Assignment - 01

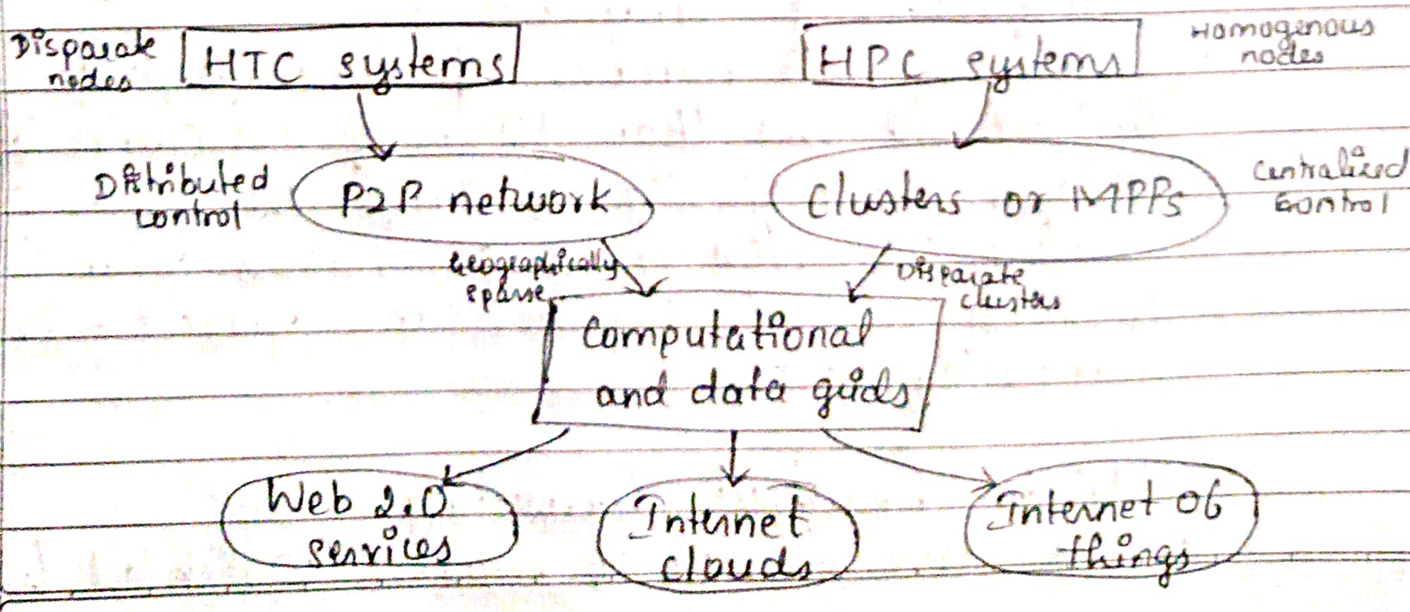
01. Explain about Internet Computing advancements over ages.

→ Billions of people use Internet every day. As a result, supercomputer sites and large data centers must provide high-performance computing services to huge numbers of Internet users concurrently. The purpose is to advance network-based computing and web services with emerging new technologies.

a) The Platform Evolution:

- Computer technology has gone through five generations of development:

→ 1950 to 1970, a handful of mainframes
→ 1960 to 1980, lower-cost minicomputers
→ 1970 to 1990, VLSI microprocessors
→ 1980 to 2000, Portable computers and pervasive computers / devices
→ Since 1990, HPC and HTC systems



- Above figure illustrates the evolution of HPC and HTC Systems.
- On the HPC side, supercomputers (massively parallel processors or MPPs) are gradually replaced by clusters of cooperative computers. The cluster is often a collection of homogeneous compute nodes that are physically connected.
- On the HTC side, peer-to-peer (P2P) networks are formed for distributed file sharing and content delivery applications. P2P, cloud computing and web service platforms are more focused on HTC applications that on HPC applications.

b) High-Performance Computing:
- For many years, HPC systems emphasize the raw speed performance.
- The speed of HPC systems has increased from Gflops in the early 1990s to now Pflops in 2010.
- This improvement was driven mainly by the demands from scientific, engineering and manufacturing communities.
- However, the number of supercomputer users is limited to less than 10% of all computer users.
- Today, the majority of computers users are using desktop computers or large servers when they conduct Internet searches and market-driven computing tasks.

c) High-Throughput Computing:
- The development of market-oriented high-end

computing systems is undergoing a strategic change from an HPC to HTC paradigm.

