database Machine) distributes whole database into a set of functionally independent databases. The partition of database partly solves the degradation problem of system performance caused by the contention for resources. The hierarchical ring structure of HRDM is motivated by this.

HRDM serves as a database server in a computer network. All hosts in the computer network may send packets of queries to HRDM in the form of a base instruction language query graph.

HRDM executes the Join operations in the linear time using a special purpose device, called J-ring. In processing queries, all basic operations in queries are performed using the dataflow concept.

And the flow of data between the operations is implemented in the form of stream, whose elements are instances of an attribute in a relation. An experiment to measure the processing time of HRDM shows that the attribute-based representation method can reduce total processing time of queries than the relation-based representation method.

All operational mechanisms previously discussed have been simulated in order to test the functional behavior of them. The performance of the proposed system was measured. And HRDM is being implemented.

REFERENCES

1. Date, C.J., An Introduction to Database Systems, 3rd Edition, Addison-Wesley, Vol.2, pp.341-369, 1983.
2. Ozkarahan,E.A., Schuster,S.A., et al., "RAP - An Associative Processor for Data Base Management," Proceedings of NCC, pp.379-387, May 1975.
3. Su,S.Y.W. and Emand,A., "CASDAL : CASSM's Data Language," ACM TODS, pp.57-91, Mar. 1978.
4. Banerjee,J., Hsiao,D.K., et al., "Concepts and Capabilities of a Database Computer," ACM TODS, pp.347-384, Dec. 1978.
5. Maryanski,F.J., "Backend Database Systems," ACM Computing Survey, No.1, pp.3-25, Mar. 1980.
6. Boral,H. and DeWitt,D.J., "Applying Data Flow Techniques to Data Base Machines," IEEE Computer, No.8, pp.57-63, Aug. 1982.
7. Boral,H. and DeWitt,D.J., "Processor Allocation Strategies for Multiprocessor Database Machines," ACM TODS, Vol.6, No.2, pp.227-254, June 1981.
8. Boral,H. and DeWitt,D.J., "Design Considerations for Dataflow Database Machines," ACM, pp.94-104, May 1980.
9. Bic,L. and Herendeen,M., "An Architecture for a Relational Dataflow Database," ACM SIGPLAN, pp.136-145, 1981.
10. Tanaks,Y., "A Data Stream Database Computer," Computer Science & technologies 1982 in Japan, OHMSHA Ltd. & NORTH-HOLLAND, pp.265-286, 1982.
11. Ubell,M., "The INtelligent Database Machine," Database Engineering, Vol.1, IEEE Computer Society Press, pp.30-32, 1983.
12. Arvind, Gostelow,K.P., et al., "An Asynchronous Programming Language and Computing Machine," Tech. Report #114A, Dept. of Inf. and Comp. Science, UC. Irvine, Dec. 1978.
13. Dennis J.B. and Misunas,D.P., "A Preliminary Architecture for a Basic Data-Flow Processor," Proceedings of the 2nd Annual Conference on Computer Architecture. New York, ACM, pp.126-132, 1975.
14. Myers,G.J., Advances in Computer Architecture, 2nd Edition, John Wiley & Sons, pp.463-494, 1982.
15. Winston P.H. and Horn,B.K.P., LISP, Addison-Wesley, pp.337-387, June 1970.
16. Kim, W., private communication.
17. Woodfill, J., et al., Ingres Version 7 Reference Manual, Univ. of California, Berkeley, 1981.
18. Epstein,R., A Tutorial on Ingres, Memorandum No. ERL-M77-25, Univ. of California, Berkeley, Dec. 1977.
19. Wong, E. and Youssefi,K., "Decomposition-A Strategy for Query Processing," ACM TODS, Vol.1, No.3, pp.223-241, Sept. 1976.
20. Codd, E.F., "A Relational Model of Data for Large Shared Data Banks," CACM, Vol.13, No.6, pp.377-387, June 1970.
21. Denning, P.J., and Buzen,J.P., "The Operational Analysis of Queueing Network Models," Computing Survey, Vol.10, No.3, pp.225-261, Sep. 1978.