Distributed Systems (HDSDEV_SEPOL_YR2)

Lecturer: Haider Khalid
hkhalid@staff.ncirl.ie

Week 3: 05/10/2022

Student Online Teaching Advice Notice

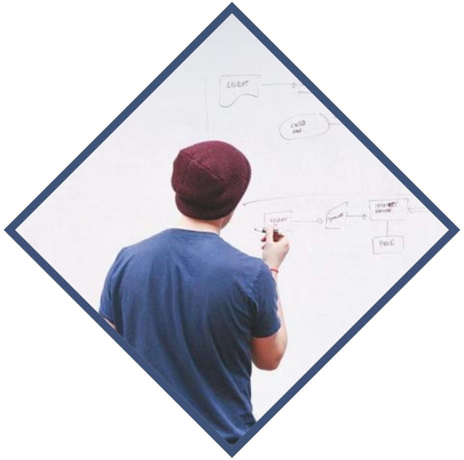
Instruction

- The **materials and content presented** within this session are intended solely for use in a context of **teaching and learning at NCI**.
- Any **session recorded** for subsequent review is made available solely for the purpose of **enhancing student learning**.
- Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at NCI.

- Further information on data protection, privacy policy is available at:

https://www.ncirl.ie/Legal/Privacy

https://ncisupporthub.ncirl.ie/hc/en-ie/articles/4406614299164-Data-Protection-Policy



HELLO!

I am Haider Khalid

I am here Associate Faculty Lecturer.

You can reach me out at hkhalid@staff.ncirl.ie

3

Distributed Systems

Recap of Week 2

- General Concept of communication Protocol.
- TCP (Transmission Control Protocol) and REST concepts:
  - TCP characteristics.
  - TCP connection establishment.
  - Three-way handshake example.
- The Hypertext Transfer Protocol (HTTP) concept
  - HTTP request (client and server side communication)
  - HTTP Methods (GET, PUT, POST, DELETE)
  - HTTP request status code (three digit code)
- JavaScript Object Notation (JSON) concepts
  - JSON Syntax rules

  - TCP Example REST

Today's Agenda (week-3)

This week we will discuss the underlying concepts of **distributed systems and UDP**, a very common protocol **for real-time communications**

- Introduction to **types of cloud systems**?
  - What is **cloud Computing**?
  - Real world **use case**
  - Types of **Cloud Systems**
- The **Client-Server Model**
- Cloud Computing Services:
  - **Iaas, Paas, Saas**
- **Parallel and Distributed Programming Paradigm**
- Difference Between **Parallel and Distributed Computing**
- **UDP Concepts (user diagram protocol)**
- A Web Socket example.

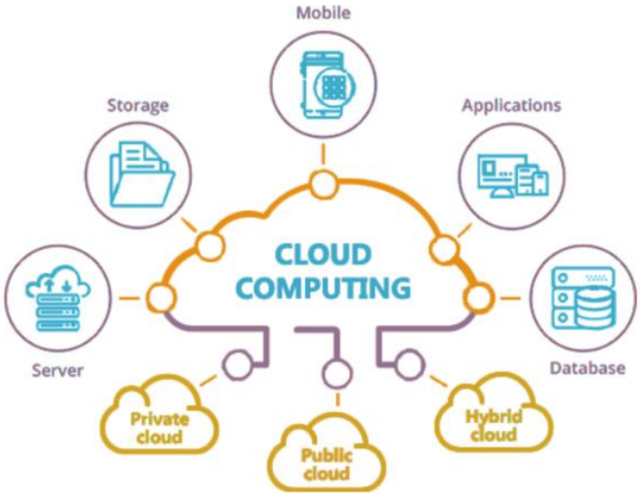
What is Cloud Computing

Definition:

Cloud computing is the **delivery of computing services** such as **software, databases, servers and networking, over the internet.** This means end users are able to access software and applications from wherever they are.

