High performance Computing :-

→ It is the use of Super Computer & parallel processing techniques for solving Complex computational problems. HPC technology focuses on developing parallel processing algorithms & systems by incarporating both administration & parallel Computational techniques.

→ It is typically used for solving advanced problems & performing rusarch activities through Computer Modeling, Simulation & Analysis.

Overview of Grid Technology :-

→ Grid Computing is a group of Computers physically connected (over a n/w or with Internet) to perform a dedicated tasks together, such as analysing E-commerce data & solve a complex problem

→ The main Concept is to make numerous autonomous m/c's that may be in different physical ~~Condition~~ Locations, act like they are Single virtual m/c. (machine)

→ In grid Computing, end users are given access to shared storage capacity & use Computing power from their desktop & shared Computer in the grid.

Abhishek.

→ Processor Architecture that combines computer resources from various domains to reach a main objective the computers on the n/w can work on a task together, thus functioning as a super computer.

→ It is ideal for applications that require large amount of computational power, that is why it is mostly used by research collaborations referred to virtual organization

→ It enables applications to be easily scaled since additional computers can be added to the grid.

→ Since the technology being used are open source, trust & Transparency is encouraged

→ Allows the sharing of computer resources accross n/w, Thus increasing the computational power available to program & reducing the no. of computer needed by the organization.

→ Grids are a form of super virtual computers that solve a particular application. The grid size may vary from small to large enterprise n/w.

→ Grid is a type of parallel & distributed

Overview of Grid Computing Technology:-

→ Grid computing is a group of computers physically connected (over a n/w or with internet) to perform a dedicated tasks together, such as analysing e-commerce data & solve a complex problem. The main concept is to make numerous autonomous m/cs that may be in different physical locations act like they are a single virtual m/c.

→ Flexible, secure, coordinated resource sharing among dynamic collection of individuals institutions & resource.

→ In grid computing, end users are given access to shared storage capacity & use computing power from their desktop & shared computer in the grid.

→ Processor architecture that combines computer resources from various domains to reach a main objective. The computers on the n/w can work on a task together, thus functioning as a super computer.

→ It is ideal for applications that require large amt. of computational power, that is why it is mostly used by research collaborations referred to virtual organizations.

→ It enables applications to be easily scaled since additional computers can be added to the grid.

→ Since the tech. being used are open source, trust & transparency is encouraged.

→ Allows the sharing of computer resources across n/w, thus increasing the computational power available to program & reducing the no. of computer needed by the organization

Grid Computing :-

* It is a distributed computing system where a grou of computers are connected. To create and wor as one large virtual computing power, stora database, application & service.

Comparison :-

Serial	Parallel	Grid
→ Fetch/store	Fetch/store	Fetch/store
→ Compute	→ Compute /	Discovery of resources
	→ cooperative game.	→ Interaction with remote application.
		→ Authentication / Authorization
		→ Security
		→ Compute/ Communicate

Cluster Computing :-

It is a form of computing in which bunch of computers (often called nodes) that are connected through a LAN, so that they behave like a single Machine.

A computer cluster help to solve complex operations more efficiently with much faster processing speed, better data integrity than a single computer and they only used for mission-critical applications.

→ The clustering methods have identified as HPC IAAS, HPC PAAS, that are more expensive and difficult to setup & maintain than a single computer.

Some of the critical applications of cluster computers are Google Search Engine, petroleum Reservoir simulation, Earthquake Simulation, Weather forecasting. Image Rendering

---

Cluster can be classified into two category.

open & close

open :- All nodes in O.C are need IP, and thatar accessible throught internet / web, that cause more security concern. More flexible

close :- On the other hand close cluster are hide behind the gateway node & provide security. Need less IPs

Good for compiting tasks.

Shinde

Cluster Computing :- It is a form of computing in which bunch of computers (called nodes) that are connected through a LAN so that, they behave like a single m/c. A computer cluster help to solve complex operations more efficiently with much faster processing speed, better data integrity than a single computer & they only used for mission-critical applications.

Advantages :- Cost Efficiency Expandability
Processing speed High availability of resources

Most widely used cluster computers are Petroleum Reservoir Simulation, Google Search Engine, Earthquake Simulation, Weather forecasting.

→ A loosely connected or tightly coupled computers that effort together so that they can be worked as a single system by the end user.

→ Amazon cloud, MS Azure, Google cloud provides Grid computing capabilities

(Grid)

Applications :- Distributed SuperComputing
High throughput computing
On-demand "
Data-intensive "
Collaborative "

Types of cluster computing :-

(1) Load-balancing cluster :- As the name implies, this system is used to distribute workload across multiple computers. That system distributes the processing load as possible across a cluster of computers.

(2) High availability (HA) cluster :- HA cluster were the bunch of computers that can suitably utilise for redundant operations in the event of nodes failure in cluster computing.

(3) High performance (HP) :- This computer H/W ing methodology use supercomputer & cluster computing to solve advanced computation problems.

Advantages :-

(1) Cost efficiency
(2) Processing speed
(3) Expandability, handle much power at a time, adding more computers
(4) High availability of resource.

Disadvantages

(1) Cost is higher since the the cluster needs good hardware & a design, it will be costly comparing to a non-clustered server management design.

(2) Since clustering needs more servers & h/w to establish one, monitoring & mainterance is hard. Thus increase the infrastructure.

Peer-to-Peer Computing :- P2P

The peer to peer computing architecture contains nodes that are equal participants in data sharing. All tasks are equally divided b/w all the nodes. The nodes interact with each other as required as share resources.

Characteristics :-

→ peer to peer n/w are usually formed by groups of a dozen or less computers. These computers all store their data using individual security but also share data with all the other nodes.

→ The nodes in peer to peer n/w both use resources & provide resources. So, if the node increase, then the resource sharing capacity of the n/w increases. This is different than client server n/w where the server gets overwhelmed if the node increase.

→ Most modern OS such as Windows & Mac contain s/w to implement P2P n/w.

Advantages :-

→ Each computer manages itself. So, the n/w is quite easy to set up & maintain.

→ Cost of server is saved.

→ Easy to scale & add more nodes.

→ None of the nodes are dependent on other.

Disadvantages :-

→ Difficult to backup the data cz of no central server & data is stored in different systems.

→ Difficult to provide overall security as independent & contain own data. (each system).

Internet Computing:

Internet Computing is the foundation on which e-business runs. It is the only architecture that can run all facets (aspects) of business, from supplier collaboration and merchandise purchasing, to distribution & store operations, to customer sales & services. It is the only architecture that supports all information flows & processes over the Internet - providing access to all applications. With Internet computing, all a user need is a standard web browser & security clearance.

→ On-demand n/w access to a shared pool of configurable computing resources (e.g. n/w, servers, storage, applications & services).

→ This Internet model promotes availability & is composed of five essential characteristics, three service models & four deployment models.

Essential Characteristics:

→ On-demand Self Service:- As needed automatically without requiring human interaction with each service's provider.

→ Broad n/w Access:- Capabilities are available over the n/w & accessed through standard mechanism that promote use by thin or thick client platforms (phones, laptops etc)

→ Resource Pooling:- Resources are pooled to serve multiple consumers using a multi-tenant model, with different physical & virtual resources dynamically assigned & reassigned according to consumer demand. Example of resources include storage, processing, memory, n/w, bandwidth & virtual m/c.

→ ~~Rapid Elasticity~~:

There are three tiers provide the benefit of centralized data that supports a unified view of the retailer's financial, human resources, inventory, logistics, trading partner & customer info. The business layer (logic) accesses & transacts the data. The user interface is simple, ~~non~~- web browser. No complexity resides on the user's device. Universal access to any person with a browser.

→ support scalability at much lower cost.

→ Internet is a collection of information. It is a service which is on top of the Internet computing.

→ Internet computing is infrastructure (architecture).

Grid Computing Models :-

(1) Distributed Super- Computing :-

⇒ Aggregate computational resources to tackle problems that cannot be solved by a Single System.

(2) High-Throughput Computing :-

⇒ Schedule large numbers of independent tasks to exploit unused CPU cycles.

(3) On demand Computing :-

→ use Grid capabilities to meet short-term requirements for resources that cannot conveniently be located locally.

