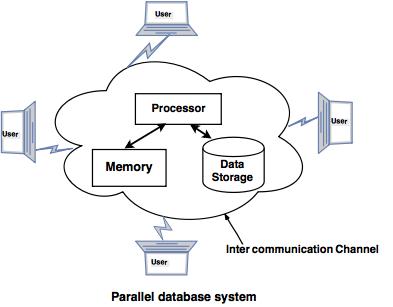
**Chapter 04**

**Parallel database**

**Introduction of Parallel Database**

**:-**

**ParallelDatabases :**  
Nowadays organizations need to handle a huge amount of data with a high transfer rate. For such requirements, the client-server or centralized system is not efficient. With the need to improve the efficiency of the system, the concept of the parallel database comes in picture. A parallel database system seeks to improve the performance of the system through parallelizing concept.



**Need**   
Multiple resources like CPUs and Disks are used in parallel. The operations are performed simultaneously, as opposed to serial processing. A parallel server can allow access to a single database by users on multiple machines. It also performs many parallelization operations like data loading, query processing, building indexes, and evaluating queries.

**Advantages**   
Here, we will discuss the advantages of parallel databases. Let’s have a look.

1. **PerformanceImprovement**  
   By connecting multiple resources like CPU and disks in parallel we can significantly increase the performance of the system.
2. **Highavailability –**  
   In the parallel database, nodes have less contact with each other, so the failure of one node doesn’t cause for failure of the entire system. This amounts to significantly higher database availability.
3. **Proper resource utilization –**  
   Due to parallel execution, the CPU will never be ideal. Thus, proper utilization of resources is there.
4. **Increase Reliability –**  
   When one site fails, the execution can continue with another available site which is having a copy of data. Making the system more reliable.

**Performance Measurement of Databases :**  
Here, we will emphasize the performance measurement factor-like Speedup and Scale-up. Let’s understand it one by one with the help of examples.

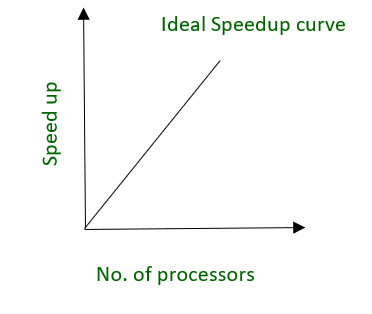
**Speedup –**  
The ability to execute the tasks in less time by increasing the number of resources is called Speedup.

**Speedup=time original/time parallel**

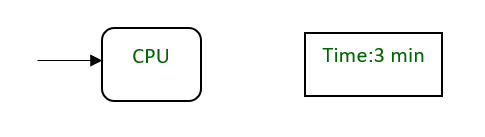
Where ,

time original = time required to execute the task using 1 processor

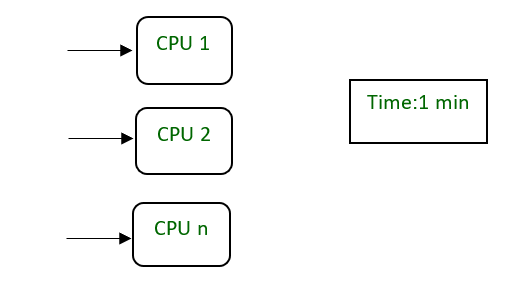
time parallel = time required to execute the task using 'n' processors



**Example –**



*fig. A CPU requires 3 minutes to execute a process*



*fig. ‘n’ CPU requires 1 min to execute a process by dividing into smaller tasks*

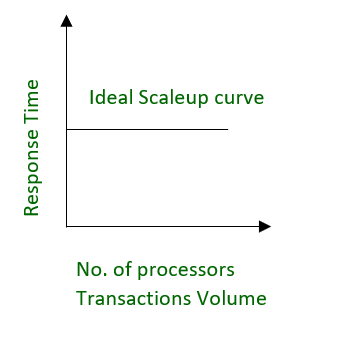
**Scale-up –**  
The ability to maintain the performance of the system when both workload and resources increase proportionally.

**Scaleup = Volume Parallel/Volume Original**

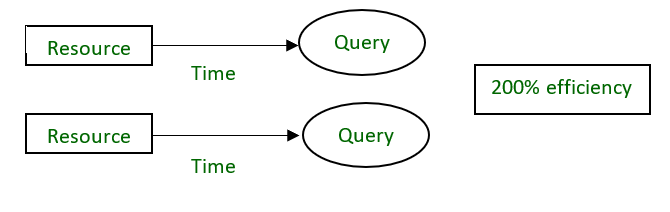
Where ,

Volume Parallel = volume executed in a given amount of time using 'n' processor

Volume Original = volume executed in a given amount of time using 1 processor



**Example –**  
20 users are using a CPU at 100% efficiency. If we try to add more users, then it’s not possible for a single processor to handle additional users. A new processor can be added to serve the users parallel. And will provide 200% efficiency.



Parallel Computer Architecture is the method of organizing all the resources to maximize the performance and the programmability within the limits given by technology and the cost at any instance of time. It adds a new dimension in the development of computer system by using more and more number of processors. This tutorial covers the basics related to Parallel Computer Architecture, discussing the various concepts and terminologies associated with the topic.