Examples:

- **Cloud Storage:** Dropbox, Gmail, Facebook
- **Marketing Cloud Platforms:** Maropost for Marketing, Hubspot, Adobe Marketing Cloud
- **Cloud Computing in Education:** SlideRocket, Ratatype, Amazon Web Services.
- **Cloud Computing in Healthcare:** ClearDATA, Dell's Secure Healthcare Cloud, IBM Cloud



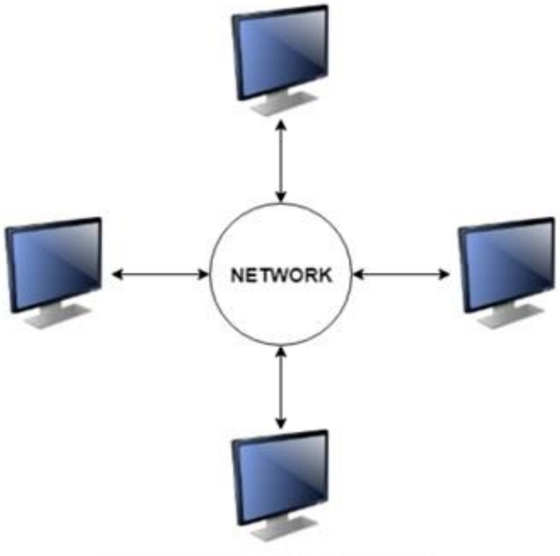
Distributed System

Definition:

A **distributed system is a system** with **multiple components located on different machines** that **communicate and coordinate** actions in order to appear as a **single coherent system to the end user**.

Middleware enables the various component of the distributed system to communicate and manage data.

Middleware **often enables interoperability** between **applications** that run on **different operating systems** or **different programming languages** by **supplying services to the application can exchange data** in the standard-based way.



DISTRIBUTED OPERATING SYSTEM

Types of Public Cloud System

Cloud System Types:

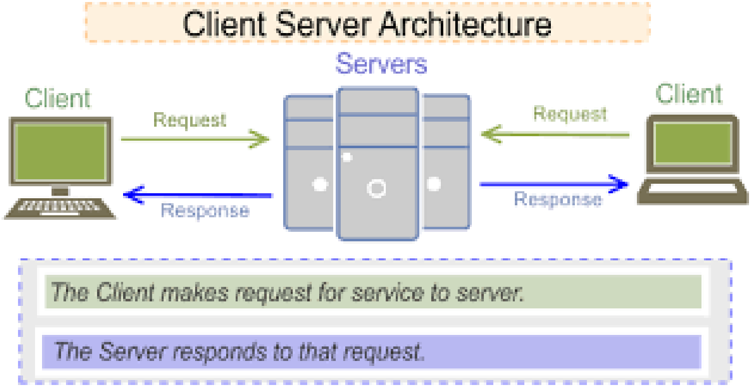
- **Public Cloud:** off-premises provision of computing services by a third party provider to the general public.
- **Community Cloud:** Cloud System used by distinct groups of organizations that shares a common set of concerns.
- **Private Cloud:** Cloud system housed within an organization own datacenter
  - Maybe necessary due to legal consideration
  - Can leverage cloud capabilities
  - Requires significant resources and development costs
- **Hybrid Cloud:** Combines private cloud with a public cloud, allowing data and applications to be shared between them.

Client-Server Model

Client Server Architecture:

Client-server architecture, **alternatively called a client-server model**, is a **network application that breaks down tasks and workloads between clients and servers** that reside on the same system or are linked by a computer network.

- Simply put, **two factors are involved** :
  - A server is the one **who provides requested services**.
  - Clients are the **ones who request services**.



The Client makes request for service to server.

The Server responds to that request.

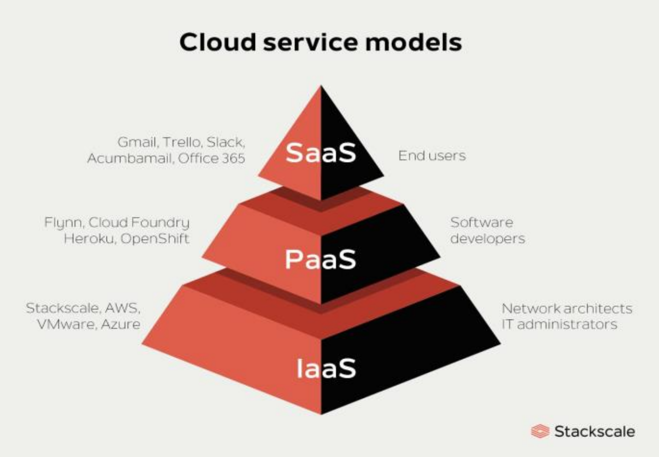
Cloud Computing Services

Understand and compare the three most popular cloud computing service models

IaaS, PaaS and SaaS are the **three most popular types of cloud service offerings**. (They are sometimes referred to as cloud service models or cloud computing service models.)

