CLOUD COMPUTING

UNIT- I

1. DISTRIBUTED MODELS AND ENABLING TECHNOLOGIES:

Introduction:

Discussions about 30 years of Cloud Computing

We study both high-performance and high-throughput computing systems in parallel computers appearing as

- computer clusters
- service-oriented architecture
- computational grids
- peer-to-peer networks
- Internet clouds
- Internet of things.

These systems are distinguished by their

- Hardware architectures
- OS platforms
- Processing algorithms
- Communication protocols and
- Service models

Essential issues:

- Scalability
- Performance
- Availability
- Security
- Energy efficiency in distributed systems.

Important Points:

1. **LincBenchmark : used to calculate the performance of a super computer.(FLOPs)**
2. **HPC is meant for science and HTC is meant for Business.**
3. **According to ELI, The IT Guy of everymanit.com Cloud Computing is the separation of an Application from its OS , in turn from its Hardware.**
4. **Hypervisors: KVM, Xen, Hyper-V, ESXi**

Peer-to-peer (**P2P**) computing or **networking** is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a **peer-to-peer network** of nodes.

A **WEB CRAWLER** (also known as a **web** spider or **web** robot) is a program or automated script which browses the World Wide **Web** in a methodical, automated manner. This process is called **Web crawling** or spidering. Many legitimate sites, in particular search engines, use spidering as a means of providing up-to-date data.

In computers, **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.

1.1 SCALABLE COMPUTING OVER THE INTERNET

- Centralized Computing issues
- Distributed Computing benefits

1.1.1.1 The Age of Internet Computing

- High Performance Computing (HPC)
- High Throughput Computing(HTC)

1.1.1.1.1 The Platform Evolution

5 Generations of computers:

- **1950-70 : Mainframes : IBM 360 and CDC 6400**
- **1960-80: Low Cost Mini Computers: DEC PDP11 and VAX Series**
- **1970-90: PCs with VLSI Micro processors**
- **1980-00: Portable and Pervasive Devices: Wired and Wireless Applications.**
- **1990: HPC and HTC**

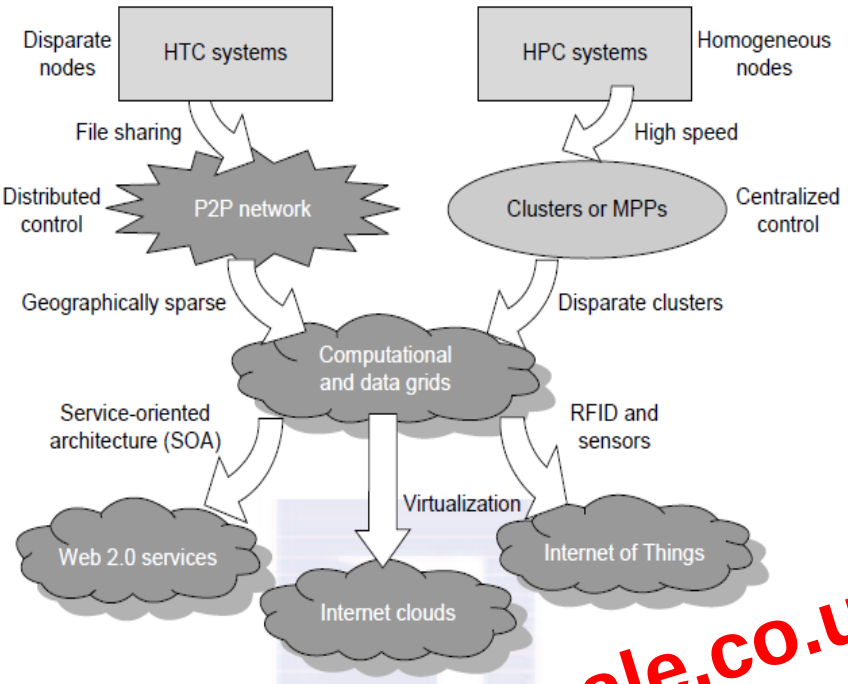


FIGURE 1.1

Evolutionary trend toward parallel, distributed and cloud computing with clusters, MPPs, P2P networks, grids, clouds, web services, and the Internet of Things.

1.1.1.2 High-Performance Computing:

High Performance Computing most generally refers to the practice of aggregating **computing** power in a way that delivers much higher **performance** than one could get out of a typical desktop **computer** or workstation in order to solve large problems in science, engineering, or business.

Eg:- Super Computer

Applications:-

High performance computing tends to be performed by scientists and researchers in the biotech, geological, and astronomy spaces. Things like gene sequencing, oil discovery, and weather forecasting.

