Zero to Mastery in

DISTRIBUTED COMPUTING

Dr. R.K . Jain

• Shadab Saifi (***Illustrator***) • Ayaz Uddin (***Editor***)

(An ISO 9001:2008 Certified Company)

Vayu Education of India

2/25, Ansari Road, Darya Ganj, New Delhi-110 002

First Edition: 2022

Published by:

AN ISO 9001:2008 CERTIFIED COMPANY

VAYU EDUCATION OF INDIA
2/25, ANSARI ROAD, DARYA GANJ, NEW DELHI-110 002
PH.: 011-41564440, MOB. 09910115201

Contents


BLOCK CHAIN
INTRODUCTION TO
BLOCKCHAIN
• Dr. R.K. Jain
aws
INTRODUCTION TO
CLOUD
COMPUTING
• Dr. R.K. Jain
INTRODUCTION TO
INTERNET
OF THINGS
• Dr. R.K. Jain

Unit- I

Distributed Computing

1.1 Overview of Cloud Computing

1.2 Characteristics of Cloud Computing

1.3 Advantages of Cloud Computing

1.4 Challenges of Cloud Computing

1.5 Applications of Cloud Computing

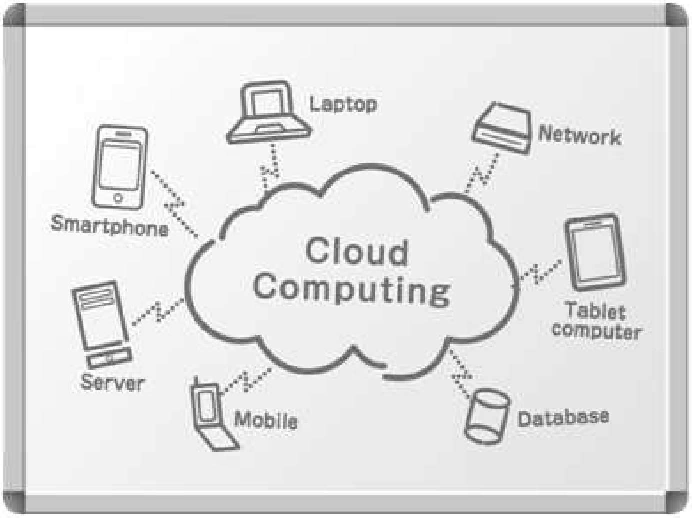
UNIT-I

1

BASIC CONFIGURATION OF ROBOTICS AND ITS WORKING

1.1 OVERVIEW OF CLOUD COMPUTING

Cloud computing is computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid.



Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network.

Cloud computing focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach should maximize the use of computing power thus reducing environmental damage as well since less power, air conditioning, rack space, etc. are required for a variety of functions. With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications.

The term "moving to cloud" also refers to an organization moving away from a traditional CAPEX(capital expenditure) model (buy the dedicated hardware and depreciate it over a period of time) to the OPEX(operational expenditure) model (use a shared cloud infrastructure and pay as one uses it).

It was observed that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. It was also observed that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand .Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing.

Cloud vendors are experiencing growth rates of 50% per annum.

1.2 CHARACTERISTICS OF CLOUD COMPUTING

Cloud computing exhibits the following key characteristics:

1. Application programming interface (API) accessibility to software that enables machines to interact with cloud software in the same way that a traditional user interface (e.g., a computer desktop) facilitates interaction between humans and computers.
2. Cost reductions claimed by cloud providers. A public-cloud delivery model converts capital expenditure to operational expenditure. This purportedly lowers barriers to entry, as infrastructure is typically provided by a third party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained, with usage-based options and fewer IT skills are required for implementation (in-house). The e-FISCAL project's state-of-the-art repository contains several articles looking into cost aspects in more detail, most of them concluding that costs savings depend on the type of activities supported and the type of infrastructure available in-house.
3. Device and location independence enable users to access systems using a web browser regardless of their location or what device

they use (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.

4. Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.
5. Multitenancy refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple tenants. A tenant is a group of users sharing the same view on the software they use. Multitenancy enables sharing of resources and costs across a large pool of users thus allowing for:
    - Centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
    - Peak-load capacity increases (users need not engineer for highest possible load-levels)
    - Utilization and efficiency improvements for systems that are often only 10–20% utilized.
6. Performance is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.
7. Productivity may be increased when multiple users can work on the same data simultaneously, rather than waiting for it to be saved and emailed. Time may be saved as information does not need to be re-entered when fields are matched, nor do users need to install application software upgrades to their computer.
8. Reliability improves with the use of multiple redundant sites, which makes well-designed cloud computing suitable for business and disaster recovery.
9. Scalability and elasticity via dynamic ("ondemand") provisioning of resources on a fine-rained, self-service basis in near real-time without users having to engineer for peak loads.
10. Security can improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford to tackle. However, the complexity of security is greatly increased when data is distributed over a wider area or over a greater number of devices, as well as in multi-tenant systems

shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

Essential characteristics

1. On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

2. Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

3. Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

4. Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.

5. Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

1.3 ADVANTAGES OF CLOUD COMPUTING

Cost Efficient: - Cloud computing is probably the most cost efficient method to use, maintain and upgrade. Traditional desktop software costs companies a lot in terms of finance. Adding up the licensing fees for multiple users can prove to be very expensive for the establishment concerned. The cloud, on the other hand, is available at much cheaper rates and hence, can significantly lower the company's IT expenses. Besides, there are many one-time-payments, pay-as-you-go and other scalable options available, which make it very reasonable for the company in question.

Almost Unlimited Storage: - Storing information in the cloud gives you almost unlimited storage capacity. Hence, you no more need to worry about running out of storage space or increasing your current storage space availability.

Backup and Recovery: - Since all your data is stored in the cloud, backing it up and restoring the same is relatively much easier than storing the same on a physical device. Furthermore, most cloud service providers are usually competent enough to handle recovery of information. Hence, this makes the entire process of backup and recovery much simpler than other traditional methods of data storage.

Automatic Software Integration: - In the cloud, software integration is usually something that occurs automatically. This means that you do not need to take additional efforts to customize and integrate your applications as per your preferences. This aspect usually takes care of itself. Not only that, cloud computing allows you to customize your options with great ease. Hence, you can handpick just those services and software applications that you think will best suit your particular enterprise.

Easy Access to Information: - Once you register yourself in the cloud, you can access the information from anywhere, where there is an Internet connection. This convenient feature lets you move beyond time zone and geographic location issues.

Quick Deployment: - Lastly and most importantly, cloud computing gives you the advantage of quick deployment. Once you opt for this method of functioning, your entire system can be fully functional in a matter of a few minutes. Of course, the amount of time taken here will depend on the exact kind of technology that you need for your business.

Disadvantages of Cloud Computing

In spite of its many benefits, as mentioned above, cloud computing also has its disadvantages. Businesses, especially smaller ones, need to be aware of these cons before going in for this technology.

Technical Issues: - Though it is true that information and data on the cloud can be accessed anytime and from anywhere at all, there are times when this system can have some serious dysfunction. You should be aware of the fact that this technology is always prone to outages and other technical issues. Even the best cloud service providers run into this kind of trouble, in spite of keeping up high standards of maintenance. Besides, you will need a very good Internet connection to be logged onto the server at all times. You will invariably be stuck in case of network and connectivity problems.

Security in the Cloud: - The other major issue while in the cloud is that of security issues. Before adopting this technology, you should know that you will be surrendering all your company's sensitive information to a third-party cloud service provider. This could potentially put your company to great risk. Hence, you need to make absolutely sure that you choose the most reliable service provider, who will keep your information totally secure.

1.4 CHALLENGES OF CLOUD COMPUTING

1. Security and Privacys

The main challenge to cloud computing is how it addresses the security and privacy concerns of businesses thinking of adopting it. The fact that the valuable enterprise data will reside outside the corporate firewall raises serious concerns. Hacking to cloud infrastructure would affect multiple clients even if only one site is attacked. These risks can be mitigated by using security applications, encrypted file systems, data loss software, and buying security hardware to track unusual behavior across servers.

2. Service Delivery and Billing

It is difficult to assess the costs involved due to the on-demand nature of the services. Budgeting and assessment of the cost will be very difficult unless the provider has some good and comparable benchmarks to offer. The service-level agreements (SLAs) of the provider are not adequate to guarantee the availability and scalability. Businesses will be reluctant to switch to cloud without a strong service quality guarantee.

3. Portability

Businesses should have the leverage of migrating in and out of the cloud and switching providers whenever they want, and there should be no lock-in period. Cloud computing services should have the capability to integrate smoothly with the on-premise IT.

4. Reliability and Availability

Cloud providers still lack round-the-clock service; this results in frequent outages. It is important to monitor the service being provided using internal or third-party tools. It is vital to have plans to supervise usage, SLAs, performance, robustness, and business dependency of these services.