- This HTC paradigm pays more attention to high-flux computing. The main application for high-flux computing in in Internet and web services by millions or more users simultaneously.
- The performance goal thus shifts to measure high throughput or the number of tasks completed per unit of time.

d) Three New Computing Paradigms:

- With the introduction of SOA, Web 2.0 services become available.
- Advances in virtualization make it possible to see the growth of Internet clouds as a new computing paradigm.
- The maturity of radio-frequency identification (RFID), Global Positioning System (GPS) and sensor technologies has triggered the development of Internet of Things (IOT)

e) Computing Paradigm Distinctions:

- In general, distributed computing is the opposite of centralized computing. The field of parallel computing overlaps with distributed computing and cloud computing overlaps with distributed, centralized and parallel computing.

→ Centralized computing is a paradigm by which all computer resources are centralized in one physical system. All resources are fully shared and tightly coupled within one integrated OS.

→ Parallel computing is a computing paradigm in which all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory

→ Distributed computing is a paradigm which consists of multiple autonomous computers, each having its own private memory, communicating through message passing

→ cloud computing is an Internet cloud of resources; can be either centralized or distributed computing system. The cloud applies parallel or distributed computing, or both. Clouds can be built with physical or virtualized resources.

Q2. Give a brief information on multicore CPU, Many-core GPU architectures and multithreading technologies.

→ • Multicore CPUs may increase from the tens of cores to hundreds or more in the future.
• But the CPU has reached its limit in terms of exploiting massive DLP due to afforementioned memory wall problem
• This has triggered the development of many-core GPU's with hundreds or more thin cores.
• Now, x-86 processors have been extended to serve HPC and HTC systems in some high-end server processors.

- Many RISC processors have been replaced with multicore x-86 processors and many-core GPUs in Top 500 systems.
- The GPU also has been applied in large clusters to build supercomputers in MPPs.
- The GPU also has been applied in large clusters, In the future, the processor industry is also keen to develop asymmetric or heterogeneous chip multiprocessors that can house both fat CPU cores and thin GPU cores on same chip.

Multithreading technologies:

- There are five processor categories with five independent threads of instructions to four pipelined data paths (functional units)

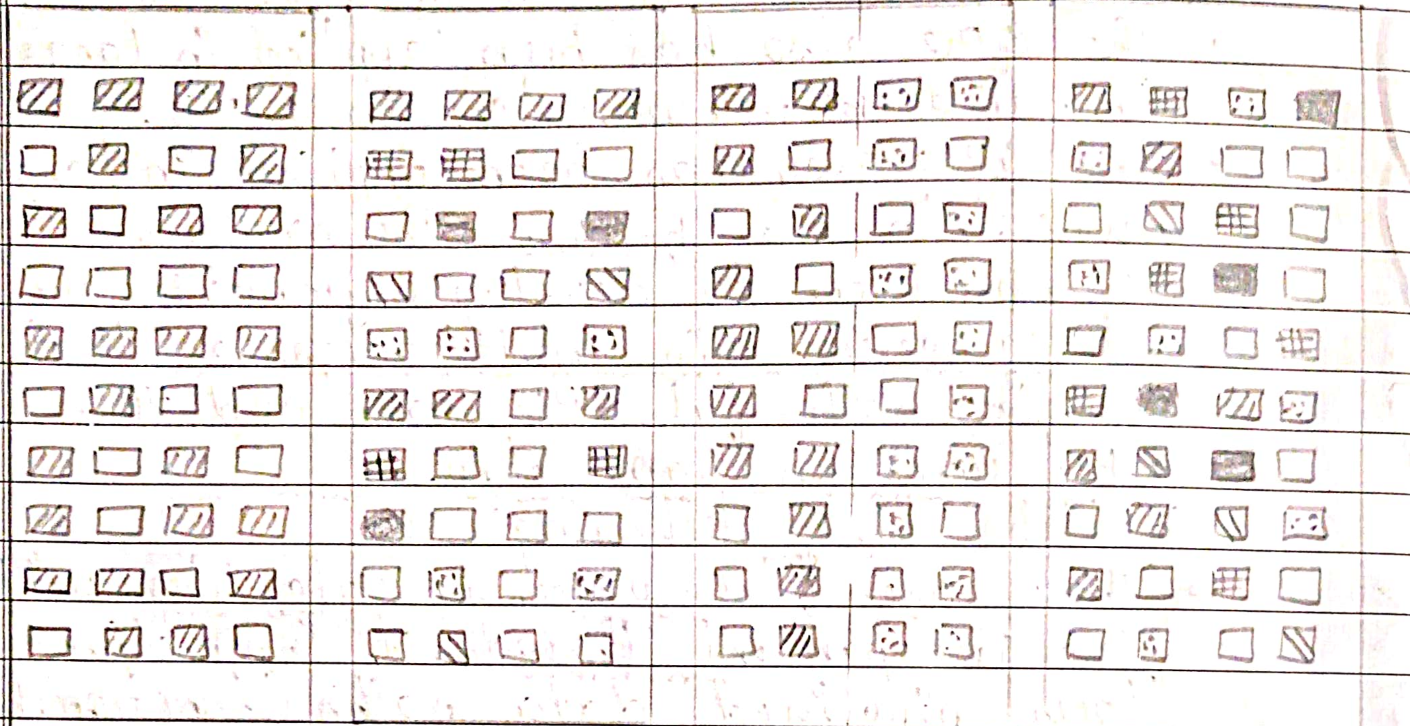
a) The superscalar processor is a single-threaded with four functional units. Only instructions from same thread are executed in this type of processor

b) The fine-grain multithreaded processor switches the execution of instructions from different threads per cycle.

