DSC-204A: Scalable Data Systems, Winter 2024

Network - 1

Lecturer: Hao Zhang *Scribe: Mithilesh Kinjalkar, Nihal Reddy*

1 Review Questions

1. **Briefly explain 1 pro and 1 con of On-Demand vs Spot instances on AWS.**

   On-Demand

   - Pro: Compute resources are provisioned to users based on their requirements.
   - Con: It is a costly option. Cloud computing vendors designate this tier for their high-priority customers, necessitating users to pay accordingly.

   Spot

   - Pro: It offers significant cost savings, typically priced at around one-third of the On-Demand Cloud Compute rates.
   - Con: With this tier, there is a high probability of resources being reclaimed and jobs preempted to accommodate high-priority, on-demand users.

2. **Briefly explain 2 pros and 2 cons of cloud vs on-premise clusters.**

   Cloud
   Pros:

   - Low cost of maintenance.
   - Flexible - horizontally elastic, allowing hassle-free up/down scaling as per demand.

   Cons:

   - Potential compromise in system security and reduced privacy.
   - Infrastructure resources are owned by the service vendor thus making it difficult to customize things at a low level.

   On-Premise
   Pros:

   - High level of security and data privacy.
   - Complete, lifetime ownership of the infrastructure.

   Cons:

   - High cost of maintenance.
   - Upgrading hardware and software can be challenging, resulting in a rigid implementation that makes scaling up or down difficult.

2 Why do we Need Cloud?

We have reached a juncture where a single computer has hit its physical limitations, unable to deliver the necessary performance due to the increasing number of application users and generated data. Hence, there arises a need for enhanced compute and storage capabilities, necessitating the distribution of tasks across a cluster of machines to enable parallel processing. Cloud computing addresses these needs by providing increased compute and storage capacities, complemented by networking capabilities conducive to distributed processing. Moreover, it offers high flexibility and elasticity, facilitating seamless scaling up and down as required. Additionally, it offers cost-efficiency and simplified management, owing to the abstraction level provided to end-users.

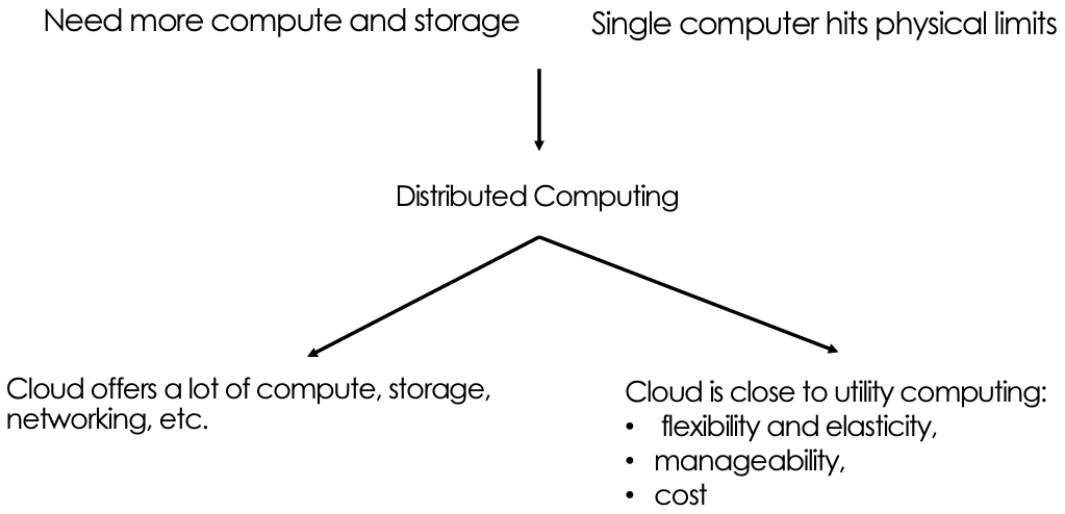

Figure 1: Cloud Computing - Utilities

2.1 Open Question

Google has pioneered and created many distributed systems and technologies that shape today's cloud computing, but why Amazon (and even Microsoft) wins over Google Cloud (GCP) on Cloud computing market shares?

Professor's Argument: Google tackles the problem using a *monolithic* architecture, where every incremental development (services, business logic, etc.) is bundled together in a single repository. New applications or processes are then built using this encapsulation from the single repository as a reference interface.

Amazon employs a pure *microservices-based* architecture, where each service resides in a separate repository, addressing a specific task or purpose. This architectural approach embraces service disaggregation, aligning with the current trend in cloud computing, which emphasizes increased disaggregation. As a result, user experience is enriched, offering greater flexibility and convenience.

