Parallel Query Processing

Dr.N.Sairam & Dr.R.Seethalakshmi
School of Computing,
SASTRA Univeristy,
Thanjavur-613401.

Contents

1. Parallel Query Processing

Query is processed using single processor or multiple processors. When executing a composite query it is decomposed into segments. These segments if it is executed in single processor it takes time. Further when the query involves I/O, it is going to be the bottleneck. Hence, composite query can be executed using parallel processors.

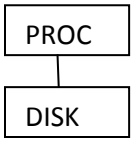

Figure3.1: Single CPU System

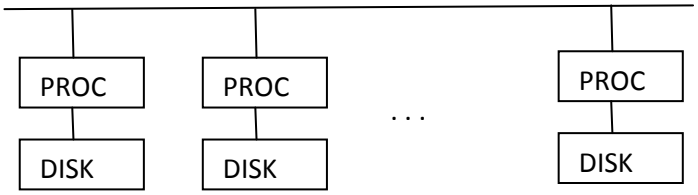

Figure 3.2: Parallel query processing

When multiple CPUs are available, SQL Server provides parallel queries to optimize query execution and index operations for these computers. Because SQL Server can perform a query or index operation in parallel by using several operating system threads, the operation can be completed quickly and efficiently.

SQL Server looks for queries or index operations that might benefit from parallel execution, during query execution. For such queries, SQL Server inserts exchange operators into the query execution plan to prepare the query for parallel execution. An exchange operator is an operator in a query execution plan that provides process management, data redistribution, and flow control. The exchange operator includes the Distribute Streams, Repartition Streams, and Gather Streamslogical operators as subtypes, one or more of which can appear in the Showplan output of a query plan for a parallel query.

Once the exchange operators are inserted, the result obtained is a parallel-query execution plan. A parallel-query execution plan can use more than one thread. A serial execution plan, used by a nonparallel query, uses only one thread for its execution. The actual number of threads used by a parallel query is determined at query plan execution initialization and is determined by the complexity of the plan and the degree of parallelism. Degree of parallelism determines the maximum number of CPUs that are being used.It does not mean the number of threads that are being used. The degree of parallelism value is set at the server level and can be modified by using the **sp_configure** system stored procedure.

The SQL Server query optimizer does not use a parallel execution plan for a query if any one of the following conditions is true:

- The serial execution cost of the query is not high

- A serial execution plan is considered faster than any possible parallel execution plan
- The query contains scalar or relational operators that cannot be run in parallel.

There are several multiprocessor architectures possible for parallel query processing

They are

Shared Memory Architectures

Shared Disk Architectures

Shared Nothing Architectures

Hybrid Architectures

1.1 Shared Memory Architectures

In this architecture there are multiple processors are available. There is only one memory module. Any processor can access the memory module or disk unit. Thus the queries can be decomposed into fragments and these fragments are executed in parallel. Figure 3.3 shows the architecture of this model. The advantages of this model are simplicity and load balancing. The disadvantages of this model are high cost, little extensibility and low availability.

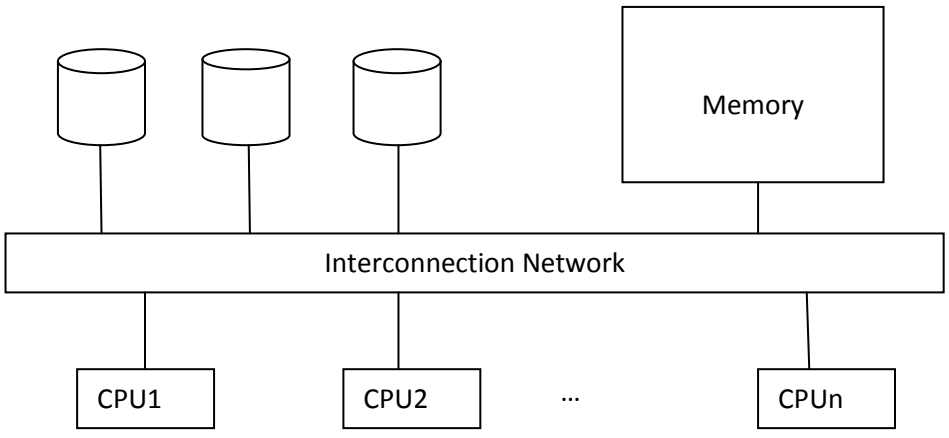

Figure 3.3: Shared Memory Architectures

1.2 Shared Disk Architecture

In contrast to the shared memory architecture the shared disk architecture consists of multiple memory modules. Each CPU has its own memory module. Using the interconnection network the multiple CPUs are able to access the disk modules. Figure 3.4 shows the shared disk architecture. The advantages of this model are low cost, extensibility, load balancing, availability and easy migration. But this model has potential performance problem. This model has higher complexity.

