**Parallel Processing & Parallel Databases**

This chapter introduces parallel processing and parallel database technologies, which offer great advantages for online transaction processing and decision support applications. The administrator's challenge is to selectively deploy this technology to fully use its multiprocessing power.

*Definition*

Parallel processing divides a large task into many smaller tasks, and executes the smaller tasks concurrently on several nodes. As a result, the larger task completes more quickly.

Some tasks can be effectively divided, and thus are good candidates for parallel processing. Other tasks, however, do not lend themselves to this approach.

*For example,* in a bank with only one teller, all customers must form a single queue to be served. With two tellers, the task can be effectively split so that customers form two queues and are served twice as fast-or they can form a single queue to provide fairness. This is an instance in which parallel processing is an effective solution.

*By contrast,* if the bank manager must approve all loan requests, parallel processing will not necessarily speed up the flow of loans. No matter how many tellers are available to process loans, all the requests must form a single queue for bank manager approval. No amount of parallel processing can overcome this built-in bottleneck to the system.

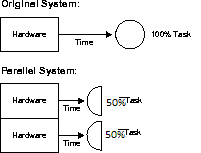
Objective or goal or need for parallel processing for Database

Three issues are the driving force to implement parallel processing for database:

*Speed-up:*

Database sizes are increasing day by day and are becoming more complex. So to fetch the data from various users at a time there is a need to speed up the performance.

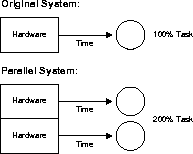
*Speed-up* is the extent to which more hardware can perform the same task in less time than the original system. With added hardware, speed-up holds the task constant and measures time savings.



*Scale-up:*

This requirement goes hand-in-hand with performance. Databases often grow rapidly, and companies need a way to easily and cost-effectively scale their systems to match that growth.

*Scale-up* is the factor m that expresses how much more work can be done in the same time period by a system n times larger. With added hardware, a formula for scale-up holds the time constant, and measures the increased size of the job which can be done.



*Availability:*

*High availability* refers to the need to keep a database up and running with minimal or no downtime. With the increasing use of the Internet, companies need to accommodate users at all hours of the day and night.

*Advantages of Parallel database*

If an organization has a large user base and millions of records to process, it may turn to a parallel database approach. Parallel databases are fast, flexible and reliable.

* **Speed**

The main advantage to parallel databases is speed. The server breaks up a user database request into parts and dispatches each part to a separate computer. They work on the parts simultaneously and merge the results, passing them back to the user. This speeds up most data requests, allowing faster access to very large databases.

* **Reliability**

A parallel database, properly configured, can continue to work despite the failure of any computer in the cluster. The database server senses that a specific computer is not responding and reroutes its work to the remaining computers.

* **Capacity**

As more users request access to the database, the computer administrators add more computers to the parallel server, boosting its overall capacity. A parallel database, for example, allows a large online retailer to have thousands of users accessing information at the same time. This level of performance is not possible with a single server.

Types of Parallel Database Architecture

Shared memory

Shared disk

Shared nothing