(4) Data - Intensive Computing :- (reliability & availability of data) terabyte & petabyte

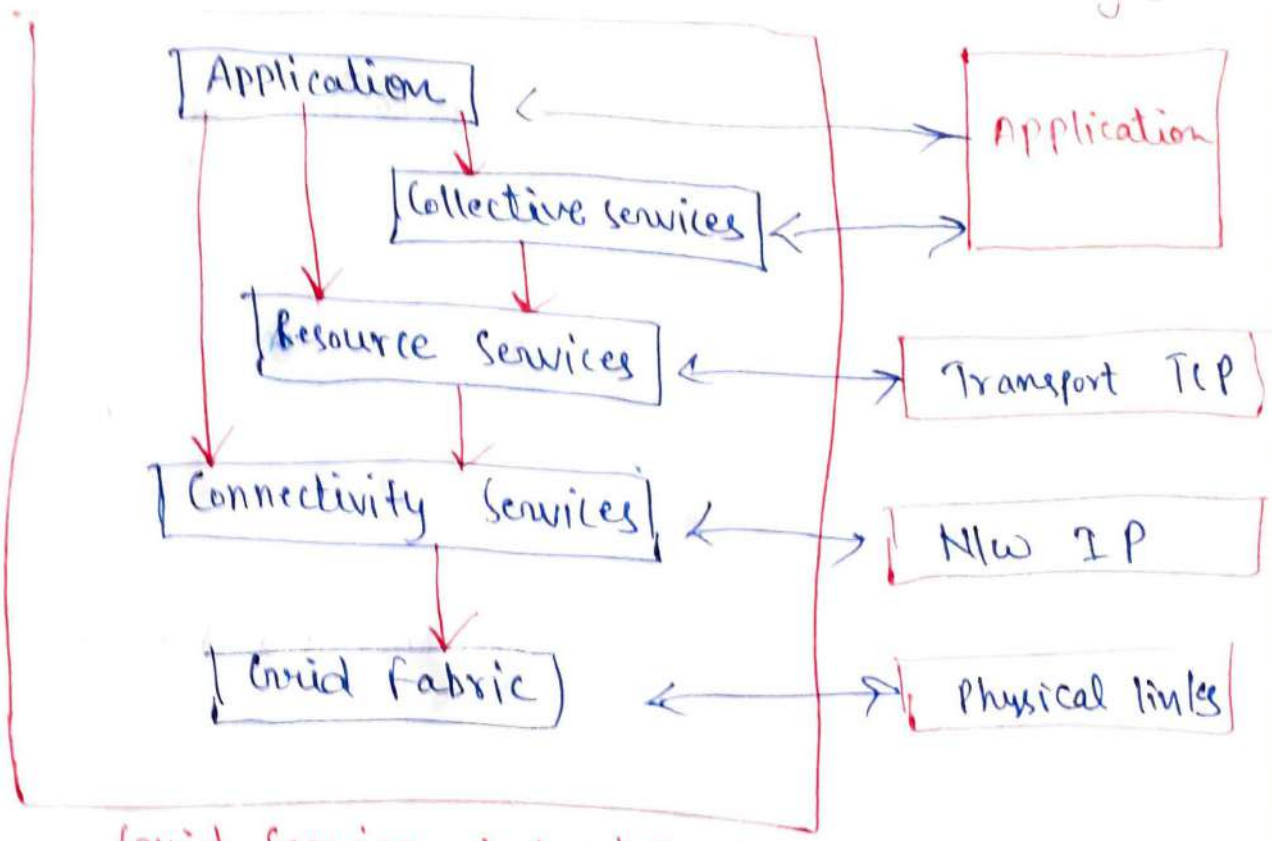
→ Synthesize data in geographically distributed repositories. (Process large volume of data). (big data)

(5) Calloborative Computing :- bridge b/w locations.

* Enable shared use of data archives & Simulations.

Diverse collection of information technologies.

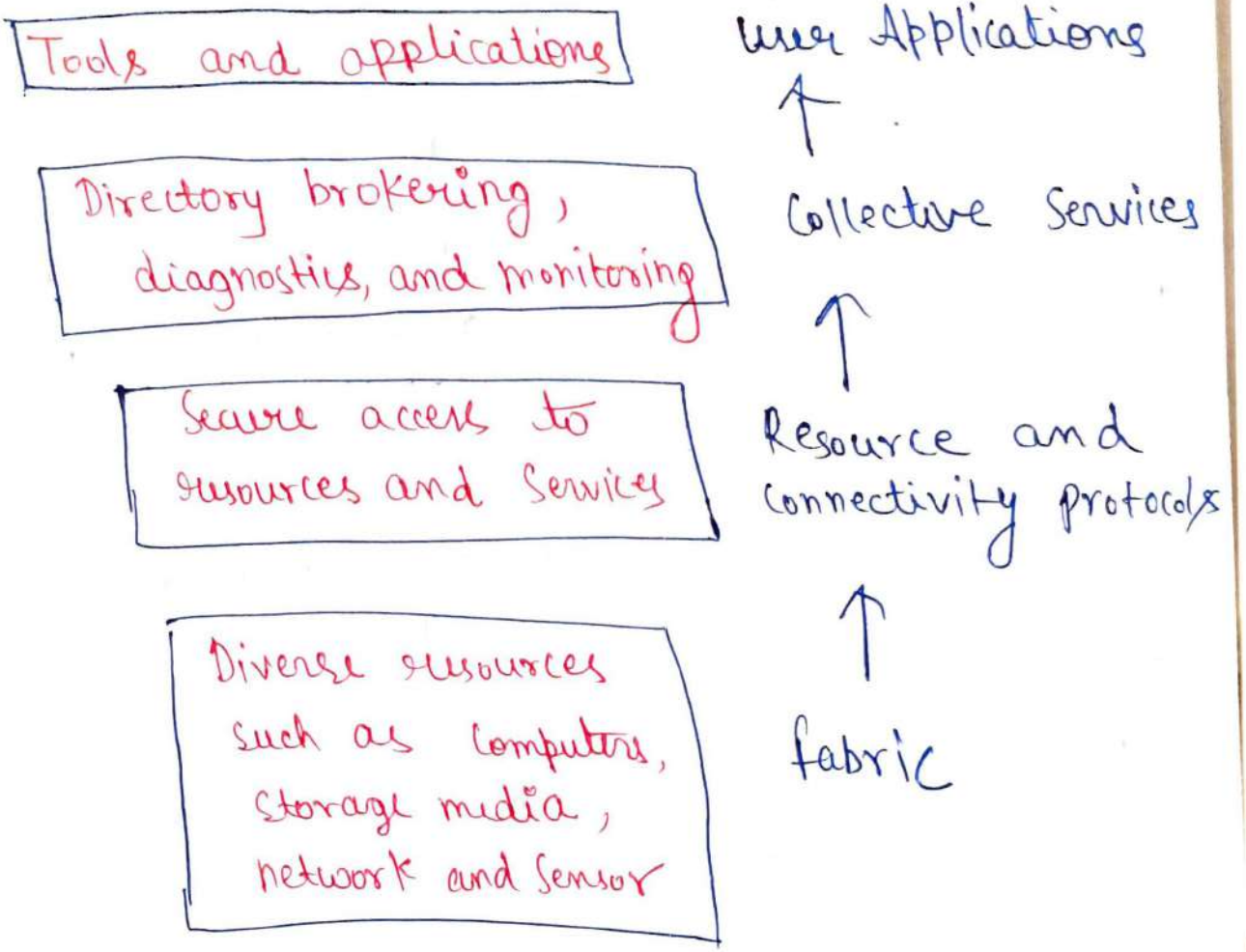
→ Improve workforce productivity & creativity by enabling readily access with each other.



Grid Service protocol stack

layer grid Service protocols and their relationship with the Internet Service protocols

Grid Architecture :-



Working of layers :-

fabric :- The lowest layer Job is used to make a common interface on all possible kinds of resources available. Access by higher layers is granted via standardized processes.

Resources and Connectivity protocols :- The connectivity layer defines the basic communication and authentication protocols which are needed by the grid. While the communication protocols allow the exchange of files between different resources connected by the first layer, the authentication protocols allow to communicate confidentially and to ensure the identity of two partners.

→ Collective Services :-

The purpose of this layer is the coordination of multiple resources. Access to these resources doesn't happen directly but merely via the the underlying protocols and interfaces.

→ User Application :-

To this layer belong all those applications which are operating in the environment of a virtual organization. Jobs of the lower layers get called by application and can use resources transparently.

Types of Grid :-

Computational Grid :- These grids provide secure access to huge pool of shared processing power suitable for high throughput applications & computation intensive computing.

Data grid :- Data grids provide an infrastructure to support data storage, data discovery, data handling, data publication, and data manipulation of large volumes of data actually stored in various heterogeneous databases and file systems.

N/w grid :- A N/w grid provides fault-tolerant and high-performance communication services.

Desktop grid :- Desktop grid computing is a form of distributed computing in which an organization uses its existing computers to handle long-running computational tasks.

Advantages of High Performance Computing :-

→ **Speed**: With many processing cores, often with high performance specs, than a typical desktop or laptop, HPC systems can offer significant speed up.

→ **Volume**:- Many HPC systems have both the processing memory (RAM) & disk storage to handle very large amt. of data. TeraB. of RAM & PB of storage are available.

