Faculty of Engineering & Applied Science

# SOFE4790U – Distributed Systems

## Homework: Networking and Virtualization

## Due Date: 10/02/2022

| First Name | Last Name | Student ID |
|---|---|---|
| Abdul | Bhutta | 100785884 |

**Part 1: Networking:**
Review your course on computer networks. Can you try to summarize what you learned in that course? Focus on the 7 layers of the OSI interconnectivity model? What different detailed functionality does each layer provide? how does this help two applications to communicate?

The OSI layer helps two different applications communicate through the 7 layers. Each layer contains a protocol data unit used to communicate among two different nodes at each layer. When computer A is trying to communicate with computer B, the data or message being sent is managed by the first three-layer (Application, Presentation, Session) and then segmented at the transport layer. The network layer will produce a packet which will then proceed to the data link layer, which creates a frame for the packets that were received. The physical layer will send the information from computer A to computer B by converting it to bits and sending it over the medium. At computer B, it will unframe the packets at each layer following the same protocol used by computer A until the message is received. The seven OSI layers are shown below, with examples of devices being used at certain levels and protocols.

| OSI Layer Level | Description | Example |
|---|---|---|
| **Physical Layer (Layer 1)** | Transmits bits over a medium and determines throughput, latency, error rate of a network link. | Ethernet Cables, Coax Cable, fiber cable, HUB |
| **Datalink Layer (Layer 2)** | It provides communication between two nodes while sending **frames** over a network link; within the frame is the data that is transmitted from one end to another end (Wire) which is encapsulated. The data link layer is divided into two subsections: MAC (Media access control) and LLC (Logical Link Layer). MAC allows for control of the frame and provides any detection of a collision of the frames while the LLC provides error checking such as checksum, parity bit, and CRC (cyclic redundancy codes). | MAC Address, Switch |

| | | |
|---|---|---|
| **Network Layer (Layer 3)** | It provides the ability to move **packets** between the networks using routing protocols. The network layer allows other devices or networks to communicate with each other over multiple links (end-to-end) through packets and allows for network addressing (IP Address) | IP, Routing Protocols |
| **Transport Layer (Layer 4)** | The transport layer (4th layer) provides with reliable data transfer while segmenting them into packets. It provides error checking and flow control in an **end-to-end** network whereas in the link layer it provides error checking at the physical layer. The main protocol being used at the transportation layer is TCP/UDP. TCP is a connection-oriented protocol and reliable which requires an acknowledgement packet (HTTP) while UDP is a connection-less oriented protocol and unreliable which does not require any form of acknowledgement packet (VOIP). | TCP/UDP |
| **Session Layer (Layer 5)** | The session layer maintains the ongoing session (**ports**) while keeping it active and provides synchronization which allows the user to reconnect to the same checkpoint in case of a failure. | OS, Remote Procedure call (RPC), Gateway |
| **Presentation Layer (Layer 6)** | The encryption and data compression are done at the presentation layer such as ASCII coding. | ASCII, signed integer, unsigned integer |
| **Application Layer (Layer 7)** | Provides the user access to network resources/applications and allows protocols to communicate with each other. | HTTP, FTP. SMTP, DNS |

## Part 2: Virtualization

## SnowFlock

Cloud computing allows an end user to rent servers in the cloud and does not need a physical device such as a server, leading to lower costs. The main component of a cloud is providing a virtual machine (VM), which imitates a physical machine. Still, the end users are not observant of the fact that the devices are being shared by many other users or the location of the resources. The cloud provides user access and location transparency.

Cloud computing aims to help users set up their servers within seconds compared to purchasing, installing, and configuring the servers, which may take weeks or months. However, the cloud systems are not stand-alone and require a pool of resources to be added. To configure the servers, a user still needs to manage the cluster and add new servers. Snowflock solution to this issue is virtual machine cloning which uses API calls. It can solve the problems with a single logical operation with cloning, resource allocation, and cluster management. The VM makes copies of the cloud servers, which store the parent's operating system and application cache while adding them to a private cluster. These resources can be used at any time, depending on the number of resources required from the user. Snowflock has allowed the creation of VM in under 5 seconds and can process the load across many physical hosts. It has also integrated into the cloud-based-elastic servers, allowing the new worker node to come online and be loaded up 20 times faster. SnowFlock implements a similar process used in UNIX called a fork, which creates a new process of the parent (duplicate) within the system. In Snowflock, it copies the memory, processes, filesystem, and virtual devices, which can be used to create multiple copies and executed in a parallel sequence.

The way SnowFlock makes its approach is by using the following concepts: Virtualization (which allows for the cloud and machine cloning); Lazy Propagation (VM doesn’t clone unless necessary); Multicast (Clones will all have similar attributes); Page Faults (Clone execution with missing memory won’t run until page arrives); and Copy on Write (when memory is copied before being overwritten). With all this, it makes it possible to alleviate stress on the cloud servers and help with ensuring that they do not need to constantly boot new instances from the same template and aids with parallel computing, data mining, and serving web pages. Cloned virtual machines are missing most of their memory state by the time they are created by the descriptor:

Memserver process stops the VM so that its memory can settle. Once copied, clones can reconnect to the external world in their new enclosed VM. The memserver provides the bits of information the clone needs from its parent. Copy on write is used to circumvent the issue of corrupted memory, where the writing access from pages in memory is removed. A page fault will occur if a write is tried, and that notifies the hypervisor to copy the page. Importantly the parent VM can write to that page to keep running, but the memserver needs to use the copied read-only page.