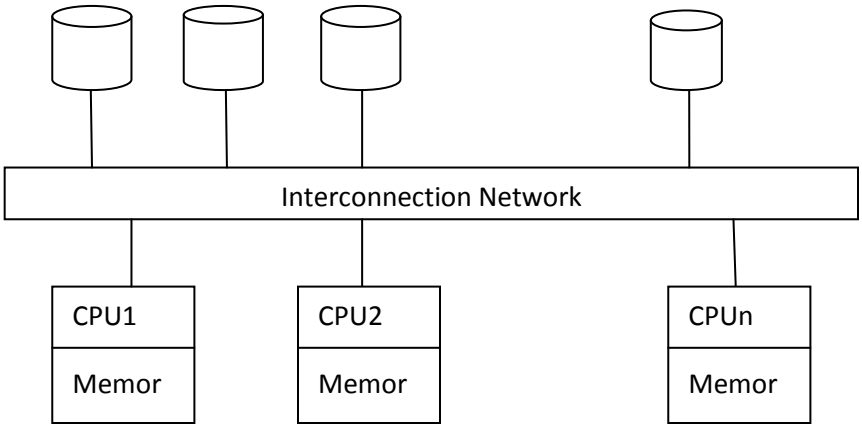

Figure 3.4: Shared Disk Architectures

1.3 Shared Nothing Architectures

In this architecture each CPU will have its own Memory and disk. The interconnection network is still used to establish communication between various CPUS. The architecture is shown in the diagram below.

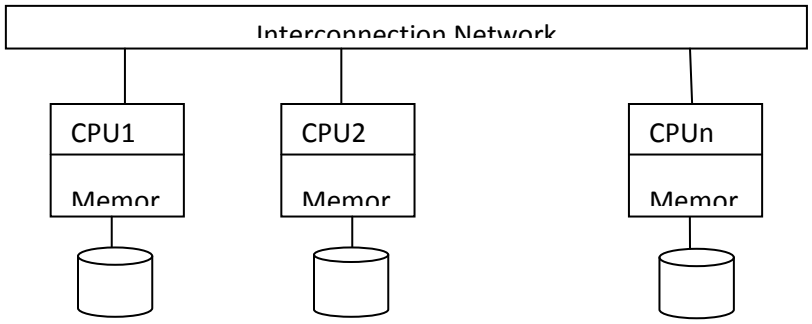

Figure 3.5: Shared Nothing Architecture

The advantages of this model are low cost, extensibility and availability. The disadvantages are higher complexity and load balancing.

Animation: Parallel Query Processing

Note: Only can be viewed Acrobat 9.0 and above

1.4 Hybrid Architectures

The hybrid architectures combine the advantages of different architectures. It uses different processing elements in the system. The system is nothing but shared nothing architecture where each node is a multicomputer system of any architecture.

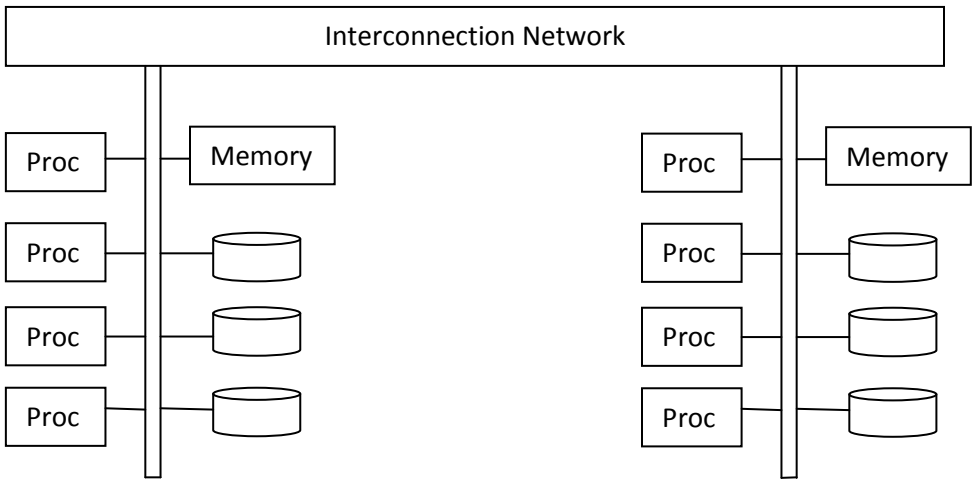

Figure 3.6 Hybrid Architectures

1.5 Parallel Relational Operators

There are three parallel relational operators. They are data partitioning, parallelization of relational operators and Join.

1.5.1 Data partitioning

The data partitioning is nothing but the distribution of tuples of a relation over several disks. The goal of the data partitioning is allowing parallel databases to exploit the I/O bandwidth of multiple disks by reading them and writing them in parallel. Relations are horizontally partitioned using three functions namely:

Round-robin

Range index

Hash function.

1.5.1.1 Round Robin Partitioning.

It maps the i^{th} tuple to the disk I mod n. It sequentially scans all the tuples in each query.

1.5.1.2 Range index partitioning

It clusters all the tuples with same attributes in the same partition. There is a sequential scan of all tuples in each query. It also performs associative search for data and clustering of data.

1.5.1.3 Hash partitioning

This method uses a hash function to group the tuples. Hash randomizes the data rather than cluster it. It searches the data associatively.

1.6 Parallel Relational Operators

Relational algebra allows parallel processing due to its properties. It is a set oriented processing. The relational operators perform simple operations. These operations are limited in nature. The basic idea is that it uses the parallel data streams and uses the sequential relational operators in parallel. Each relational operator has a set of input ports on which the input tuples arrive and an output port to which the operator's output stream is sent. The parallel dataflow works by partitioning and merging data streams into these sequential ports.