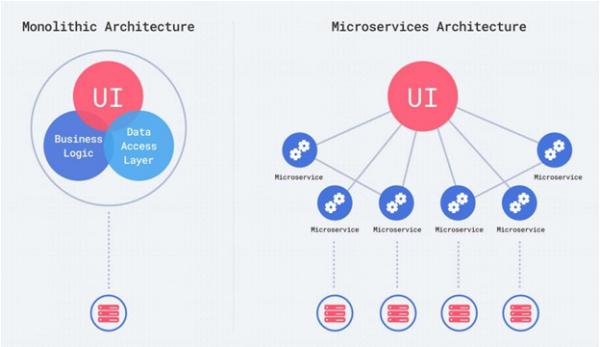

Figure 2: Monolithic v/s Microservices-Based Architecture

3 Cloud Computing Layers

As cloud computing continues to gain prominence, many companies are constructing comprehensive ecosystems atop the cloud infrastructure. The cloud can be conceptualized as a stack composed of the following layers:

1. **Hardware:** This layer accounts for the hardware resources necessary to support this architecture.
2. **Infrastructure:** At this layer, the virtualization of resources provisioned in the hardware layer occurs.
3. **Platform:** This layer represents platforms or services designed to leverage the virtualized resources.
4. **Application:** At this layer, user-facing applications are built by utilizing platforms that service one layer below. This layer is also referred to as the Software layer.

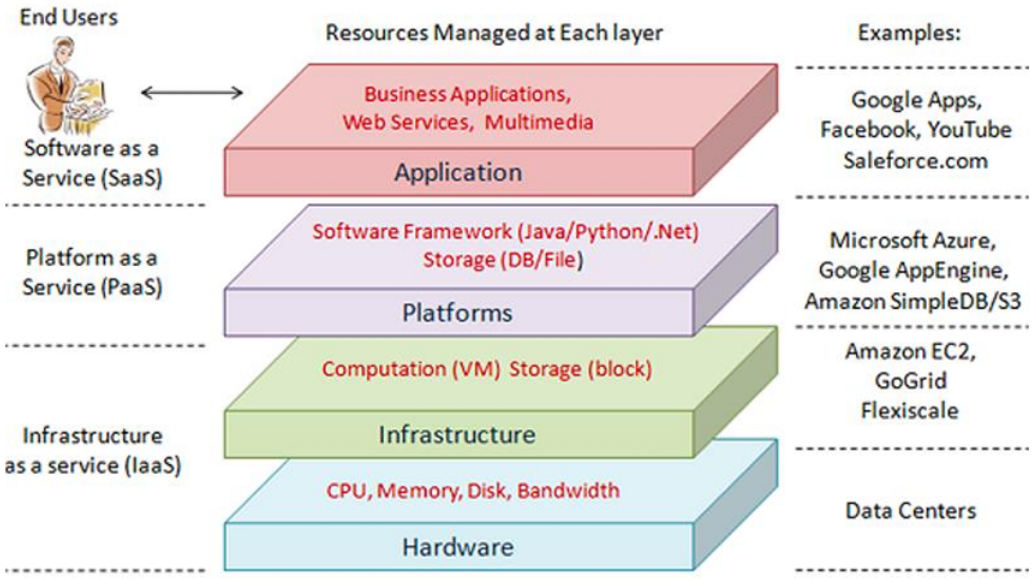

Figure 3: Cloud Computing Layers

4 Cloud - Tipping Point

In the current era, having distributed compute power is paramount for handling computationally-intensive tasks. As a result, many companies operating above the hardware layer have become heavily reliant on hardware providers, particularly chip makers. Among these providers, NVIDIA has emerged as the dominant force, monopolizing this market. It supplies GPUs to major tech companies in need of hardware for their compute requirements. With tasks becoming increasingly complex, these companies consistently demand more compute power, leading to a surge in GPU prices.

4.1 New Trends

4.1.1 GPU Reservation

Given the high demand for GPUs, providing them on-demand to individual users has become challenging, as major tech companies prioritize acquiring these resources. Cloud providers have implemented reservation systems for GPU leasing, similar to those in place for other compute resources - prioritizing users that pay more. As a result, businesses have emerged in this space to facilitate general-purpose users in accessing GPUs for spot usage through GPU vertical clouds and community clouds. This approach offers the added benefit of reduced prices compared to reserving GPUs from major cloud platforms.

4.1.2 On-Premise Supercomputers

As GPUs continue to increase in cost, some major players opt not to lease them at high reservation rates. Instead, they allocating their resources towards developing in-house chips and constructing personalized supercomputer centers. This strategic move aims to reduce dependency on GPU providers, whether they are GPU manufacturing firms or major cloud platforms.