5. Performance and Bandwidth Cost

Businesses can save money on hardware but they have to spend more for the bandwidth. This can be a low cost for smaller applications but can be significantly high for the data-intensive applications. Delivering intensive and complex data over the network requires sufficient bandwidth. Because of this, many businesses are waiting for a reduced cost before switching to the cloud.

All these challenges should not be considered as road blocks in the pursuit of cloud computing. It is rather important to give serious consideration to these issues and the possible ways out before adopting the technology.

1.5 APPLICATIONS OF CLOUD COMPUTING

Cloud computing applications, or apps, are the cloud-based services also known as Software as a Service (SaaS). Programs that once had to be installed on computers individually are now offered online, and the only thing a person needs to access the program is an account and password. These apps can do everything from keeping track of notes to accounting.

1. Infrastructure as a service (IaaS) and platform as a service (PaaS)

When it comes to IaaS, using an existing infrastructure on a pay-per-use scheme seems to be an obvious choice for companies saving on the cost of investing to acquire, manage and maintain an IT infrastructure. There are also instances where organizations turn to PaaS for the same reasons while also seeking to increase the speed of development on a ready-to-use platform to deploy applications.

2. Private cloud and hybrid cloud

Among the many incentives for using cloud, there are two situations where organizations are looking into ways to assess some of the applications they intend to deploy into their environment through the use of a cloud (specifically a public cloud). While in the case of test and development it may be limited in time, adopting a hybrid cloud approach allows for testing application workloads, therefore providing the comfort of an environment without the initial investment that might have been rendered useless should the workload testing fail.

Another use of hybrid cloud is also the ability to expand during periods of limited peak usage, which is often preferable to hosting a large infrastructure that might seldom be of use. An organization would seek to have the additional capacity and availability of an environment when needed on a pay-as you-go basis.

3. Test and development

Probably the best scenario for the use of a cloud is a test and development environment. This entails securing a budget, setting up your environment through physical assets, significant manpower and time. Then comes the installation and configuration of your platform. All this can often extend the time it takes for a project to be completed and stretch your milestones.

With cloud computing, there are now readily available environments tailored for your needs at your fingertips. This often combines, but is not limited to, automated provisioning of physical and virtualized resources.

4. Big data analytics

One of the aspects offered by leveraging cloud computing is the ability to tap into vast quantities of both structured and unstructured data to harness the benefit of extracting business value.

Retailers and suppliers are now extracting information derived from consumers′ buying patterns to target their advertising and marketing campaigns to a particular segment of the population. Social networking platforms are now providing the basis for analytics on behavioral patterns that organizations are using to derive meaningful information.

5. File storage

Cloud can offer you the possibility of storing your files and accessing, storing and retrieving them from any web-enabled interface. The web services interfaces are usually simple. At any time and place you have high availability, speed, scalability and security for your environment. In this scenario, organizations are only paying for the amount of storage they are actually consuming, and do so without the worries of overseeing the daily maintenance of the storage infrastructure.

There is also the possibility to store the data either on or off premises depending on the regulatory compliance requirements. Data is stored in virtualized pools of storage hosted by a third party based on the customer specification requirements.

6. Disaster recovery

This is yet another benefit derived from using cloud based on the cost effectiveness of a disaster recovery (DR) solution that provides for a faster recovery from a mesh of different physical locations at a much lower cost that the traditional DR site with fixed assets, rigid procedures and a much higher cost.

7. Backup

Backing up data has always been a complex and time-consuming operation. This included maintaining a set of tapes or drives, manually collecting them and dispatching them to a backup facility with all the inherent problems that might happen in between the originating and the backup site. This way of ensuring a backup is performed is not immune to problems such as running out of backup media , and there is also time to load the backup devices for a restore operation, which takes time and is prone to malfunctions and human errors.

IMPORTANT QUESTIONS

A: 2 Marks Questions

1. What is cloud computing?
2. What is the meaning of CAPEX?
3. What is the meaning of OPEX?
4. What is the approximate growth rate of cloud vendors?
5. API stands for ……………………………………………………..
6. How cost is reduced by cloud computing?
7. Maintenance of cloud computing application is easier. Why?
8. What is Multitenancy?
9. Explain any one essential characteristics of clod computing.
10. How security is achieved in cloud computing?
11. What is Iaas?
12. What is Paas?
13. Differentiate between private clod and private cloud.
14. What is resource pooling in cloud computing?
15. What is disaster recovery in clod computing?

B: 4 Marks Questions

16. Explain cloud computing in details.
17. Write any four characteristics of characteristics of cloud computing.
18. Discuss the advantage of cloud computing.
19. What are challenges to cloud computing?
20. Explain applications of cloud computing.

C: 10 Marks Questions

21. What is cloud computing? Explain its characteristics in detail.
22. Explain the advantages and disadvantages of cloud computing.
23. Explain the applications of cloud computing.

Unit- II

Cloud Computing Service Models and Deployment Models

2.1 **Service Model**
  - Saas
  - Paas
  - Iaas

2.2 **Deployment Models**
  - Private Cloud
  - Public Cloud
  - Hybrid Cloud
  - Community Cloud

UNIT-I

2

CLOUD COMPUTING SERVICE MODELS AND DEPLOYMENT MODELS

2.1 SERVICE MODEL- SAS, PAS, IAS

Cloud computing providers offer their services according to several fundamental models.

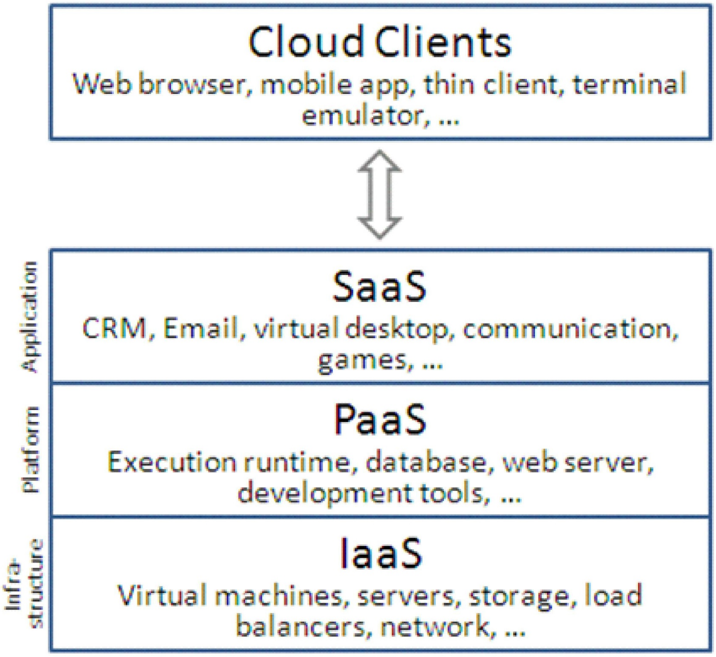


Fig. 2.1 Structure as a service (IaaS)

In the most basic cloud-service model & according to the IETF (Internet Engineering Task Force), providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. (A hypervisor, such as Xen, Oracle VirtualBox, KVM,VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.) IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on-demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed.

Infrastructure as a Service (IaaS) is one of the three fundamental service models of cloud computing alongside Platform as a Service (PaaS) and Software as a Service (SaaS). As with all cloud computing services it provides access to computing resource in a virtualized environment, "the Cloud", across a public connection, usually the internet. In the case of IaaS the computing resource provided is specifically that of virtualized hardware, in other words, computing infrastructure. The definition includes such offerings as virtual server space, network connections, bandwidth, IP addresses and load balancers. Physically, the pool of hardware resource is pulled from a multitude of servers and networks usually distributed across numerous data centers, all of which the cloud provider is responsible for maintaining. The client, on the other hand, is given access to the virtualized components in order to build their own IT platforms.

In common with the other two forms of cloud hosting, IaaS can be utilized by enterprise customers to create cost effective and easily scalable IT solutions where the complexities and expenses of managing the underlying hardware are outsourced to the cloud provider. If the scale of a business customer's operations fluctuate, or they are looking to expand, they can tap into the cloud resource as and when they need it rather than purchase, install and integrate hardware themselves.

The following are salient examples of how IaaS can be utilized by enterprise:

- Enterprise infrastructure; by internal business networks, such as private clouds and virtual local area networks, which utilize pooled server and networking resources and in which a business can store their data and run the applications they need to operate day-to-day. Expanding businesses can scale their infrastructure in accordance with their growth whilst private clouds (accessible only by the business itself) can protect the storage and transfer of the sensitive data that some businesses are required to handle.

- Cloud hosting; the hosting of websites on virtual servers which are founded upon pooled resources from underlying physical servers. A website hosted in the cloud, for example, can benefit from the redundancy provided by a vast network of physical servers and on demand scalability to deal with unexpected demands placed on the website.
- Virtual Data Centers (VDC); a virtualized network of interconnected virtual servers which can be used to offer enhanced cloud hosting capabilities, enterprise IT infrastructure or to integrate all of these operations within either a private or public cloud implementation.

