3a) Negatives of virtualization (Effects of virtualization on security?)

(i) the negative effect on performance due to additional overhead

(ii) need for ~~po~~ more powerful system to run multiple virtual machines

One of the most important virtues of virtualization is that the complete state of an OS running under a VM is captured by VM. This state can be saved in a file and then the file can be copied and shared. There are several implications regarding this fact:

1.) Ability to support IaaS delivery model when user selects an image matching the local environment used by the application and then uploads and runs the application on the cloud using this image

2) Increased reliability: An OS with all apps running under it can be replicated and switched to standby in case of system failure

3) Improved intrusion prevention and detection. In a virtual environment a clone can look for known patterns in system activity and detect intrusion. The operator can switch to a hot standby when suspicious events are detected.

4) Secure logging and intrusion protection: Intrusion detection can be disabled and logging can be modified by an intruder when implemented at OS level. When these services are implemented at the VMM (hypervisor level, the services can't be disabled (modified. VMM may be able to log only events of interest for a post-attack analysis

5) More efficient and flexible software testing: Virtualization allows the multitude of OS instances to share a small no. of physical systems

6) i) To balance load of a system, an OS and apps running on it can be moved to another server when the load on the current server exceeds a high-water mark.

ii) To reduce power consumption, the Load of lightly loaded servers can be moved to other servers and then these servers can be turned off or set on standby mode.

Undesirable effects of virtualization leads to the diminshed ability of an organization to manage it's systeme and track their status

1) the no. of physical systems in the inventory of an organization is limited by cost, space, energy consumption, and human support. Creating a VM reduces ultimately to copying a file ∴ the explosion in the no. of VM's is a fact of life. The only limitation for the no. of VMs is the amount of storage space available

2) In addition to quantity, there is also qualitative aspect to the explosion in the number of VMs. Traditionally, organizations install and maintain the same version of the system software. In a virtual environment such a homo geneity cant be enforced; thus the no. of differnt OS, their versions, and the patch status of each version will be very diverse, and this heterogeneity will tax the support team.

3) It poses problems in software life cycle. the traditional assumption. The traditional assumption is that the Software life cycle is a straight line, so patch managment.

is based on a monotonic forward progress. However, the virtual execution maps to a tree, structure than that of a line: Indeed at any point in time multiple instances of the VM can be created and then each one of them can be updated, different patches installed, and so on.

This problem has serious implications on security.

→ Some of the infected VMs may be damant at the time when the measures to clean up the systems are taken and then, at a later time, they could wake up and infect other systems. This senario can repeat itself and guarantee that infection will last indefinitly. But in non-virtualized environments: once an infection is detected the systems are quarantined and cleaned up. The systems will behave normally until the next episode of the infection occur.

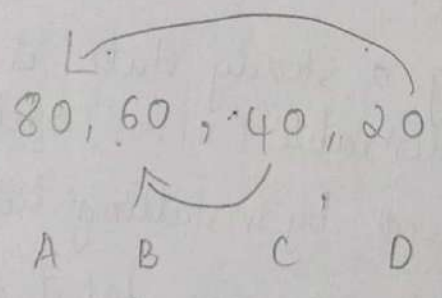
→ In traditional computing a steady state is reached which might be desirable or undesirable (virus infected systems). The desirable state is achieved by installing latest version of the system software and then applying latest patches to all the systems. Due to lack of control, a virtual environment may never reach a steady state and in a non-virtual environment the security can be comprimised when an infected laptop is connected to network protected by firewall. A side-effect of ability to record in a file the complete VM state poses challenge. OTP's are transmitted in clear and no protection. If system runs the S/KEY password system he can replay rolled-back versions and acess past sniffed passwords. Some cryptographic protocols are required

3c) Load balancing and minimising the costs are interrelated. It ensures that the load is equally distributed among servers and also VM's within server. No sever should be over burdend and some servers only doing little work as that would lead to less energy optimization. So load balancing should be done on VM's.

And with this energy optimization no VM is under or overloaded.

Conside 4 MM's are being used The CPU. utilization of 4 of them is 80, 60, 40, 20 respectively.

Now instead of having 4 VM's running that will waste time, space, energy or power consumption.



Two VM's C and D are shutdown and migrated to B and A respectively saving the cost. It ensures that least no. of servers server the users. CMS or central management system takes care of this.

2c) HDFS fault tolerance:

→ One of the main aspects of HDFS is it's fault tolerance characteristic.

→ Since Hadoop is designed to be deployed on low cost hard-ware by default, a hardware failure in this system is considerd to be common rather than exception.

→ Therefore, Hadoop considers the following issues to fulfill reliability requirements of file system:-

(i) Block replication: Input file is split and is replicated to multiple data nodes. If the data in data node is corrupted it informs the namenode, then the next copy of the same input split is fetched and the mapper starts working on the cop.