c) The course-grain multithreaded processor executes many instructions from the same thread for a quite few cycles before switching to another thread.

d) The dual-core processor is a two processing cores, each a single-threaded two-way superscalar processor.

e) The multicore simultaneous multithreaded processor allows simultaneous scheduling of instructions from different threads in same cycle

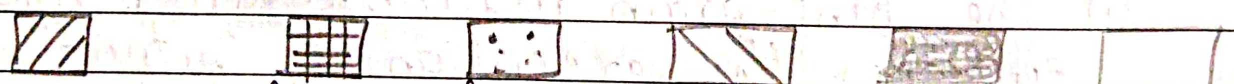


(a) 4-issue superscalar processor

(b) Fine-grain multithreaded processor

(c) Dual-core (2-processor (MP)

(d) Simultaneous multithreaded processor

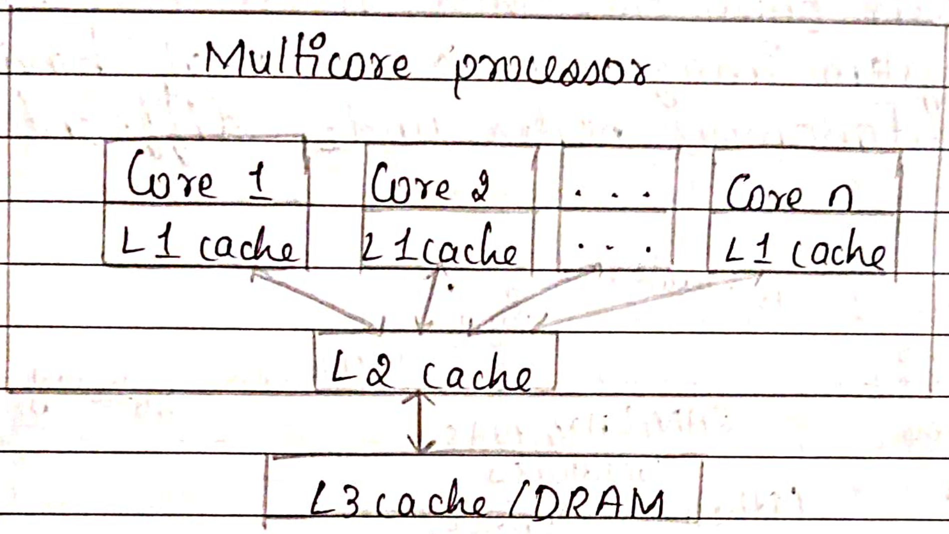


Thread 1 Thread 2 Thread 3 Thread 4 Thread 5 Idle slot

- Instructions from different threads are distinguished from specific shading patterns for instructions from five independent threads
- These execution patterns closely mimic an ordinary program.
- The blank squares correspond to no available instructions for an instruction data path at a particular processor cycle.

Advances in CPU processors:

- Today, advanced CPUs or microprocessor chips assume a multicore architecture with dual, quad, six or more processing cores. These processors exploit parallelism at ILP and TLP levels.
- Both multicore CPU and many-core GPU processors can handle multiple instruction threads at different magnitudes.



Architecture of typical modern multicore processor.

Q3. Explain the clusters of Cooperative Computers.

