**CH1**

**Distributed system**

A distributed system is an assembly of autonomous, potentially heterogeneous computers or nodes connected through one or more networks. it operates as a single coherent system, Communication and coordination among these nodes are achieved through synchronous or asynchronous message passing. The primary objective is to enable access to shared resources and services across the network, ensuring system-wide integration and functionality.

**Openness:** the degree to which a system is able to add new resources, integrate them, so that they work together regardless of the different underlying environments.

**Scalability**: Scalability is about the ease of expansion of a system to accommodate an increased number of users, more rams and so on and continue working effectively.

**Coarse-Grained:** describes a scenario where operations or data manipulations are done in large units or blocks. The process is divided into fewer tasks, but each task is relatively large. This approach can reduce the overhead of communication and synchronization between tasks.

**Fine-Grained:** refers to operations or data manipulations done in smaller units. In parallel computing. The process is divided into a larger number of smaller tasks. This approach can offer more flexibility.

**heterogeneity**

presents a fundamental challenge in the development and operation of distributed systems by introducing diversity in platforms and standards. **Middleware** serves as a critical technology that bridges this diversity, offering solutions for seamless integration and interoperability.

**Deadlock:** occurs when a group of processes or threads are each waiting for another member of the group to release a resource they need to continue, but none of them can proceed because the resource they are waiting for is being held by another waiting process.

**Livelock**:It is when two or more processes are stuck in a loop responding to each other's actions without any progress.

**P2P (Peer-to-Peer) Computing**

decentralized model where each participant (peer) in the network shares a part of their resources, directly with other participants without the need for centralized coordination by servers. This model is often used for file sharing, distributed computing projects, and cryptocurrencies.

**Cluster Computing**

Cluster computing involves connecting multiple computers (nodes) through a local area network (LAN) so they work together as a single system. Clusters improve performance, reliability, and scalability. used for high-performance computing tasks.

**Utility Computing**

Utility computing is a provisioning model in which computing resources, such as processing power and storage, are provided to users as a metered service similar to traditional utilities like electricity and water. Users pay only for the resources they consume. (grid, cloud)

**Grid Computing**

connects computers from multiple locations, to reach a common goal, utilizing the unused processing cycles of all connected computers. Unlike cluster computing, which is usually localized and homogenous, grid computing is heterogeneous. It's used for large-scale problems that require significant computing resources, such as analyzing large datasets.

**Cloud Computing**

It is a computing model for enabling everywhere, convenient, ondemand network access to a shared pool of configurable resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

**Jungle Computing**

refers to a highly heterogeneous computing environment that combines various types of computing architectures, such as grids, clouds, clusters, and supercomputers, to tackle extremely complex computing tasks.

**CH2**

**cloud computing**

It’s a computational environment that provides transparent access to a shared pool of computational resources matching user need.

**IaaS (Infrastructure as a Service)**

* **Provides**: gives users access to networking features, computers, and data storage space.
* **Control Level**: Users have control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).
* **Use Cases**: IaaS is suitable for creating virtual data centers for companies that need a high level of flexibility and control over their IT resources, especially for businesses looking to manage their infrastructure without the expenses and complexities of physical hardware.

**PaaS (Platform as a Service)**

* **Provides**: Platforms and environments for developers, designed for easier web or mobile apps creation without worrying about setting up or managing underlying infrastructure.
* **Control Level**: Users control the applications (and possibly the application hosting environment configurations), but not the underlying infrastructure.
* **Use Cases**: PaaS is ideal for developers.

**SaaS (Software as a Service)**

* **What It Provides**: Software applications over the internet, on a subscription basis.
* **Control Level**: Users have no control over the infrastructure or platform but can configure the software application within the limits set by the provider.
* **Use Cases**: used for services such as email, customer relationship management (CRM), and document management.
* **IaaS** offers the most flexibility and management control over your resources.
* **PaaS** removes the need for your organization to manage the underlying infrastructure and allows you to focus on the deployment and management of your applications.
* **SaaS** provides you with a completed product that the service provider runs and maintains.

**1. Public Cloud**

* **Description**: provisioned for open use by the public. It may be owned, managed, and operated by a business, academic, or government organization, or a combination. exists on premises.
* **Characteristics**: High scalability, reliability, and cost-effectiveness.
* **Examples**: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).

**2. Private Cloud**

* **Description**: provisioned for exclusive use by a single organization comprising multiple consumers, may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
* **Characteristics**: Offers greater control and data security.
* **Examples**: VMware and OpenStack.

