RINDA - A RELATIONAL DATABASE PROCESSOR FOR NON-INDEXED QUERIES

Ushio INOUE, Haruo HAYAMI, Hidoki FUKUOKA, and Kenji SUZUKI

NTT Communications and Information Processing Laboratories
1-2356 Take, Yokosuka-Shi, Kanagawa 238-03, Japan

ABSTRACT

A new relational database processor called RINDA is presented. RINDA performs key database operations, such as search and sort, at very high speed with specialized hardware. It is attached to general purpose computers and controled by database management software. In this paper, the architecture of RINDA is described from the viewpoint of hardware and software. Performance evaluation shows RINDA accelerates non-indexed queries of relational databases by 10 to 100 times compared with conventional database management software.

KEYWORDS

database machines, computer architecture, database management, specialized hardware

1. INTRODUCTION

Development of database machines has stopped into the commercial stage from the experimental stage. Commercial database machines are classified into database computers and database processors. Database computers, such as IDM/BL series[3], DBC/1012[13], and RDM[9], are backend processors and all database management functions are performed by the backend.

Database processors, such as CAFS[1] and IDP[8], are processors attached to host computers, and controlled by database management software running on the host. Database processors, in general, perform some database management functions much faster than the host, and the host performs the remaining functions by regular database software. For example, CAFS provides only search and filtering functions for disk storage, and IDP provides only set operations on main storage. The database processor approach is suitable for users with installed computers, because their hardware and software can continue to be used after new database processors are connected to their systems.

RINDA** is a new database processor that aims to accelerate non-indexed queries of relational databases. It performs essential and computationally intense operations[5], such as search and sort, at a very high speed with specialized hardware. RINDA was developed to be attached to the DIPS series of general purpose computers[7,10], and controlled by the relational database management software called DEIMS-5**. The functions of RINDA are limited to the execution of non-indexed queries, and all other functions, such as creation and update of databases, are performed by DEIMS-5 using ordinary software. Indexed queries are also executed by software. The application program interface is based on SQL[6] and RINDA can be utilized easily by existing database users.

RINDA is expected to be used in various NTT sections in order to improve the performance of business database systems.

This paper presents the architecture of RINDA from the viewpoint of hardware and software. In Section 2, problems of conventional relational database management software and solutions by RINDA are described. In Section 3, hardware and software architecture of the RINDA system are described. In Section 4, the performance improvement achieved by RINDA is evaluated.

2. BASIC CONCEPTS

2.1 PROBLEMS OF CONVENTIONAL SYSTEMS

The performance of relational database management software has been improved by the effective use of indexes. However, some queries

** RINDA stands for a Relational Database processor.

** DEIMS stands for Dend6(NTT's name in Japanese) Information Management Systems.
to databases, in practice, can not take advantage of indexes. Conventional database software consumes too much CPU time and processing takes too long when performing non-indexed queries. Cases where the use of indexes is inefficient are as follows:

- Text retrieval is performed for many table rows, because queries include LIKE predicates of SQL.
- Various queries must be executed which specify different columns of the table.
- Maintenance of indexes is impractical because insertion of table rows is very frequent.
- Many table rows are selected by indexes because indexed columns have identical values or the predicate is a range query type.

2.2 SOLUTIONS BY RINDA

In relational database systems, search and sort are the principal operations. Search operations select rows and columns from a table according to predicates specified by users. If the table is stored on disk storage, disk reads are also needed for conducting searches. Searches are performed as the first step of query execution. On the other hand, sort operations are employed to join two tables or calculate aggregate functions. Many other procedures can be performed efficiently using a sort operation. However, both search and sort are very heavy loads for general purpose computers. That is, they cost much CPU and I/O time(5).

To solve the problem, RINDA uses specialized hardware for search and sort operations. RINDA performs on-the-fly search and order-N sort at the data transfer speed of data channels. Both the CPU and I/O time of the host computer are dramatically reduced.

3. ARCHITECTURE

3.1 DESIGN CONSIDERATION

The goal of RINDA is to relieve the DEIMS-5 relational database management software of heavy loads caused by non-indexed queries. Therefore, the following was decided for the design of RINDA.

- Data channels are used to connect RINDA and host computers because of flexibility, expansibility, and independence from the host.
- Database conversion is unnecessary, and users can access the same database at the same time with and without RINDA.
- The application program interface is never changed by introducing RINDA to an existing database system.
- Multiple disk volumes can be searched simultaneously by RINDA in order to shorten disk read time.

The last decision makes the most of RINDA's functions and capabilities.

3.2 HARDWARE

The RINDA system organization is shown in Figure 1. RINDA is composed of CSPs (Content Search Processors) and optional ROPs (Relational Operation Accelerating Processors)(12). Each CSP and ROP has a channel interface and is connected to a DIPS series host computer. They are controlled by channel commands and control data generated by DEIMS-5 running on the host.

A CSP searches tables stored in disk volumes, selects rows and columns specified in a query, and transfers the result to the host. A CSP performs the majority of search functions needed by relational database systems, and has the capability of on-the-fly search for most queries. The seek and search time of disk storage is dramatically reduced by reading multiple tracks at once.

An ROP sorts a table transferred from the host, and transfers the result back to the host. The sort processing is performed in a pipeline at data channel speed. Duplicated rows can be counted and removed during processing.

