: Is parallel database technology needed for data-driven DSS?

by Dan Power

Editor, DSSResources.COM

Few managers have heard of parallel computer processing and parallel databases. Nonetheless the desire of managers for more and better historical data is increasing the need for such capabilities. The call for papers for the ACM Eighth International Workshop on Data Warehousing and OLAP states that "Data Warehouse (DW) and Online Analytical Processing (OLAP) technologies are the core of current Decision Support Systems. ... Research in data warehousing and OLAP has produced important technologies for the design, management and use of information systems for decision support."

Norman and Thanisch with Bloor Research Group argue the future of commercial databases "is bound up with the ability of databases to exploit hardware platforms that provide multiple CPUs." They also note "There is a tremendous amount of confusion in the market over parallel database technology, among both customers and the vendors. The root of this is a general lack of understanding of the technical issues on both sides. Although the workings of a parallel database are more complex than an ordinary database, understanding it requires more of a change in mind set than an astronomically high IQ. Unfortunately, few decision makers have much understanding of parallel database, and consequently, it is open season for database and hardware marketing people to confuse the market with technical mumbo-jumbo, that they don't fully understand themselves." Parallel database technology makes it possible to process very large databases for data-driven decision support.

What is the history of parallel database technology?

According to James Gray, "During the 1970s there was great enthusiasm for database machines -- special-purpose computers that would be much faster than general-purpose systems running conventional databases. The problem was that general purpose systems were improving at 50% per year, so it was difficult for customized systems to compete with them. By 1980, most researchers recognized the futility of special-purpose approaches, and the database machine community switched to research on using arrays of general purpose processors and disks to process data in parallel. The University of Wisconsin

was home to the major proponents of this idea in the US. Funded by the government and industry, they built a parallel database machine called Gamma. That system produced ideas and a generation of students who went on to staff all the database vendors. Today the parallel systems from IBM, Tandem, Oracle, Informix, Sybase, and AT&T all have a direct lineage from the Wisconsin research on parallel database systems. The use of parallel database systems for data mining is the fastest-growing component of the database server industry."

Also, Gray notes "projects at UCLA gave rise to Teradata." Today NCR Teradata (www.teradata.com) is the premier vendor of parallel database software. Most of my experience with parallel database software occurred at NCR Teradata Partners conferences. The current NCR Massively Parallel Processing (MPP) platform is designed to run the Teradata Database software efficiently for data warehousing and decision support.

What is parallel database processing?

This question is challenging to answer for a broad audience. I'll try to limit the buzz words and technical jargon. I'll also emphasize 2 nontechnical examples.

Let's start with a simple generalization. Parallel processing divides a computing task into smaller tasks that can be processed independently. Hence, the larger task is completed more quickly. Parallel relational database systems store data that is spread across many storage disks and accessed by many processing units. Whatis.com states massively parallel processing "is the coordinated processing of a program by multiple processors that work on different parts of the program, with each processor using its own operating system and memory."

A Teradata Warehouse technical overview includes the following example of parallel processing: "Imagine that you were handed a shuffled stack of playing cards and were not allowed to scan the cards beforehand. Then you were asked a simple question, 'How many aces are in the stack?' The only way to get the answer would be to scan the entire deck of cards. Now imagine that the same cards were

distributed among four people, each receiving one-fourth of the cards. The time required to answer this same query is now reduced by four times. Each person would simply have to scan their cards, and the four totals would be aggregated for the correct answer. In this simple example, we can refer to these four people as parallelized units of work. As you can see, more available parallelized units of work will result in faster query processing. The larger the data volume and the more complex the queries, the bigger the payoff from using parallel processing. It’s also important to note that the most efficient way to distribute the playing cards (or data) is to distribute them evenly among the four people (or parallelized units of work)."

Mahapatra and Mishra provide another example: "Your local grocery store provides a good, real-life analogy to parallel processing. Your grocer must collect money from customers for the groceries they purchase. He could install just one checkout stand, with one cash register, and force everyone to go through the same line. However, the line would move slowly, people would get fidgety, and some would go elsewhere to shop. To speed up the process, your grocer doubtless uses several checkout stands, each with a cash register of its own. This is parallel processing at work. Instead of checking out one customer at a time, your grocer can now handle several at a time."

So imagine many relational databases linked together where each database has the same data organization and individual questions are simultaneously asked of all the databases and individual answers are then summarized.

Is parallel database technology critical to the future success of data-driven DSS?

YES. According to Todd Walter of NCR Teradata, three issues are driving the increasing use of parallel processing in database environments: the need for increased speed or performance for large databases, the need for scalability and the need for high availability. Finally, Mahapatra and Mishra (2000) conclude "Intra-query parallelism is very beneficial in decision support system (DSS) applications, which often have complex, long-running queries. As DSS have become more widely used, database vendors have been increasing their support for intra-query parallelism."

In general, parallel processing is necessary to provide timely results from complex, decision support database queries needed by managers in data intensive organizations.

References

Abdelguerfi, M. and K. Wong, Parallel Database Techniques, Wiley-IEEE Computer Society Press, July 1998.

Barney, B., "Introduction to Parallel Computing," Livermore Computing,
URL http://www.llnl.gov/computing/tutorials/parallel_comp/

DeWitt, D. J. and J. Gray, "Parallel Database Systems: The Future of High Performance Database Processing", Communications of the ACM, Vol. 36, No. 6, June 1992, http://www.cs.wisc.edu/~dewitt/includes/paralleldb/cacm.pdf

Gray, J. N., "Database Systems: A Textbook Case of Research Paying Off," 1997,
URL http://www.cs.washington.edu/homes/lazowska/cra/database.html

Mahapatra, T. and S. Mishra, "Oracle Parallel Processing," O'Reilly, 2000,
URL http://www.oreilly.com/catalog/oraclepp/chapter/ch01.html

Norman, M. G. and P. Thanisch, "Parallel Database Technology: An Evaluation
and Comparison of Scalable Systems," Bloor Research Group, URL http://www.dpu.se/blopdt_e.html

Teradata Warehouse Technical Overview: Teradata Pioneered Data Warehousing, EB-3025,
September 2005, URL
http://www.teradata.com/t/pdf.aspx?a=83673&b=84876 .

Walter, T., "Scalability, Performance, Availability," Teradata Magazine Online,
URL http://www.teradata.com/t/go.aspx/index.html?id=115886

Citation: Power, D., "Is parallel database technology needed for data-driven DSS?" DSS News, Vol. 7, No. 8, April 9, 2006.

Author: Daniel Power
Last update: 2007-03-02 12:05