5 Networking

In today's interconnected world, network interactions are commonplace, involving multiple components across diverse domains. Let's consider a simple scenario: triggering a search query from a mobile device. This action sets off a chain of communication. The device first consults a domain name server to locate the address of the system responsible for retrieving relevant details. Once identified, the device attempts to reach the destination system, often via a load balancer, which efficiently distributes incoming queries while managing high volumes of requests. Subsequently, the query is dispatched to the destination system, which retrieves and sends back the required information.

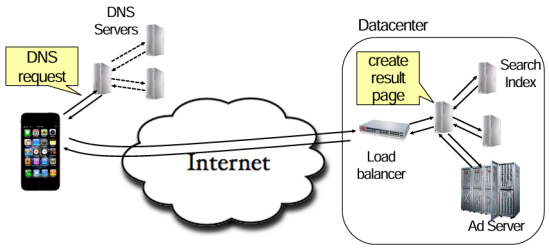

Figure 4: A day-to-day networking example

6 Networking Basics

Networking is a fundamental part of computer systems, functioning as a bridge between components. Each component of a computer system, including networking, is represented in two forms: software and hardware.

6.1 Network Hardware

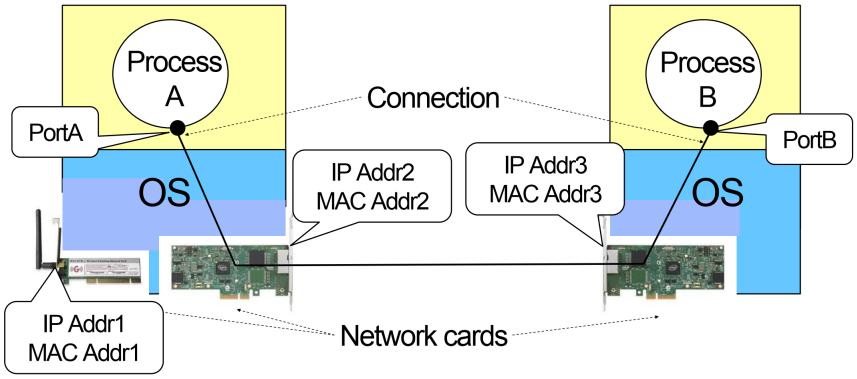

Figure 5: Inter Process Communication

As shown in the 5 the Operating System (OS) serves as the foundational layer, with Processes operating above it. Networking Hardware, an essential component within the computer, is orchestrated by the OS. Network Cards are integral to this setup, and come in various forms - wired and wireless.

6.1.1 Network Interface Card/Controller

The Network Interface Card (NIC) is hardware responsible for physically connecting a computer to a network. A huge industry is dedicated to evolving these network controllers. Despite the variety of network controllers available, industry standards ensure compatibility and interoperability among different network controller.

6.1.1.1 Network Addresses

Each Network Interface Card is associated with two distinct types of addresses:

1. **Media Access Control (MAC) Address (Physical Address)**
   - The MAC address, unique to each network card, ensures that the operating system recognizes the hardware.
   - It is a 48-bit identifier, distinctively assigned to the network card by the manufacturer.
2. **Internet Protocol (IP) Address (Logical or Virtual Address)**
   - The IP address may be common to network cards on the same host, hence not exclusively unique to each card.

- Depending on the network version, the IP address can be a 32-bit identifier for IPv4 or a 128-bit identifier for IPv6. It is assigned by the network administrator or is dynamically allocated when the computer connects to a network.

Question: Why Do We Need IP Addresses in Addition to MAC Addresses?
MAC addresses, while unique, are tied to the hardware and originate from various vendors, leading to a vast and diverse range of identifiers. This diversity can complicate the management and monitoring of devices, especially in large-scale networks. IP addresses streamline this process by providing a structured and hierarchical addressing system. This system is not only essential for managing devices within a network but is also crucial for enabling communication over the internet and larger networks.

6.1.1.2 Connection

A connection is a communication channel between two processes across a network. The operating system uses the MAC address for local network interactions and assigns an IP address to the device for network-layer identification. Processes establish connections using ports, which, when combined with an IP address, form a unique socket for the connection. This unique pairing of an IP address and a port number, as depicted in Figure 5, allows for distinct communication endpoints. Each endpoint is identified by a port number.

6.1.1.3 Common Port Numbers

Application	**Port number**
Wake-on-LAN	9
FTP data	20
FTP control	21
SSH	22
Telnet	23
DNS	53
HTTP	80
SNMP	161
...	...