A typical Infrastructure as a Service offering can deliver the following features and benefits:

- Scalability; resource is available as and when the client needs it and, therefore, there are no delays in expanding capacity or the wastage of unused capacity
- No investment in hardware; the underlying physical hardware that supports an IaaS service is set up and maintained by the cloud provider, saving the time and cost of doing so on the client side
- Utility style costing; the service can be accessed on demand and the client only pays for the resource that they actually use
- Location independence; the service can usually be accessed from any location as long as there is an internet connection and the security protocol of the cloud allows it
- Physical security of data centre locations; services available through a public cloud, or private clouds hosted externally with the cloud provider, benefit from the physical security afforded to the servers which are hosted within a data centre
- No single point of failure; if one server or network switch, for example, were to fail, the broader service would be unaffected due to the remaining multitude of hardware resources and redundancy configurations. For many services if one entire data center were to go offline, never mind one server, the IaaS service could still run successfully.

Platform as a service (PaaS)

In the PaaS models, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PaaS offers like Microsoft Azure and Google App Engine, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. The latter has also been proposed by an architecture aiming to facilitate real-time in cloud environments.

Platform as a service (PaaS) provides a computing platform and a key chimney. It joins with software as a service (SaaS) and infrastructure as a service (IaaS), model of cloud computing.

Platform as a Service, often simply referred to as PaaS, is a category of cloud computing that provides a platform and environment to allow developers to build applications and services over the internet. PaaS services are hosted in the cloud and accessed by users simply via their web browser.

Platform as a Service allows users to create software applications using tools supplied by the provider. PaaS services can consist of preconfigured features that customers can subscribe to; they can choose to include the features that meet their requirements while discarding those that do not. Consequently, packages can vary from offering simple point-and-click frameworks where no client side hosting expertise is required to supplying the infrastructure options for advanced development.

The infrastructure and applications are managed for customers and support is available. Services are constantly updated, with existing features upgraded and additional features added. PaaS providers can assist developers from the conception of their original ideas to the creation of applications, and through to testing and deployment. This is all achieved in a managed mechanism.

As with most cloud offerings, PaaS services are generally paid for on a subscription basis with clients ultimately paying just for what they use. Clients also benefit from the economies of scale that arise from the sharing of the underlying physical infrastructure between users, and that results in lower costs.

Below are some of the features that can be included with a PaaS offering:

- Operating system
- Server-side scripting environment
- Database management system
- Server Software
- Support
- Storage
- Network access
- Tools for design and development
- Hosting

Software developers, web developers and businesses can benefit from PaaS. Whether building an application which they are planning to offer over the internet or software to be sold out of the box, software developers may take advantage of a PaaS solution. For example, web developers can use individual PaaS environments at every stage of the process to develop, test and ultimately host their websites. However, businesses that are developing their own internal software can also utilize Platform as a Service, particularly to create distinct ring-fenced development and testing environments.

Below are some of the benefits of PaaS to application developers:

- They don't have to invest in physical infrastructure; being able to 'rent' virtual infrastructure has both cost benefits and practical benefits. They don't need to purchase hardware themselves or employ the expertise to manage it. This leaves them free to focus on the development of applications. What's more, clients will only need to rent the resources they need rather than invest in fixed, unused and therefore wasted capacity.
- Makes development possible for 'non-experts'; with some PaaS offerings anyone can develop an application. They can simply do this through their web browser utilizing one-click functionality. Salient examples of this are one-click blog software installs such as Word Press.
- Flexibility; customers can have control over the tools that are installed within their platforms and can create a platform that suits their specific requirements. They can 'pick and choose' the features they feel are necessary.
- Adaptability; Features can be changed if circumstances dictate that they should.
- Teams in various locations can work together; as an internet connection and web browser are all that is required, developers spread across several locations can work together on the same application build.
- Security; security is provided, including data security and backup and recovery.

In summary, a PaaS offering supplies an operating environment for developing applications. In other words, it provides the architecture as well as the overall infrastructure to support application development. This includes networking, storage, and software support and management services. It is therefore ideal for the development of new applications that are intended for the web as well as mobile devices and PCs.

Software as a service (SaaS)

In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis. SaaS providers generally price applications using a subscription fee.

In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. Cloud applications are different from other applications in their scalability—which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access point. To accommodate a large number of cloud users, cloud applications can be *multitenant*, that is, any machine serves more than one cloud user organization.

The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point. Proponents claim SaaS allows a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and personnel expenses, towards meeting other goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software. One drawback of SaaS is that the users' data are stored on the cloud provider's server. As a result, there could be unauthorized access to the data. For this reason, users are increasingly adopting intelligent third-party key management systems to help secure their data.

SaaS, or Software as a Service, describes any cloud service where consumers are able to access software applications over the internet. The applications are hosted in "the cloud" and can be used for a wide range of tasks for both individuals and organizations. Google, Twitter, Facebook and Flickr are all examples of SaaS, with users able to access the services via any internet enabled device. Enterprise users are able to use applications for a range of needs, including accounting and invoicing, tracking sales, planning, performance monitoring and communications (including webmail and instant messaging).

SaaS is often referred to as software-on-demand and utilising it is akin to renting software rather than buying it. With traditional software applications you would purchase the software upfront as a package and then install it onto your computer. The software's license may also limit the number of users and/or devices where the software can be deployed. Software as a Service users, however, subscribe to the software rather than purchase it, usually on a monthly basis. Applications are purchased and used online with files saved in the cloud rather than on individual computers.

There are a number of reasons why SaaS is beneficial to organisations and personal users alike:

- No additional hardware costs; the processing power required to run the applications is supplied by the cloud provider.
- No initial setup costs; applications are ready to use once the user subscribes.
- Pay for what you use; if a piece of software is only needed for a limited period then it is only paid for over that period and subscriptions can usually be halted at any time.
- Usage is scalable; if a user decides they need more storage or additional services, for example, then they can access these on demand without needing to install new software or hardware.
- Updates are automated; whenever there is an update it is available online to existing customers, often free of charge. No new software will be required as it often is with other types of applications and the updates will usually be deployed automatically by the cloud provider.
- Cross device compatibility; SaaS applications can be accessed via any internet enabled device, which makes it ideal for those who use a number of different devices, such as internet enabled phones and tablets, and those who don't always use the same computer.
- Accessible from any location; rather than being restricted to installations on individual computers, an application can be accessed from anywhere with an internet enabled device.
- Applications can be customised and whitelabelled; with some software, customisation is available meaning it can be altered to suit the needs and branding of a particular customer.

Office software is the best example of businesses utilising SaaS. Tasks related to accounting, invoicing, sales and planning can all be performed through Software as a Service. Businesses may wish to use one piece of software that performs all of these tasks or several that each perform different tasks. The required software can be subscribed to via the internet and then accessed online via any computer in the office using a username and password. If needs change they can easily switch to software that better meets their requirements. Everyone who needs access to a particular piece of software can be set up as a user, whether it is one or two people or every employee in a corporation that employs hundreds.

2.2 DEPLOYMENT MODELS

Public Cloud

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free or offered on a pay-per-usage model. Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called "AWS Direct Connect" and "Azure ExpressRoute" respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

The deployment of a public cloud computing system is characterized on the one hand by the public availability of the cloud service offering and on the other hand by the public network that is used to communicate with the cloud service. The cloud services and cloud resources are procured from very large resource pools that are shared by all end users. These IT factories, which tend to be speciûcally built for running cloud computing systems, provision the resources precisely according to required quantities. By optimizing operation, support, and maintenance, the cloud provider can achieve signiûcant economies of scale, leading to low prices for cloud resources. In addition, public cloud portfolios employ techniques for resource optimization; however, these are transparent for end users and represent a potential threat to the security of the system. If a cloud provider runs several datacenters, for instance, resources can be assigned in such a way that the load is uniformly distributed between all centers.

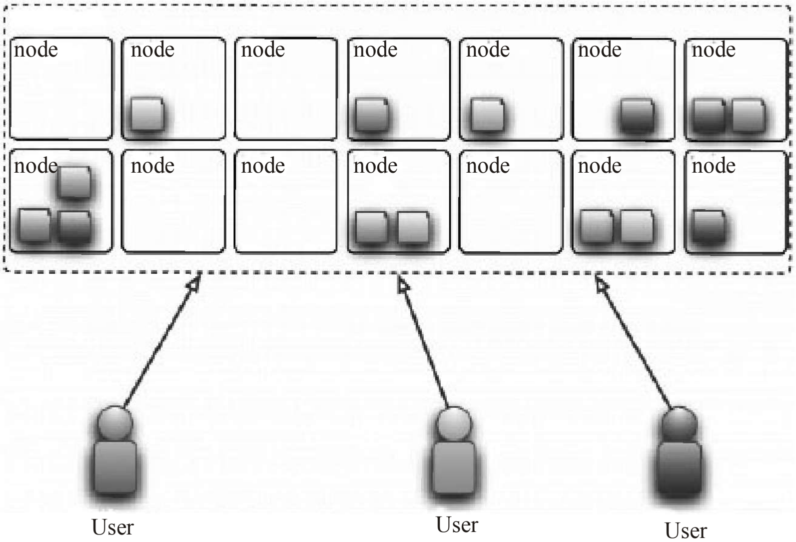


Fig. 2.2: Three users accessing a public cloud

Some of the best-known examples of public cloud systems are Amazon Web Services (AWS) containing the Elastic Compute Cloud (EC2) and the Simple Storage Service (S3) which form an IaaS cloud offering and the Google App Engine with provides a PaaS to its customers. The customer relationship management (CRM) solution Salesforce.com is the best-known example in the area of SaaS cloud offerings.