**3. Hybrid Cloud**

* **Description**: Hybrid clouds combine public and private clouds, bound together by technology that allows data and applications to be shared between them. provides greater flexibility.
* **Characteristics**: Businesses can keep critical applications and sensitive data in a private while computational resources of a public cloud for non-critical applications. better scalability.
* **Examples**: web-based email.

**4. Community Cloud**

* **Description**: provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns. may be owned, managed, operated by one or more of the organizations in the community, a third party, or a combination, may exist on or off premises.
* **Characteristics**: providing a middle ground between the public and private cloud.
* **Examples**: Government agencies or a consortium of financial services firms

**Multi-cloud**

* **Definition**: refers to the use of cloud services from more than one cloud provider. This can include any combination of public, private, or community cloud resources across different service providers. To avoid reliance on a single cloud provider and to benefit from the unique services or better pricing that different providers may offer. (same type, diff vendors)

**Challenges**:

**1. Complexity in Management 2. Security and Compliance**

**1. Cost Efficiency: Reduced Capital Expense and operation expense** eliminates the need for significant capital investment in hardware and software. Pay-as-you-go.

**2. Scalability 3. ease of access 4. reduce business risks and maintenance costs**

**5. energy saving:** Greenhouse gas emissions decreased

**Elasticity**: Degree to which a system can adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand to avoid Over- Provisioning or Under Provisioning.

**IBM Bluemix (now IBM Cloud)**: Offers a wide range of compute Bluemix is designed to support a variety of computing needs from simple apps to complex, scalable cloud applications. Strong focus on AI and machine learning with Watson services. Also notable for its blockchain and IoT services.

**Google App Engine**: Provides a fully managed platform that abstracts the underlying infrastructure. Developers can focus on code while Google handles deployment. Excels in data analytics and machine learning with seamless integration with Google's BigQuery, ML Engine, and other AI services.

**Microsoft Azure:** Offers a broad set of compute services. Known for its extensive support for hybrid cloud configurations, IoT, and AI services through Azure Cognitive Services.

**service Level Agreement (SLA):** A contract between the user and the Provider. Service SLA contains terms and conditions to ensure the rights of the users, providers. Its role is identifying the user’s needs and creating a relationship between the user and the service provider.

**Cloud Bursting:** It is the practice of automatically and temporarily adding computing resources from a public cloud to a private cloud to handle spikes in demand.

**CH3**

**Virtualization**: a technology to run multiple same or different isolated OSs on a single physical system by abstracting and partitioning its physical resources into multiple virtual machines with different workloads to improve IT throughput and cost.

**Dual System**: A computer system in which two operating systems are installed on the same hard drive, allowing either operating system to be loaded and given control.

**Emulation System**: A system that pretends to be another system.

**Virtualization System**: A system that pretends to be two or more of the same system.

**Virtual Machine Manager (VMM) or hypervisor** creates and runs virtual machines by providing interface that is identical to the host.

**Guest**: process provided with virtual copy of the Host

**Hypervisor**: A software that allows multiple OSs (guests) to share a single hardware host.

**control** host processor and resources, **allocating** what is needed to each operating system, and **making sure** that the guest operating systems (called Virtual Machines (VMs)) cannot disrupt each other.

**Vertical Virtualization:** containerization technologyvirtualizes the operating system kernel and application programs.

**Horizontal Virtualization:** full virtualization virtualizes entire operating system and hardware resources.

**Distributed systems operate best with coarse grain granularity" justify بقي**

Coarse graine: larger units of work which take longer time to complete individually but required less communication overhead machines in OS , Communication between machines in distributed system can be slow so DS work with coarse grin.

**Benefits of Virtualization:**

• Consolidation: Operate different OS's and applications on one single server

• Sharing of resources

• Isolation: Virtual machines are isolated from each other as if they are physically separated

• Encapsulation: Virtual machines encapsulate a complete computing environment

• Hardware Independence: Virtual machines run independently of underlying hardware

• Portability: Virtual machines can be migrated between different hosts.

**OS Level virtualization**: creates VM that allows single machine to act as many.

**Full Virtualization:** share resources of a single hardware across multiple environments

**Para Virtualization:** host (OS) provide abstraction layer for running virtual guest OS.

**Small Scale Consolidation**: Operate different OS's and applications on one single server.

**Production Consolidation**: A company can achieve greater efficiency and increase profitability by selling all or part of its manufacturing operations. - higher profitability for the company.