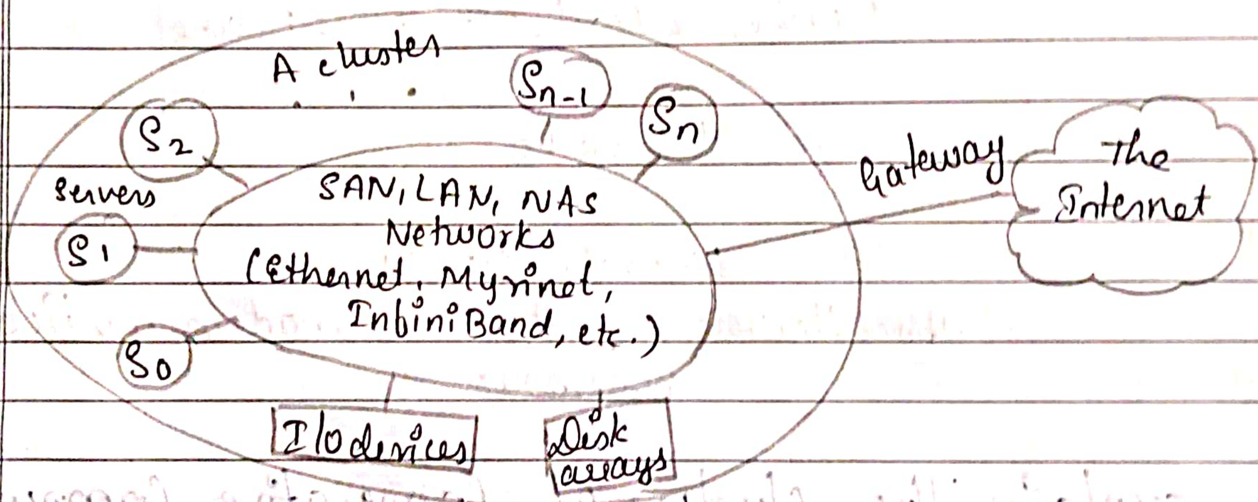
→ • A computing cluster consists of interconnected stand-alone computers which work cooperatively as a single integrated computing resource.

Cluster architecture:

- It is an architecture of typical server cluster built around low latency, high-bandwidth interconnection network. This network can be as simple as SAN or a LAN.
- Through hierarchial construction using a

SAN, LAN or WAN, one can build scalable clusters with an increasing number of nodes.

- The cluster is connected to Internet via a Virtual private network (VPN) gateway. The gateway IP address locates the cluster.
- The system image of computer is decided by the way OS manges the shared cluster resources.
- All resources of a server node are manged by their own OS. Thus, most clusters have multiple system images as a result of having many autonomous nodes under different OS control



Single - System Image:

- Single-System Image (SSI) is an ideal cluster merging multiple system images into one.
- Cluster designers desire a cluster operating system or some middleware to ssupport SSI at various levels, including sharing of CPUs, memory and I/O across all cluster nodes.
- An SSI is an illusion created by software

or hardware that presents a collection of resources as one integrated, powerful resource.
- SSI makes cluster appear like a single machine to the user.

Hardware, Software and Middleware support:
- Clusters exploring massive parallelism are commonly known as MPPs. Almost all HPC clusters in the Top 500 list are also MPPs.
- The building blocks are computer nodes, special communication software such as PVM or MPI and a network interface card in each node.
- Most clusters run under the Linux OS. The computer nodes are interconnected by a high-bandwidth network.
- Special cluster middleware supports are needed to create SSI or high availability (HA)
- Users may want all distributed memory to be shared by all servers by forming distributed shared memory (DSM).
- Many SSI features are expensive or difficult to achieve at various cluster operational levels.
- Instead of achieving SSI, many clusters are loosely coupled machines. Using virtualization, one can build many virtual clusters dynamically.

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.

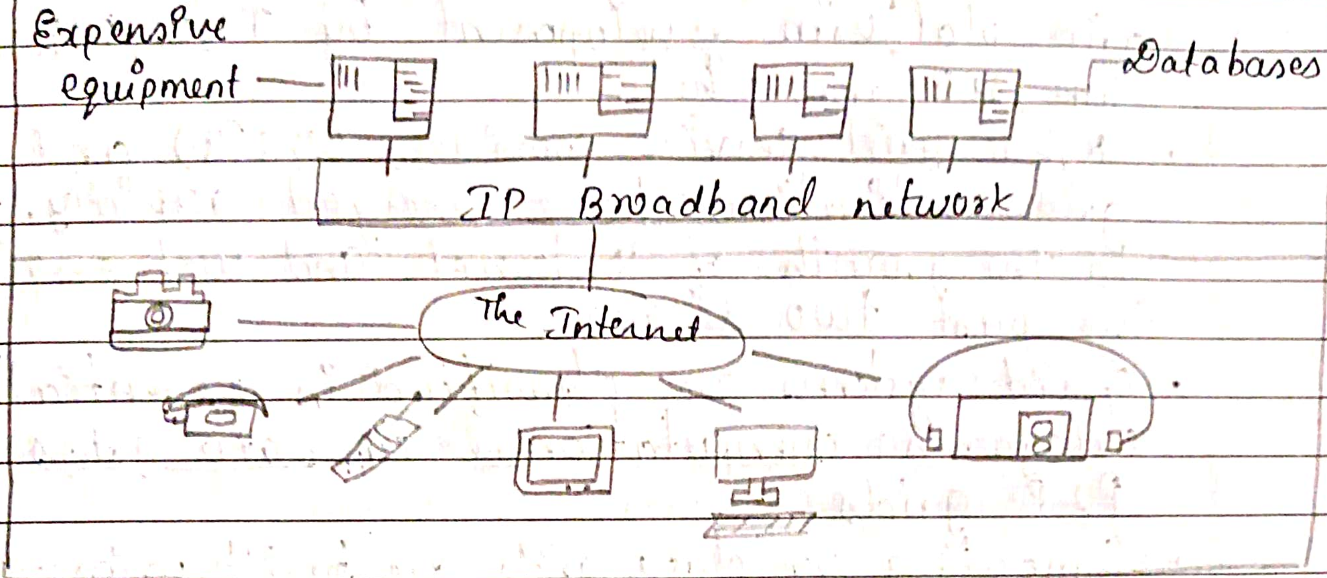
- Without this middleware, cluster nodes cannot work together effectively to achieve cooperative coopera computing.
- The software environments and applications must rely on middleware to achieve high performance.
- The cluster benefits come from scalable performance, efficient message passing, high system availability, seamless fault tolerance and cluster-wide job management.

