

# The Race for Data Access

Parallel processing is neither cheap nor simple. How do you know when your data access requirements justify it?

---

By Bill Roberts

---

**H**as data access got your enterprise stumped? Consider the New England farmer who faced a similar problem clearing his field: a stump he couldn't budge. He retired his mule for the day, borrowed two teams of draft horses and harnessed the power of four to solve his problem.

The late Grace Hopper (the U.S. Navy's first female admiral, who was instrumental in bringing advanced computing to the armed forces) often used that anecdote to explain *parallel processing*, the harnessing of two or more CPUs to return a query or solve a calculation faster than one processor can. Once the sole domain of expensive scientific computers, parallel processing has plunged in price, and in response commercial applications have proliferated. It now replaces many mainframe systems for online transaction processing and online analytical processing, including decision support.

A parallel computer paired with a parallel database engine increasingly is touted by vendors, analysts and users as the preferred solution for the rapid access, flexible queries, modeling and analysis needed in decision support systems. It finds many ready uses. A stock trader who can analyze derivatives faster than competitors makes more money. Large retailers can study the buying trends of groups or individuals. Banks can detect credit card fraud. Airlines can analyze travel patterns. Pharmaceutical companies can tailor sales pitches to individual doctors based on their prescription writing patterns. Because decision support applications like these are mission-critical, parallel processing earns consideration.

Before getting into the technical details, an organization should identify its database and data query needs. Once IS staff and business managers understand the data they have or will have, as well as their business processes, what access they need and how much they are willing and able to spend, they can decide whether parallel processing may be the answer. If that decision is affirmative, they are ready to choose a database engine and a hardware architecture.