**Virtualization Over Cloud Computing Benefits**:

o Reduce capital expenses (CAP-EX)

o Reduce maintenance and operation expenses (OP-EX) through server consolidation,

o Reduce physical space needed in data centers.

o Resource Management, Migration, Maintainability, High availability, and Fault tolerance

**Advantages of Virtualization Over Cloud Computing**

• Zero downtime maintenance

• Freedom from vendor-imposed upgrade cycles

• Instant provisioning

• Pooling hardware resource

• Virtual hardware supports legacy operating systems efficiently

• Dynamic resource sharing

• Security and fault isolation

• Business continuity, backups, and automated restoration

**Virtualization & Green ICT:** reduce energy use and CO2, reduce space (better for environment)

**Transparent**: invisibility of the processes, complexities, or the actual inner workings of a system to user.

**Available:** degree to which a system, service is operational and accessible when required by user.

**Tightly Coupled Systems**: components are highly dependent on each other. Changes to one component often require changes to others, and components communicate or interact with one another directly.

**Loosely Coupled Systems**: designed such that individual components have little or no knowledge of the definitions of other separate components. Components interact through interfaces and messages.

**Resource Scheduling:** involves allocating available resources to tasks or processes over time, ensuring that resources are utilized effectively to meet specific objectives.

**Resource Provisioning:** refers to the process of making resources available for use, often involving the setup and allocation of the necessary infrastructure, software, and services required.

**Virtual Machines (VMs)**: an emulation of a computer system that provides the functionality of a physical computer. It runs an entire operating system (OS) stack.

**Containers**: package an application and all its dependencies into a single executable unit. Unlike VMs, containers share the host system's kernel.

**Data locality**: the strategy of managing and accessing data based on its location within a computing system to improve performance and efficiency.

|  |  |
| --- | --- |
| **Full Virtualization** | **Para Virtualization** |
| all software (including all OS's)  capable of executing on the raw (bare) hardware | These hypervisors run on a conventional operating system (Host OS) just as other computer programs do. |
| It directly sitting on top of the bare hardware device | Para hypervisors abstract Guest OSs from the Host OS. |
| Hypervisors Enable to run multi unmodified guest OS | Improves performance & Lower overhead. |
| Guest OS is not aware that it is being virtualized. | A Guest OS runs as a process on the Host OS. |

|  |  |
| --- | --- |
| **Server without virtualization** | **Server with virtualization** |
| Single OS can run at time | Can run multiple OSs |
| Software & hardware tightly coupled | Hardware independence of OS & applications |
| Running multiple apps on same machine creates conflict | Save electricity, initial cost to buy server |
| Hardware changes require manual effort |  |

|  |  |
| --- | --- |
| Centralized Systems | Decentralized Systems |
| System shared by users all the time | Multiple autonomous components |
| All resources accessible | Some resources may not be accessible |
| Software runs in a single process | Components shared by users |
| Single physical location, point of control, point of failure | Multiple physical locations, point of control, point of failure |
| Example: Airplane booked, Banks | No Global time & No Shared memory Software can run in concurrent processes on different processors |
|  | Example: Gird, Cloud |

A table with text on it

Description automatically generatedA screenshot of a cloud

Description automatically generated

**CH4**

**Migration**: the process of moving a virtual machine from: One host server to another, Storage location to another, One data center to another.

Why? Load balancing — Maintenance —Recovery from host failure.

**Hotspot**: any resource exceeds a threshold (or SLA violation) for a sustain period.

**Hotspot Detection (Sandpiper):** A Black-box Monitoring CPU /processes Network, Memory (swap) Gray-box Monitoring Gather OS level statistics and application logs.

**Memory Migration**: Pre-Copy » (Warm-Up Stop-and-Copy), Post-Copy.

**File System Migration**: copy only changed local files to the destination using VM Manager’s API.

**Network Migration:** If both source and destination are on same LAN switch an Address Resolution Protocol (ARP) reply from the migrating host is provided, if both source and destination are on a switched network The migrating OS can keep its original Ethernet MAC address, relying on the network switch to detect its move to a new port.

**Downtime:** Time during which the VM on the source host is suspended (not available).

**Migration Time**: Minimize total end-to-end migration time, Predictability of migration time.

**Guest Penalty**: Minimize performance loss, minimize downtime (maximize availability).

**Atomicity**: Avoid dependence on multiple volumes (for replication fault domains)

**Ideally,** we want migrations to: Complete successfully, Limit the impact of migration on Guest and the local network.

**Kernel-Based Virtual Machine (KVM)** : virtualization module in the Linux kernel that allows the kernel to function as a hypervisor.