→ **Efficiency** :- Many HPC systems operate a pool of resources that are drawn on by many users. In most cases, when pool is large & diverse enough, the resources on the systems are used almost constantly.

→ **Cost**:- Bulk purchasing & govt. funding mean that the cost to the research community for using these system is significantly less.

→ **Convenience**:- less time taking for calculations.

Disadvantages:- Complex
Energy Consumption
Requires Maintainance

Open Grid System Architecture

Chapter-2

Unit-2

OGSA :- The OGSA describes an architecture for a service-oriented grid computing environment for business and scientific use, developed within the Global Grid forum (GGF).

* OGSA is based on several other Web Service technologies, notably WSDL and SOAP.
* OGSA is a distributed interaction and computing architecture based around services, assuring interoperability on heterogeneous systems so that different types of resources can communicate and share information.
* OGSA has been described as a refinement of the emerging Web services architecture, specifically designed to support Grid requirements.
* OGSA has been adopted as a Grid architecture by a number of grid projects including the Globus Alliance.

SOAP →	WSDL → Web Service Description Language
Simple object Access protocol. or procedure-oriented information.	It is a format for describing N/W Services as a set of endpoints operating on message containing either document-oriented,

Open Grid Services Architecture :-

(2)

* Service - Oriented Architecture
  → Key to virtualization, discovery, composit
  local-remote Transparency.
* Leverage Industry standards :-
  → Internet, Web Services.
* Distributed Service management
  → A "Component model for Web Services"
  or a Service model for grid.
* A framework for the definition of
  Composable, Interoperable Services

OGSA defines standards for

* What Grid Services are
* What they should be capable of.
* What type of Technologies they should be
  based on.

* OGSA does not give a technical and
  detailed specification. They use WSDL.

OGSA is developed by Global,
Grid. forum. (GGF)

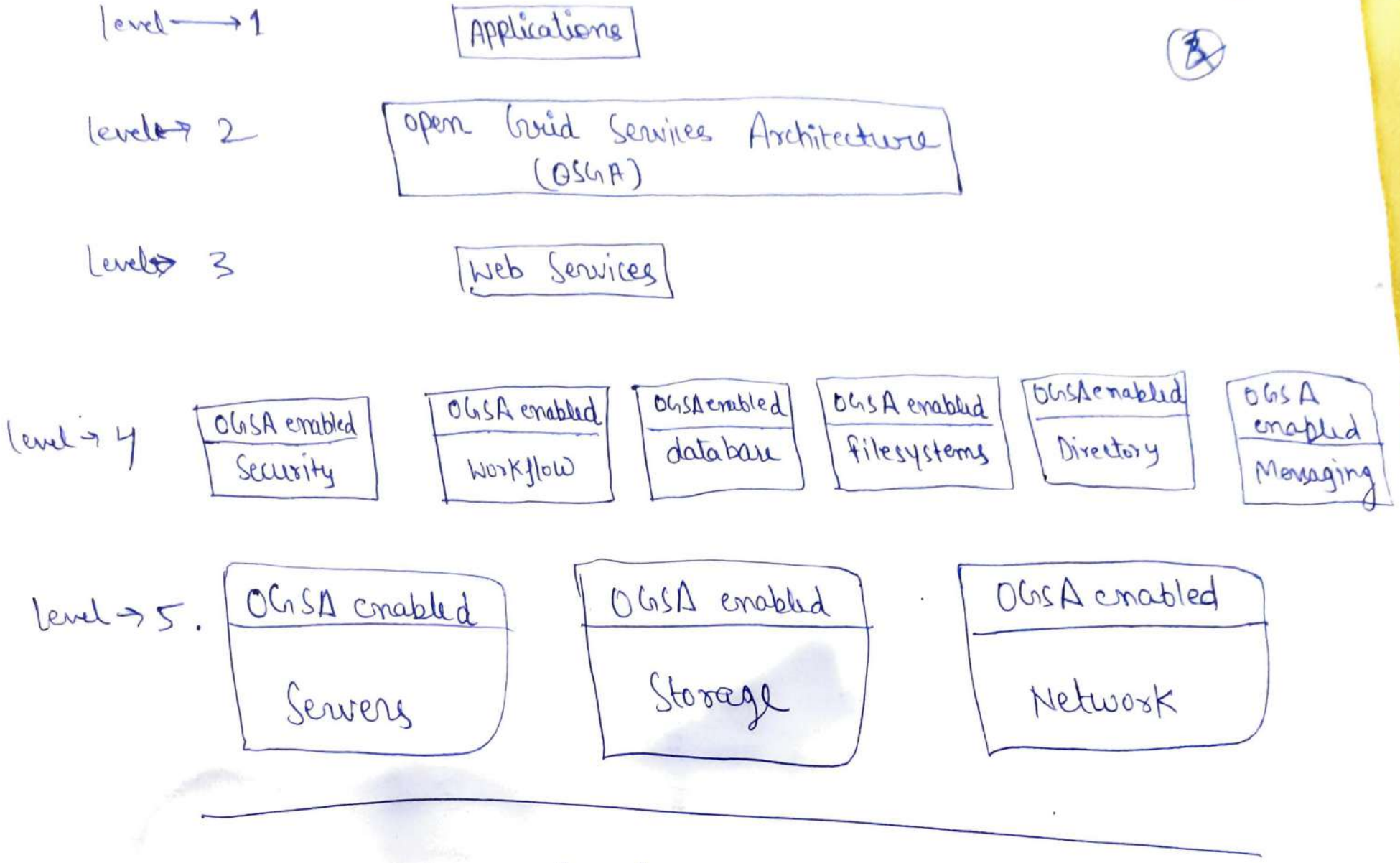
~~High Performance Application Development Environment:~~

Open Grid Service Architecture :-

Several Grid based middleware & applications, such as grid portals, search engines, data grids & authorization services have recently begun to provide services on the Grid infrastructure. The services are separate & not inter-operable. The need for integration & interoperability among the increasing no. of applications is driving grid technologies towards an open Grid architecture offering an extensible set of services that virtual organization can aggregate in various ways. To achieve this goal, OGSA defines the grid service concept, based on principles & technologies from both the Grid computing & Web services communities.

~~Gpra~~ In General, Grid environments tend to be heterogeneous & distributed, encompassing a variety of hosting environment, operating systems, devices & services provided by the various venders. In addition, Grid environments are frequently intended to be long-lived & dynamic. ~~& may therefore~~

OGSA must enable interoperability b/w such diverse, heterogeneous & distributed resources and services as well as reduce the complexity of systems (administration)



Open Grid Service Architecture.

OGSA Requirements :

→ Resource Sharing Access Organizations :

Resource sharing & utilization across administrative domains, whether different work units within an enterprise or even different institutions. Sharing resources across organizations also implies various security requirements.

→ Optimization :

It refers to techniques used to allocate resources effectively to meet consumer & supplier requirements. Resource utilization can be improved by flexible resource allocation policies such as advance reservation of resources with a bounded time period & pooling of backup resources.

→ Quality of Services (QOS) Assurance :

Key QOS dimensions include availability, security & performance. QOS expectations should be expressed using measurable terms. Services such as job execution & data services must provide agreed-upon QOS.

→ Job Execution :

OGSA must provide manageability for execution of user-defined work (jobs) throughout their lifetime. Functions such as ~~sch~~ scheduling, provisioning, job control & exception handling of jobs must be supported.

→ Data services:

→ Efficient access & movement & management of large quantities of data in a distributed environment. Also require support for the sharing & integration of distributed data with appropriate security assurance.

→ Security:

Safe administration requires controlling access to services through robust security protocols & a/c to provided security policy.

→ Administrative Cost Reduction:

The complexity of administering large-scale distributed, heterogenous systems increases administration costs & risk of human errors. Support for administration tasks, by automating administrative opⁿ & consistent mgt. of virtualized resource is needed.

→ Scalability:

Needs to scale to potentially thousands of resources of a widely varied nature.

Globus Tool kit (1998)

→ An open architecture, open source set of s/w services & libraries that support computational grids.

→ Components can be used independently or together to develop useful grid applications.

