

Short Paper

Evaluating the Performance of a Distributed Database of Repetitive Elements in Complete Genomes

MING-HUI JIN AND JORNG-TZONG HORNG

Department of Computer Science and Information Engineering

National Central University

Chungli, 320 Taiwan

The original version of the Repeat Sequence Database (RSDB¹) [2] was created based on centralized database systems (CDBSs). RSDB presently includes an enormous amount of data, with the amount of biological data is increasing rapidly. Distributed RSDB (DRSDB) is developed to yield better performance. This study proposed many approaches to data distribution and experimentally determine the best approach to obtain good performance of our database. Experimental results indicate that DRSDB performs well for particular types of queries.

Keywords: databases, genomes, repetitive elements, query processing, distributed databases

1. INTRODUCTION

The amount of biological data available in the post-genomic era is increasing rapidly. Large data stores are thus required. The Repeat Sequence Database (RSDB) [2] is based on CDBSs architecture [1]. The motivation to create an RSDB follows Li et al. [9] who stated that about 43% of the human genome consists of repetitive elements. Approximately 51% of the rice genome is occupied by repetitive elements. Analyzing repetitive elements reveals that repetitive elements in our genome may play an important role in evolutionary genomics.

Distributed database systems (DDBSs) represent the trend for storing a large amount of data [6, 7] owing to its ability to improve system performance. Data distribution and extents of data replication are key factors in determining the performance of distributed database systems [4, 5, 8]. Assumptions are made concerning the distribution of data and replication in a system to simplify the evaluation of performance measures. A data distribution and replication model is specified by four parameters. While the data distribution and replication models appear to be simple, the results generated by them are close to those from complex models.

Received September 5, 2001; accepted April 15, 2002.

Communicated by Jang-Ping Sheu, Myongsoon Park and Makoto Takizawa.

¹ <http://rsdb.csie.ncu.edu.tw>

Data placement is important for achieving high performance shared-nothing parallel database systems and DDBSs [3]. A poor data displacement strategy can result in a non-uniform distribution of data load. However, no consensus exists on the most efficient data placement algorithm, and placement is still performed manually by a database administrator with periodic reorganization.

Another database, “Distributed RSDB”, based on DDBSs architecture is developed to improve on the performance of the original RSDB. The performance of CDBSs is compared to that of DDBSs, and appropriate data distribution approaches for RSDB, to implement DRSDb with higher performance, are found.

The rest of this paper is organized as follows. In section 2, we briefly describe RSDB. In section 3, we give the design of experiments. Section 4 shows the performance evaluation of the experiments designed in section 3. Finally, we draw a summary in section 5.

2. REPEAT SEQUENCES DATABASE (RSDB)

2.1 System Architecture

Table 1 lists the number of repetitive elements for each of the 24 organisms in RSDB. Fig. 1 shows the system architecture of RSDB.

Table 1. The amount of data in RSDB.

Identical Repeats	71 millions
Copies of Repeats	218 millions
(Repeat) Sequences	59 millions

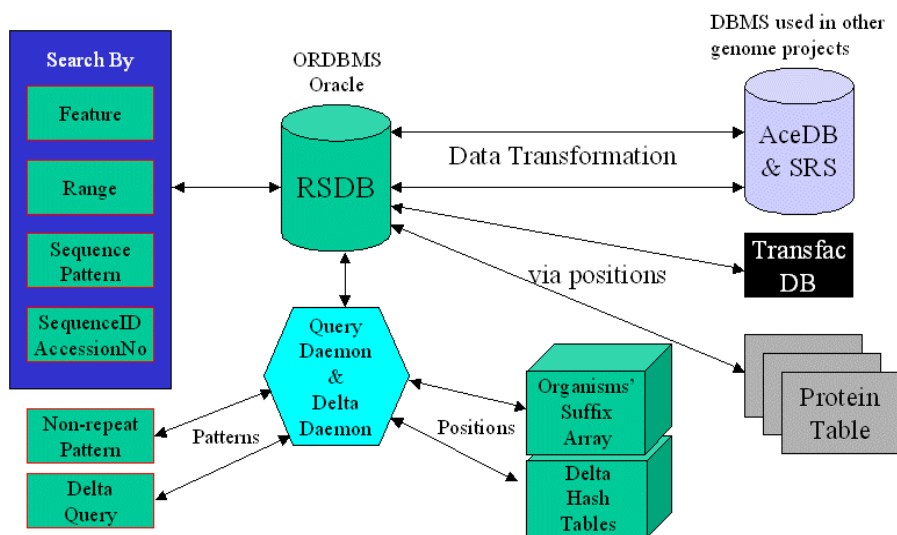


Fig. 1. System architecture of RSDB.