Private Cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management, essentially "[lacking] the economic model that makes cloud computing such an intriguing concept".

Private cloud computing systems emulate public cloud service offerings within an organization's boundaries to make services accessible for one designated organization. Private cloud computing systems make use of virtualization solutions and focus on consolidating distributed IT services often within data centers belonging to the company. The chief advantage of these systems is that the enterprise retains full control over corporate data, security guidelines, and system performance. In contrast, private cloud offerings are usually not as large-scale as public cloud offerings resulting in worse economies of scale.

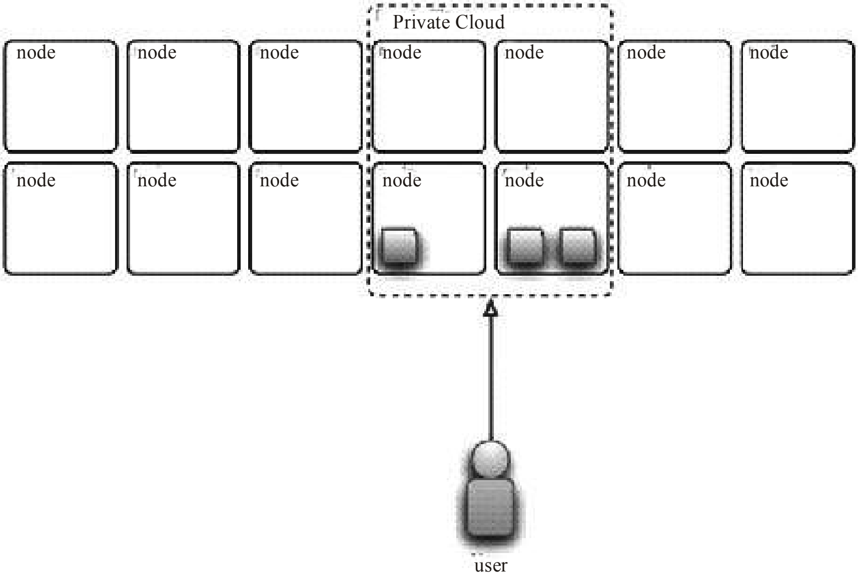


Fig. 2.3: A user Accessing a Private Cloud

Community Cloud

In a community cloud, organizations with similar requirements share a cloud infrastructure. It may be understood as a generalization of a private cloud, a private cloud being an infrastructure which is only accessible by one certain organization.

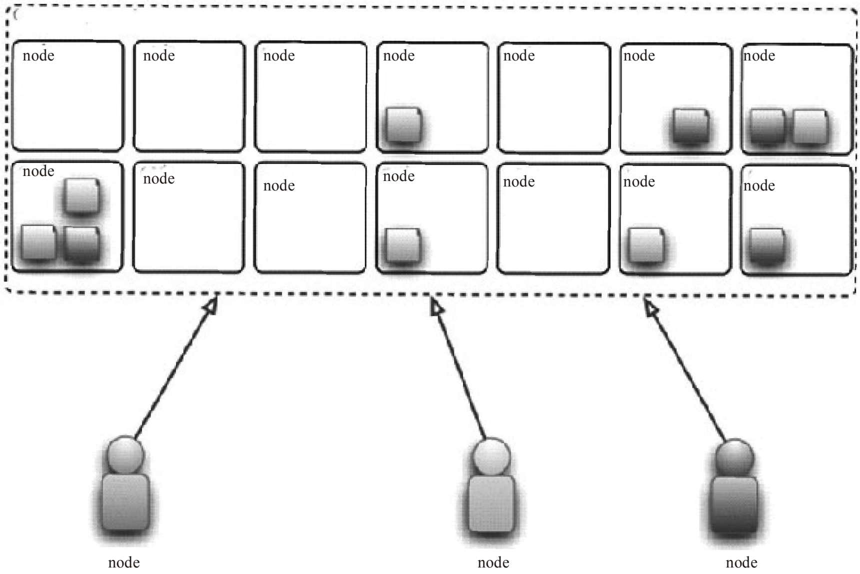


Fig. 2.4: Three Users Accessing a Community Cloud

Hybrid Cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources.

Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers.[64] A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

Varied use cases for hybrid cloud composition exist. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service. This example of hybrid cloud extends the capabilities of the enterprise to deliver a specific business service through the addition of externally available public cloud services.

Another example of hybrid cloud is one where IT organizations use public cloud computing resources to meet temporary capacity needs that cannot be met by the private cloud. This capability enables hybrid clouds to employ cloud bursting for scaling across clouds. Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases. A primary advantage of cloud bursting and a hybrid cloud model is that an organization only pays for extra compute resources when they are needed. Cloud bursting enables data centers to create an in-house IT infrastructure that supports average workloads, and use cloud resources from public or private clouds, during spikes in processing demands.

A hybrid cloud service deployment model implements the required processes by combining the cloud services of different cloud computing systems, e.g. private and public cloud services. The hybrid model is also suitable for enterprises in which the transition to full outsourcing has already been completed, for instance, to combine community cloud services with public cloud services.

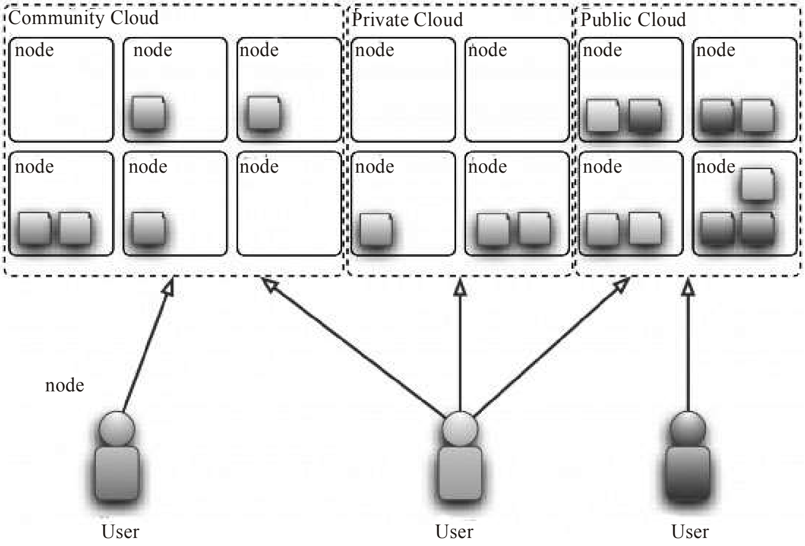


Fig. 2.5: Hybrid Cloud Usage

IMPORTANT QUESTIONS

A: 2 Marks Questions

1. What are service model of cloud computing?
2. What is Iaas?
3. What is Paas?
4. What is Iaas?
5. Write deployment models.
6. What is Private cloud?
7. What is Public cloud?
8. What is Hybrid cloud?
9. What is Community Cloud?

B: 4 Marks Questions

10. Explain service model of cloud computing in details.
11. Differentiate between Iaas, Paas, and Iaas?
12. Explain deployment models.
13. Differentiate between Private Cloud and Public Cloud.
14. Differentiate between Hybrid Cloud and Community Cloud.

C: 10 Marks Questions

15. Explain all service models in details
16. Explain deployment models in details.
17. Differentiate between Private Cloud, Public Cloud, Hybrid Cloud and Community Cloud.

Unit- III

Grid Computing

UNIT-I

3

GRID COMPUTING

Grid Computing

Grid computing is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing in that grid computers have each node set to perform a different task/application. Grid computers also tend to be more heterogeneous and geographically dispersed (thus not physically coupled) than cluster computers. Although a single grid can be dedicated to a particular application, commonly a grid is used for a variety of purposes. Grids are often constructed with general-purpose grid middleware software libraries.

The term grid computing originated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid.

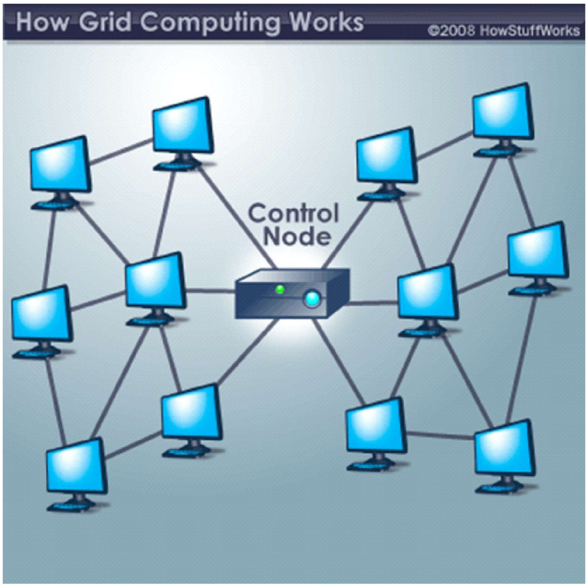


Fig. 3.1: Hybrid Cloud Usage

Overview of Grid Computing

Grid computing combines computers from multiple administrative domains to reach a common goal, to solve a single task, and may then disappear just as quickly.

