HW-5

1) Rack-mounted servers:

Rack Servers are rack-mounted servers that could specifically be fitted in a server rack. Rack Servers are servers that are made with efficient configurations to support a wide range of requirements. They are also known for their shape and structure as unlike traditional servers shaped in a PC structure, it is much wider and flatter which could be fit into any server rack. Rack servers are considered beneficial if there’s a requirement of a small number of servers as they pose low upfront costs being highly economical.

Blade servers:

A blade server is a modular server that allows multiple servers to be housed in a smaller area. These servers are physically thin and typically only have CPUs, memory, integrated network controllers, and sometimes storage drives built in. Any video cards or other components that are needed will be facilitated by the server chassis. Which is where the blades slide into. Blade servers are often seen in large data centers. Due to their ability to fit so many servers into one single rack and their ability to provide a high processing power.

The motivation behind the two forms of server placement is to reduce the complexity in connection, space occupancy, and to provide flexibility.

* Flexibility: A blade server provides a switch through which the servers within connect to the external network. Worth noting here is that the chassis also fits into a rack much like a rack-mounted server.
* Space: A blade server is even more compact than a rack-mounted server. They are optimized to reduce their physical foot print and interconnection complexity. Such optimization is necessary in the face of an ever-increasing number of servers that need to be put in the constrained space of a data center.
* Computing Power: The smaller form factor is achieved by eliminating pieces that are not specific to computing – such as cooling. As a result, a blade may amount to nothing more than a computer circuit board that has a processor, memory, I/O, and an auxiliary interface. Such a blade certainly cannot function on its own. It is operational only when inserted into a chassis that incorporates the missing modules. The chassis accommodates multiple blades.

*Sources: Rack server vs blade server(* [*https://www.racksolutions.com/news/data-center-optimization/blade-server-vs-rack-server/*](https://www.racksolutions.com/news/data-center-optimization/blade-server-vs-rack-server/) *), Wiki (*[*https://en.wikipedia.org/wiki/Blade\_server*](https://en.wikipedia.org/wiki/Blade_server)*)*

2) Servers in a data center need to be interconnected with each other as well as with external world. Top-of-rack(TOR) and end-of-row(EOR) are two different types of connectivity leading to 2 different cabling options. We use ethernet technologies to implement both these options. Ethernet is particularly important for data centres because of its potential to eliminate employing separate transport mechanisms (e.g. FC) for storage and inter processor traffic.

*Sources: Wiki (* [*https://en.wikipedia.org/wiki/Data\_center\_bridging*](https://en.wikipedia.org/wiki/Data_center_bridging) *)*

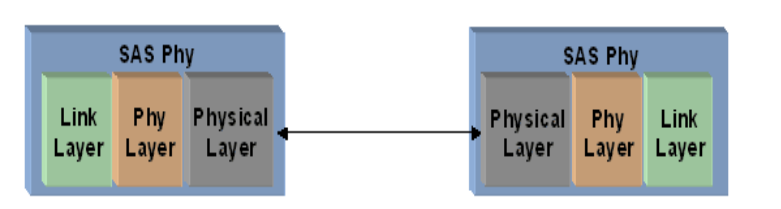
3) While NAS (Network Attached Storage) and SAN (Storage Area Network) both reside across a network while DAS (Direct-Attached Storage), directly attaches to the processor through a point to point link. DAS is built for and dedicated to storage traffic in the case of SAN. The main difference between NAS and SAN is that one is files and objects, and other is disk objects, respectively. The other difference is that SAN relies on special transport, FC which is optimized for storage traffic while NAS does not require anything special apart from IP network. Since DAS is not affected to network delay it can keep local data like boot and swap images. Depending on the location of storage device in view of the host, DAS can be both internal and external.

*Sources: DAS vs. NAS vs. SAN: Which is best for virtual storage? (* [*https://www.computerweekly.com/tip/DAS-vs-NAS-vs-SAN-Which-is-best-for-virtual-storage*](https://www.computerweekly.com/tip/DAS-vs-NAS-vs-SAN-Which-is-best-for-virtual-storage) *)*

4) A phy layer is concerned with the line coding out of band signals as well as other preparations needed for serial transmissions. It has an 8-bit identifier that is unique for each device. A management function assigns identifiers.

A phy, as defined in SAS, is a combination of the physical layer, phy layer and link layer functions. A minimum of two phys (one at the initiator, the other at the target) is required to complete a SAS physical connection pathway, as shown in Figure.

By contrast, a physical layer deals with the physical and electrical characteristics of cables, connectors, and transceivers.