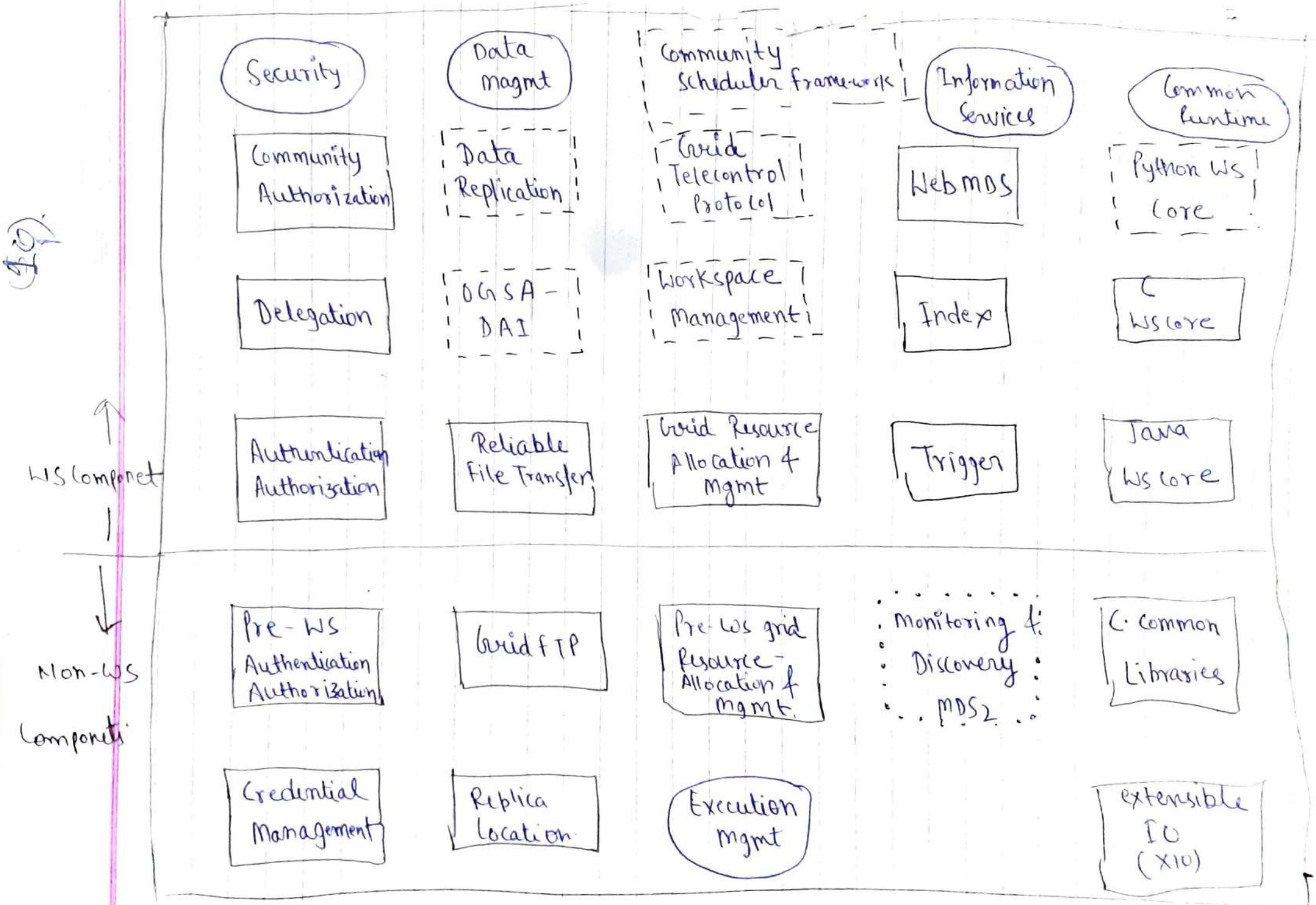
→ It is a fundamental enabling technology for the 'grid', letting people share computing power, database & other tools securely online across corporate, institutional & geographical boundaries without ~~scar~~ sacrificing local autonomy.

→ The toolkit includes s/w services & libraries for resource monitoring, discovery & management, plus security & file management. The Globus Toolkit is a substrate on which leading IT companies are building significant commercial Grid products.

→ The Globus Tool kit was conceived to remove obstacles that prevent seamless collaborations. Its core services, interfaces & protocols allow users to access remote resources as if they were located within their own m/c room while simultaneously preserving local control over who can use resources & when.

→ The Globus Toolkit has grown through an open-source strategy similar to the linux OS. This encourages broader, more rapid adoption & leads to greater technical innovation, as the open-source community provides continual enhancements to the product.

(Q)


Security
Data mgmt
Community Scheduler Framework
Information Services
Common Runtime
Community Authorization
Data Replication
Grid Telecontrol Protocol
WebMDS
Python WS Core
Delegation
OGSA-DAI
Workspace Management
Index
C WS core
Authentication Authorization
Reliable File Transfer
Grid Resource Allocation & Mgmt
Trigger
Java WS core
WS Component
Non-WS Component
Pre-WS Authentication Authorization
GridFTP
Pre-WS grid Resource Allocation & Mgmt.
Monitoring & Discovery MDS2
C Common Libraries
Credential Management
Replica Location
Execution Mgmt
extensible IO (XIO)

[ ] → Core GT Components

[- - -] → Contribution Tech/Preview :- public interfaces may change b/w incremental release.

[. . .] → Deprecated Component → will be dropped in a future release.

Common Runtime :- The Common Runtime Components provide a set of fundamental libraries and tools which are needed to build both WSF non W-S services.

Security :- Using the security components will allow us to manage large sets of data in our virtual organization.

- Information Services :- Information Services, more commonly referred to as the Monitoring & Discovery Services (MDS), includes a set of components to discover and monitor resources in a virtual organization. Note that GT4 also includes a non-WS version of MDS (MDS2) for legacy purpose. This component is deprecated and will surely disappear in future release of the toolkit.

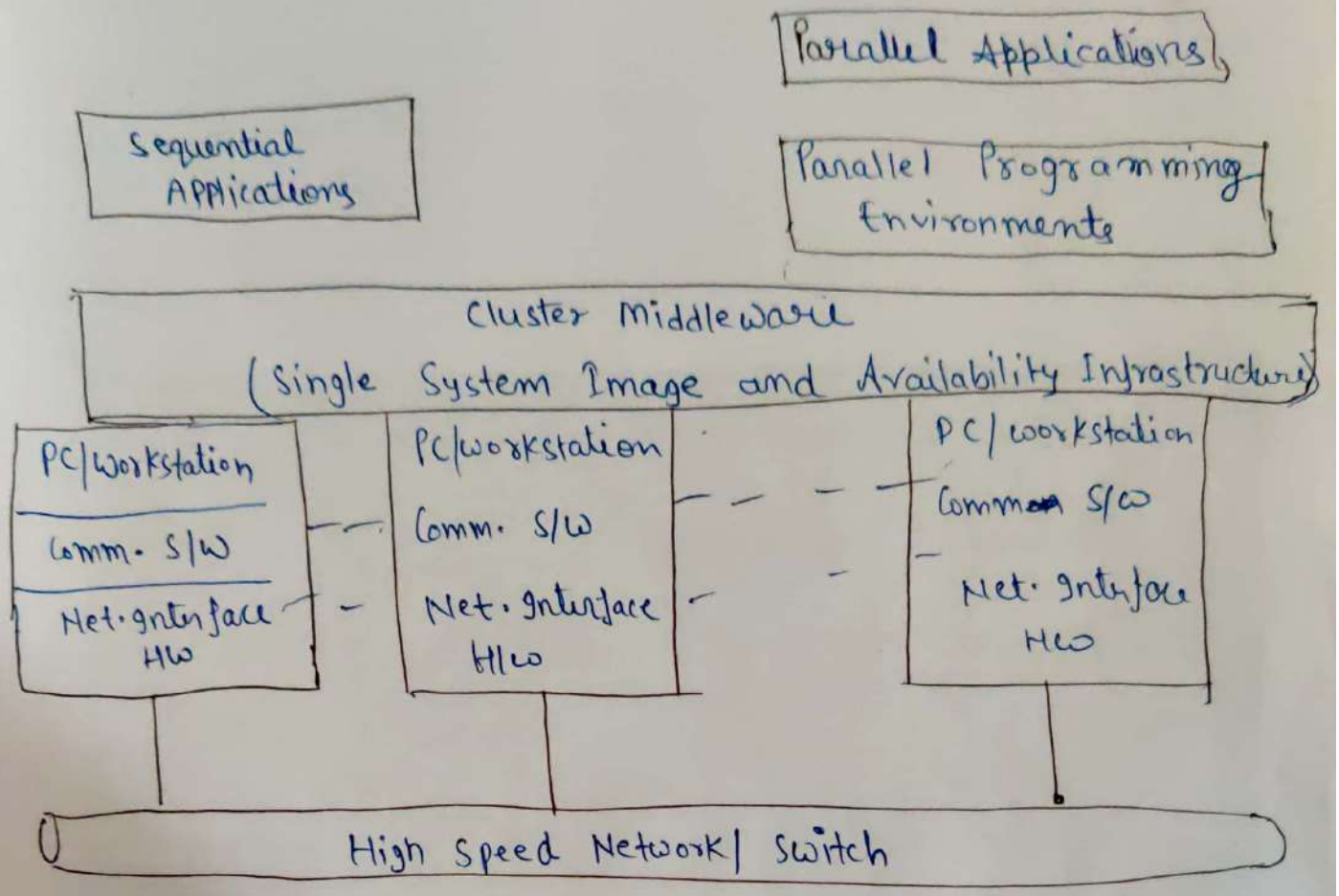
Execution Mgmt :- Exe. Mgmt components deals with the initiation, monitoring, management, scheduling & coordination of executable programs, usually called Jobs, in a grid.