One of the main strategies of grid computing is to use middleware to divide and apportion pieces of a program among several computers, sometimes up to many thousands. Grid computing involves computation in a distributed fashion, which may also involve the aggregation of large-scale clusters.

The size of a grid may vary from small—confined to a network of computer workstations within a corporation, for example—to large, public collaborations across many companies and networks. "The notion of a confined grid may also be known as intra-nodes cooperation whilst the notion of a larger, wider grid may thus refer to inter-nodes cooperation".

Grids are a form of distributed computing whereby a "super virtual computer" is composed of many networked loosely coupled computers acting together to perform very large tasks. This technology has been applied to computationally intensive scientific, mathematical, and academic problems through volunteer computing, and it is used in commercial enterprises for such diverse applications as drug discovery, economic forecasting, seismic analysis, and back office data processing in support for e-commerce and Web services.

Comparison of Grids and Conventional Supercomputers

"Distributed" or "grid" computing in general is a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface producing commodity hardware, compared to the lower efficiency of designing and constructing a small number of custom supercomputers. The primary performance disadvantage is that the various processors and local storage areas do not have high-speed connections. This arrangement is thus well-suited to applications in which multiple parallel computations can take place independently, without the need to communicate intermediate results between processors. The high-end scalability of geographically dispersed grids is generally favorable, due to the low need for connectivity between nodes relative to the capacity of the public Internet

There are also some differences in programming and deployment. It can be costly and difficult to write programs that can run in the environment of a supercomputer, which may have a custom operating system, or require the program to address concurrency issues. If a problem can be adequately parallelized, a "thin" layer of "grid" infrastructure can allow conventional, standalone programs, given a different part of the same problem, to run on multiple machines. This makes it possible to write and debug on a single conventional machine, and eliminates complications due to multiple instances of the same program running in the same shared memory and storage space at the same time.

Design Considerations and Variations

One feature of distributed grids is that they can be formed from computing resources belonging to multiple individuals or organizations (known as multiple administrative domains). This can facilitate commercial transactions, as in utility computing, or make it easier to assemble volunteer computing networks.

One disadvantage of this feature is that the computers which are actually performing the calculations might not be entirely trustworthy. The designers of the system must thus introduce measures to prevent malfunctions or malicious participants from producing false, misleading, or erroneous results, and from using the system as an attack vector. This often involves assigning work randomly to different nodes (presumably with different owners) and checking that at least two different nodes report the same answer for a given work unit. Discrepancies would identify malfunctioning and malicious nodes. However, due to the lack of central control over the hardware, there is no way to guarantee that nodes will not drop out of the network at random times. Some nodes (like laptops or dialup Internet customers) may also be available for computation but not network communications for unpredictable periods. These variations can be accommodated by assigning large work units (thus reducing the need for continuous network connectivity) and reassigning work units when a given node fails to report its results in expected time.

The impacts of trust and availability on performance and development difficulty can influence the choice of whether to deploy onto a dedicated cluster, to idle machines internal to the developing organization, or to an open external network of volunteers or contractors. In many cases, the participating nodes must trust the central system not to abuse the access that is being granted, by interfering with the operation of other programs, mangling stored information, transmitting private data, or creating new security holes. Other systems employ measures to reduce the amount of trust "client" nodes must place in the central system such as placing applications in virtual machines.

Public systems or those crossing administrative domains (including different departments in the same organization) often result in the need to run on heterogeneous systems, using different operating systems and hard ware architectures. With many languages, there is a tradeoff between investment in software development and the number of platforms that can be supported (and thus the size of the resulting network). Cross-platform languages can reduce the need to make this trade off, though potentially at the expense of high performance on any given node (due to run-time interpretation or lack of optimization for the particular platform). There are diverse scientific and commercial projects to harness a particular associated grid or for the purpose of setting up new grids. BOINC is a common one for various academic projects seeking public volunteers; more are listed at the end of the article.

In fact, the middleware can be seen as a layer between the hardware and the software. On top of the middleware, a number of technical areas have to be considered, and these may or may not be middleware independent. Example areas include SLA management, Trust and Security, Virtual organization management, License Management, Portals and Data Management. These technical areas may be taken care of in a commercial solution, though the cutting edge of each area is often found within specific research projects examining the field.

Market segmentation of the grid computing market

For the segmentation of the grid computing market, two perspectives need to be considered: the provider side and the user side:

The provider side

The overall grid market comprises several specific markets. These are the grid middleware market, the market for grid-enabled applications, the utility computing market, and the software-as-a-service (SaaS) market.

Grid middleware is a specific software product, which enables the sharing of heterogeneous resources, and Virtual Organizations. It is installed and integrated into the existing infrastructure of the involved company or companies, and provides a special layer placed among the heterogeneous infrastructure and the specific user applications. Major grid middlewares are Globus Toolkit, gLite, and UNICORE.

Utility computing is referred to as the provision of grid computing and applications as service either as an open grid utility or as a hosting solution for one organization or a VO. Major players in the utility computing market are Sun Microsystems, IBM, and HP.

Grid-enabled applications are specific software applications that can utilize grid infrastructure. This is made possible by the use of grid middleware, as pointed out above.

Software as a service (SaaS) is "software that is owned, delivered and managed remotely by one or more providers." (Gartner 2007) Additionally, SaaS applications are based on a single set of common code and data definitions. They are consumed in a one-to-many model, and SaaS uses a Pay As You Go (PAYG) model or a subscription model that is based on usage. Providers of SaaS do not necessarily own the computing resources themselves, which are required to run their SaaS. Therefore, SaaS providers may draw upon the utility computing market. The utility computing market provides computing resources for SaaS providers.

The user side

For companies on the demand or user side of the grid computing market, the different segments have significant implications for their IT deployment strategy. The IT deployment strategy as well as the type of IT investments made are relevant aspects for potential grid users and play an important role for grid adoption.

Advantages of Grid Computing

1. Can solve larger, more complex problems in a shorter time
2. Easier to collaborate with other organizations
3. Make better use of existing hardware
4. No need to buy large six figure SMP servers for applications that can be split up and farmed out to smaller commodity type servers. Results can then be concatenated and analyzed upon job(s) completion.
5. Much more efficient use of idle resources. Jobs can be farmed out to idle servers or even idle desktops. Many of these resources sit idle especially during off business hours. Policies can be in place that allows jobs to only go to servers that are lightly loaded or have the appropriate amount of memory/cpu characteristics for the particular application.
6. Grid environments are much more modular and don't have single points of failure. If one of the servers/desktops within the grid fail there are plenty of other resources able to pick the load. Jobs can automatically restart if a failure occurs.
7. Policies can be managed by the grid software. The software is really the brains behind the grid. A client will reside on each server which send information back to the master telling it what type of availability or resources it has to complete incoming jobs.
8. This model scales very well. Need more compute resources? Just plug them in by installing grid client on additional desktops or

servers. They can be removed just as easily on the fly. This modular environment really scales well.

9. Upgrading can be done on the fly without scheduling downtime. Since there are so many resources some can be taken offline while leaving enough for work to continue. This way upgrades can be cascaded as to not effect ongoing projects.
10. Jobs can be executed in parallel speeding performance. Grid environments are extremely well suited to run jobs that can be split into smaller chunks and run concurrently on many nodes. Using things like MPI will allow message passing to occur among computer resources.
11. Grid computing enables organizations to aggregate resources within an entire IT infrastructure no matter where in the world they are located. It eliminates situations where one site is running on maximum capacity, while others have cycles to spare.
12. Organizations can dramatically improve the quality and speed of the products and services they deliver, while reducing IT costs by enabling transparent collaboration and resource sharing.
13. Grid computing enables companies to access and share remote databases. This is especially beneficial to the life sciences and research communities, where enormous volumes of data are generated and analysed during any given day.
14. Grid computing enables widely dispersed organizations to easily collaborate on projects by creating the ability to share everything from software applications and data, to engineering blueprints.
15. Grid computing can create a more robust and resilient IT infrastructure better able to respond to minor or major disasters.
16. A grid can harness the idle processing cycles that are available in desktop PCs located in various locations across multiple time zones. For example, PCs that would typically remain idle overnight at a company's Asian manufacturing plant could be utilized during the day by its European operations.

Disadvantages of Grid Computing

1. Grid software and standards are still evolving
2. Learning curve to get started
3. Non-interactive job submission
4. For memory hungry applications that can't take advantage of MPI you may be forced to run on a large SMP.