A typical procedure to execute queries with RINDA is shown in Figure 2. First, selected rows and columns are transferred to the main memory of the host by CSPs. Next, they are sorted as they make the round trip between the CSP: Content Search Processor
ROP: Relational Operation Accelerating Processor
DKC: Disk Controller

Figure 1. RINDA system organization

-383-
main memory and an ROP. Finally, DEIMS-5 transfers the result to the application program requesting the query. When not equipped with the optional ROP, DEIMS-5 performs the ROP functions by software.

3.3 SOFTWARE

The DEIMS-5 database management software organization is shown in Figure 3. Queries, in the form of SQL statements, written by database users are analyzed, optimized, and executed by the database language processing subsystem. New optimization functions for RINDA were implemented in the subsystem.

The RINDA control subsystems, newly developed, provide functions to execute queries with RINDA. Their primary functions are input/output control from/to RINDA, buffer management, and basic relational database operations such as selections and joins. Parallel CSP search of a table distributed over multiple disk volumes is possible for a query.

The database control subsystems provide functions to execute queries without RINDA. Their primary functions are simple read/write of table rows and index management. They also provide functions for centralized database control, such as transaction management, concurrency control, and resource management. Therefore, they are also employed during execution of queries through RINDA.

4. PERFORMANCE

4.1 ENVIRONMENTS

The benchmark database table is based on the extended Wisconsin Benchmark[2,4]. It consists of 10,000 rows, and each row is 208 bytes long except system control data. The RINDA system tested consists of a DIPS-V30 processor[7], two CSPs, and two 500 megabytes disk drives. There was no optional ROP, because its development was not completed in time for testing.

The queries executed are shown in Table 1. The first three queries in the table are standard to the benchmark. The others were executed to test pure selection performance without transfer of selected rows, and to test performance for text retrieval applications. Indexes were not used, because the objectives of RINDA are non-indexed queries. The join, projection, and non-scalar aggregate function queries were not executed, because the ROP devices were unavailable.

4.2 RESULTS

The performance improvement by RINDA is shown in Figure 4. The database processing time is elapsed time from starting execution of the SQL statement to returning the last selected row to the application program. Three main conclusions can be drawn from Figure 4. First, most of the query processing burden imposed on the host CPU is offloaded onto RINDA and performed on-the-fly. The ratio of offload is larger as the rows selected are fewer, and it exceeds 90% when the returned value is a
count(*). Second, RINDA dramatically reduces disk read time needed for the table search. The reduction is caused by the parallel CSP search and multiple tracks read. Finally, performance improvement by RINDA is revolutionary, and RINDA accelerates non-indexed queries by 10 to 100 times compared with the conventional software system.

The performance of RINDA is compared with other database machines[4,11] in Table 2, for reference. In this environment, all selected rows were inserted into a temporary table instead of being returned to the application program. Table 2 indicates that RINDA is quite a bit faster than other machines for the queries evaluated.

Table 1. Executed queries in the test

<table>
<thead>
<tr>
<th>Queries</th>
<th>Corresponding SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-indexed 1% selection</td>
<td>SELECT FROM tank WHERE unique2 =&gt; 5000 and unique2 &lt; 5100</td>
</tr>
<tr>
<td>Non-indexed 10% selection</td>
<td>SELECT FROM tank WHERE unique2 =&gt; 5000 and unique2 &lt; 6000</td>
</tr>
<tr>
<td>Non-indexed MIN</td>
<td>SELECT MIN(unique2) FROM tank</td>
</tr>
<tr>
<td>Non-indexed scalar COUNT</td>
<td>SELECT COUNT(*) FROM tank WHERE unique2 =&gt; 5000 and unique2 &lt; 6000</td>
</tr>
<tr>
<td>LIKE predicate</td>
<td>SELECT COUNT(*) FROM tank WHERE stringu2 LIKE 'XABCX'</td>
</tr>
<tr>
<td>scalar COUNT</td>
<td>SELECT COUNT(*) FROM tank WHERE stringu2 LIKE '%ABC%'</td>
</tr>
</tbody>
</table>

Table 2. Query execution time (in seconds)

<table>
<thead>
<tr>
<th>Machines (year)</th>
<th>Non-index 1% sel.</th>
<th>Non-index 10% sel.</th>
<th>Non-index scalar MIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDM/500 dac (85)</td>
<td>19.9</td>
<td>23.4</td>
<td>19.0</td>
</tr>
<tr>
<td>DBC/1012 (87)</td>
<td>6.86</td>
<td>15.97</td>
<td>4.21</td>
</tr>
<tr>
<td>RINDA (89)</td>
<td>0.80</td>
<td>1.25</td>
<td>4.30</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

This paper has presented a new relational database processor, RINDA. The goal of RINDA is to relieve the database management software of heavy loads caused by non-indexed queries. RINDA performs search and sort operations at very high speed with specialized hardware. Performance improvement by RINDA is revolutionary, and non-indexed queries are accelerated by 10 to 100 times compared with conventional software.

RINDA can be used in a large number of database application systems, such as information retrieval, traffic data analysis, and management of various objects. RINDA is expected to be used in various NTT sections in the near future.

ACKNOWLEDGMENT

The authors wish to thank Hideaki Takada, Jinnosuke Nakamura, Toshio Nakamura, Tadashi Kitamura, and the many others involved in the RINDA project.

REFERENCES