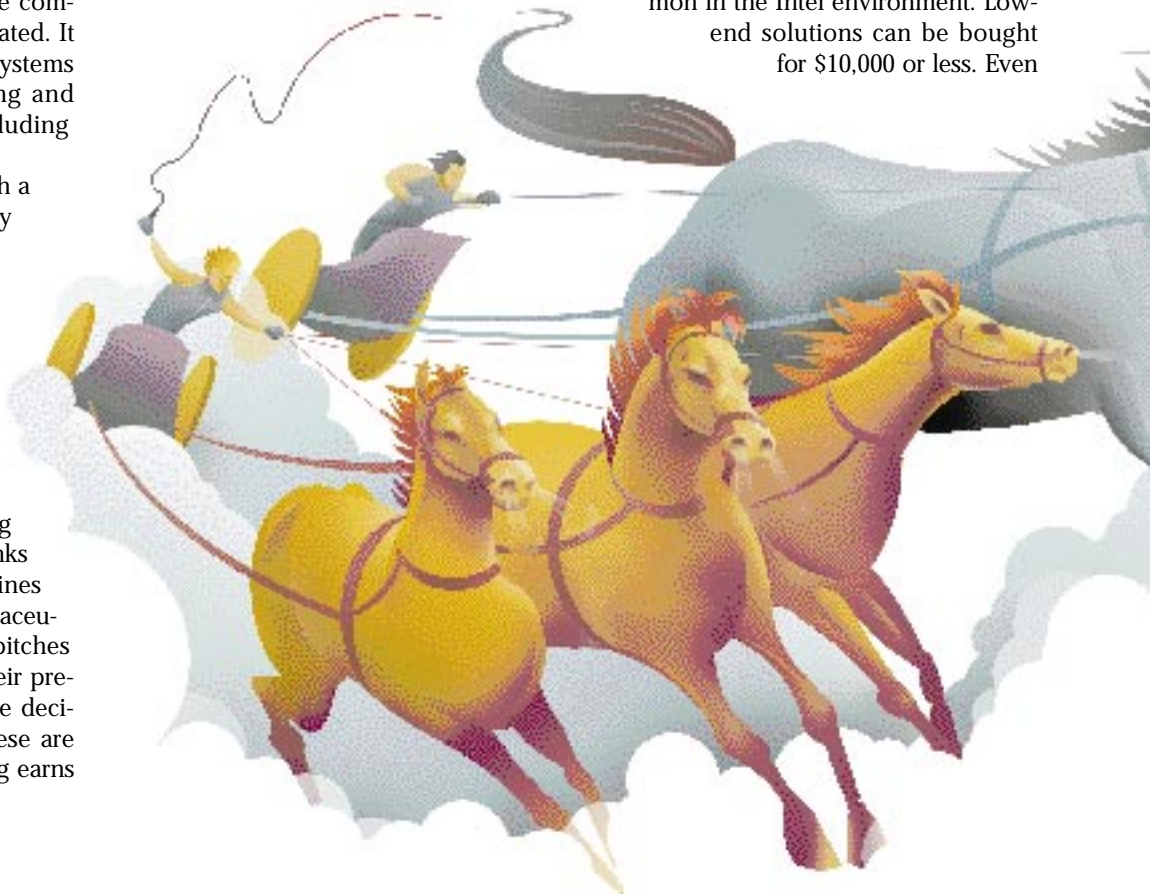
## A Commercial Challenge

Parallel processing is based on a simple idea. Software splits large problems into smaller subtasks and distributes them to different processors, which simultaneously handle the subtasks and so reach a solution faster than a uniprocessor system does (see diagram on opposite page).

Various trends are driving commercial parallelism. Data in legacy systems is not easily accessible. Multiple-processor architectures have become increasingly fast, scalable and easier to administer. Tools have improved to install and administer the multithreaded operating systems and database engines for these platforms. Application development is less onerous now.

More than any of these reasons, the driving force is that databases are growing. Databases of 2GB to 3GB, once considered large, are dwarfed by those of 10GB and more. Some enterprise-wide databases exceed 50GB.

From the hardware perspective, parallel processing is almost a moot question. Database servers increasingly have parallel capability. Many servers that run the Unix operating system house two, four or more CPUs. Two CPUs are common in the Intel environment. Low-end solutions can be bought for \$10,000 or less. Even



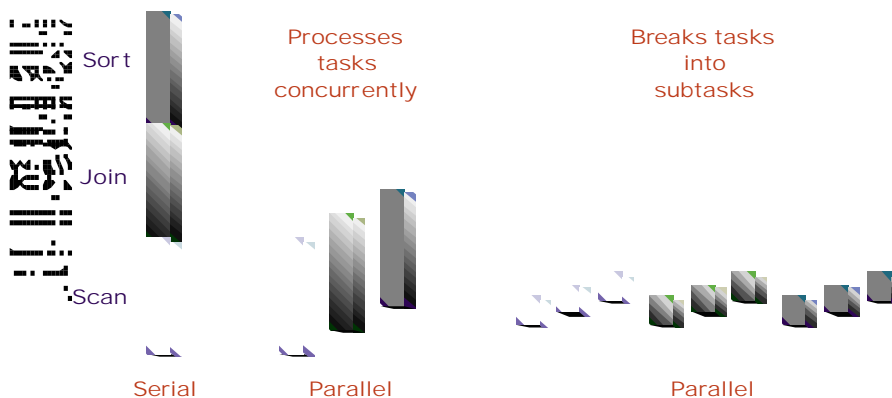
so, just because you have the capacity for more processors doesn't mean you must buy and use them. If you're not ready for parallel processing today, you could lay out a migration path to your next database server.

In tandem with data growth, more users demand access to databases across the company, and their queries are getting more complex, more ad hoc and more sophisticated. The benefit of parallel data querying depends on the number of queries, their complexity, the complexity of the data and the number of users accessing the system. In a typical executive information system, the usual hit rate is low, because only a few people use it, but the queries might be especially complex and the data could be large. Once decision support spreads to the masses, the database will sustain an enormous number of hits. When users

find out what kind of questions they can ask, they start doing more complicated queries. One vendor calls it the "potato chip syndrome"—the more you eat, the more you want.

Parallel processing can be less efficient for applications that let users ask a series of questions of small subsets of data, drilling deeper with each subsequent query.

In a parallel system, each query would scan far more data than needed. For these applications, data indexing that leads more directly to the categories most often used could achieve quicker



The parallel model divides database processing in two complementary ways.

Source: Informix

response times. But for dealing with large amounts of data and complex analysis, parallel processing can be the answer. If your query requires multiple operations and you want to improve response time, you probably need parallel processing.