Data Mgmt :- These components will allow us to manage large set of data in our virtual organizations.

(1)

Unit-3

Cluster Computer Architecture



Cluster Computer Architecture.

Some prominent components of cluster computers :-

* Multiple High Performance computers (PC, Workstation, or SMPs)
* State-of-the-art operating System (Layered or Micro-kernel based)
* High performance H/w & switches (Gigabit Ethernet & Myrinet)
* Network Interface Cards (NICs)
* fast communication Protocols & Services.
* Cluster middleware.
  H/w (Digital (DEC) Memory channel, H/w DSM, and SMP Techniques.)

Scanned by CamScanner

→ (2) Operating System Kernel or Gluing Layer (such as solaris MC. and GLU-nix).

→ Parallel programming environment & Tools such as Compilers, PVM, MPI (Message passing Interface)

Application:-

* Parallel
* Distributed
* Sequential.

* N/w Interface h/w acts as a communication processor and is responsible for transmitting & recieving packets of data b/w Cluster nodes via a n/w or switch.

* Communication S/w offers a means of fast & reliable data communication among cluster nodes and to the outside world.
Myrinet use communication protocols such as active messages for fast Commtn. among its nodes.

* Cluster Middleware is responsible for offering an illusion of a unified system image (single System image) and availability out of a Collection on independent but interconnected Computers.

* Programming Environment can offer portable, efficient, and easy-to-use tools for development of applications. They include message passing libraries, debuggers, & profilers.

Scanned by CamScanner

Components for Cluster:

There are some of the hardware & software components commonly used to build cluster & nodes.

1. Processors: Popular processors includes Intel, Digital Alpha, IBM PowerPC, Sun SPARC, SGI MIPS. Computer systems based on these processors have also been used as clusters.

2. Memory & Cache: The amount of memory needed for the cluster is likely to be determined by the cluster target applications. Programs that are parallelized should be distributed such that the memory is distributed b/w processors for scalability.

3. Disk & I/O: Grand challenge applications often need to process large amounts of data & data sets.

4. System Bus

5. Cluster Interconnects:- The nodes in a cluster communicate over high-speed n/w using a standard networking protocol such as TCP/IP or a low level protocol. In most facilities, it is likely that the interconnection will be nic standard Ethernet. Some high performance n/w technologies are Ethernet, fast ethernet, Gigabit Ethernet, ATM & SCI.

6. Operating System: This section focuses on the various OS available for workstations & PCs. OS technology is maturing & can easily be extended & new subsystems can be added. UNIX & its variants are properly used on workstations. (LINUX, Solaris, Windows NT)

Scanned by CamScanner

Cluster Middleware & Single System Image (SSI):

If a collection of interconnected computers is designed to appear as a unified resource, we say it possesses a single system Image (SSI). The SSI is supported by a middleware layer that resides b/w the OS & user-level environment. This middleware consists of essentially two sublayers of S/w infrastructure :-

→ SSI infrastructure.

→ System Availability Infrastructure.

The SSI infrastructure glues together operating systems on all nodes to offer unified access to system resources. The system availability infrastructure enables cluster services such as checkpointing, automatic failover, recovery from failure & fault - tolerant support among all nodes of the cluster.

→ Single System Image levels / Layers;

A SSI is defined as the illusion, created by H/w or S/w, that presents a collection of resources as one, more powerful resources. The SSI concept can be applied to application specific subsystems, or the entire cluster. These services can be offered by one or more of the following levels / Layers:

→ Hardware

→ OS Kernel – Underware

→ Applications & subsystems – Middleware

Scanned by CamScanner

& migration. The resource scheduler

It should also be noted that programming & runtime systems like PVM can also serve as cluster middleware.

The SSI layers support both cluster-aware & non-aware applications.
(parallel applications) (sequential program).

The SSI & system availability services play a major role in the success of clusters.

Hardware Layer :- Systems such as Digital Memory channel & hardware DSM offer SSI at hardware level & allow the user to view cluster as a shared memory system.

OS Kernel (Onderware) or Gluing Layer : Cluster OS support an efficient execution of parallel applications in an environment shared with sequential applications. A goal is to pool resources in a cluster to provide better performance for both sequential & parallel applications. To realize this goal, OS must support gang-scheduling of parallel programs.

A full cluster-wide SSI allows all physical resources & kernel resources to be visible & accessible from all nodes within the systems.

Applications & Subsystems Layer (Middleware): SSI can also be supported by applications & subsystems, which presents multiple, cooperating components of an application to the user/administrator as a simple application.

Scanned by CamScanner

→ Single System Image, Benefits:-

→ provides a simple, straightforward view of all system resources & activities, from any node of the cluster.

→ It frees the end user from having to know where an application will run.

→ It lets the user work with familiar interface & commands & allows the administrator to manage the entire cluster as a single entity.

→ It presents multiple, cooperating components of an application to the administrator as a single application.

→ It provides location-independent message communication.

→ It helps track the location of all resources.

The benefits of a SSI also apply to system programmers. It reduces the time, effort & knowledge required to perform tasks, & allows current staff to handle larger or more complex systems.....

Resource Management & Scheduling (RMS):

It is the act of distributing applications among computers to maximize their throughput. It also enables the effective & efficient utilization of the resource available. The software that performs the RMS consists of 2 components: a resource manager & a resource scheduler. The resource manager component is concerned with problems, such as locating & allocating computational resources, authentication, as well as tasks

Scanned by CamScanner

such as process creation & migration. The resource scheduler component is concerned with tasks such as queuing applications, as well as resource location & assignment.

The basic RMS architecture is a client-server system. Each computer sharing computational resources runs a server daemon. These daemons maintain up-to-date tables, which store info. about the RMS environment in which it resides. A user interacts with the RMS environment via a client program.

Process Migration :- Process can be suspended, moved & restarted on another computer within the RMS environment

Checkpointing :- Snapshot of an executing program's state is saved & can be used to restart the program from the same point at a later time.

Fault Tolerance :- RMS system can provide various levels of fault tolerance.

Load Balancing, Multiple Application Queues.

Programming Environments & Tools :-

The availability of standard programming tools & utilities have made clusters a practical alternative as a parallel-processing platform. Some most popular tools are:

→ Thread: Threads are a popular paradigm for concurrent programming on uniprocessor as well as multiprocessors machines. Unlike forked process, thread creation is cheaper & easier to manage. Threads communicate using shared variables as they are created within their parent process address space.

Scanned by CamScanner

Threads have been extensively used in developing both applications & system software.

3.

→ Message Passing Systems (MPI & PVM):

Message passing libraries allow efficient parallel programs to be written for distributed memory system. These libraries provide routines to initiate & configure the messaging environment as well as sending & receiving packets of data.

PVM (Parallel Virtual M/c) is both an environment & a message passing library, which can be used to run parallel applications on systems ranging from high-end supercomputers through to cluster of workstations. Whereas MPI is a message passing specification, designed to be standard for distributed memory parallel computing using explicit message passing.

→ Distributed Shared Memory (DSM) System:

Distributed virtual memory over many geographical locations communicating with each other.

→ Parallel Debuggers & Profilers:-

To develop correct & efficient high performance applications it is highly desirable to have some form of easy-to-use parallel debugger & performance profiling tools. Should work in heterogeneous environment. Ex:- Total View.

Scanned by CamScanner

4: 1:10 pm

→ Performance Analysis Tools:

The basic purpose of this is to help a programmer to understand the performance characteristics of an application. It should analyze & locate parts of an application that exhibit poor performance & create program bottlenecks.

→ Cluster Administration Tools:

Monitoring clusters is a challenging task that can be eased by tools that allow entire clusters to be observed at different levels using a GUI. Good management s/w is crucial for exploiting a cluster as a HPC platform.

Clusters :- SMILE, PARMON, Berkeley NOW. (68) (181)

↓

K-CAP (admin. tool)

Cluster Applications :-

→ A cluster can cope up with grand challenge or super-computing applications. GCA (Grand challenge application) are fundamental problems in science & engineering with broad economic & scientific ~~effect~~ impact. A typical example of this is simulation of some phenomena that cannot be measured through experiments such as global climate modeling, protein dynamics.

Scanned by CamScanner

What is a cluster

⇒ A group of systems that are linked together for one or more of the following purposes :-

* Redundancy * Load-balancing
* Parallel processing.

