High Performance Database Processing

David Taniar
Clayton School of Information Technology
Monash University
Clayton, Victoria, Australia
e-mail: david.taniar@monash.edu

Abstract— **The sizes of databases have seen exponential growth in the past, and such growth is expected to accelerate in the future, with the steady drop in storage cost accompanied by a rapid increase in storage capacity. Many years ago, a terabyte database was considered to be large, but nowadays they are sometimes regarded as small, and the daily volumes of data being added to some databases are measured in terabytes. In the future, petabyte and exabyte databases will be common.**

With such volumes of data, it is evident that the sequential processing paradigm will be unable to cope; for example, even assuming a data rate of 1 terabyte per second, reading through a petabyte database will take over 10 days. To effectively manage such volumes of data, it is necessary to allocate multiple resources to it, very often massively so. The processing of databases of such astronomical proportions requires an understanding of how high-performance systems and parallelism work. Besides the massive volume of data in the database to be processed, some data has been distributed across the globe in a Grid environment. These massive data centres are also a part of the emergence of Cloud computing, where data access has shifted from local machines to powerful servers hosting web applications and services, making data access across the Internet using standard web browsers pervasive. This adds another dimension to such systems.

This talk, based on our recent published book [1], discusses fundamental understanding of parallelism in data-intensive applications, and demonstrates how to develop faster capabilities to support them. This includes the importance of indexing in parallel systems [2-4], specialized algorithms to support various query processing [5-9], as well as object-oriented scheme [10-12].

Parallelism in databases has been around since the early 1980s, when many researchers in this area aspired to build large special-purpose database machines – databases employing dedicated specialized parallel hardware. Some projects were born, including Bubba, Gamma, etc. These came and went. However, commercial DBMS vendors quickly realized the importance of supporting high performance for large databases, and many of them have incorporated parallelism and grid features into their products. Their commitment to high-performance systems and parallelism, as well as grid configurations, shows the importance and inevitability of parallelism.

There have been an increase number of researches in high performance parallel database processing in the last five years (2008-12). Data partitioning is still the fundamental issue in high performance database processing [13, 14]. The data itself is getting more complex, including XML-based data [15, 16], bio-informatics data [17, 18], and data streams [19, 20]. These new data types require new approaches to parallel processing. In addition, database transactions [21, 22] are still a major focus in many high performance database systems, such as grid transactions. We also see an increasing growth of new application domains, broadly categorized as data-intensive applications, including data warehousing and online analytic processing (OLAP) [23-25].

Therefore, it is critical to understand the underlying principle of data parallelism, before specialized and new application domains can be properly addressed.

Keywords-parallel databases; parallel query processing; grid databases; cloud databases; data partitioning.

REFERENCES

[1] D. Taniar, C. H. C. Leung, W. Rahayu, and S. Goel, High Performance Parallel Database Processing and Grid Databases, John Wiley & Sons 2008.

[2] D. Taniar, and W. Rahayu, "A Taxonomy of Indexing Schemes for Parallel Database Systems", Distributed and Parallel Databases, 12(1): 73-106, 2002.

[3] D. Taniar, and W. Rahayu, "Global parallel index for multi-processors database systems", Information Sciences, 165(1-2): 103-127, 2004.

[4] D. Taniar, and W. Rahayu, "Parallel database sorting", Information Sciences, 146(1-4): 171-219, 2002.

[5] D. Taniar, W. Rahayu, and R. B-N, Tan, "Parallel algorithms for selection query processing involving index in parallel database systems", Comput. Syst. Sci. Eng. 19(2):, 2004.

[6] D. Taniar, R. B-N. Tan, C. H. C. Leung, and K. H. Liu, "Performance analysis of "Groupby-After-Join" query processing in parallel database systems", Information Sciences, 168(1-4): 25-50, 2004.

[7] D. Taniar, and W. Rahayu, "Parallel group-by query processing in a cluster architecture", Comput. Syst. Sci. Eng. 17(1): 23-39, 2002.

[8] D. Taniar, and W. Rahayu, "Parallel sort-merge object-oriented collection join algorithms", Comput. Syst. Sci. Eng. 17(3): 145-158, 2002.

[9] D. Taniar, and W. Rahayu, "Parallel sort-hash object-oriented collection join algorithms for shared-memory machines", Parallel Algorithms Appl. 17(2): 85-126, 2002.

[10] D. Taniar, C. H. C. Leung, "The impact of load balancing to object-oriented query execution scheduling in parallel machine environment", Information Sciences, 157: 33-71, 2003.

[11] D. Taniar, and W. Rahayu, "Performance Analysis of Parallelization Models for Path Expression Queries", Information Sciences, 117(1-2): 107-142, 1999.

1550-445X/12 $26.00 © 2012 IEEE
DOI 10.1109/AINA.2012.140

IEEE computer society

[12] D. Taniar, and C. H. C. Leung, “Query execution scheduling in parallel object-oriented databases”, Information & Software Technology, 41(3): 163-178, 1999.

[13] R. V. Nehme, and N. Bruno, “Automated partitioning design in parallel database systems”, Proceedings of the ACM SIGMOD Conference, pp: 1137-1148, 2011.

[14] L. I. Rusu, W. Rahayu, and D. Taniar, “Partitioning methods for multi-version XML data warehouses”, Distributed and Parallel Databases, 25(1-2): 47-69, 2009.

[15] I. Machdi, T. Amagasa, and H. Kitagawa, “Parallel holistic twig joins on a multi-core system”, International Journal of Web Information Systems, 6(2): 149-177, 2010.

[16] I. Machdi, T. Amagasa, and H. Kitagawa, “XML data partitioning schemes for parallel holistic twig joins”, International Journal of Web Information Systems, 5(2): 151-194, 2009.

[17] T. Rognes, “Faster Smith-Waterman database searches with inter-sequence SIMD parallelisation”, BMC Bioinformatics, 12: 221, 2011.

[18] H. P. Lee, T-F. Sheu, and C. Y. Tang, “A parallel and incremental algorithm for efficient unique signature discovery on DNA databases”, BMC Bioinformatics, 11: 132, 2010.

[19] I. Kamel, Z. Al Aghbari, and T. Awad, “MG-join: detecting phenomena and their correlation in high dimensional data streams”, Distributed and Parallel Databases, 28(1): 67-92, 2010.

[20] A-A Safaeei, and M. S. Haghjoo, “Parallel processing of continuous queries over data streams”, Distributed and Parallel Databases, 28(2-3): 93-118, 2010.

[21] D. Taniar, and S. Goel, “Concurrency control issues in Grid databases”, Future Generation Computer Systems, 23(1): 154-162, 2007.

[22] C. B. Colohan, A. Ailamaki, J. G. Steffan, and T. C. Mowry, “Incrementally parallelizing database transactions with thread-level speculation”, ACM Transactions on Computer Systems, 26(1):, 2008.

[23] A. Lima, C. Furtado, P. Valduriez, and M. Mattoso, “Parallel OLAP query processing in database clusters with data replication”, Distributed and Parallel Databases, 25(1-2): 97-123, 2009.

[24] P. Furtado, “A Survey of Parallel and Distributed Data Warehouses”, International Journal of Data Warehousing and Mining, 5(2): 57-77, 2009.

[25] Y. Chen, F. Dehne, T. Eavis, and A. Rau-Chaplin, “PnP: sequential, external memory, and parallel iceberg cube computation”, Distributed and Parallel Databases, 23(2): 99-126, 2008.