- **IaaS,** or infrastructure as a service, is on-demand access to cloud-hosted physical and virtual servers, storage and networking - the backend IT infrastructure for running applications and workloads in the cloud. **Example Vmware, AWS.**
- **PaaS,** or platform as a service, is on-demand access to a complete, ready-to-use, cloud-hosted platform for developing, running, maintaining and managing applications. **Example Windows Azure, Cloud bees.**
- **SaaS,** or software as a service, is on-demand access to ready-to-use, cloud-hosted application software. **Example mail chimp, google docs.**

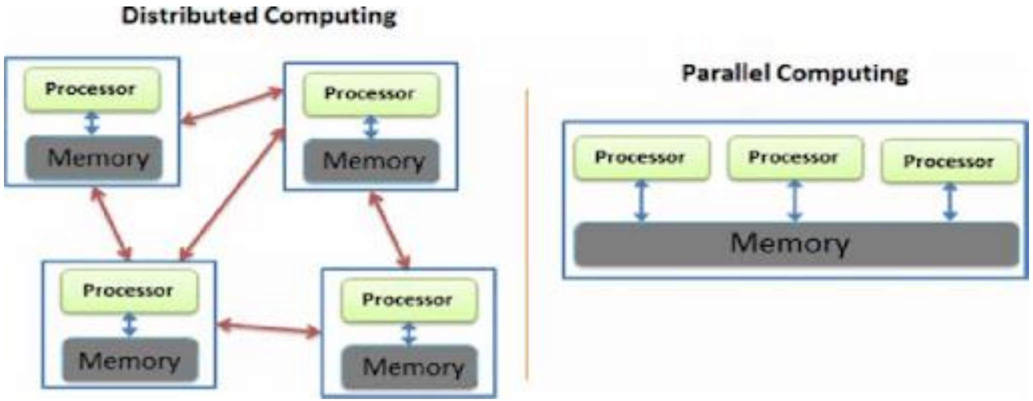
https://www.ibm.com/cloud/learn/iaas-paas-saas


Cloud service models
Gmail, Trello, Slack, Acumbamail, Office 365
SaaS
End users
Flynn, Cloud Foundry Heroku, OpenShift
PaaS
Software developers
Stackscale, AWS, VMware, Azure
IaaS
Network architects IT administrators
Stackscale

Difference Between Parallel and Distributed Computing

Parallel and Distributed Computing

The **main difference between parallel and distributed computing** is that **parallel computing allows multiple processors to execute tasks simultaneously** while distributed computing divides a **single task between multiple computers to achieve a common goal.**



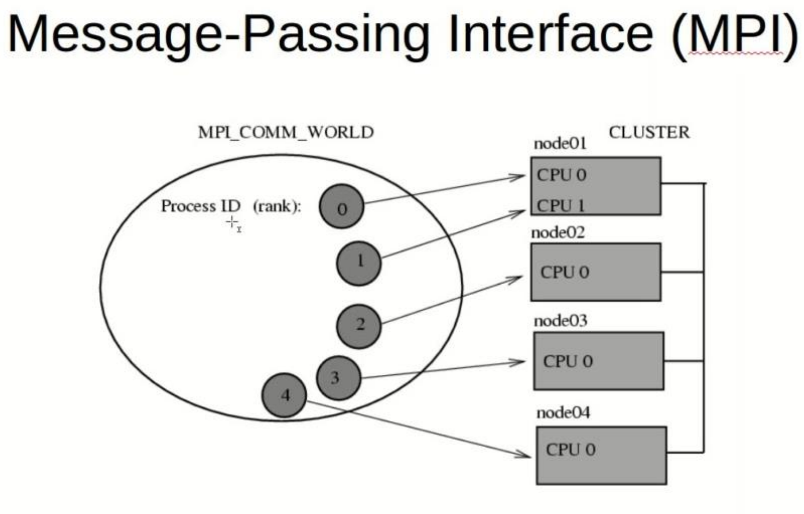
Parallel and Distributed Programming Paradigm

In the **context of processing data**, parallel and distributed programming **is the simultaneous use of more than one computational engine** (not necessarily connected via a network) to run a job or an application.