2.2 Queries on RSDB

RSDB currently supports four types of queries:

- Search By Feature
- Search By Range
- Search By Pattern
- Search By Accession Number or Sequence ID

3. EXPERIMENTAL DESIGN

3.1 Experimental Environment

The experimental environment for DRSDb includes three SMP PCs. The bandwidth of the communication network is 100MB/s. The operating system is RedHat Linux 6.2 and the version of Oracle DBMS is 8.1.7. Fig. 2 shows the system architecture of DRSDb.

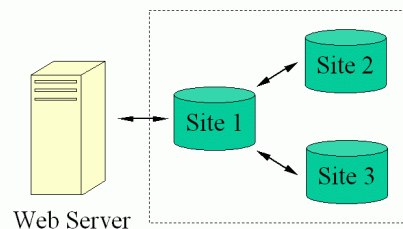


Fig. 2. System architecture of DRSDb.

3.2 Data Distribution

We understand the data distribution is an important issue of DDBSs. The decision of how to distribute data is important in obtaining good load balancing. Herein, according to the statistics gathered in RSDB's data warehouse, we design two data distribution approaches used to distribute repeat data.

(A) Distribution by interleaving each record of a column value (Type 1)

Many records of a given column value may exist. This method distributes each record of this value to each site. For example, many repeats exist in CE's chromosome X. The first record of a chromosome value of "X" is stored at site 1; the second at site 2, and so on. Consider the column "copy" in the "REPEAT" table as another example. Many CEs repeat twice. The first record of the "copy" value of two is stored at site 1, the second record at site 2, and so on. Fig. 3 illustrates how data is distributed according to this method.

(B) Distribution by statistics in the RSDB's data warehouse (Type 2)

According to the statistics in the RSDB's data warehouse, the identical sequence to which repeats usually refer is 13 or 14 base pairs long. These data may thus be stored at a single site, and repeats of fewer than 13 bp or more than

14 bp in length are stored at the two other sites. Besides, repeats with two copies represent half of all the identical repeats. Thus, all these data may be best stored at one site. Fig. 4 shows how data is distributed according to this type.

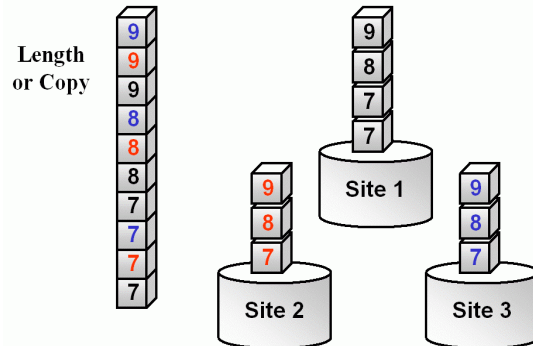


Fig. 3. Distribution by interleaving with each record of a column value.

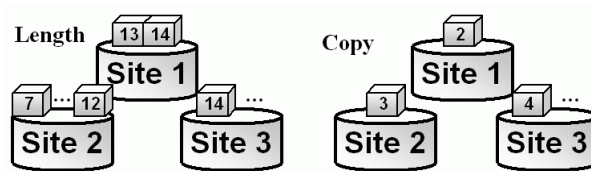


Fig. 4. Distribution by statistics in RSDB's data warehouse.