Beowulf cluster :-

* A group of computers running linux
* The computers are linked together via Ethernet
* One computer is designated as the server, the other computers are the nodes

What will a Beowulf cluster do :-

* It is a parallel processing that will decrease the amount of time required for processor-intensive tasks.

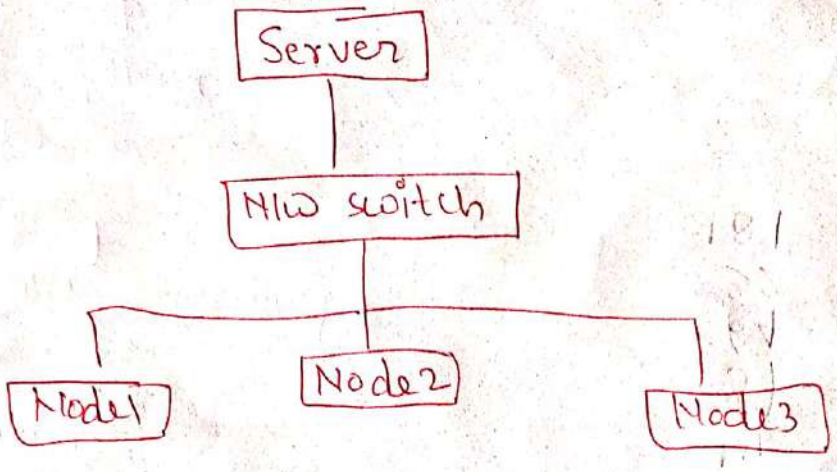
Time needed to complete Task

N — Single Processor machine

N/4 — N1 | N2 | N3 | N4

Scanned by CamScanner

Layout of a Beowulf cluster



How it is Implemented :-

* The Server is configured with Linux and two network cards.
* One N/W card connects to the outside world and the other connects to the internal n/w for the cluster.
* The nodes are configured with Linux and given rights to access the Server.
* The /home and /usr/Local directory on each node is mapped to a directory on the server.
* programs are executed on the Server which then spawn Individual processes to the nodes.
* The nodes execute the processes and send the results back to the Server.

Scanned by CamScanner

Advantages :-

* Low price-to-performance ratio.
* Easy to expand for increased performance.
* Can be implemented with typical personal computers
* Most Linux S/w is completely free.

Disadvantages :-

* Can be somewhat tricky to implement
* programs must be written specifically for parallel systems in order to take advantage of the cluster.

Scanned by CamScanner

Beowulf Cluster:-

A Beowulf cluster is a computer cluster of what are normally identical, commodity-grade computers networked into a small local area n/w with libraries & programs installed which allow processing to be shared among them. The result is a high-performance parallel computing cluster from inexpensive personal computer hardware.

The Name 'Beowulf' comes from the old English epic poem.

Beowulf clusters normally run a UNIX-like OS such as BSD, Linux or Solaris. Commonly used parallel processing library include MPI & PVM. Both of these permit the programmer to divide a task among a group of networked computers, & collect the result of processing.

→ It is a multi-computer architecture which can be used for parallel computations. It is a system which usually consists of one server node, & one or more client nodes connected via Ethernet or some other network.

→ Nodes are dumb & configured and controlled by the server node, & do only what they are told to do.

→ It is a technology of clustering computers to form a parallel, virtual supercomputer.

Scanned by CamScanner

The Beowulf Model :- MPP (Massively Parallel Processing)

Clusters can be built with anything from desktop PCs to high end SMP servers for processing & anything from Ethernet to ATM for an interconnect.

Beowulf systems define a class of clustered computing that focuses on maximizing the price-to-performance ratio of the overall system without compromising its ability to perform the computational work for which it is being built. This design model typically results in machines that are an order of magnitude less expensive than any other. In this respect, the Beowulf model has almost come to define low cost supercomputing.

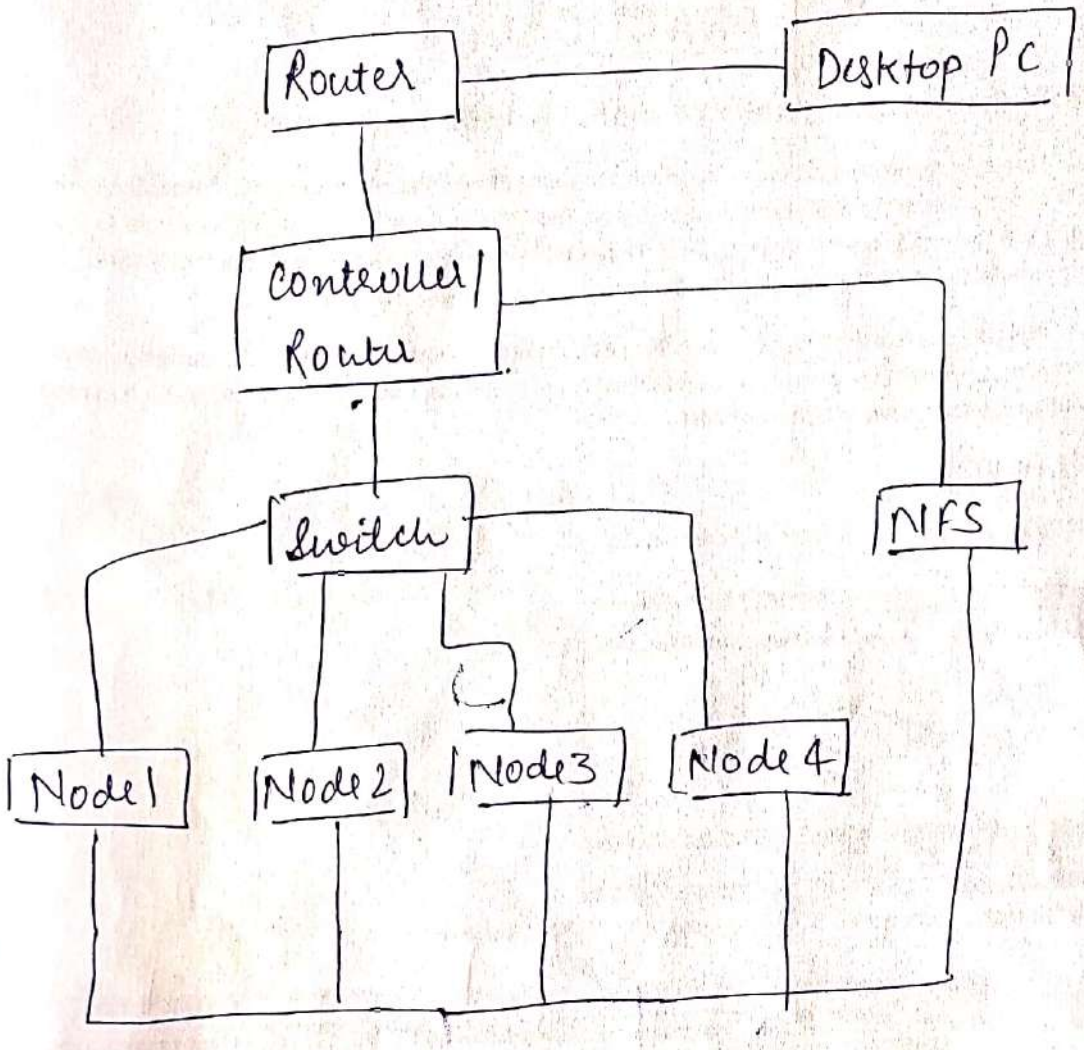
→ Cost-optimization is the most notable aspect.

→ Beowulf, by tracking the evolution of commodity h/w, are able to incorporate the very latest technology advancement well before proprietary parallel machines.

→ When you move from one Beowulf generation to next, all your s/w stays the same & the programming model doesn't change. A first generation Beowulf program will compile & run on a fourth generation systems, only faster.

Scanned by CamScanner

Beowulf System Architecture:



→ Beowulf do not impose a fixed system architecture. The primary system components that drive the architecture can be decomposed into processor, memory, n/w & secondary storage systems. But the processor & network have the most visible impact on the changes in Beowulf architecture.

→ Processing power was a dominant limiting factor on the performance of initial Beowulf systems. But memory bandwidth has now replaced the role of the processor as a performance bottleneck.

→ The N/W has always been a performance limiting factor since the early days. However the n/w still impose a strict limit on the scaling of Beowulf systems.