*Sources: SAS overview(* [*https://www.snia.org/sites/default/education/tutorials/2007/spring/networking/SAS-Overview.pdf*](https://www.snia.org/sites/default/education/tutorials/2007/spring/networking/SAS-Overview.pdf) *)*

5) Generic file related systems calls are for close, open, read and write. In the NFS there is no RPC invocation for the close <file> system call because of following reasons:

i) In order to facilitate crash detection, HFS’s protocols have a stateless design of servers that does not keep track of previous requests, resulting in it lacking close routine.

ii) There is no modification of files.

A remote file operation even with an RPC component may not lead to an RPC invocation. Invocations of theta kind are not needed when data is stored in the client cache, leading to lessening the number of remote procedure calls and increasing efficiency. However, caching makes it harder to maintain file consistency.

*Sources: File system call (* [*http://www2.cs.uregina.ca/~hamilton/courses/330/notes/unix/filesyscalls.html*](http://www2.cs.uregina.ca/~hamilton/courses/330/notes/unix/filesyscalls.html) *), Remote procedure call(* [*https://www.techtarget.com/searchapparchitecture/definition/Remote-Procedure-Call-RPC*](https://www.techtarget.com/searchapparchitecture/definition/Remote-Procedure-Call-RPC) *)*

6) The following connection topologies are supported in FC-2M:

a) Point-to-point

b) Fabric

c) Arbitrated loop

The fabric topology is the most flexible topological connection. It involves a set of ports attached to a network of interconnecting FC switches through separate physical links. The switching network (or fabric) has a 24-bit address space with hierarchical structure, as per domains and areas. An attached port is assigned a unique address during the fabric login procedure. The exact address typically depends on the physical port of attachment on the fabric (or switch, to be precise). The fabric routes frames individually based on the destination port address in each frame header

*Sources: Textbook*

7) ENode chooses compatible FCF, based on advertisements and a discovery solicitation is sent by it at which a capability negotiation commences. At the receipt of the solicitation, the FCF responds to ENode with solicitated discovery advertisement to confirm the negotiated capabilities. Once it receives the solicited discovery advertisement, the ENode sets up a virtual link to the FCF. This is the same procedure as the FC fabric login method. A successful login completion of the procedure leads to creation of virtual port on ENode, a virtual port on the FCF and a virtual link between them.

*Sources: Textbook*

8)

i) The feature of TCP that are leverage in iSCSI include the multiple iSCSI nodes that maybe reachable at the same address and the same iSCSI node can be reached at multiple address. Hence, we can use multiple TCP connections for a communication session between a pair of iSCSI nodes to obtain greater throughput.

ii) These features are required because of reliable in-order delivery, automatic re-transmission of unacknowledged packets, and congestion control.

iii) The Stream Control Transmission Protocol(SCTP) is similar to the TCP as it supports the features essential to the SCSI operations similar to the TCP. However, at the time of standardization of iSCSI, the SCTP was considered too new to be depended upon.

iv) iSCSI by itself doesn’t provide any mechanisms to protect a connection or a session. All native iSCSI communication is clear and subject to eavesdropping and active attacks. In an untrusted environment, iSCSI should be used along with IPsec

*Sources: Textbook, Why does iSCSI use TCP instead of IP ? (* [*https://etherealmind.com/why-does-iscsi-use-tcp/*](https://etherealmind.com/why-does-iscsi-use-tcp/) *)*

9) iSCSI makes use of a scheme knows as connection allegiance in order to reduce complexity. In this scheme the initiator of the connection can use any connection to issue commands but must then stick to that same connection for all subsequent communications. The iSCSI sessions need management this is carried out by the login procedure. Successful completion of the login procedure leads either to a new connection or adds a new user to an already existing session.

*Sources: Textbook*

10) A credential is basically in essence an cryptographically protected tamper proof capability that involves the keyed-Hash Message Authentication Code (HMAC) of a capability with a shared key. Its use in ubiquitous today yet its not used as proof for access control. It can be represented as:

< Capability, Object Storage Identifier, Capability Key>

Where,

Capability key == HMAC (Secret Key, Capability || Object Storage Identifier)

*Example*: The standardized scheme derives a proof based on the capability key. This proof is a quantity calculated with capability key over selective request components according to the negotiated security method.

*Sources: Textbook, ANSI INCITS 458-2011(* [*http://webstore.ansi.org/standards/incits/incits4582011*](http://webstore.ansi.org/standards/incits/incits4582011) *)*

11) The three approaches to block level virtualization are as follows:-

i) Host-based: here a volume manager that is part of the operating system handles virtualization. The volume manager is responsible for mapping the native blocks to the logical blocks and also keep track of overall storage utilization. The drawback we face in this approach per-host control is unfavourable to optimal storage utilization in any multi-host environment.

ii) Storage Device based: here a controller of storage system handles virtualization. Since this is located close to the storage, performance is good.

iii) Network-based: here a special function of the storage network, which maybe a component of switch handles virtualization. This approach is transparent to hosts and storage systems as long as they support the appropriate storage network protocols. It can be classified into in-band (symmetric) or out-of-band (asymmetric) on the basis of how control traffic and application traffic are handled.