04. Differentiate between
a) Different Grid families

→ • Grid computing is envisioned to allow close interaction among applications running on distant computers simultaneously
- The evolution from Internet to web and grid services is certainly playing a major role in this growth.

* Computational grids -
- A computing grid offers an infrastructure that couples computers, software/middleware, special instruments and people and sensors together.
- The grid is often constructed across LAN, WAN or Internet backbone networks at a regional, national or global scale.
- They can also be viewed as virtual platforms to support virtual organizations.

- The computers used in a grid are primarily workstations, servers, clusters and supercomputers.



- The resource sites offer complementary computing resources, including workstations, large servers and Linux clusters to satisfy a chain of computational needs.
- The grid is built across various IP broadband networks including LANs and WANs already used by enterprises or organizations over the Internet
- The grid is presented to users as integrated resource pool.
- At the server end, the grid is a network. At the client end, we see wired or wireless terminal devices.
- The grid integrates the computing, communication, contents and transactions as rented services.

* Grid Families -

- Grid technology demands new distributed computing models, software/middleware support, network protocols and hardware infrastructures.

- National grid projects are followed by industrial grid platform development by IBM, Microsoft, Sun, HP and others.
- New grid service providers (GSPs) and new grid applications have emerged rapidly, similar to the growth of Internet and web services in the past two decades.
- Grid systems are classified in essentially two categories: computational or data grids and P2P grids
- Computing or data grids are built primarily at the national level.

b) Grids versus clouds

→ • The boundary between grids and clouds are getting blurred in recent years.

- For web services, work-flow technologies are used to coordinate or orchestrate services with certain specifications used to define critical business process models such as two-phase transactions.
- In all the approaches of workflow, one is building a collection of services which together tackle all or part of a distributed computing problem
- In general, a grid system applies static resources, while a cloud emphasizes elastic resources.
- For some researchers, the differences between grids and clouds are limited only in dynamic resource allocation based on virtualization and autonomic

computing.

- One can build a grid out of multiple clouds.
- This type of grid can do better job than a pure cloud, because it can explicitly support negotiated resource allocation.
- Thus one can may end up building with system of systems: such as a cloud of clouds, a grid of clouds, a cloud of grids or inter-clouds.

05. Explain cloud-computing as on-demand computing paradigm with a neat diagram.

→ • Cloud Computing as on-demand computing paradigm resolves or relieves us from problems of constant system maintainance, poor utilization and increasing costs associated with hardware/software upgrades.

- There are three services provided by cloud:

a) Infrastructure as a Service (IaaS)-

- This model puts together infrastructures demanded by users-namely servers, storage, networks and data center fabric
- The user can deploy and run on multiple VMs running guest OSes on specific applications.
- The user does not manage or control the underlying cloud infrastructure, but can specify when to request and release the needed resources.