The motivation for **Parallel and Distributed Programming, improve processing time and to leverage underlying resources more efficiently**.

Message-Passing Interface (MPI)

In parallel computing, **multiple computers – or even multiple processor cores within the same computer – are called nodes.** Each node in the parallel arrangement typically works on a portion of the **overall computing problem**. The challenge then is to **synchronize the actions of each parallel node**, exchange **data between nodes**, and provide **command and control over the entire parallel cluster.**



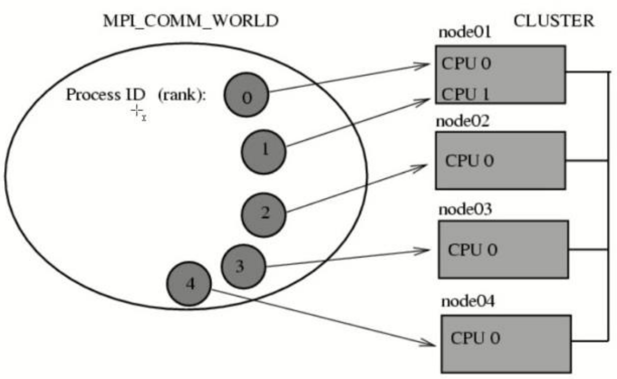
Message-Passing Interface (MPI)

Message-Passing Interface (MPI)

The **message passing interface (MPI)** defines a **standard suite of functions for these tasks.** The term message passing itself typically refers to the **sending of a message to an object**, parallel process, subroutine, function or thread, which is then used to start another process.

Message Passing Interface (MPI) is a **specification for software developers** used to make **use of a cluster of computers.** A set of libraries exist for using this standard on modern day (High Performance Computing) HPC Clusters.

Message-Passing Interface (MPI)