3.3 Experimental Queries

Herein, many “Search By Feature”, “Search By Range”, and “Search By Pattern” queries are considered because sometimes much time is required to execute such of queries. Besides, CE's repeat data are used as test data because CE includes the most data. A single repeat has several features. Users may specify or not specify these features to select desired repeats. Herein, some representative queries are designed, and each query specifies different features to evaluate the performance of this query under various conditions. Table 2 shows queries used to evaluate “Search By Feature” in CDBSs and DDBSs.

Table 2. Queries used to evaluate Search By Feature.

Query	Search for CE's repeats with the following features
Q1	Palindromic
Q2	Length = 14, AT = 50%, Direct
Q3	Copy = 2, AT = 100%, Bi-directional
Q4	Length = 20, Copy = 7, AT = 75%, Bi-directional
Q5	Length = 11 ~ 30, Copy = 2 ~ 5, AT = 25%, Intrachromosomal
Q6	Length = 31 ~ 100, Copy = 6 ~ 10, AT = 70% ~ 100%, Direct, Interchromosomal
Q7	Copy = 2 ~ 8, AT = 50%, Palindromic

4. PERFORMANCE EVALUATION

Fig. 5 shows that approaches to data distribution affect the performance of the queries. This Figure shows that queries are more poorly responded to if data is distributed by the Type 2 approach. Fig. 6 shows the same results. According to the experimental results for this type of query, the query performs very well if it includes a column in which data is distributed using the Type 1 approach to distributing data. However, some queries, if the column, "LENGTH", is chosen to distribute data and the query is executed without the feature, "LENGTH".

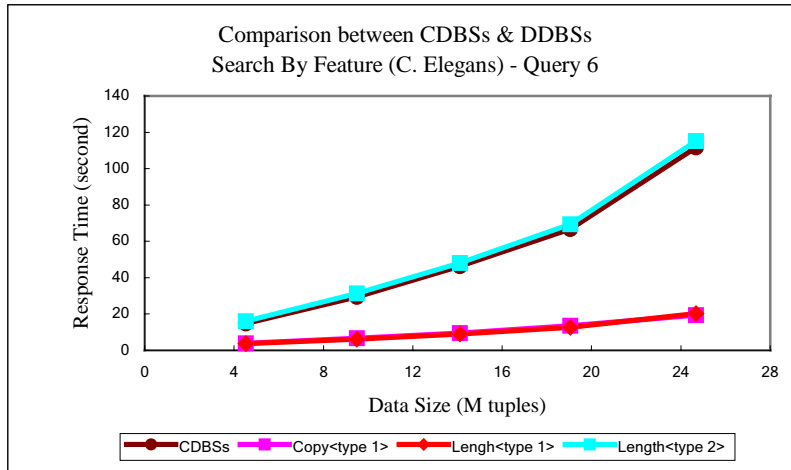


Fig. 5. Comparison between CDBSs and DDBSs of Query 6.

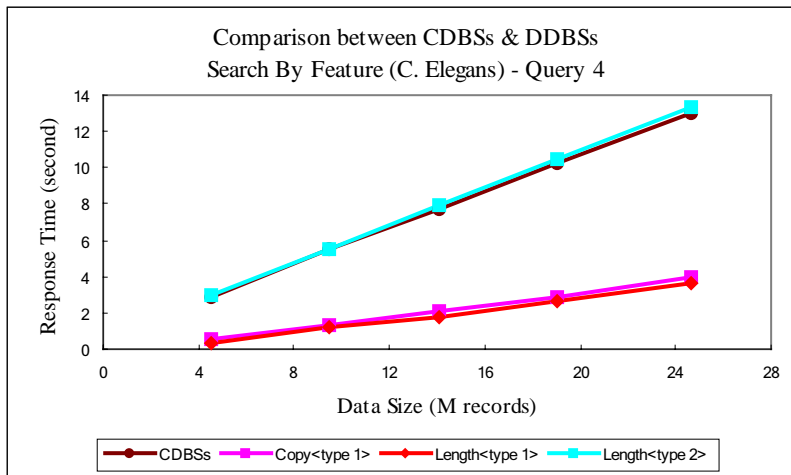


Fig. 6. Comparison between CDBSs and DDBSs of Query 4.