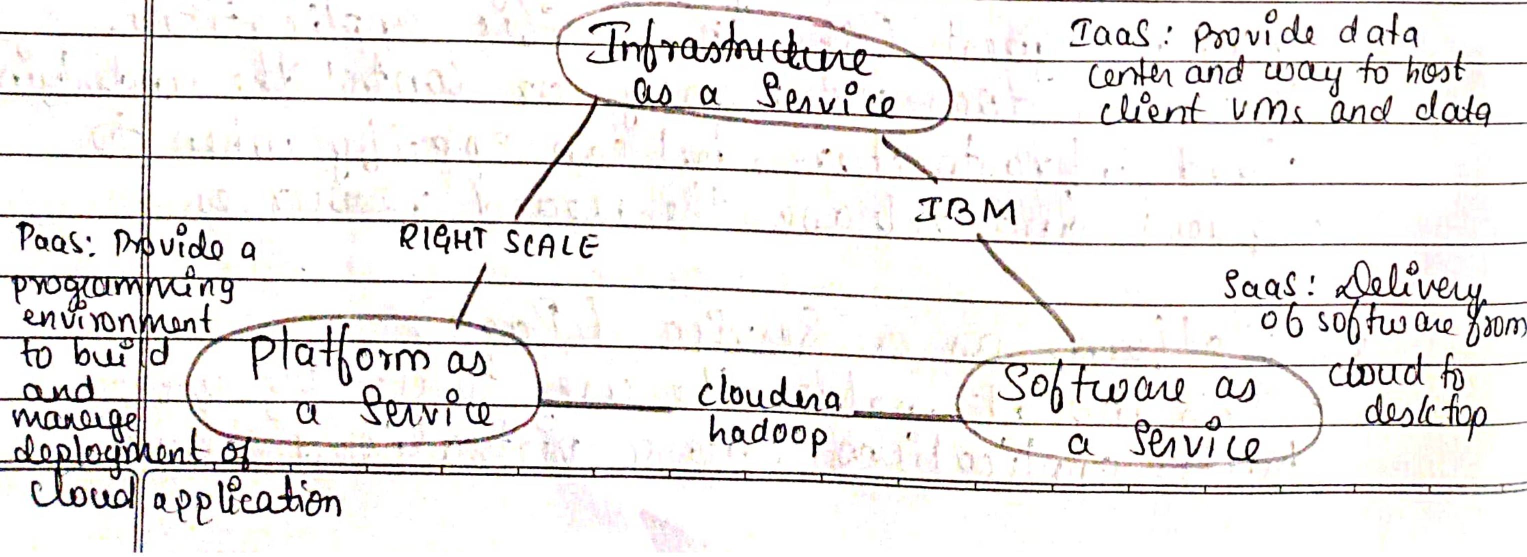
b) Platform as a Service (PaaS)-

- This model enables the users to deploy user-built applications onto a virtualized cloud platform.

- PaaS includes middleware, databases, development tools and some runtime support such as Web 2.0 and Java.
- The platform includes both hardware and software integrated with specific programming interfaces
- the provider supplies the API and software tools. The user is freed from managing the cloud infrastructure.

c) Software as a Service (SaaS) -

- This refers to browser-initiated application software over thousands of paid cloud customers.
- The SaaS model applies to business processes, industry applications, consumer relationship management (CRM), human resources (HR) and collaborative applications
- On the customer side, there is no upfront investment in servers or software licensing.
- On the provider side, costs are rather low, compared with conventional hosting of user applications.



- Internet clouds offer four deployment modes: private, public, managed and hybrid
- These modes demand different levels of security implications
- Eight reasons to adapt cloud for upgraded Internet applications and web services:
  - → Desired location in areas with protected space and higher energy efficiency
  - → Sharing of peak-load capacity among a large pool of users
  - → Seperation of infrastructure maintenance duties from domain-specific application development.
  - → Significant reduction in cloud computing cost
  - → Cloud computing programming and application development
  - → Service and data discovery and content/service distribution
  - → Privacy, security, copyright and reliability issue
  - → Service agreements, business models and pricing policies

06. Explain the layered architecture of Web Services and Grids.

→ • The entity interfaces correspond to Web Services Description Language (WSDL), Java method and CORBA interface definition language (IDL) specifications in these example distributed systems

• The communication systems, SOAP, RMI and IIOP, support features including particular message patterns, fault recovery and specialized routing.

- In case of fault tolerance, the features in Web Services Reliable Messaging (WSRM) framework mimic OSI layer capability modified to match different abstractions at entity levels.
- Security is a critical capability that either uses or reimplements the capabilities seen in concepts such as Internet Protocol Security and secure sockets in OSI layers.
- The CORBA Trading Service, UDDI, LDAP and ebXML are other exaples of discovery and information services
- Managements services include service state and lifetime support
- The latter can have performance advantages and offers a "shared memory" model allowing more convenient exchange of information.
- The distributed model has two critical advantages: higher performance and cleaner separation of software functions.

Layers	
Application specific services/grids	Higher level services
Generally useful services & grids	
Workflow	Service context
Service management	
Service discovery & information	Service Internet
Service Internet transport → Protocol	
Service interface	
Base hosting environment	
Protocol HTTP FTP DNS ...	Bit level Internet
Presentation XDR ...	
Session SSH ...	
Transport TCP/UDP ...	
Network IP	

Q7. Write a note on the Evolution of SOA.

→ • SOA (Service-oriented architecture) has evolved over the years and applies to building grids, clouds, grids of clouds, clouds of grids, clouds of clouds and system of systems.

• A large number of sensors provide data-collection services, denoted as SS (sensor service). A sensor can be ZigBee device, a Bluetooth device, a WiFi access point, a personal computer among other things.