**Advantages of shared storage:**

Complete mobility of VMs with /live migration

Can scale the compute nodes and the storage nodes independently.

Simpler compute nodes: little or no local storage required.

Central point of volume management, Central point of backup / Data Recovery.

**Disadvantages of shared storage:**

Storage becomes single point of failure.

Network becomes single point of failure.

Network bandwidth can be a bottleneck.

Network /latency can impact performance, Network security.

Risk of accidentally starting two VMs using the same disk image!

**Burst migration Benefits**: Load balancing, Disaster recovery, Hardware maintenance, Fault takeover.

**Cold (Regular) migration** (offline migration): VM is completely power off before migration to remote end. The virtual machines are not required to be in shared storage.

**Warm (Suspended/Paused) Migration**: Transfer VM from one physical server to another without shutting down it. State of VM saved in hard disk or RAM for short time, Suspend VM on Host 1, copy across RAM and CPU registers Continue on Host 2 (some seconds later).

**Hot (Live) (Real Time) Migration**: The movement of a virtual machine from one physical host to another while being powered on. Requires shared memory between VIMs.

**Migration Time**: Total amount of time required to transfer a virtual machine from source to destination node without affecting its availability.

**Measured metrics of live migration performance**

**Preparation** **Time**: Resources are reserved on the destination which performed various operations. **Down Time**: Time during which the VM on the source host is suspended (not available). **Resume Time:** instantiation of VM on the destination with the same state as suspended source. **Pages Transferred**: Total amount of memory "dirty“ pages transferred. **Total Migration Time**: The total time taken in completion of all these phases. **Application Degradation:** When VM migrated from one host to another, the application performance is degraded which is running on that VM.

**Advantages of Live Migration**

Allows High Availability (because of minimum Down Time in It Guest). facilitates proactive maintenance in case of failure. Can be used for load balancing in which job is shared among computers. Migration takes place without any noticeable effect from the end user’s point of view. Energy efficiency: rearrange loads to reduce A/C needs.

**Pre-Copy Memory Transfer**

**Warm-Up Technique:** Transferring memory pages to the destination host over several iterations without stopping the execution of the VM in the source. Then, VM is transferred to the destination, Dirty Pages must send again to the destination host.

**Dirty Pages**: Memory pages that have been modified in the source host since the last page transfer. If the rate of updating of pages is very high, migration time will rise to a very high value.

**Stop and Copy Technique**: Transferring memory pages to the destination host over a number of iterations without stopping the execution of the VM in source, The VM will suspend in source and the remaining dirty pages will be copied to the destination, Then, VM will be resumed in destination.

**Advantage of Pre-Copy**: All updates are available at the destination. can be activated any time.

**Disadvantage of Pre-Copy**: VM that has some set of pages update very frequently, they are poor candidates for pre-copy migration.

**Post-Copy Memory Transfer**

Suspend the migrating of VM at the source side, copy minimal subset of the execution state of the VM (CPU state, registers and, optionally non-pageable memory) state to the destination host Resuming VM migration, Concurrently, the source actively pushes the remaining memory pages of the VM to the target. At the target, if the VM tries to access a page that has not yet been transferred, it generates a page-fault (known as network fault) and trapped at the target and redirected to the source, which responds with the faulted page.

Too many network faults can degrade the performance of applications running inside the VM.

VM migration Fulfill the requirements of on-demand computing resources.

**VM consolidation:** concerns about reducing the number of active Physical Machine (PM) by migrating VMs into lesser number of active Physical Machines (PMs), so that PMs with no VM can be converted into sleep state to reduce energy consumption.

**VM Placement**: the process of selecting the appropriate host for the given VM while considering maximizing resource utilization and QoS of this host.

**VM Migration**: carried out after the initial VM placement in order to reduce the number of running physical machines by migration of few VMs and consolidate them into reduced number of PMs.

**static consolidation**: a new VM is placed to PM for processing and no migration takes place here. **dynamic consolidation**: VMs are migrated from one PM to another whenever a necessity occurs.

**Advantages**

Reduce the amount of hardware, Reduce the data center footprints, indirectly reduce power consumption, Grantee Green ICT, Cost reduction and Reduce staffing needs.

**Consolidation Challenge**

accurately characterizing an app’s resource requirements and resource usage over a period.