

Computer Technology Review®

THE STANDARD OF EXCELLENCE FOR STORAGE TECHNOLOGY INFORMATION

AMPP: Combining SMP and MPP to Speed Database Queries

By Foster Hinshaw

In the current age of data warehousing, the timely processing and retrieval of vast amounts of data is vital to the decision-making process. However, just as important as timeliness is the depth of data analysis possible.

According to Greg's Law, it is estimated that on average a company's data doubles every nine months. With the growing size of the average data warehouse, the task of data analysis has become increasingly difficult; already, terabyte-sized data warehouses are fairly common.

The Challenge

Most organizations today experience exponential growth in the amount of data they want to store. That's not just because they're growing their base of customers. It's because they're storing much more information about almost everything:

- Lots of detail about purchases and transactions
- Click stream data encapsulating the interests displayed by visitors to their website
- Supply chain management data
- Competitive intelligence
- Employee knowledge and expertise
- Demographic information about customers

If it can be accessed effectively, this information becomes valuable intelligence, capable of improving business operations across the board, from marketing to merchandising. If the data can't be stored or retrieved efficiently, the

organization risks losing critical ground to those competitors who have better intelligence.

In addition to experiencing exponential growth in data storage, organizations are also experiencing growth in user demand for that data. Through their Intranet, Extranet, and Internet sites, they are providing concurrent access to their data to more people for more hours in the day than ever before. For example:

- Over an Intranet, marketers access demographic information to perform finer segmentation and to achieve better targeting for their new marketing strategies and tactics.
- Over an Extranet, suppliers analyze purchases by location and time to manage inventories and capitalize on transient sales opportunities at the local geographical level.
- Over the Internet, customers are invited to view the hottest products and track order status.

Lots of people now have access to lots of data. And all of these people want answers quickly. If decisions are made in real-time, then the information required for those decisions must be available in real-time. Waiting an hour or more for a query to return results is unacceptable; users want answers in seconds.

Leading software vendors have made great strides in accelerating database queries, but these applications cannot address the inherent bottlenecks of existing data warehousing architectures. As demand for access and analysis continues to grow, users must look

to new database architectures to provide improved performance.

Traditional Database Architectures

Many databases currently use either symmetric multiprocessing (SMP) or massively parallel processing (MPP) to handle queries. SMP uses multiple CPUs with a common memory. While additional CPUs can be added to improve performance, the SMP architecture often creates system bottlenecks. With MPP, queries are sent to multiple processors, each with their own storage device. This prevents the bottlenecks experienced with SMP, but requires additional programming to enable all of the segments to communicate with each other. The higher cost and administrative hassle of the no-shared-resources approach makes pure MPP systems infrequent in practice. Typical MPP systems are implemented virtually in clusters of SMPs. This preserves some of the performance and scalability advantages of MPP while reducing cost and communication latencies. However, sharing resources to any degree imposes coordination overhead that ultimately limits performance and scalability.

The most common argument against the effectiveness of MPP approaches is that uneven data distributions mitigate the inherent advantages of parallelism. While this is true in theory, most MPP-based systems use a hash distribution scheme that spreads data evenly across all nodes, for most queries. In any event, a similar concern applies to architectures like SMP and clusters that share

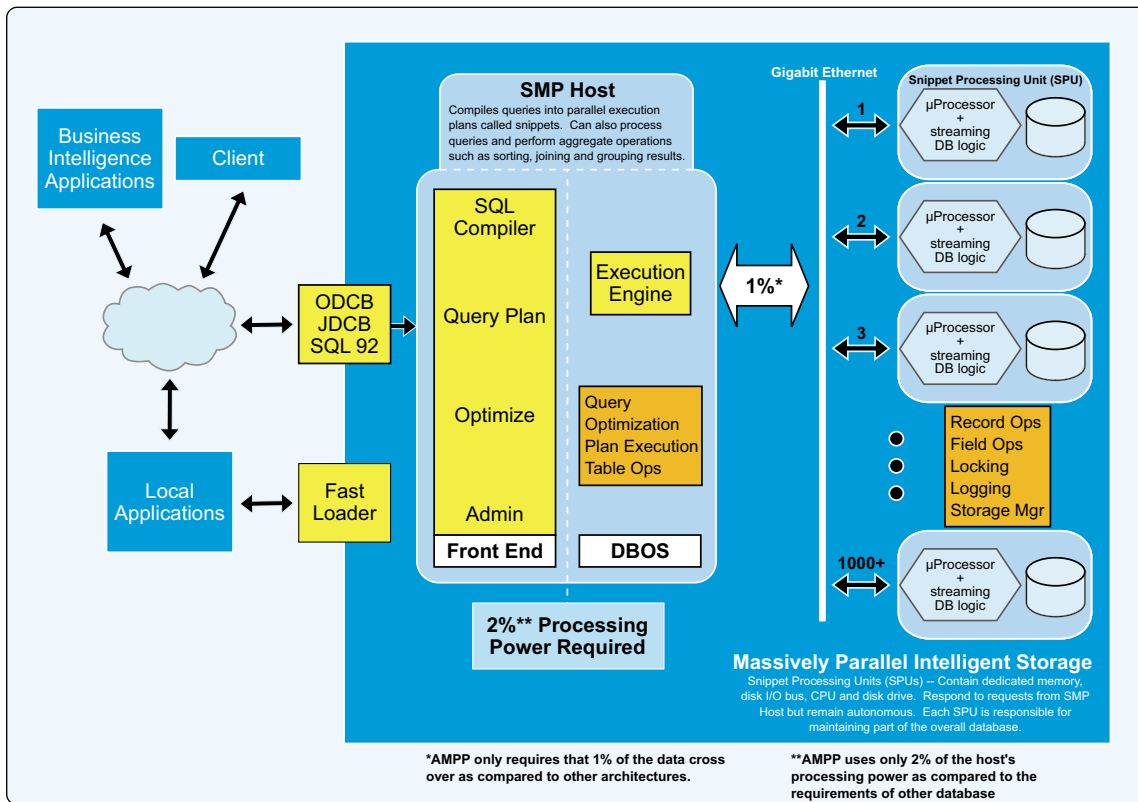
disks. In these cases, commonly accessed data may reside on selected sections of the shared disk. As multiple processors try to update the same disk blocks, they experience delays as they coordinate their locks. Fundamentally, neither SMP nor MPP are successful on their own, due to these constraints.