The overall system performance in response to the search queries was measured. The experimental results are considered below. Fig. 7 shows a particular experimental result. The data distribution is not uniform and measuring performance of CDBSs and DDBSs is therefore difficult. However, the workload of CDBSs exceeds that for DDBSs for the same concurrent requests.

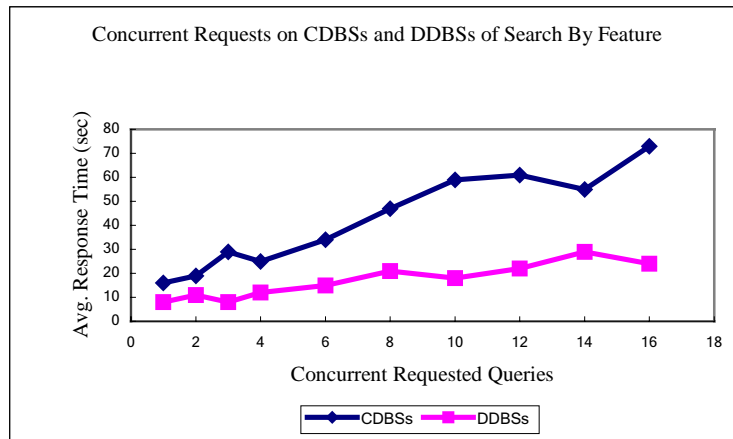


Fig. 7. Concurrent requests of Search By Feature.

5. SUMMARY

Approaches to data distribution are important to distributed database systems. A poor data distribution strategy can lead to a non-uniform distribution of the load, as proven experimentally using the “Distribution by statistics in the RSDB’s data warehouse” approach. Choosing an appropriate approach to distribution data for Search By Feature is important; determining the best approach is not easy. Therefore, user query logs can be analyzed to determine the approach to distributing data of greatest benefit.

ACKNOWLEDGEMENTS

The authors would like to thank Li-Wei Liu for his providing the infrastructure of RSDB and Hu-Chia Chang for his implementation of experiments of distributed RSDB. The authors also would like to thank the National Science Council of the Republic of China and Asia Bioinnovation Corporation for financially supporting this research. The authors are grateful to Prof. Cheng-Yan Kao for his contribution to this research.

REFERENCES

1. R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 2nd ed., Addison-Wesley Publishing Company, Menlo Park, 1994.

2. J. T. Horng, J. H. Lin, and C. Y. Kao, "RSDB – A database of repetitive elements in complete genomes," in *Proceedings of the Atlantic Symposium on Computational Biology and Genome Information Systems & Technology*, 2000, pp. 220-223.
3. M. Mehta and D. J. DeWitt, "Data placement in shared-nothing parallel database systems," *The VLDB Journal*, Vol. 6, 1997, pp. 53-72.
4. R. Mulkamala, "Measuring the effect of data distribution models on performance evaluation of distributed database systems," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, 1989, pp. 494-507.
5. M. Nicola and M. Jarke, "Performance modeling of distributed and replicated databases," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, 2000, pp. 645-672.
6. M. T. Özsu and P. Valduriez, "Distributed and parallel database systems," *ACM Computing Surveys*, Vol. 28, 1996, pp. 125-128.
7. M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems*, 2nd ed., Prentice-Hall, 1999
8. A. M. Tamhankar and S. Ram, "Database fragmentation and allocation: An integrated methodology and case study," *IEEE Transactions on System, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 28, 1998, pp. 288-305.
9. W. H. Li, Z. Gu, H. Wang, and A. Nekrutenko, "Evolutionary analyses of the human genome," *Nature*, Vol. 409, 2001, pp. 847-849.

Ming-Hui Jin (金明輝) is currently a PhD student in the Department of Computer Science and Information Engineering at National Central University.

Jorng-Tzong Horng (洪炯宗) was born in Nantou, Taiwan, on April 10, 1960. He received the Ph.D. degree in Computer Science and Information Engineering from National Taiwan University, Taipei, in April 1993.

He is currently an Associate Professor, Department of Computer Science and Information Engineering, National Central University, Chungli, Taiwan. His current research interests include database systems, data mining, genetic algorithms, and bioinformatics.