5. You may need to have a fast interconnect between compute resources (gigabit ethernet at a minimum). Infiband for MPI intense applications
6. Some applications may need to be tweaked to take full advantage of the new model.
7. Licensing across many servers may make it prohibitive for some apps. Vendors are starting to be more flexible with environment like this.
8. Grid environments include many smaller servers across various administrative domains. Good tools for managing change and keeping configurations in syncwith each other can be challenging in large environments. Tools exist to manage such challenges include systemimager, cfengine, Opsware, Bladelogic, pdsh, cssh, among others.

VIRTUAL ORGANIZATION IN GRID COMPUTING

In grid computing, a virtual organization (VO) refers to a dynamic set of individuals or institutions defined around a set of resource rules and conditions. All these virtual organizations share some commonality among them, including common concerns and requirements, but may vary in size, scope, duration, sociology, and structure.

The collaborations involved in grid computing of the early 2000s lead to the emergence of multiple organizations that function as one unit through the use of their shared competencies and resources for the purpose of one or more identified goals.

Virtual organization members (A–J) can group together dynamically, to use resources such as a database

A virtual organization has the characteristics of a formal organization while not being one. It comprises a complex network of smaller organizations which each contribute a part of the production process. Boundaries between organizations are fuzzy; control is generally by market forces, reinforced by the certainty of long- term contracts.

The XtreemOS project promised to support virtual organizations

Virtual organisations (VOs) are groups of researchers with similar scientific interests and requirements, who are able to work collaboratively with other members and/or share resources (e.g. data, software, expertise, CPU, storage space), regardless of geographical location.

Researchers must join a VO in order to use grid computing resources provided by EGI. Each virtual organisation manages its own membership list, according to the VO's requirements and goals.

EGI provides support, services and tools to allow VOs to make the most of their resources.

EGI currently hosts more than 200 VOs for communities with interests as diverse as Earth Sciences, Computer Sciences and Mathematics, Fusion, Life Sciences or High-Energy Physics.

Grid Computing Applications

- Application partitioning that involves breaking the problem into discrete pieces
- Discovery and scheduling of tasks and workflow
- Data communications distributing the problem data where and when it is required
- Provisioning and distributing application codes to specific system nodes
- Results management assisting in the decision processes of the environment
- Autonomic features such as self-configuration, self-optimization, self-recovery, and self-management

Let us now explore some of these Grid applications and their usage patterns. We start with schedulers, which form the core component in most of the computational grids.

Schedulers

Schedulers are types of applications responsible for the management of jobs, such as allocating resources needed for any specific job, partitioning of jobs to schedule parallel execution of tasks, data management, event correlation, and service-level management capabilities. These schedulers then form a hierarchical structure, with meta-schedulers that form the root and other lower level schedulers, while providing specific scheduling capabilities that form the leaves. These schedulers may be constructed with a local scheduler implementation approach for specific job execution, or another meta-scheduler or a cluster scheduler for parallel executions. Figure shows this concept.

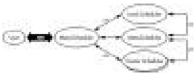


Figure The scheduler hierarchy embodies local, meta-level, and cluster schedulers.

The jobs submitted to Grid Computing schedulers are evaluated based on their service-level requirements, and then allocated to the respective resources for execution. This will involve complex workflow management and data movement activities to occur on a regular basis. There are schedulers that must provide capabilities for areas such as (but not limited to):

- Advanced resource reservation
- Service-level agreement validation and enforcement
- Job and resource policy management and enforcement for best turnaround times within the allowable budget constraints
- Monitoring job executions and status
- Rescheduling and corrective actions of partial failover situations

Resource Broker

The resource broker provides *pairing* services between the service requester and the service provider. This pairing enables the selection of best available resources from the service provider for the execution of a specific task. These resource brokers collect information (e.g., resource availability, usage models, capabilities, and pricing information) from the respective resources, and use this information source in the pairing process.

Load Balancing

The Grid Computing infrastructure load-balancing issues are concerned with the traditional load-balancing distribution of workload among the resources in a Grid Computing environment. This load-balancing feature must always be integrated into any system in order to avoid processing delays and over commitment of resources. These kinds of applications can be built in connection with schedulers and resource managers.

Grid Portals

Grid portals are similar to Web portals, in the sense they provide uniform access to the grid resources. For example, grid portals provide capabilities for Grid Computing resource authentication, remote resource access, scheduling capabilities, and monitoring status information. These kinds of portals help to alleviate the complexity of task management through customizable and personalized graphical interfaces for the users. This, in turn, alleviates the need for end users to have more domain knowledge than on the specific details of grid resource management.

Some examples of these grid portal capabilities are noted in the following list:

- Querying databases or *LDAP* servers for resource-specific information
- File transfer facilities such as file upload, download, integration with custom software, and so on

- Manage job through job status feedbacks
- Allocate the resources for the execution of specific tasks
- Security management
- Provide personalized solutions

In short, these grid portals help free end users from the complexity of job management and resource allocation so they can concentrate more on their domain of expertise. There are a number of standards and software development toolkits available to develop custom portals. The emerging Web services and Web service portal standards will play a more significant role in portal development.

Integrated Solutions

Many of the global industry sectors have witnessed the emergence of a number of integrated grid application solutions in the last few years. This book focuses on this success factor.

These integrated solutions are a combination of the existing advanced middleware and application functionalities, combined to provide more coherent and high performance results across the Grid Computing environment.

Integrated Grid Computing solutions will have more enhanced features to support more complex utilization of grids such as coordinated and optimized resource sharing, enhanced security management, cost optimizations, and areas yet to be explored. It is straightforward to see that these integrated solutions in both the commercial and noncommercial worlds sustain high values and significant cost reductions. Grid applications can achieve levels of flexibility utilizing infrastructures provided by application and middleware frameworks.

IMPORTANT QUESTIONS

A: 2 Marks Questions

1. What is grid computing?
2. Write any two advantages of grid computing?
3. Write any two disadvantages of grid computing?
4. What is Virtual organization in grid computing?
5. Write applications of grid computing.
6. What is Resource Broker?
7. What is Load Balancing?

B: 4 Marks Questions

8. Explain grid computing in details.
9. Explain virtual organization in grid computing.
10. Write advantages of grid computing.
11. Write applications of grid computing.
12. Compare grids and conventional supercomputers.
13. Discuss design consideration in grid computing.

C: 10 Marks Questions

14. Discuss market segmentation of the grid computing market.
15. Explain advantages and disadvantages of grid computing.
16. Write applications of grid computing.
17. Discuss grid computing with its design considerations and variations.

Unit- IV

Other Technologies

4.1 Cluster Computing
4.2 Peer to Peer Networks
4.3 Utility Computing
4.4 Ubiquitous computing
4.5 Comparison of Grid, Cluster and Cloud Computing

UNIT

4

OTHER TECHNOLOGIES

4.1 CLUSTER COMPUTING

Cluster computing can be described as a fusion of the fields of parallel, high-performance, distributed, and high-availability computing. Cluster computing has become a hot topic of research among academic and industry community including system designers, network developers, and language designers, standardizing forums, algorithm developers, graduate students and faculties.

The use of clusters as computing platform is not just limited to scientific and engineering applications; there are many business applications that can benefit from the use of clusters. There are many exciting areas of development in cluster computing with new ideas as well as hybrids of old ones being deployed for production as well as research systems. Clusters are based on the communication between nodes and designing fast and low latency networks is a must for clusters to become the configuration of the future. Plenty of work being done in this are in both hardware (Infiniband, SCI, Myrinet, QSnet) and software (Low latency protocols such as VIA). For scientific application, clusters can be used in grand challenge or supercomputing applications, such as earthquakes or hurricanes prediction, complex crystallographic and microtomographic structural problems, protein dynamics and biocatalysts, relativistic quantum chemistry of actinides, virtual materials design and processing including crash simulations, and global climate modeling. For the commercial applications, cluster can be best used in e-commerce as superserver, which consolidates web server, ftp server, e-mail server, database server, etc. Clusters can also be used in data mining applications to provide the storage and data management services for the data set being mined and computational services required by the data filtering, preparation and mining tasks. Other commercial applications include image rendering, network simulation, etc.

4.2 PEER TO PEER NETWORKS

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

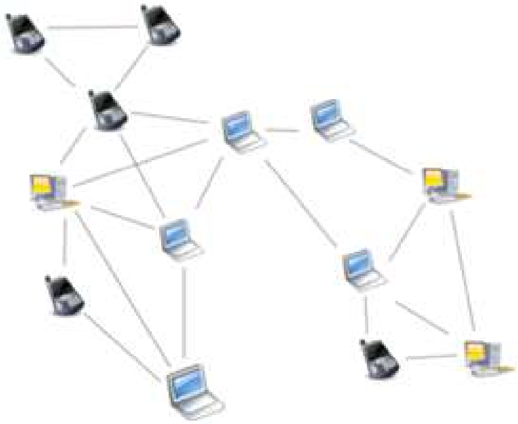
Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model in which the consumption and supply of resources is divided. Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual community thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers, yet that are beneficial to all the peers.