It's not an easy path to choose. It can be costly. Applications can require recoding. Moving data to a parallel architecture is work, although tools exist to help. You must train people for the environment. A key issue is figuring how to spread your data properly, which requires understanding the nature of the data and how it will be used.

Given the increasing reliance of many organizations on data manipulation, a boom in parallel processing for data querying probably is on the way. "Any scalable database application will demand parallel processing," says John Oltsik, an analyst at Forrester Research in Cambridge, MA. "Applications tend to grow, and data tends to grow a lot faster than anticipated. You need to plan for the future."

## Comparing Platforms

A decision support database is oriented toward large-scale sorting and searching in large databases, usually those comprised of historical data. I/O is large and sequential. Queries typically involve full-table scans requiring minimal indexing. In this environment, speed is everything.

All the major vendors of open systems databases strive to enhance the performance of their products as new processors and higher clock speeds appear on the market. Oracle of Redwood Shores, CA, was first to market with a parallel database—albeit as an add-on—and the first to support 300MHz and 350MHz processors, though Sybase of Emeryville, CA, wasn't far behind. Informix Software of Menlo Park, CA, has a uniquely architected parallel processing database. Red Brick Systems of Los Gatos, CA, focuses on the data warehouse.

According to Rob Tholemeier, vice president and database analyst for the Meta Group in Burlingame, CA, Informix, Oracle and Red Brick products offer "dynamic parallel processing," which does



not depend heavily on the data layout and is able to determine which CPUs are loafing and optimize individual queries as appropriate. Dynamic parallel processing requires a symmetric multiprocessing (SMP) architecture, also called "shared memory" or "shared everything." AT&T Teradata, IBM and Tandem Computers sell solutions that are examples of what Tholemeier calls "static parallel processing," which requires a more coherent layout of data and runs in a massively parallel processing (MPP) architecture, sometimes called "shared nothing." The problem with static parallelism is that, as data changes, so do the layout needs. "It's

relatively easy to do the first load [of the database in such a system]," he says. "The continuing administration is much more difficult."

Commercial parallel databases from Informix, Oracle and other vendors perform best in SMP environments. "Running lots of jobs where all of them share memory and share resources, SMP is the biggest bang for the buck," says Tholemeier, an SMP proponent. A recent report he authored concludes, "Users who can predefine a path to data for time-critical transactions can use shared nothing architectures (MPP) by effectively partitioning data. Otherwise, shared everything (SMP)

architectures will support both defined path and more flexible database designs."

SMP uses shared memory and disk I/O subsystems to service more than one CPU. SMP systems run a single copy of the operating system and share a single copy of the application. CPUs can be added without impacting the operating system, the application or the data. Tasks or processes are automatically distributed among CPUs through dynamic load balancing. As more CPUs are added, applications need not be retuned, and system management gets no more complicated. The bus bandwidth eventually gets used

(CONTINUED ON PAGE 30)

### Schubert Ticketing Services Conducting Sales Harmoniously

**S**elling tickets for theater, opera and sporting events might seem like basic transaction processing. But the core of this business is seat analysis, which rivals decision support in complexity. Here's a sample request: "I need the best available orchestra seats for a party of 10 for *Phantom of the Opera* on any Wednesday night over the next two months, except the third Wednesday of the month. One member of our party is in a wheelchair and will need an aisle seat." Two other factors add to the complexity. Dozens of similar queries can hit the system at once, and the definition of "best available seating" changes constantly.

To meet this kind of demand over and over, New York City-based Schubert Ticketing Services, which sells tickets to most Broadway and Off-Broadway productions and numerous other events, has adopted a parallel database on parallel processing hardware. "What parallel processing can do for us is gather up lots of seats simultaneously—if you have your data partitioned properly," says David Andrews, director of operations and chief technology executive. "A database that can support parallel updates can fire off multiple data requests that can be handled in parallel. We will do as many things in parallel as possible so we can get one request out of the way and the next one under way."

Schubert Ticketing chose a clustered SMP architecture of Digital Equipment Corp. servers running Digital Unix with Informix's OnLine Dynamic Server as the

database engine. Clients will be 486 PCs running Windows 95. "The data is centralized; the clients are all over the place in six states," says Andrews. The firm opted for the RISC Alpha chips because they support 64-bit processing, and only DEC was ready with its 64-bit architecture on Schubert's timetable. "Informix's partitioning scheme meant superior performance," Andrews adds.

