Tutorial 3 Answers

1. Explain the difference between a Distributed System and a Parallel System. When is a Distributed System also a Parallel System, and vice versa?

Topic 5 – 5,6 Slides

Topic 1 – 6,7,8,28

Distributed computing involves any computing system, where computational activity is divided across multiple host machines, linked by a digital communications medium of some type.

While a parallel computing system may be distributed, many distributed systems do not closely fit the model of parallel computing, as the components of the distributed system and applications may be dissimilar in many ways.

Numerous distributed schemes in wide use today, many up to decades old by design:

1. Client / Server – socket applications, file serving, web serving;

2. Parallel Processing with homogenous workloads – clusters running MPI, Grids running MPI or other parallel applications;

3. Parallel Processing with heterogeneous workloads – clusters for web serving, Grids with some mixed applications; Clouds with mixed workloads;

Parallel applications led to parallel computing:

1. Coarse-grained systems – SMP (Symmetrical Multi Processing);

2. Medium-grained systems – Clusters;

3. Fine-grained systems – MPP (Massively Parallel Processing);

• The intention behind all of these schemes is to divide up large computational tasks across large numbers of processors;

• Performance improvements intended to be proportional to the number of processors, but limited by Amdahl's Law;

• These schemes were well established by the 1990s but with the exception of SMP, usage was mostly limited to specialised applications, mainly in scientific computing.

Topic 1 – Slide 28

• Client-Server – well established protocols e.g. HTTP, running over socket connections.

• Client-Server – Remote Procedure Calls (RPC) where client can execute program on server, NFS over RPC.

• Client-Server – OO schemes such as CORBA (Common Object Request Broker) where client can create, destroy, execute objects on a remote or local server.

• Parallel Distributed Schemes – Clusters, Grids, Clouds.

From Stackoverflow:

Parallelism is generally concerned with accomplishing a particular computation as fast as possible, exploiting multiple processors. The scale of the processors may range from multiple arithmetical units inside a single processor, to multiple processors sharing memory, to distributing the computation on many computers. On the side of models of computation, parallelism is generally about using multiple simultaneous threads of computation internally, in order to compute a final result. *Parallelism* is also sometimes used for real-time reactive systems, which contain many processors that share a single master clock; such systems are fully deterministic.

Distributed computing studies separate processors connected by communication links. Whereas parallel processing models often (but not always) assume shared memory, distributed systems rely fundamentally on message passing. Distributed systems are inherently concurrent. Like concurrency, distribution is often part of the goal, not solely part of the solution: if resources are in geographically distinct locations, the system is inherently distributed. Systems in which partial failures (of processor nodes or of communication links) are possible fall under this domain.

Grid computing is the most distributed form of parallel computing. It makes use of computers communicating over the Internet to work on a given problem. Because of the low bandwidth and extremely high latency available on the Internet, distributed computing typically deals only with embarrassingly parallel problems. Many distributed computing applications have been created, of which SETI@home and Folding@home are the best-known examples.[36]

2. How do the latency and throughput (bandwidth) of the interconnect in a cluster influence the performance of the cluster?

Topic 4 – Slide 18

Notice the term "Broadband" - it refers to how wide the pipe is, not how fast. For our purposes, it is the amount of time it takes a packet to travel from source to destination. Together, **latency** and **bandwidth** define the speed and capacity of a network. **Latency** is normally expressed in milliseconds.

https://zoompf.com/blog/2011/12/i-dont-care-how-big-yours-is

In principle, the limits to the number of cores in a cluster are determined by the performance limits of the “fabric” interconnecting the machines forming the cluster;

In constructing clusters, which are the basic building block in Grids and clouds, the performance of the interconnecting fabric is critically important;

• Two parameters are of interest, these are latency and bandwidth in interconnecting processes running on cores. **The inter process communication speeds depend on the latency and bandwidth.**

3. Explain the limitations of metrics such as FLOPS, MOPS and MIPS [Hint: reread FIT2069 Topics 11 and 12]. Explain the term Scalability. How is it measured or calculated?

FLOPS are floating-point operations per second. Floating-point is, according to IBM, "a method of encoding real numbers within the limits of finite precision available on computers." Using floating-point encoding, extremely long numbers can be handled relatively easily.

MOPS – Million Operations Per Second

Million instructions per second **(MIPS)** is an older, obsolete measure of a computer's speed and power, MIPS measures roughly the number of machine instructions that a computer can execute in one second.

https://en.wikipedia.org/wiki/FLOPS

Topic 4 – Slide 19,20

• Maximum aggregate performance of the system can be measured in terms of Maximum aggregate

floating-point operations: **P = N*C*F*R**

Where:

• P performance in flops,
• N number of nodes,
• C number of CPUs,
• F floating point ops per clock period,
• R clock rate.
• The other measure is for integer operations – using MOPS/MIPS

These metrices give an idea about the peak performance levels of a system. They do not give you an accurate measure of performance practically.

For example, all CPUs in each node may not be running the program that you want to run. Even if they run your program, they may not be executing your program using their full computational power.

Other computer performance metrics (things to measure) include availability, response time, channel capacity, latency, completion time, service time, bandwidth, throughput, relative efficiency, scalability, performance per watt, compression ratio, instruction path length and speed up. All these need to be taken into account if you are to perform a proper performance measurement.

Performance also depend on the type of software you are running. If they are not optimized to run on the given distributed system, the system performance may reduce.

Topic 4 – Slide 22

• Scalability is computed thus: S = T(1) / T(N)

• Where T(1) is the wall clock time for a program to run on a single processor

• T(N) is the runtime over N processors

• A scalability figure close to N means the program scales well

• Stability metric helps estimate the optimal number of processors for an application

4. Explain the di_erences between a Distributed Storage System and a Conventional Storage System.

Topic 5 – Slide 33

Traditional mass storage schemes involve the use of local disk storage on machines, which may be individual disk drives, or RAID (Redundant Array of Independent Disks) storage arrays;

- Where the dataset is large, and significant disk bandwidth is required for applications, the local interface to the disk or disk storage array will become a performance bottleneck;

- The idea of distributed storage is to spread a large dataset across a very large number of host machines, so that the disk I/O traffic is also spread, across many disk interfaces, thus removing the interface bottleneck of a large storage device or array;