Architecture

A peer-to-peer network is designed around the notion of equal *peer* nodes simultaneously functioning as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client–server model where communication is usually to and from a central server. A typical example of a file transfer that uses the client-server model is the File Transfer Protocol (FTP) service in which the client and server programs are distinct: the clients initiate the transfer, and the servers satisfy these requests.

Peer-to-peer networks generally implement some form of virtual overlay network on top of the physical network topology, where the nodes in the overlay form a subset of the nodes in the physical network. Data is still exchanged directly over the underlying TCP/IP network, but at the application layer peers are able to communicate with each other directly, via the logical overlay links (each of which corresponds to a path through the underlying physical network). Overlays are used for indexing and peer discovery, and make the P2P system independent from the physical network topology. Based on how the nodes are linked to each other within the overlay network, and how resources are indexed and located, we can classify networks as *unstructured* or *structured* (or as a hybrid between the two).

Unstructured networks



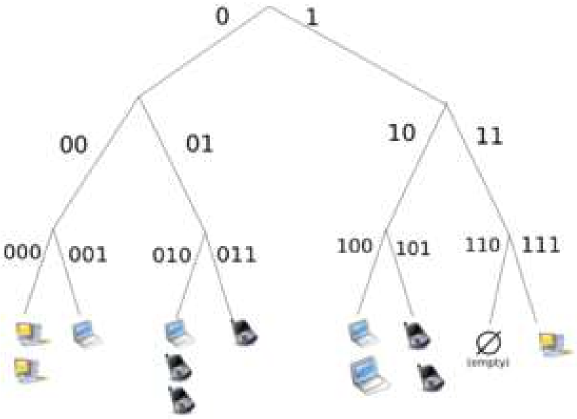
Overlay network diagram for an **unstructured P2P network**, illustrating the ad hoc nature of the connections between nodes

Unstructured peer-to-peer networks do not impose a particular structure on the overlay network by design, but rather are formed by nodes that randomly form connections to each other. (Gnutella,Gossip, and Kazaa are examples of unstructured P2P protocols).

Because there is no structure globally imposed upon them, unstructured networks are easy to build and allow for localized optimizations to different regions of the overlay.

However the primary limitations of unstructured networks also arise from this lack of structure. In particular, when a peer wants to find a desired piece of data in the network, the search query must be flooded through the network to find as many peers as possible that share the data. Flooding causes a very high amount of signaling traffic in the network, uses more CPU/memory (by requiring every peer to process all search queries), and does not ensure that search queries will always be resolved. Furthermore, since there is no correlation between a peer and the content managed by it, there is no guarantee that flooding will find a peer that has the desired data. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful.

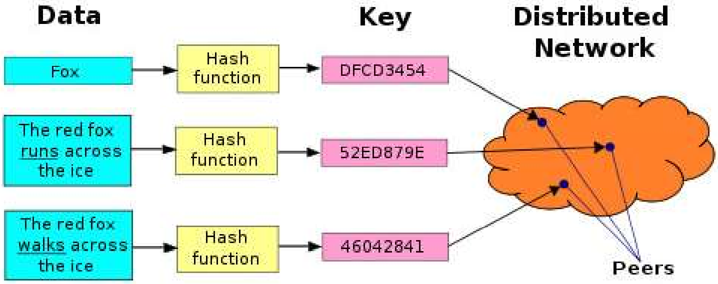
Structured networks



Overlay network diagram for a structured P2P network, using a distributed hash table (DHT) to identify and locate nodes/resources

In *structured peer-to-peer networks* the overlay is organized into a specific topology, and the protocol ensures that any node can efficiently search the network for a file/resource, even if the resource is extremely rare.

The most common type of structured P2P networks implement distributed (DHT), in which a variant of consistent hashing is used to assign ownership of each file to a particular peer. This enables peers to search for resources on the network using a hash table: that is, (*key*, *value*) pairs are stored in the DHT, and any participating node can efficiently retrieve the value associated with a given key.



Distributed hash tables

However, in order to route traffic efficiently through the network, nodes in a structured overlay must maintain lists of neighbors that satisfy specific criteria. This makes them less robust in networks. More recent evaluation of P2P resource discovery solutions under real workloads have pointed out several issues in DHT-based solutions such as high cost of advertising/ discovering resources and static and dynamic load imbalance.

Hybrid models

Hybrid models are a combination of peer-to-peer and client-server models. A common hybrid model is to have a central server that helps peers find each other. Spotify is an example of a hybrid model. There are a variety of hybrid models, all of which make trade-offs between the centralized functionality provided by a structured server/client network and the node equality afforded by the pure peer-to-peer unstructured networks. Currently, hybrid models have better performance than either pure unstructured networks or pure structured networks because certain functions, such as searching, do require a centralized functionality but benefit from the decentralized aggregation of nodes provided by unstructured networks.

Routing attacks

Also, since each node plays a role in routing traffic through the network, malicious users can perform a variety of "routing attacks".. Examples of common routing attacks include "incorrect lookup routing" whereby malicious nodes deliberately forward requests incorrectly or return false results, "incorrect routing updates" where malicious nodes corrupt the routing tables of neighboring nodes by sending them false information, and "incorrect routing network partition" where when new nodes are joining they bootstrap via a malicious node, which places the new node in a partition of the network that is populated by other malicious nodes.[38]

Corrupted data and malware

The prevalence of malware varies between different peer-to-peer protocols. Studies analyzing the spread of malware on P2P networks found, for example, that 63% of the answered download requests on the Limewire network contained some form of malware, whereas only 3% of the content on OpenFT contained malware. In both cases, the top three most common types of malware accounted for the large majority of cases (99% in Limewire, and 65% in OpenFT). Another study analyzing traffic on the Kazaa network found that 15% of the 500,000 file sample taken were infected by one or more of the 365 different computer viruses that were tested for.

Corrupted data can also be distributed on P2P networks by modifying files that are already being shared on the network. For example, on the FastTrack network, the RIAA managed to introduce faked chunks into downloads and downloaded files (mostly MP3 files). Files infected with the RIAA virus were unusable afterwards and contained malicious code. The RIAA is also known to have uploaded fake music and movies to P2P networks in order to deter illegal file sharing. Consequently, the P2P networks of today have seen an enormous increase of their security and file verification mechanisms. Modern hashing, chunk verification and different encryption methods have made most networks resistant to almost any type of attack, even when major parts of the respective network have been replaced by faked or nonfunctional hosts.

4.3 UTILITY COMPUTING

Utility computing is a model in which a service provider makes computing resources and infrastructure management available to the customer as needed, and charges them for specific usage rather than a flat rate. Like other types of on-demand computing (such as grid computing), the utility model seeks to maximize the efficient use of resources and/or minimize associated costs. Utility is the packaging of computing resources, such as computation, storage and services, as a metered service. This model has the advantage of a low or no initial cost to acquire computer resources; instead, computational resources are essentially rented.

This repackaging of computing services became the foundation of the shift to "on demand" computing, software as a serviceand cloud computing models that further propagated the idea of computing, application and network as a service.

There was some initial skepticism about such a significant shift. However, the new model of computing caught on and eventually became mainstream.

IBM, HP and Microsoft were early leaders in the new field of Utility Computing with their business units and researchers working on the architecture, payment and development challenges of the new computing model. Google, Amazon and others started to take the lead in 2008, as they established their own utility services for computing, storage and applications.

Utility Computing can support grid computing which has the characteristic of very large computations or a sudden peaks in demand which are supported via a large number of computers.

"Utility computing" has usually envisioned some form of virtualization so that the amount of storage or computing power available is considerably larger than that of a single time-sharing computer. Multiple servers are used on the "back end" to make this possible. These might be a dedicated computer cluster specifically built for the purpose of being rented out, or even an under-utilized supercomputer. The technique of running a single calculation on multiple computers is known as distributed.

The term "grid computing" is often used to describe a particular form of distributed computing, where the supporting nodes are geographically distributed or cross administrative domains. To provide utility computing services, a company can "bundle" the resources of members of the public for sale, who might be paid with a portion of the revenue from clients.

One model, common among volunteer computing applications, is for a central server to dispense tasks to participating nodes, on the behest of approved end-users (in the commercial case, the paying customers). Another model, sometimes called the Virtual Organization (VO) is more decentralized, with organizations buying and selling computing resources as needed or as they go idle.

4.4 UBIQUITOUS COMPUTING

Ubiquitous computing (*ubicomp*) is a concept in software engineering and computer science where computing is made to appear everywhere and anywhere. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format. A user interacts with the computer, which can exist in many different forms, including laptop computers, tablets and terminals in everyday objects such as a fridge or a pair of glasses. The underlying technologies to support ubiquitous computing include Internet, advanced middleware, operating system, mobile code, sensors, microprocessors, new I/O and user interfaces, networks, mobile protocols, location and positioning and new materials.