The database engine for the ticketing application is just part of a \$7 million migration to a client/server system. But ticketing—the mission-critical business process—is driving everything else. Schubert has more than 600 PC clients accessing the database from dozens of outlets in six states. It sells about 11.5 million tickets—worth about \$600 million—per year.

Andrews is leading his 35-member IS staff through a two-year migration. "Today we're on the ragged edge. I wouldn't want to deploy it," he says. The application is still under development, and Andrews says it will need all the performance that the Alpha and OnLine Dynamic Server will offer by the time he's ready to deploy. "When we spec'd what we needed a year ago, that's where we are on the performance curve today, but we want 50 percent above that for our safety margin," he says. "That will be here a year from now."

Schubert presently uses a Control Data mainframe with a proprietary database. "It's finely honed and tuned to ticketing but in a single-threaded manner," Andrews explains. "So we are dependent

on a single CPU for our analysis." There is enormous administrative overhead in this system. "Every time you make a change in the database, you have to write a program to restructure all the in-memory data pointers."

Although plans aren't final yet, he anticipates using six to eight Alphas in three separate boxes. As a hedge against growth, each box can handle as many as 12 CPUs. Andrews plans on 1GB of RAM with a 100GB database. "We typically have about 20 million-plus seats available for sales at any one time and a customer phone database of about three million entries per year. Today we're limited in historical data analysis, and it's not cost-effective." That will change. Andrews envisions a data warehouse for decision support using extra processors.

"To get the power out of SMP you have to be cognizant of your business activity and its data," he advises. "Then be proactive about how you partition your database and how you spread it across CPUs and disk drives, in order to maximize the performance of the database."

The company has not brought in many consultants, because the IS development staff of 10 understands its data and how people use it. "Customer-service-oriented solutions are becoming more important. You want someone to pick up the phone in 20 seconds and get you on your way quickly," Andrews says. "IT decisions have to flow from the business process. For Schubert, everything is driven from ticket sales."

This page  
intentionally left  
blank



(CONTINUED FROM PAGE 28)

up, but it can be optimized by taking advantage of the efficiencies of sharing.

Analysts and vendors estimate that uniprocessor and SMP systems together

account for 80 to 90 percent of the commercial database server market. Increasingly the larger share is SMP. The most powerful SMP chips can handle most commercial database applications, but of

course there are trade-offs. Due to systems overhead, efficiency of systems declines after reaching a certain number of processors—eight or 16 in most Unix systems. At that point, the choice becomes

### Pyxis Corp. Parallel Is Premature

Just as a child outgrows shoes, a company outgrows the small IT systems that carried it during its early years. Pyxis Corp., a seven-year-old manufacturer of automated medication and supply dispensing systems for health care providers, based in San Diego, recently faced this issue. It needed a system that would synchronize all company divisions with consistent information.

"Right now we have piecemeal systems," says David Nebo, business application manager for the \$200 million, publicly held company, which ranks among *Fortune's* top 10 growth companies. There is one system in contracts, a different one in manufacturing and a third in manufacturing, and sales and distribution have had virtually nothing. "They couldn't support our growth and didn't talk to each other," says Nebo. "We needed to take a step up, and the next logical step was into a Unix environment."

The company decided to keep its LAN but add several Hewlett-Packard Unix servers to build an enterprise solution. On top of that will be Informix's OnLine Dynamic Server as the database engine and the R3 accounting and business software from SAP America of Philadelphia. Pyxis opted for four HP 9000 servers: one for the database, two for applications and one for the development group. The Novell LAN will remain for file and print sharing. "We wanted an integrated solution for the corporation to handle current requirements and all future growth and expansion," says Nebo.

The company database—about 5GB in size—currently resides on a NetWare server. Nebo figures the new database server will handle 40GB to 50GB, which leaves Pyxis room to grow. But the company's not moving to parallel processing yet. Both the HP Unix server and the

Informix software are capable of it, but Nebo doesn't figure Pyxis is ready. "Parallel is a migration path. That's one reason we went that way," he says. "It provides us with the infrastructure to move to that arena if we need it." Nebo expects to roll out the entire new system before the end of this year.