Scanned by CamScanner

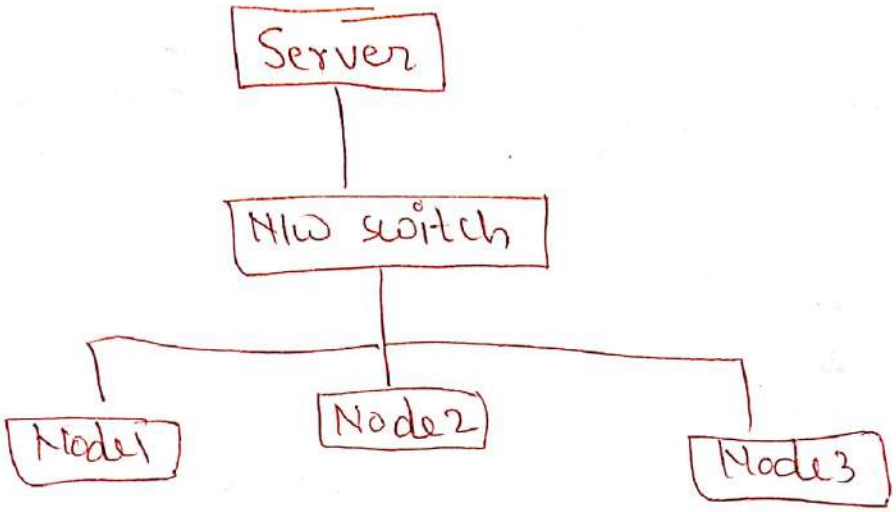
Difference Between Beowulf Cluster & Cluster of Workstations (COW) (4)

→ One of the main differences b/w Beowulf & COW is that Beowulf behaves more like a single m/c rather than many workstations. In most cases client nodes do not have keyboards or monitors & are accessed only via remote login or possibly serial terminal.

→ Beowulf nodes can be thought of as a CPU + memory package which can be plugged into the cluster, just like a CPU or memory module can be plugged into a motherboard.

Scanned by CamScanner

Layout of a Beowulf cluster



How it is Implemented :-

* The Server is configured with Linux and two network cards.
* One N/W card connects to the outside world and the other connects to the internal n/w for the cluster.
* The nodes are configured with Linux and given rights to access the Server.
* The /home and /usr/Local directory on each node is mapped to a directory on the server.
* Programes are executed on the Server which then spawn individual processes to the nodes.
* The nodes execute the processes and send the results back to the Server.

Scanned by CamScanner

What is a cluster (25.Nov.17)

⇒ A group of systems that are linked together for one or more of the following purposes :-

* Redundancy * Load-balancing
* Parallel processing.

Beowulf cluster :-

* A group of computers running linux
* The computers are linked together via Ethernet
* One computer is designated as the server, the other computers are the nodes

What will a Beowulf cluster do :-

* It is a parallel processing that will decrease the amount of time required for processor-intensive tasks.

Time needed to complete Task

N — Single Processor machine

N/4 — N1 | N2 | N3 | N4

Scanned by CamScanner

Advantages :-

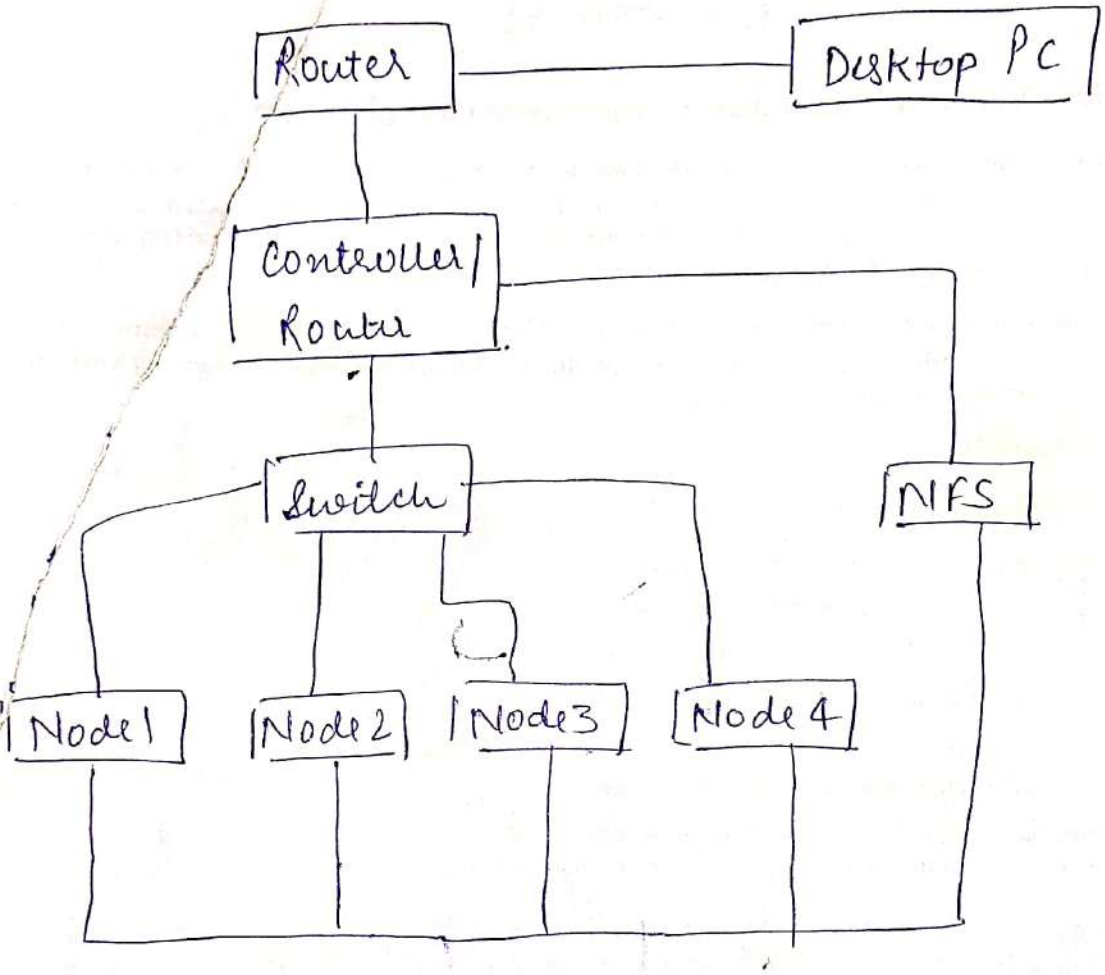
* Low-price-to-performance ratio.
* Easy to expand for increased performance.
* Can be implemented with typical personal computers
* Most Linux S/W is completely free.

Disadvantages :-

* Can be somewhat tricky to implement
* programs must be written specifically for parallel systems in order to take advantage of the cluster.

Scanned by CamScanner

Beowulf System Architecture:



→ Beowulf do not impose a fixed system architecture. The primary system components that drive the architecture can be decomposed into processor, memory, n/w & secondary storage systems. But the processor & network have the most visible impact on the changes in Beowulf architecture.

→ Processing power was a dominant limiting factor on the performance of initial Beowulf systems. But memory bandwidth has now replaced the role of the processor as a performance bottleneck.

→ The N/W has always been a performance limiting factor since the early days. However the n/w still impose a strict limit on the scaling of Beowulf systems.

Scanned by CamScanner

Beowulf Cluster:-

(1) UNIT - 4

A Beowulf cluster is a computer cluster of what are normally identical, commodity - grade computers networked into a small local area n/w with libraries & programs installed which allow processing to be shared among them. The result is a high - performance parallel computing cluster from inexpensive personal computer hardware.

The Name 'Beowulf' comes from the Old English epic poem.

Beowulf clusters normally run a UNIX-like OS such as BSD, Linux or Solaris. Commonly used parallel processing library include MPI & PVM. Both of these permit the programmer to divide a task among a group of networked computers, & collect the result of processing.

→ It is a multi-computer architecture which can be used for parallel computations. It is a system which usually consists of one server node, & one or more client nodes connected via Ethernet or some other network.

→ Nodes are dumb & configured and controlled by the server node, & do only what they are told to do.

→ It is a technology of clustering computers to form a parallel, virtual supercomputer.

Scanned by CamScanner

The Beowulf Model :-

MPP (Massively Parallel Processing)

Clusters can be built with anything from desktop PCs to high end SMP servers for processing & anything from Ethernet to ATM for an interconnect.

Beowulf systems define a class of clustered computing that focuses on maximizing the price-to-performance ratio of the overall system without compromising its ability to perform the computational work for which it is being built. This design model typically results in machines that are an order of magnitude less expensive, than any other. In this respect, the Beowulf model has almost come to define low cost supercomputing.

- Cost-optimization is the most notable aspect.
- Beowulf, by tracking the evolution of commodity h/w, are able to incorporate the very latest technology advancements well before proprietary parallel machines.
- When you move from one Beowulf generation to next, all your s/w stays the same & the programming model doesn't change. A first generation Beowulf program will compile & run on a fourth generation systems, only faster.

Scanned by CamScanner

Difference Between Beowulf Cluster & Cluster of Workstations (COW): (4)

→ One of the main differences b/w Beowulf & COW is that Beowulf behaves more like a single m/c rather than many workstations.. In most cases client nodes do not have keyboards or monitors & are accessed only via remote login or possibly serial terminal.