Mcdist is used to distribute a packet to various receivers. It does this through IP multicasting which allows for network hardware parallelism, which in turn lowers the memory server load. Mcdist does not need to be a reliable method as the packets do not need to be received in a specific order, where the server multicasts for responses, and the clients timeout if/when they do not get a reply for a request they made and have to try again. There are 3 optimizations of mcdist:

- **Lockstep:** when temporal locality occurs, various clones will request the same page, and it will be done almost at the same time, and mcdist will ignore all but requests but the first one that reaches.
- **Flow control:** The server will limit its sending rate to be a weighted average of the client's receive rate. If this is not done, then the receiver will be overwhelmed with too many pages sent from the server.
- **End game:** At the time the server has transmitted most of its pages, the server will go back to unicast responses. Usually, anything at this point is requested retries, and having retries sent to every clone is a waste.

## Compute Resource Consolidation Pattern

Grouping multiple tasks and operations into one unit to maximize resource utilization will allow the user to cut the cost and expenses of their business. Running various tasks on a different system with limited usage and charges per resource is not optimal. Grouping similar jobs with a similar scalability profile and processing requirement will help them scale together as one unit. The idlest situation is running systems that are not cost-effective and are frequently idle. When combining multiple tasks, tasks that require scalability or the computational level that fluctuates depending on the traffic should not be grouped with functions that do not require as many resources, which may lead to more tasks being done. When implementing the compute resource consolidation pattern, it is recommended to think about the topics below before implementing the design,

| Topics | Description |
| --- | --- |
| **Scalability and Elasticity** | Do not group tasks that require the same level of scalability for a resource. |
| **Lifetime** | Design the tasks to have checkpoints and restart the tasks when interrupted or resource is resumed. |
| **Release cadence** | If one code requires to be updated or the system to be restarted, all other tasks will have to stop and be restarted as well! |

| | |
|---|---|
| **Security** | Each task running should have trust between each other without causing an issue or corrupting each other. |
| **Fault Tolerance** | If one system fails, all the tasks within the system are at a halt as well. |
| **Contention** | Do not use tasks that require the same resources such as two tasks using a lot of memory. |
| **Complexity** | Adding multiple tasks within the same system increases the complexity of the code. |
| **Stable logical architecture** | Design the code in a manner where it does not need to change even if the physical environment has changed. |

## Data Partitioning Guidance

In big data and distributed system, it is impossible to keep all the data files on one resource, which is why partitioning is significant. Partitioning allows the data to be stored in separate storage and can have many benefits shown below,

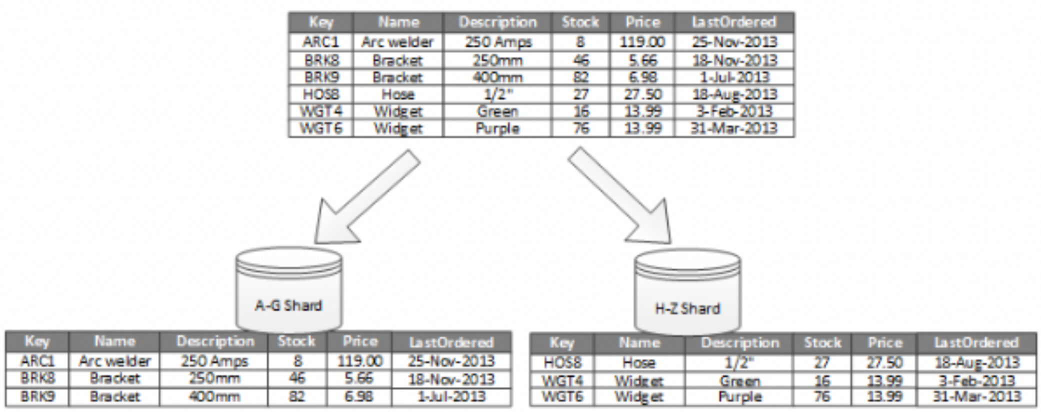
| Improvement | Description |
|---|---|
| **Scalability** | Increasing the data storage on a single server is limited where having partitions of many drives spread throughout on different nodes will achieve almost unlimited data storage. |
| **Performance** | Allows the user to access multiple files or operation in parallel. |
| **Security** | User can partition the data with sensitive material and apply various security protocols. |
| **Provide management** | Allow the user to manage, monitor, failure protection, and other tasks based on the partition and what each partition contains. |

| | |
|---|---|
| **Availability** | Allows the user to have access to files through redundancy or even if one partition fails, they will still have access to other partitions. |
| **Match the data store to the pattern of use** | Partitioning lets each individual partition to have a different type of data store. In other words, files could be stored using a document database, and binary data would be put into blob storage. |