→ Maximum of 3 copies of each block is kept. 3 copies of the same split.

→ One copy is in one data node in one rackspace. Second copy is in another data node in same as first rack space as it is easy to fetch.

→ Third copy is in another datanode in some other rackspace.

→ A rackspace is huge server with multiple racks or nodes.

→ Namenode has metadata (location of datanode, structure of data node, size of data node).

→ HDFS is configured by users.

The replication factor is set by the user and is three by default.

2). Replica placement: The placement of replicas is another factor to fulfill the desired fault tolerance in HDFS.

→ Storing replicas on different data nodes on different racks across whole cluster provides more reliability.

→ It is sometimes ignored as the cost of communication b/w two nodes in different racks is relatively high in comparision with that of different nodes located in the same rack.

→ for the default one with factor of 3, HDFS stores one replica in the same node the original data is stored; one replica on different node to provide 3 copies of data.

3) Heart beat and BlockReport messages:

→ These are periodic messages sent to the namenode by each data node in the cluster.

→ Reciept of heartbeat implies that the data node is functioning properly, while each block report contains a list of all blocks on a datanode.

→ The namenode recieves such messages because it is the sole decision maker of all replicas in the system.

→ Namenode polls input splits in data node for heart beats whereas namenode is indicated about node corruption in case of block report.

2b) Trust isn't that easy to build. (Trust)

→ Trust in the part of the cloud computing is intimately related to the general problem of trust in online activities.

→ There are two conditions which must exist for a trust to develop and we as CSP should abide by it.

i) Risk: The perceived probability of loss, indeed trust would not be necessary if there were no risk involved, if there is a certainity that an action can succeed.

ii) Interdependence: The idea that interest of one entity cannot be achieved without reliances on other entities.

It has 3 phases:

1) Building phase, when trust is formed,

→ During SLA is signed both CSP and customer agrees.

2) Stability phase, when trust exits.

→ After the customers has used CSP and now trust is stable.

3) Dissolution phase, when trust declines.

→ when customers end his relationship with entity or CSP.

And there are different reasons for and forms of trust, Utilization reasons could be based on the belief that costly penalties for breach of trust exceed any potential benefits from opportunistic behaviour and it is the essence of deterence -based trust.

→ Involving the other party is in the self-interest of the party and it a calculus based trust.

→ Policies and reputation are two ways of determing trust. Policies reveal the condition to obtain trust and action to be taken.

→ Reputation is quality attributed to an entity based on relatively long history of interactions with or possibly observation of entity

→ An entity must work very hard to build trust, may loose the trust very easily.

→ Internet trust:

i) Obscures or lacks entirely the dimension of character and personality, nature of relationship and institutional character of traditional trust.

→ offers individual the ability to cancel their identity.

→ Indentity is critical for developing trust relations, it allows us to base our trust on past history and interactions with entity.

→ Anonymity causes mistrust because identity is associated with accountability and in absence of identity, accountability cannot be enforced.

→ opacity, extends identity to personal characteristics. There are no guarantees that entities we transact with fully understood the thejre assumed. The anonymity reduces the uses normally used in judgements. of trust

Recommendations are based on trust decisions made by others and filtered through perspective of entity assessing the trust.

In A computer science context: Trust of party A to party B for a service X is the measurable belief of A in that B behaves dependably for specified period within a specified context.

Why distributed on Big Data Analytics?

3b) The simultaneous growth in the availability of big data and in the no. of simultaneous uses on the Internet places particular pressure on the need to carry out tasks " in parallel " or simultaneously. Parallel and distributed computing occurs across many different topic areas in computer science, including algorithms, OS, COA and software engineering.

Reasons:-

i) Distributed systems allows breaking huge problems/ Big Data to smaller pieces and have multiple computers work on them in parallel, which can help cut down on the time needed to solve / compute those problem. It takes divide and compute approach.

ii) Networks such as Internet now help in many computers communicate with each other and takes this advantage of these networked computers by arranging them to work together on a problem, thereby reducing the time needed to obtain the solution.

Ex: Map Reduce Architecture.

→ In parallel computing multiple processors perfoms multiple tasks assignmed to them simultaneously. providing concurrency and save times and mony

→ Different aspects of the distributed paradigm solve. different challenges in Big Data Analytics.

→ Distributed computing allows it to be fault tolerant.

→ This power in fast decision making in crucial times.

Types/Levels of scheduling algorithms and objectives? Pg No - 06

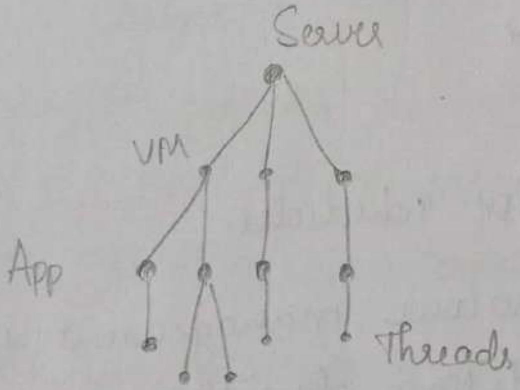
. Scheduling algorithms should be able todecide for whom to assign the task, how much and how to meet deadlines.