1.1.1.3 High-Throughput Computing:

- This HTC paradigm pays more attention to high-flux computing.
- The main application for high-flux computing is in Internet searches and web services by millions or more users simultaneously.

1.3.1.4 Major Cluster Design Issues

- Unfortunately, a cluster-wide OS for complete resource sharing is not available yet.
- Middleware or OS extensions were developed at the user space to achieve SSI at selected functional levels.
- Without this middleware, cluster nodes cannot work together effectively to achieve cooperative computing.
- The software environments and applications must rely on the middleware to achieve high performance.
- The cluster benefits come from scalable performance, efficient message passing, high system availability, seamless fault tolerance, and cluster-wide job management, as summarized below.

Table 1.3 Critical Cluster Design Issues and Feasible Implementations

Features	Functional Characterization	Feasible Implementations
Availability and Support	Hardware and software support for sustained HA in cluster	Failover, failback, check pointing, rollback recovery, nonstop OS, etc.
Hardware Fault Tolerance	Automated failure management to eliminate all single points of failure	Component redundancy, hot swapping, RAID, multiple power supplies, etc.
Single System Image (SSI)	Achieving SSI at functional level with hardware and software support, middleware, or OS extensions	Hardware mechanisms or middleware support to achieve DSM at coherent cache level
Efficient Communications	To reduce message-passing system overhead and latencies	Fast message passing, active messages, enhanced MPI library, etc.
Cluster-wide Job Management	Using a global job management system with better scheduling and monitoring	Application of single-job management systems such as LSF, Codine, etc.
Dynamic Load Balancing	Balancing the workload of all processing nodes along with failure recovery	Workload monitoring, process migration, job replication and gang scheduling, etc.
Scalability and Programmability	Adding more servers to a cluster or adding more clusters to a grid as the workload or data set increases	Use of scalable interconnect, performance monitoring, distributed execution environment, and better software tools

1.3.2 Grid Computing Infrastructures

In the past 30 years, users have experienced a natural growth path from Internet to web and grid computing services.
Internet services such as the Telnet command enables a local computer to connect to a remote computer.
A web service such as HTTP enables remote access of remote web pages.
Grid computing is envisioned to allow close interaction among applications running on distant computers simultaneously.

- This includes many popular P2P networks such as Gnutella, Napster, and BitTorrent, among others.
- Collaboration P2P networks include MSN or Skype chatting, instant messaging, and collaborative design, among others.
- The third family is for distributed P2P computing in specific applications.
- For example, SETI@home provides 25 Tflops of distributed computing power, collectively, over 3 million Internet host machines.
- Other P2P platforms, such as JXTA, .NET, and FightingAID@home, support naming, discovery, communication, security, and resource aggregation in some P2P applications.

Table 1.5 Major Categories of P2P Network Families [46]

System Features	Distributed File Sharing	Collaborative Platform	Distributed P2P Computing	P2P Platform
Attractive Applications	Content distribution of MP3 music, video, open software, etc.	Instant messaging, collaborative design and gaming	Scientific exploration and social networking	Open networks for public resources
Operational Problems	Loose security and serious online copyright violations	Lack of trust, disturbed by spam, privacy, and peer collusion	Security holes, selfish partners, and peer collusion	Lack of standards or protection protocols
Example Systems	Gnutella, Napster, eMule, BitTorrent, Aimster, KaZaA, etc.	ICQ, AIM, Groove, Magi, Multiplayer Games, Skype, etc.	SETI@home, Genome@home, etc.	JXTA, .NET, FightingAid@home, etc.

1.3.3.4 P2P Computing Challenges

- P2P computing faces three types of heterogeneity problems in hardware, software, and network requirements.
- There are too many hardware models and architectures to select from; incompatibility exists between software and the OS; and different network connections and protocols make it too complex to apply in real applications.
- We need system scalability as the workload increases. System scaling is directly related to performance and bandwidth.
- P2P networks do have these properties.
- Data location is also important to affect collective performance.
- Data locality, network proximity, and interoperability are three design objectives in distributed P2P applications.
- P2P performance is affected by routing efficiency and self-organization by participating peers. Fault tolerance, failure management, and load balancing are other important issues in using overlay networks.
- Lack of trust among peers poses another problem. Peers are strangers to one another.
- Security, privacy, and copyright violations are major worries by those in the industry in terms of applying P2P technology in business applications.