**Designing Partitions**

**1. Horizontal partitioning (Sharding)**

Each partition stores a portion of the data (rows of data), such as records 1-50, and the subsequent partition holds records 51 – 100, where each partition is known as a shard.



| Key | Name | Description | Stock | Price | LastOrdered |
|---|---|---|---|---|---|
| ARC1 | Arc welder | 250 Amps | 8 | 119.00 | 25-Nov-2013 |
| BRK8 | Bracket | 250mm | 46 | 5.66 | 18-Nov-2013 |
| BRK9 | Bracket | 400mm | 82 | 6.98 | 1-Jul-2013 |
| HOS8 | Hose | 1/2" | 27 | 27.50 | 18-Aug-2013 |
| WGT4 | Widget | Green | 16 | 13.99 | 3-Feb-2013 |
| WGT6 | Widget | Purple | 76 | 13.99 | 31-Mar-2013 |

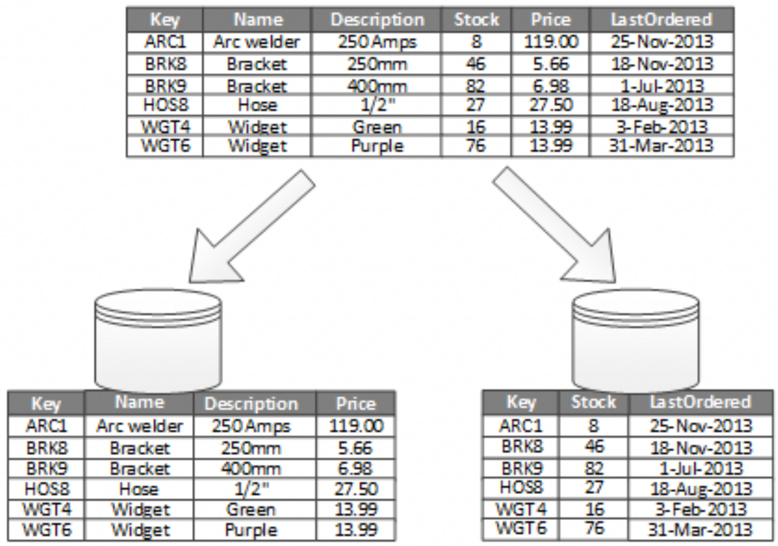


| Key | Name | Description | Stock | Price | LastOrdered |
|---|---|---|---|---|---|
| ARC1 | Arc welder | 250 Amps | 8 | 119.00 | 25-Nov-2013 |
| BRK8 | Bracket | 250mm | 46 | 5.66 | 18-Nov-2013 |
| BRK9 | Bracket | 400mm | 82 | 6.98 | 1-Jul-2013 |

| Key | Name | Description | Stock | Price | LastOrdered |
|---|---|---|---|---|---|
| HOS8 | Hose | 1/2" | 27 | 27.50 | 18-Aug-2013 |
| WGT4 | Widget | Green | 16 | 13.99 | 3-Feb-2013 |
| WGT6 | Widget | Purple | 76 | 13.99 | 31-Mar-2013 |

**2. Vertical partitioning**

Each partition holds a table or a section of the table (columns), such as partition one will hold columns 1 – 10 while partition two will hold columns 11 – 20. Another example is partition one can save the most used columns while partition two will have the lesser used columns in the database.



| Key | Name | Description | Stock | Price | LastOrdered |
|---|---|---|---|---|---|
| ARC1 | Arc welder | 250 Amps | 8 | 119.00 | 25-Nov-2013 |
| BRK8 | Bracket | 250mm | 46 | 5.66 | 18-Nov-2013 |
| BRK9 | Bracket | 400mm | 82 | 6.98 | 1-Jul-2013 |
| HOS8 | Hose | 1/2" | 27 | 27.50 | 18-Aug-2013 |
| WGT4 | Widget | Green | 16 | 13.99 | 3-Feb-2013 |
| WGT6 | Widget | Purple | 76 | 13.99 | 31-Mar-2013 |

| Key | Name | Description | Price |
|---|---|---|---|
| ARC1 | Arc welder | 250 Amps | 119.00 |
| BRK8 | Bracket | 250mm | 5.66 |
| BRK9 | Bracket | 400mm | 6.98 |
| HOS8 | Hose | 1/2" | 27.50 |
| WGT4 | Widget | Green | 13.99 |
| WGT6 | Widget | Purple | 13.99 |

| Key | Stock | LastOrdered |
|---|---|---|
| ARC1 | 8 | 25-Nov-2013 |
| BRK8 | 46 | 18-Nov-2013 |
| BRK9 | 82 | 1-Jul-2013 |
| HOS8 | 27 | 18-Aug-2013 |
| WGT4 | 16 | 3-Feb-2013 |
| WGT6 | 76 | 31-Mar-2013 |

### 3. Functional partitioning

Partition is based on the purpose of the data being used, what category it comes under, or one primary goal. For example, having read-only data in one partition for faster access and another partition for read-write data. Designing the partition is challenging and should focus on the primary goal or main function of the partition. It is recommended that all three design partitions should be integrated into the schema.