Resources can be shared at differnt level.

i) Server level.-> It should be shared among VM's

ii) VM level -> It should be shared among applications

iii) Application.-> It should be shared among threads



A scheduling algorithm should be fair, efficient, starvation free. - no process should wait indefinitly.

-> Objectives for a batch system:

i) maximize throughput

ii) minimize turnaround time.

iii) meet the deadlines.

iv) be predictable/solvable/tractable and stable

-> Objectives for real-time system.

i) to meet deadlines and to be predictable

Schedulers for systems supporting mix of tasks are subjected contradictory requirements. Types Of Requirements?

(i) Hard real time requirement

eg: Satellite launching and response should not exceed more than 2ms.

(ii) Soft-real time constrained.

eg: Developer needs to update something should be done within 3hrs.

(iii). No-time constraints.

eg: Backup and Archive

Pre-emptive and non-premptive scheduler.

* Two ditinct dimensions of resource management that should be addressed by scheduling algorithms policy:

i) The amount or quantity of resources allocated.

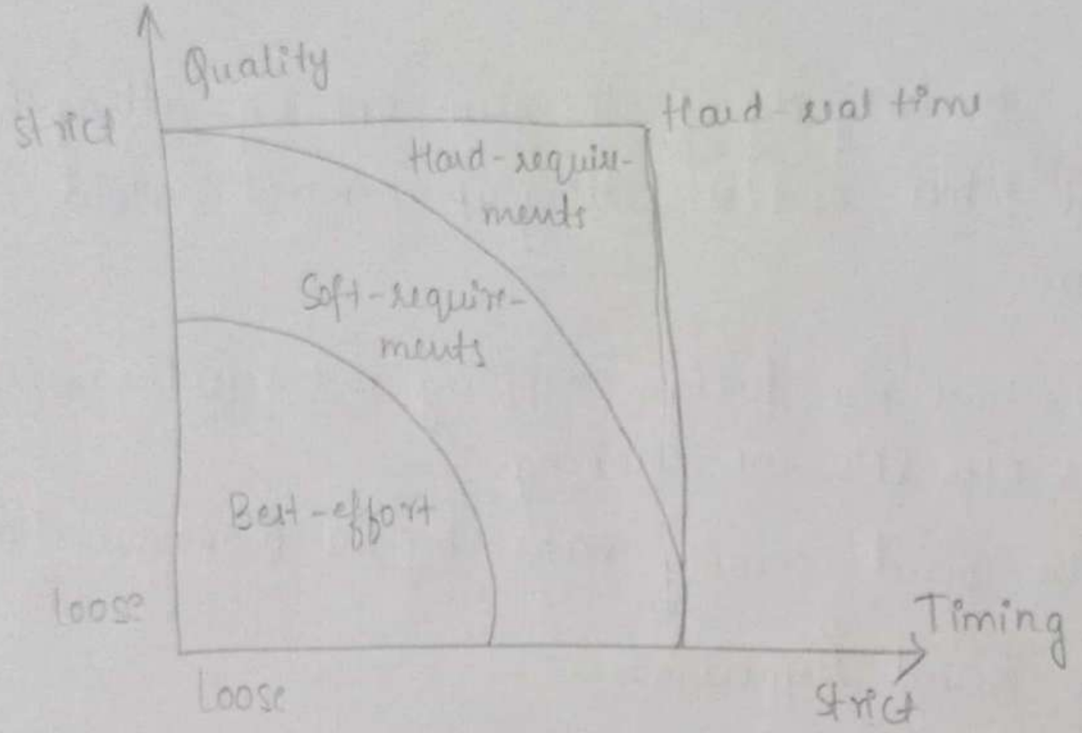
ii) The timing when access to resources are granted.

Classes of resource allocation requirements in the space defined by two dimensions

i) Best-effort

ii) Soft-requirements.

iii) Hard-requirements



classes of resource allocation requirements.

i) Best effort policies do not impose requirements regarding either the amount of resources allocated to an application or the timing when application is scheduled.

ii) Soft-requirements allocation policies require statiscally guaranted amounts and timing constrainte.

iii) Hard-real time systems are the most challenging because they require strict timing and precise amounts of resources

The QoS requirements differ for differnt classes of cloud apps and demand scheduling policies. Best-effort apps such as batch apps and analytics do not require QoS guarantees. Multimedia apps such as audio & video streaming have soft-real time constraints and require statistically guaranteed max delay & throughput. Apps with hard real-time constraint dont use a public cloud.