→ Beowulf nodes can be thought of as a CPU + memory package which can be plugged into the cluster, just like a CPU or memory module can be plugged into a motherboard.

Scanned by CamScanner

Application Domains:

The potential application of Beowulf-class systems is broad & ranges from high-end floating point intensive scientific & engineering problems to commercial data intensive tasks such as Web servers & data bases. Examples of applications include finite element & finite difference algo, conjugate gradient solvers, tree codes, ray tracing prob.

Parallel Programming with MPL (Message Passing Library) :- or MPI (Message Passing Interface)

MPL is a subroutine library from IBM for writing distributed memory programs in C or Fortran, for RISC system. MPL provides routines for message passing, mgt. of tasks & task groups, & collective communications. The programming model is MIMD (Multiple instn multiple data). The no. of tasks dedicated to run a parallel program in MPL is fixed at load time.

MPL allows both process-to-process commn & collective (or global) commn. It is a commn protocol for programming parallel computers. Its goals are high performance, scalability & portability. (Dominant Model)

Advantages are portability & speed.
↳ has been implemented for almost every distributed memory architecture.

Scanned by CamScanner

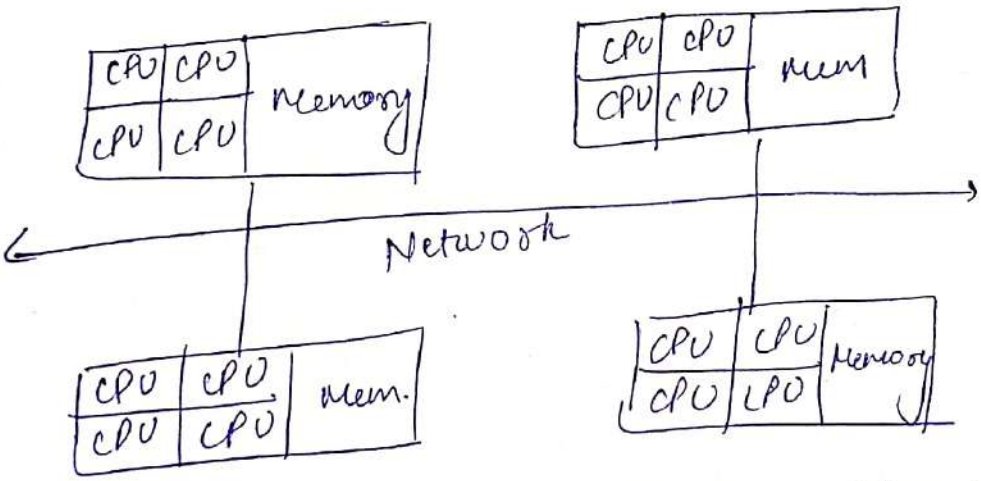
→ Reasons for using MPI :-

It uses language Independent specifications (LIS) for calls & language bindings. It meant to provide essential virtual topology, synchronization & commⁿ functionality b/w a set of processes in a language independent way.

It is based on the consensus (General Agreement) of MPI forum, which has over 40 participating organizations, including vendors, researchers, s/w library developers & users. The interface attempt to be :- Practical, Efficient, Portable, Flexible

→ Most recent version is MPI-3.x.

→ Programming model of MPI :- Basically designed for distributed memory systems.



Today it runs on virtually any h/w platform :-

Distributed Memory
Shared Memory
Hybrid

→ All parallelism is explicit: the programmer is responsible for correctly identifying parallelism & implementing parallel algorithms using MPI constructs.

Scanned by CamScanner

→ Reasons for using MPI: -

Standardization

Portability

Availability

Performance Opportunities

Functionality → over 430 routines defined

MPI library	Where	compilers
MVAPICH	Linux Clusters	GNU, Intel, PGI, Clang
Open MPI	"	"
Intel MPI	"	Intel, GNU
IBM BG/Q MPI	BG/Q Clusters	IBM, GNU
IBM Spectrum MPI	Coral Early Access	IBM, GNU, PGI, Clang

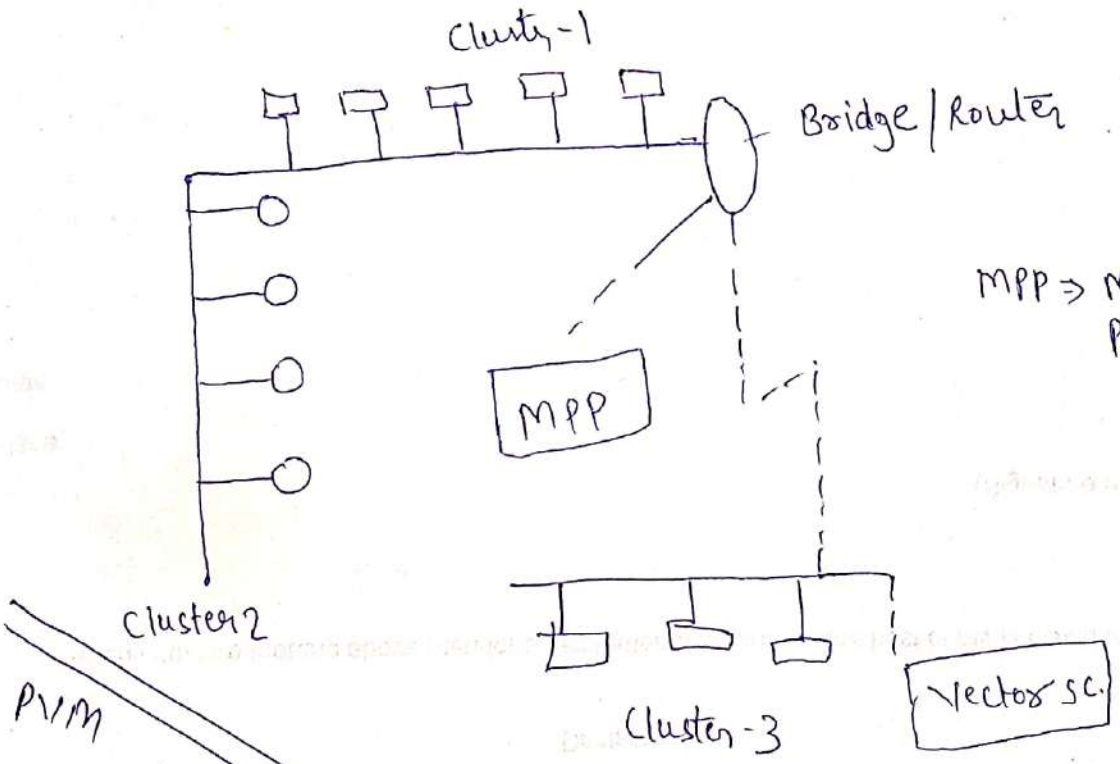
Scanned by CamScanner

Parallel Virtual Machine (PVM) :-

PVM is a s/w tool for parallel networking of computers. It is designed to allow a n/w of hetrogeneous UNIX or Windows m/c to be used as a single distributed parallel processor. Written in C. (Fault Tolerant)

PVM consists of a run-time environment & library for message passing, task & resource mgt; & fault notification.

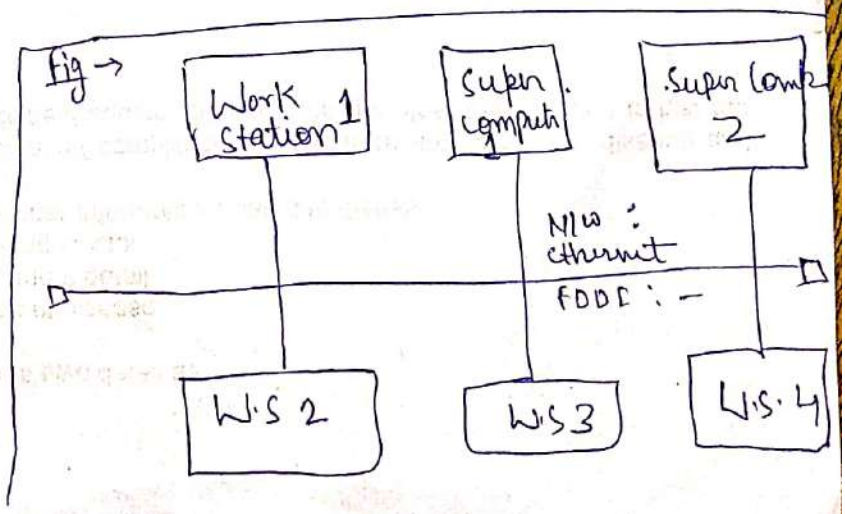
Architecture :- combine multiple Beowulf cluster



MPP ⇒ Massively Parallel Processing

PVM

uniform view of Multiprogrammed Virtual Machine



Scanned by CamScanner