1.5 PERFORMANCE, SECURITY, AND ENERGY EFFICIENCY

1.5.1 Performance Metrics and Scalability Analysis

1.5.1.1 Performance Metrics

- In a distributed system, performance is attributed to a large number of factors.
- System throughput is often measured in MIPS, Tflops (tera floating-point operations per second), or TPS (transactions per second).
- Other measures include job response time and network latency.
- An interconnection network that has low latency and high bandwidth is preferred.
- System overhead is often attributed to OS boot time, compile time, I/O data rate, and the runtime support system used.
- Other performance-related metrics include the QoS for Internet and web services; system availability and dependability; and security resilience for system defense against network attacks

1.5.1.2 Dimensions of Scalability

Users want to have a distributed system that can achieve scalable performance. Any resource upgrade in a system should be backward compatible with existing hardware and software resources. Overdesign may not be cost effective. System scaling can increase or decrease resources depending on many practical factors. The following dimensions of scalability are characterized in parallel and distributed systems:

•Size scalability This refers to achieving higher performance or more functionality by increasing the machine size. The word "size" refers to adding processors, cache, memory, storage, or I/O channels. The most obvious way to determine size scalability is to simply count the number of processors installed. Not all parallel computer or distributed architectures are equally size scalable.

For example, the IBM S2 was scaled up to 512 processors in 1997. But in 2008, the IBM BlueGene/L system scaled up to 65,000 processors.

•Software scalability This refers to upgrades in the OS or compilers, adding mathematical and engineering libraries, porting new application software, and installing more user-friendly programming environments. Some software upgrades may not work with large system configurations. Testing and fine-tuning of new software on larger systems is a nontrivial job.

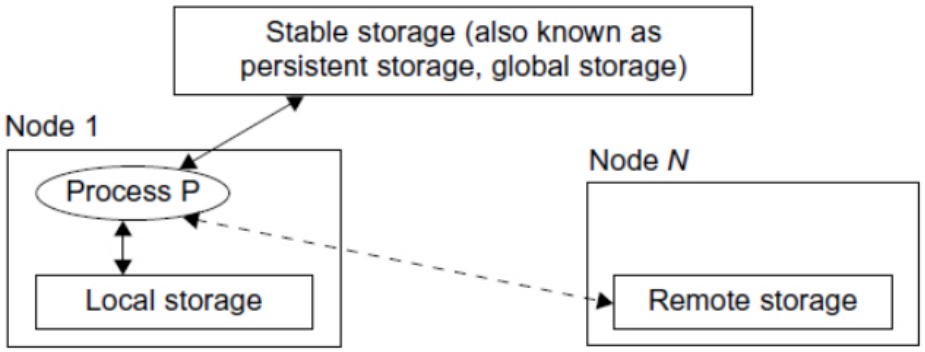
•Application scalability This refers to matching problem size scalability with machine size scalability. Problem size affects the size of the data set or the workload increase. Instead of increasing machine size, users can enlarge the problem size to enhance system efficiency or cost-effectiveness.

•Technology scalability This refers to a system that can adapt to changes in building technologies, such as the component and networking technologies discussed in Section 3.1. When scaling a system design with new technology one must consider three aspects: time, space, and heterogeneity. (1) Time refers to generation scalability. When changing to new-generation processors, one must consider the impact to the motherboard, power supply,

To run a server farm (data center) a company has to spend a huge amount of money for hardware, software, operational support, and energy every year. Therefore, companies should thoroughly identify whether their installed server farm (more specifically, the volume of provisioned resources) is at an appropriate level, particularly in terms of utilization. It was estimated in the past that, on average, one-sixth (15 percent) of the full-time servers in a company are left powered on without being actively used (i.e., they are idling) on a daily basis. This indicates that with 44 million servers in the world, around 4.7 million servers are not doing any useful work. The potential savings in turning off these servers are large—$3.8 billion globally in energy costs alone, and $24.7 billion in the total cost of running nonproductive servers, according to a study by 1E Company in partnership with the Alliance to Save Energy (ASE). This amount of wasted energy is equal to 11.8 million tons of carbon dioxide per year, which is equivalent to the CO pollution of 2.1 million cars. In the United States, this equals 3.17 million tons of carbon dioxide, or 580,678 cars. Therefore, the first step in IT departments is to analyze their servers to find unused and/or underutilized servers.

- User requests to log into the cluster (telnet cluster.cs.hku.hk)
- The DNS translates the symbolic name and returns the IP address 159.226.41.150 of the least loaded node (Host1)
- The user then logs in using this IP address
- The DNS periodically receives load information from the host nodes to make load-balancing translation decisions

➢ Single File Hierarch



- A stable storage is persistent, which means that it retains data for long time even after cluster shuts down; and it is fault-tolerant to some degree, by using redundancy and periodic backups
- Could be centralized (single point of failure and a potential performance bottleneck) or distributed (difficult to implement, but more economical, efficient, and available)

Single other resources