• Raw data is collected by sensor services. All the SS devices interact with large or small computers, many forms of grids, databases, the computer cloud, the storage cloud, the discovery cloud, the filter cloud, and so on.

• Filter services (fs) are used to eliminate unwanted raw data, in order to respond to specific requests from the web, the grid or web services

• A collection of filter services forms a filter cloud.

• SOA aims to search for, or sort out, the useful data from massive amounts of raw data items.

• Processing this data will generate useful information and subsequently knowledge for our daily use.

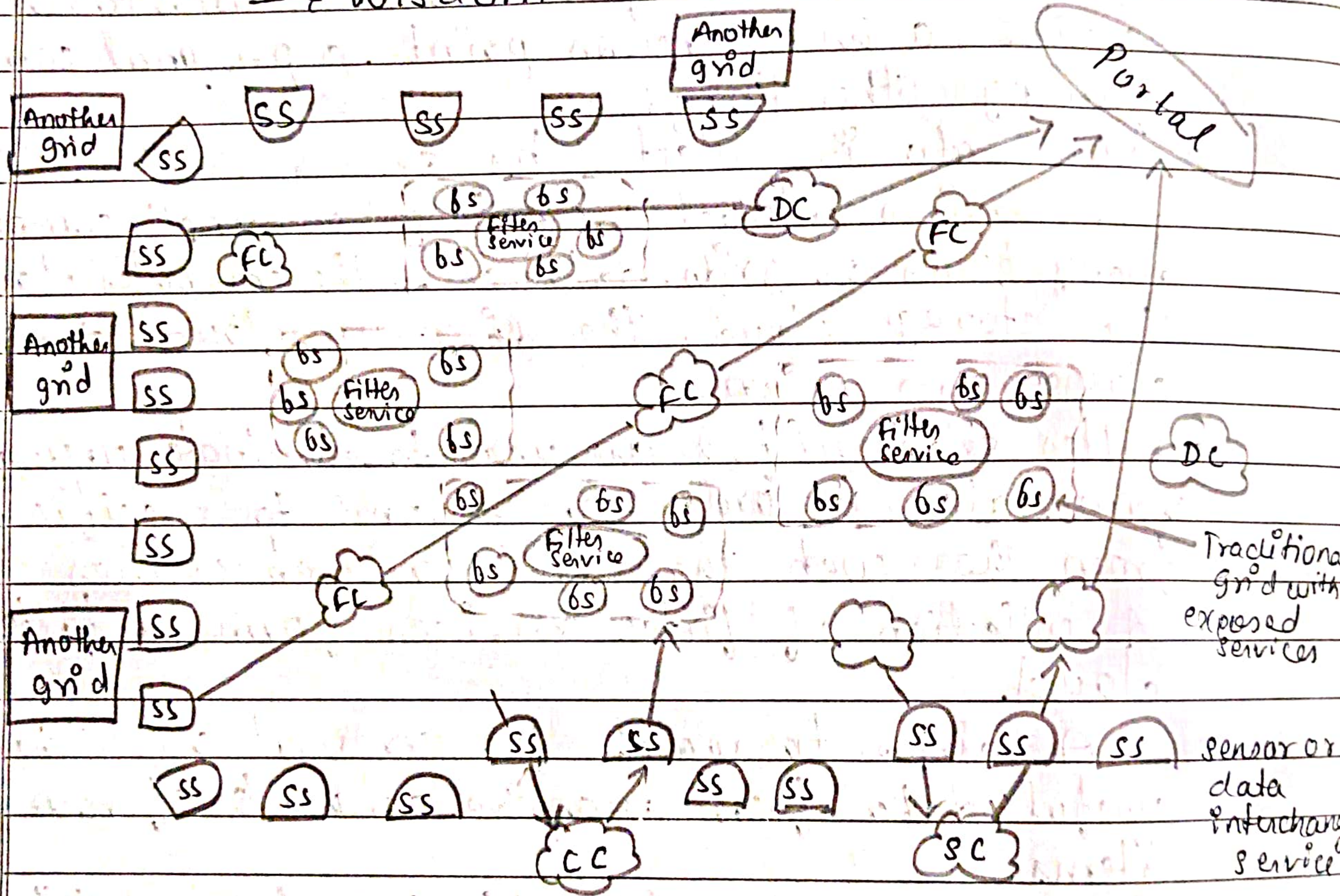
• Finally, we make intelligent decisions based on both biological and machine wisdom.

• Most distributed systems require a web interface or portal.

• For raw data collected by a large number of sensors to be transformed into useful information or knowledge, the data stream may go through a sequence of compute, storage, filter and discovery clouds.

- finally, the inter-service messages converge at the portal, which is accessed by all users.