The company retained consultants Ernst & Young to assist with implementation and added six people with Unix and database skills to its existing seven-person IS staff. "We're having key individuals on the systems side and the apps side go off and take training, then come back and train staff," says Nebo. "To get people in here with the experience level we required is cost-prohibitive. Trying to train everyone is very expensive. We're taking a bit longer on the learning curve, but it pays in the long run by training people on staff."

### Thinking Massively

If parallel processing is a certainty for your fast-growing organization, here are 10 criteria for judging whether an MPP-driven data warehouse is in your future, suggested by the Standish Group.

**Scalability.** All aspects of the data warehouse must scale: the database engine, storage, processors and the tools to manage the system. When scaling becomes necessary, it must be accomplished in a smooth fashion invisible to the end user.

**Date Management Utilities.** Database loading, data redistribution and other routine management operations should take advantage of the parallel environment, operating concurrently with online user queries.

**Performance Optimization.** To achieve high performance, time-intensive CPU operations, such as sorting, must be minimized. The system also

should be designed to enhance the location of the data and to optimize queries.

**Openness.** In this context, *open* means able to support a range of modeling and desktop analysis tools, through ODBC or other standard interfaces. It also implies the ability to gather data from legacy systems and from dispersed OLTP systems.

**Availability and Integrity.** Continuous availability (fault tolerance) is the optimum solution. The system should also support user queries of varying length and complexity in addition to data management utilities. Data integrity, meanwhile, is a prevention issue. Again, fault tolerance is the best solution. Minimally, that means RAID storage technology and/or disk mirroring.

**Mixed Workload Support.** The system should shield users from performance variations associated with the processing

of queries of varying length and/or complexity and provide the appearance of near immediacy for all users.

**Cost of Ownership.** Be sure to evaluate not only initial price/performance metrics but also the cost of expansion—additional processors, disk storage, operating software, utilities and support.

**Integration.** Vendor expertise can enhance the opportunity for successful installation and minimize the risk of failure. First-timers should consider vendor-supplied consulting to be essential.

**Technical Experience.** Look for and verify customer references with similarly sized data warehouses from the proposed provider.

**Industry Experience.** The ideal vendor will have the industry-specific experience to guide first-time users to the "low-hanging fruit" that yields early, impressive benefits.

moving to clusters of SMP boxes or to MPP. In a cluster, a few or several computers are tied together with a high-speed bus and talk to the same database. Each computer could contain one or more CPUs. Clusters now comprise between 10 and 20 percent of the commercial database market.

Open systems hardware vendors have become accomplished in SMP. "There are many terrific boxes out there that do really good SMP up to 64 CPUs," Tholemeier says. Although a high-end SMP machine can run \$50,000 and SMP clusters \$100,000 or more, a more common, less expensive SMP system is a 5GB database on a single machine with four processors, which may cost \$20,000 to \$30,000.

### **A Rule of Thumb**

An MPP architecture can have hundreds or more processors in one computer. Each node in an MPP machine is a self-contained computer with CPU and memory. Connected to the others by a high-speed bus or switch, each node functions on its own, and hardware resources are generally not shared. It's more scalable than SMP at the hardware level and good for very large data warehouses of 200GB or more. Although theoretically such a system could have thousands of nodes, commercial applications typically use less than 100 nodes. (For MPP adoption criteria, see "Thinking Massively," at left.)

Today this leading-edge technology probably has less than 2 percent of the commercial database market. The Standish Group International of Dennis, MA, which has studied the MPP market, views the leaders as AT&T Teradata, IBM and Tandem. These systems cost up to \$2 million.

SMP proponents assert that it's much easier to implement and maintain than MPP. Jim Johnson, Standish chairman, doesn't necessarily agree. "There aren't a lot of people who have done both. I haven't found an end user who will stand up and say SMP is easier." Size is the best metric, he believes. "Anybody who has 200GB should go MPP. Between 100 and 200 is no man's land. For 100GB down to 50GB, SMP is the way to go." **IT**

***Bill Roberts** is a free-lance writer who covers business, technology and management. He can be reached at [wcrober@aol.com](mailto:wcrober@aol.com).*