Ubiquitous computing touches on a wide range of research topics, including distributed computing, mobile computing, location computing, mobile networking, context-aware computing, sensor networks, human-computer interaction, andartificial intelligence.

At their core, all models of ubiquitous computing share a vision of small, inexpensive, robust networked processing devices, distributed at all scales throughout everyday life and generally turned to distinctly common-place ends. For example, a domestic ubiquitous computing environment might interconnect lighting and environmental controls with personal biometric monitors woven into clothing so that illumination and heating conditions in a room might be modulated, continuously and imperceptibly. Another common scenario posits refrigerators "aware" of their suitably tagged contents, able to both plan a variety of menus from the food actually on hand, and warn users of stale or spoiled food.

Ubiquitous computing presents challenges across computer science: in systems design and engineering, in systems modelling, and in user interface design. Contemporary human-computer interaction models, whether command-line, menu-driven, or GUI-based, are inappropriate and inadequate to the ubiquitous case. This suggests that the "natural" interaction paradigm appropriate to a fully robust ubiquitous computing has yet to emerge - although there is also recognition in the field that in many ways we are already living in a ubicomp world. Contemporary devices that lend some support to this latter idea include mobile phones, digital audio players, radio-frequency identification tags, GPS, and interactive whiteboards.

Mark Weiser proposed three basic forms for ubiquitous system devices: tabs, pads and boards.

- *Tabs*: wearable centimetre sized devices
- *Pads*: hand-held decimetre-sized devices
- *Boards*: metre sized interactive display devices.

Ubiquitous computing may be seen to consist of many layers, each with their own roles, which together form a single system:

Layer 1: *task management layer*

- Monitors user task, context and index
- Map user's task to need for the services in the environment
- To manage complex dependencies

Layer 2: *environment management layer*

- To monitor a resource and its capabilities
- To map service need, user level states of specific capabilities

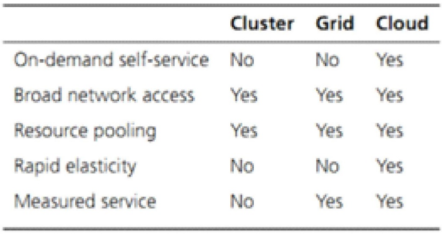
Layer 3: *environment layer*

- To monitor a relevant resource
- To manage reliability of the resources

One of the earliest ubiquitous systems was artist Natalie Jeremijenko's "Live Wire", also known as "Dangling String", installed at Xerox PARC during Mark Weiser's time there. This was a piece of string attached to a stepper motor and controlled by a LAN connection; network activity caused the string to twitch, yielding a *peripherally noticeable* indication of traffic.

4.5 COMPARISON OF GRID, CLUSTER AND CLOUD COMPUTING

Often the valid question arises if cloud computing is actually something new and, if so, what exactly distinguishes this new paradigm from former computing models like cluster and grid computing. Summarized in table 1, five essential characteristics of cloud computing are applied to cluster and grid computing. All three systems are distributed and share similar characteristics. The similarities relate to resource pooling and broad network access – two criteria that are fulfilled by all systems. Network access to cluster and grid computing systems usually takes place within a corporate network, while the services of a cloud computing system can also be accessed through public network, i.e. the Internet.



	Cluster	Grid	Cloud
On-demand self-service	No	No	Yes
Broad network access	Yes	Yes	Yes
Resource pooling	Yes	Yes	Yes
Rapid elasticity	No	No	Yes
Measured service	No	Yes	Yes

Table. 1: Comparing Cluster, Grid and Cloud Computing

The differences between cloud computing systems on the one hand and grid and cluster computing systems on the other are attributable to the system dynamics. Resources in grid and cluster environments are generally pre-reserved, while cloud computing systems are demand driven, i.e. operation of these systems is geared to consumers' actual needs. Another difference concerns the "rapid elasticity" criterion, which forms an integral part of cloud computing systems but is not normally supported by cluster or grid systems. Service usage only tends to be accurately measured in grid and cloud computing systems, whereas the majority of cluster environments simply provision rudimentary metering functions.

Compared to other distributed systems such as grids or clusters, cloud computing solutions give enterprises significantly more flexibility. They can dispense with IT infrastructures of their own and only have to pay for the resources and services they actually use ("pay-per-use"/ "pay as you go"). These can be dynamically adapted to changed business requirements and processes with the help of virtualization technologies and service oriented, distributed software systems.

Cluster Computing	Grid Computing	Cloud Computing
The computers in the cluster are normally contained in a single location or complex.	Grid are inherently distributed by its nature over a LAN, metropolitan or WAN	Clouds are mainly distributed over MAN
Characteristics of Cluster computing 1. Tightly coupled systems 2. Single system image 3. Centralized Job management & scheduling system	1: Loosely coupled Computing 1. Loosely coupled (Decentralization) 2. Diversity and Dynamism 3. Distributed Job Management & scheduling	Characteristic of cloud computing 1: Dynamic computing infrastructure 2: IT service-centric approach 3: Self-service based usage model 4: Minimally or self-managed platform 5: Consumption-based billing
In cluster computing, a bunch of similar (or identical) computers are hooked up locally (in the same physical location, directly connected with very high speed connections) to operate as a single computer	In grid computing, the computers do not have to be in the same physical location and can be operated independently. As far as other computers are concerned each computer on the grid is a distinct computer.	In cloud computing, the computers need not to be in the same physical location.
The cluster computers all have the same hardware and OS	The computers that are part of a grid can run different operating systems and have different hardware	The memory, storage device and Network communication is managed by the operating system of the basic physical cloud units. Open source software such as LINUX can support the basic physical unit management and virtualization computing.
Areas of cluster computing 1. Educational resources 2. Commercial sectors for industrial promotion	Areas of Grid Computing 1.Predictive Modeling and Simulations 2.Engineering Design and	Areas of cloud Computing 1. Banking 2. Insurance 3. Weather Forecasting 4. Space Exploration

3. Medical research	Automation 3.Energy Resources Exploration 4.Medical, Military and Basic Research 5.Visualization	5. Software as a service 6. PaaS 7. Infrastructure- as -a-Service
Size or scalability is 100s	Size or scalability is 1000s	Size or scalability is 100s to 1000s
One of the standard OSs (Linux, Windows	Any standard OS (dominated by Unix)	A hypervisor (VM) on which multiple OSs run
Dedicated, high-end with low latency and high bandwidth Interconnection Network	Mostly Internet with high latency and low Bandwidth Interconnection Network	Dedicated, high-end with low latency and high Bandwidth Interconnection Network
Traditional login/password-based. Medium level of privacy depends on user privileges.	Public/private key pair based authentication and mapping a user to an account. Limited support for privacy.	Each user/application is provided with a virtual machine. High security/privacy is guaranteed. Support for setting per-file access control list (ACL).
Membership services discovery	Centralized indexing and decentralized info services discovery	Membership services discovery

IMPORTANT QUESTIONS

A: 2 Marks Questions

1. What is cluster computing?
2. What are Peer to Peer Networks?
3. What is Utility Computing?
4. What is Ubiquitous Computing?
5. What are structured networks?
6. What are unstructured networks?
7. What are routing attacks?
8. What are Hybrid models?

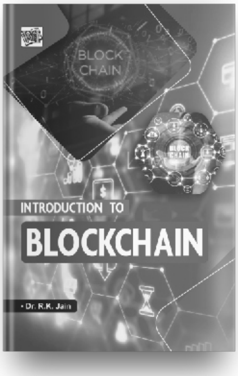
B: 4 Marks Questions

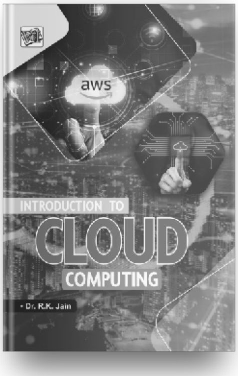
9. Explain cluster computing in details.
10. Differentiate between cluster computing and Peer to Peer Networks?
11. Discuss the architecture of Peer to Peer Networks?
12. Discuss in detail the Utility Computing?
13. What is Ubiquitous Computing? Explain.
14. Differentiate between Utility Computing and Ubiquitous Computing.

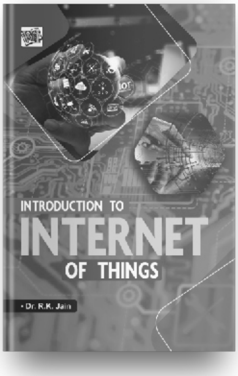
C: 10 Marks Questions

15. Compare Grid, cluster and cloud computing.
16. Explain cluster computing in details.
17. Discuss the architecture of Peer to Peer Networks?
18. **Discuss in detail the Utility Computing and Ubiquitous Computing?**

INDEX


BLOCK CHAIN
INTRODUCTION TO
BLOCKCHAIN
• Dr. R.K. Jain


aws
INTRODUCTION TO
CLOUD
COMPUTING
• Dr. R.K. Jain


INTRODUCTION TO
INTERNET
OF THINGS
• Dr. R.K. Jain