Raw data → Data → Information → Knowledge → Wisdom → Decisions



SS → sensor service   FC → filter cloud
fs → filter service   DC → Discovery cloud
SC → storage cloud   CC → compute cloud

08. Discuss the performance metrics and scalability analysis.

→ • Performance metrics are needed to measure various distributed systems.

a) Performance metrics-

- In a distributed system, performance is attributed to a large number of factors.
- System throughput is often measured in MIPS, Tflops or TPS.
- 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 runtime support system used.

b) Dimensions of Scalability -

- 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.

→ Software scalability -

- This refers to upgrades in OS or compilers, adding mathematical and engineering libraries, porting new application software and installing more user-friendly programming environments.

→ Application scalability -
- This refers to matching problem size scalability with machine size scalability.
- Instead of increasing machine size, users can enlarge problem size to enhance system efficiency.

→ Technology scalability -
- This refers to system that can adapt to changes in building technologies, such as component and networking technologies.
- When scaling a system design with new technology one must consider three aspects : time, space and heterogenity.

c) Amdahl's law -
- Amdahl's law states that speedup factor of using n-processor system over the use of single processor is expressed by:

$$\text{Speedup} = S = T / [\alpha T + (1-\alpha) T / n]$$
$$= 1 / [\alpha + (1-\alpha) / n]$$

where,
T → total execution time in minute
α → fraction of code must be executed sequentially
$(1-\alpha)$ → parallel execution
n → no of processors.
- The maximum speedup is achieved only if sequential bottleneck α is reduced to zero or code is fully parallelizable with $\alpha = 0$.

d) Gustafson's law -
- Let W be the workload in given program.

- When using n-processor system, the user scales workload to $W' = \alpha W + (1-\alpha) nW$
- This scaled workload W' is essentially the sequential execution time on single processor
- The parallel execution time of scaled workload W' is define by scaled-workload speedup as:

$$S' = W'/W$$
$$= [\alpha W + (1-\alpha)nW]/W$$
$$= \alpha + (1-\alpha)n$$

This speedup is known as Gustafson's law.

e) Problem with fixed workload -
- To execute a fixed workload on n processors, parallel processing may lead to system efficiency defined as:

$$E = S/n$$
$$= 1/[\alpha n + 1 - \alpha]$$

- Very often the system efficiency is rather low, especially when the cluster size is very large.

Q9. Discuss implementation levels of virtualization
→
- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in same hardware machine.
- After virtualization, different user applications managed by their own operating systems can run on same hardware, independent of host OS. This is done by adding additional software called virtualization layer.

- This virtualization layer is known as VMM (Virtual machine monitor) or hypervisor.
- The virtualization software creates abstraction of VMs by interposing a virtualization layer at various levels of computer system.
- The virtualization layers are as follows:

Application level JVM/.NET CLR/Panot
Library (user-level API) level WINE/WABI/vCUDA
Operating System level Jail/Virtual Environment
Hardware Abstraction level VMware/Virtual PC/Xen/L4/User mode Linux/Denali
Instruction set architecture level Bochs/Crusoe/Dynamo

a) Instruction set architecture level-

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine
- With this approach, it is possible to run a large amount of legacy binary code written for various processor.

- The basic emulation method is through code interpretation. An intrepreter program interprets the source instructions to target instructions one by by one.
- For better perfomance, dynamic binary transalation is desired, which translates basic blocks of dynamic source instructions to target instructions.
- A virtual instruction set architecture (V-ISA) thus requires adding processor-specific software translation layer to compiler.

b) Hardware Abstraction Level -
- Hardware-level virtualization is performed right on top of bare hardware.
- This approach generates a virtual hardware environment for VM and process manages underlying hardware through virtualization.
- The idea is to virtualize computer's resources, such as its processors, memory and I/O devices, and the intentio is to upgrade hardware utilization rate by multiple users concurrently

c) Operating System Level -
- This refers to abstraction layer between traditional OS and user applications
- OS level virtualization creates isolated containers on a single physical server and OS instances to utilize hardware and software in data centers. The containers behave like real servers

- OS-level virtualization is commonly used to create virtual hosting environments.

d) Library support level-
- Most applications use API's exported by user-level libraries rather than using lengthy system calls by OS.
- Virtualization with library interfaces is possible by controlling the communication link between applications and rest of system through API hooks.

e) User-application level-
- Virtualization at application level virtualizes an application as VM. On traditional OS, application runs as process.
- Therefore, application level virtualization is also called as process-level virtualization.
- The most popular approach is to deploy high level language (HLL) VMs. Any program written in HLL and compiled for this VM will be able to run on it
- JVM and .NET CLR are best examples of this class of VM.
- Other form of application-level virtualization are known as application isolation, application sandboxing or application streaming.
- The process involves wrapping application in a layer that is isolated from host OS and other applications.