Table 1: Common Network Applications and Their Port Numbers

6.2 Network Software

6.2.1 Abstract Network

A network comprises multiple nodes connected by communication links. These links facilitate the exchange of data between the nodes. Given that the underlying network hardware may vary widely, it is crucial to develop network software capable of enabling seamless communication across diverse hardware platforms. This software ensures that the nodes can effectively transmit information over the links, despite the differences in hardware.

6.2.1.1 Links: Basic Building Blocks

In a practical scenario, numerous nodes exist within a network, and while establishing a physical connection between each pair of nodes would ensure direct communication, it is impractical due to the high cost and complexity. The networking infrastructure should be designed such that, despite a limited number of physical links, each node can communicate with any other node in the network. The network hardware and software must address scalability to accommodate the growth of the network without affecting the network efficiency.

6.2.1.2 Multiplexing

The concept of multiplexing in networking involves adding interior nodes that function as switches. These intermediate nodes are strategically placed to manage traffic flow efficiently, allowing each endpoint in the network to communicate via these switch nodes. Rather than having a dedicated link for each pair of endpoints, multiple endpoints share the same physical path through the network, with switches directing traffic as needed. This method of resource sharing among multiple end nodes is central to modern networking.

6.2.1.3 Circuit switching

During the times of telephones, human operators served in the capacity of routers. These operators would consult a directory and manually establish a connection to the intended recipient.

The idea is to generate the router for this physical person. Routing is a core concept in networking.

6.2.1.4 Packet Switching

- The source sends information as self-contained packets, each with its own address.
- The source divides it into multiple smaller packets.
- Each packet navigates independently to the destination host.
- Network switches read the packet's address to determine how to forward it towards its destination.
- The technique of 'store and forward' is employed, meaning switches temporarily store the packet before forwarding.

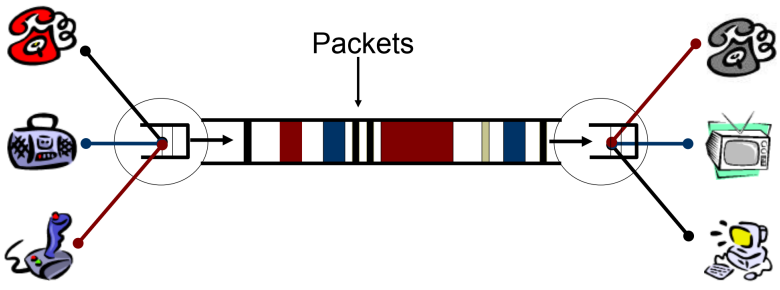

Figure 6: Packet Switching

Switches arbitrate between input sources, allowing data to be sent from any input that's ready, ensuring that network links are never idle. This approach boosts efficiency, making the best use of shared network links. One effective method employed is the FIFO (First-In-First-Out) approach for sending packets, ensuring a fair and orderly transmission of data among users.

Question: Why should the data should be broken into smaller packets?
Smaller packet sizes ensure more reliable transmission over potentially unstable networks because if an error occurs, only the affected packets need to be resent not the entire file.

Question: Why should packet sizes not be excessively too small?
Very small packets will increase the overall network overhead due to the greater number of packets required for data transmission, which can elevate latency and decrease network efficiency.

Question: What if the Network is Overloaded?
Addressing network overload involves the use of buffering and congestion control. In cases of short bursts of data, buffering comes into play, effectively forming a queue to handle incoming packets. However, if the buffer overflows, packets may be dropped. To mitigate this, congestion control comes into picture where the sender adjusts its data transmission rate to match the available network resources, ensuring efficient data transmission.

6.2.1.5 Characterizing Network Communication

Network communication is characterized by several key parameters:

- **Latency**: This measures the time it takes for the first bit of data to reach its destination.
- **Capacity (Bandwidth)**: It is maximum data transfer rate in bits per second of the network.
- **Jitter**: It assesses the variation in latency.
- **Loss / Reliability**: This parameter assesses whether the network channel can drop packets.
- **Reordering**: This refers to when network packets reach their destination out of order.

6.2.1.6 Packet Delay

$$PacketDelay = PropagationDelay + TransmissionDelay + ProcessingDelay + QueuingDelay$$

Packet delay, a critical factor in network performance, is the sum of various delay components. These components include propagation delay, which is directly related to the length of each network link, transmission delay influenced by the packet size and inversely proportional to link speed, processing delay determined by the router's speed, and queuing delay, which varies based on network traffic load and queue size. Understanding and managing these delay components is crucial for optimizing network efficiency and ensuring timely data transmission.

6.2.1.7 Throughput

When streaming packets, the network functions as a pipeline where all links concurrently forward different packets. However, the overall throughput is constrained by the slowest stage, which is known as the bottleneck link. The reasons for this bottleneck could vary, including factors like low link bandwidth or multiple users sharing the link's bandwidth.