**In-band approach**, where the virtualization function for mapping and I/O redirection is always in the path of both the control and application traffic.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| I. On the positive side, the central point of control afforded by the in-band approach simplifies administration and support for advanced storage features such as snapshots, replication and migration. | I. The virtualization function could become a bottleneck with a single point of failure. |
| II. The snapshot feature is of particular relevance to Cloud Computing. It can be applied to capture the state of a virtual machine at a certain point in time, reflecting the run-time conditions of its components (e.g., memory, disks, and network interface cards). | II. There is a trade-off as in this case the performance of other virtual machines on the same host may suffer when the snapshot of a virtual machine is being taken |

**Out-of-band approach,** where the virtualization function is in the path of the control traffic but not the application traffic. The virtualization function directs the application traffic.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| I. In comparison with the in-band approach, the approach results in better performance since the application traffic can go straight to the destination without incurring any processing delay in the virtualization function. | I. This approach does not lend itself to supporting advanced storage features. More important, it imposes an additional requirement on the host to distinguish the control and application traffic and route the traffic appropriately. As a result, the host needs to add a virtualization adaptor, which, incidentally, may also support caching of both metadata and application data to improve performance.. |
|  | II. Per-host caching, however, faces the challenging problem of keeping the distributed  cache consistent |

We can conclude **Network based approach is the most suited for cloud computing,** given its relative transparency and flexibility in storage pooling. With this approach storage can be assigned to VM hosts which in turn can allocate the assigned virtual storage to VMs through their own virtualization facilities.

12)

|  |  |
| --- | --- |
| NOR flash | NAND flash |
| I. The basic construct has similar properties to those of a NOR gate. | I. The basic construct has properties similar to those of a NAND gate. |
| II. It is fast and it can be randomly addressed to a given byte. | II. It allows random access only for units that are bigger than a byte. |
| III. It has limited storage density | III. It has big influence on consumer electronics |
|  | IV. It is more widely used than NOR flash – in digital cameras, smart phones, etc. |

13) To deploy NAND flash solid state drives in the cloud we need to overcome the following obstacles:

i) A write operation over the existing content requires that this content be erased first. (This makes Write operations much slower than Read operations).

ii) Memory cells erase out after a limited number or write-erase cycles.

iii) Erase operations are done on a block basis, while write operations on a page basis.

*Sources: Textbook*

14)

Depending on the size of DRAM available on a server, caching the workload data may need more than one server. In this case, the hash table is distributed across multiple servers, which form a cluster with aggregated DRAM. Memcached servers, by design, are neither aware of one another nor coordinated centrally. It is the job of a client to select what server to use, and the client (armed with the knowledge of the servers in use) does so based on the key of the data item to be cached.

How should the hash table be distributed so that the same server is selected for the same key? A naive scheme is described as follows:

s = H(k) mod n

where H(k) is a hashing function, k the key, n is the number of server, and s is the server label, which is assigned the remainder of the division of H(k) over n. The scheme works as long as n is constant, but it will most likely yield a different server when the number of servers grows or shrinks dynamically – as is typically the case in Cloud Computing. As a result, cache misses abound, application performance declines significantly, and all servers in the latest cluster have to be updated. Obviously, this is clearly undesirable, and so another scheme is in order. To this end, memory cached implementations usually employ variants of consistent hashing to reduce the updates required as the server pool changes and maximize the chance of having the same server for a given key. The basic algorithm of consistent hashing can be elaborated as follows:

i) Map the range of a hash function to a circle, with the largest value wrapping around to the smallest value in a clockwise fashion;

ii) Assign a value (i.e., a point on the circle) to each server in the pool as its identifier, and

iii) To cache a data item of key k, select the server whose identifier is equal to or larger than H(k).

The server selected for key k is called k's successor, which is responsible for the arc between k and the identifier of the previous server. As an example, below figure shows a circle of three servers, where server 1 is responsible for caching the associated data items for keys hashed to 6, 7, 0, and 1; server 3 for keys hashed to 2 and 3; and server 5 for keys hashed to 4 and 5.



An immediate result of consistent hashing is that a departure or an arrival of a server only affects its immediate neighbors. In other words, when a new server p joins the pool, certain keys that were previously assigned to the original p's successor will now be re-assigned to server p, while other servers are not affected. Similarly, when an old server p leaves the pool, the keys previously assigned to it will now be reassigned to p's successor while other servers are not affected. Adding a new server 7 would result in reassigning keys 6 and 7 to the new server; removing server 3 would result in reassigning keys 2 and 3 to server 5.

*Sources: Textbook*