The AMPP Approach

A more effective approach can be found by combining the strengths of SMP and MPP to create a new architecture called Asymmetric Massively Parallel Processing (AMPP). AMPP architecture is built to harness the processing power of SMP and combine it with the scalability of MPP. Simply put, AMPP applies SMP and MPP approaches in a two-tier system to the areas where they can provide the largest benefits in performance and scalability.

On the first tier, an SMP-based host compiles queries into parallel execution plans and provides the right amount of processing power to sort and aggregate large sets of query results. On the second tier, data is distributed across many nodes to minimize I/O latency and increase scalability.

A host divides a query up into a sequence of smaller requests called snippets that can be executed in parallel, and it distributes snippets to the second tier for execution. In addition to coordinating second-tier components, the host is also available for query processing on its own. It is typically called upon to perform aggregate operations like sorting, joining



and grouping intermediate results. The host makes good advantage of the SMP's shared memory model and intrinsic load balancing.

AMPP combines the best elements of SMP and MPP into a new architecture to allow a query to be processed in the most optimized way possible. It is architected to remove all the bottlenecks to data flow so that the only remaining limit is the disk speed—a “data flow” architecture where data moves at “streaming” speeds. Through standard interfaces, it is fully compatible with existing BI applications, tools and data. And it is extremely simple to use.

With AMPP, the traditional I/O bus bandwidth problem of SMP is mitigated by delegating most I/O to the second, massively parallel tier. The traditional difficulty of managing large numbers of independent components is lessened by concentrating system management and reliability functions in the SMP host(s).

The second tier of AMPP architecture consists of hundreds to thousands of query snippet processors, called snippet processing units (SPUs). Each SPU is solely responsible for managing

a slice of the overall database. To this end, it has dedicated memory, disk, I/O bus, general purpose CPU and a programmable disk controller. The massively parallel, shared-nothing set of SPUs provides the performance advantage of MPP.

The SPUs in the second tier are not directly accessible to the end user or application. They respond to requests from the host elements in the first tier. The requests sent from the host elements to the SPUs typically require significant processing on the part of the SPUs, so that the higher interprocess communication typical of MPP is dispersed over more processing. This mitigates one of the traditional disadvantages of MPP.

While the SPUs in the second tier respond to requests from the hosts in the first tier, they are highly autonomous, performing their own scheduling, storage management, transaction management, concurrency control and replication. This significant degree of autonomy relieves the host from the responsibility of coordinating these functions and synchronizing this coordination with other hosts in the first tier,

allowing full scalability in both tiers of the architecture. Using this massively parallel approach, AMPP requires only two percent of the host's processing power, as compared to other database architectures.

Simply put, AMPP architecture applies SMP and MPP approaches to the areas where they can provide the largest benefits in performance and scalability, enabling companies to quickly harness data for real-time decisions.

Conclusion

The size of the average data warehouse is increasing and showing no signs of slowing down—multi-terabyte sized data warehouses are becoming more and more common—and with this increased store of knowledge comes an increased demand to generate intelligence from data. Businesses should not need to discard customer data from two months ago because their database slows to a crawl when the data is kept. As the amount of data continues to grow, the data warehouse must be fast enough, and flexible enough, to grow as well.

While almost all large com-

panies have a growing amount of data, those organizations experiencing a true data explosion include financial services, retail, telecommunications, bioinformatics and government. Companies in these sectors are struggling to access data quickly in order to make more intelligent business decisions. An AMPP architecture will be very valuable as companies in these and other areas explore data for business purposes and begin building data warehouses for the first time.

The benefits of this type of architecture are impressive. Users who are forced to wait hours and days for analysis on traditional systems benefit from tremendous speeds with the new architecture, which can generate results in minutes and seconds rather than hours or days. That speed translates into complete freedom to conduct ad hoc and complex analyses without restraints.

In the next chapter of data warehousing, AMPP architecture will play an important role as companies continue to struggle to make sense of ever-growing oceans of data. **CTR**

Foster Hinshaw is co-founder and CTO of Netezza Corp. (Framingham, Mass.)



Reprinted by permission from the publisher of **Computer Technology Review**®, Volume 23, Issue 3. For FREE subscription information, please call 310/276-9500 or reply via the World Wide Web at <http://www.wvpi.com>.

©2003 WestWorld Productions, Inc.