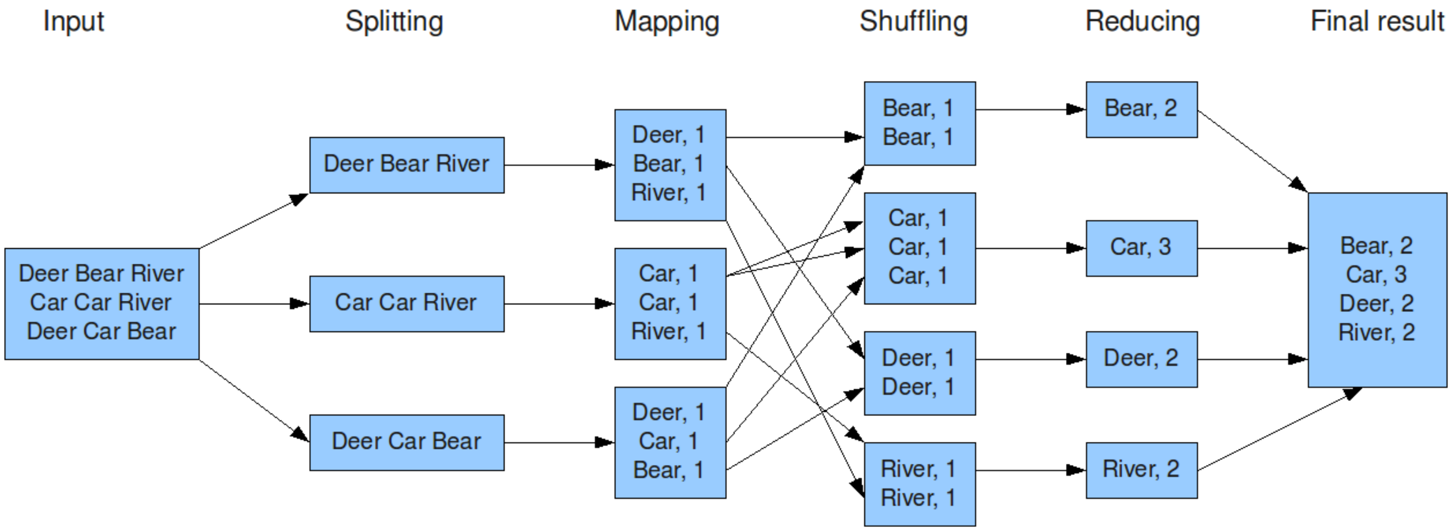


MapReduce

- **MapReduce is a web programming model** for scalable **data processing on large clusters over large data sets.**
- A MapReduce program is **composed of a map function**, which performs **filtering and sorting and a reduce method**, which performs a summary operation.
- The map reduce model is **applied mainly in web-scale search and cloud computing applications**.
- **Applications:** making item proposal Mechanisms for E-commerce inventories, examining website records, buy history, user interaction logs, etc
- MapReduce is **highly scalable**
- MapReduce can take advantage of the locality of data, processing it near the place it is stored in order to minimize communication overhead.

MapReduce

The overall MapReduce word count process



Hadoop

- Apache Hadoop is a **collection of open-source software utilities** that **facilitates using a network** of many computers to **solve problems involving massive amounts of data and computation.**
- Hadoop is an **open source implementation of MapReduce** coded and released in Java.
- Hadoop is a scalable, **economical, efficient, and reliable** tool for providing users with easy access of **commercial clusters.**
- Hadoop software library to **write and run large user applications** on vast data sets in **business applications.**
- Hadoop can easily **scale to store and process petabytes** of data in the web space.
- Hadoop is specifically designed with a **fundamental assumption that hardware failures** are **common occurrences and are handled by Hadoop.**
- Various Hadoop applications include **stream processing, fraud detection, and prevention, content management, risk management.**

UDP Concepts

User Datagram Protocol:

In computer networking, the **User Datagram Protocol is one of the core communication protocols** of the Internet protocol suite used to send messages to other hosts on an Internet Protocol network.

It is an **unreliable, connectionless protocol for applications** that do not want TCP’s sequencing or flow control and wish to provides their own

- UDP segment maybe lost delivered out of order therefore unreliable
- No handshaking between sender and receiver.

UDP Concepts

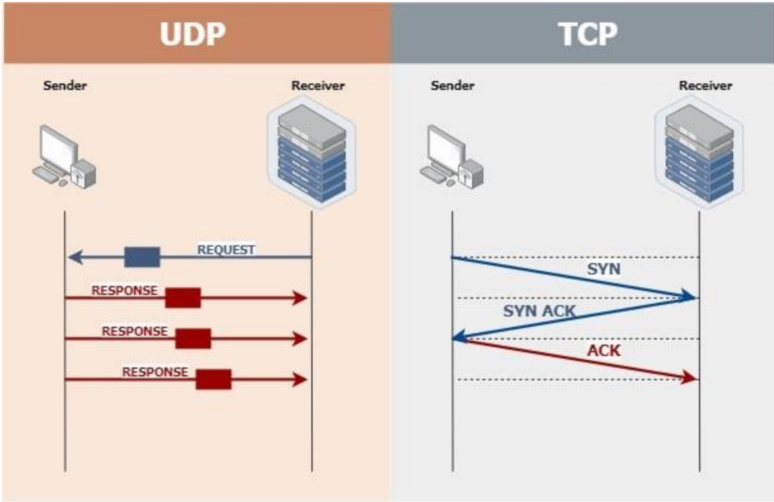
User Datagram Protocol:

It is also **widely used for one-shot**, client-server-type request reply queries and applications in which **timely delivery is more impotent than accuracy delivery.**

- Such as speech/video transmission streaming media, real-time multiplayer games.

For reliable transfer over UDP

- Applications are required to add reliability
- Applications are required to add specific error recovery



Web Socket Example (Installation)

Web Socket Example:

Steps:

- Create folder “socket-example”.
- Open socket-example folder and inside socket-example folder create two folders “server” and “client”.
- Open cmd.
- Go to “server” directory. → ……..cd server.
- Install npm → npm init
- Entry point → app.js (rent leave blank by pressing enter)
- Open both server and client folder in Atom
- Open server folder in Atom and open package.json file and change “test” to start and value node app.js

Web Socket Example:

Steps:

- Create new file “app.js” (inside server folder) which is the starting point.
- Go back to cmd and install http and socket using command the followed command.
- npm install http
- npm install socket.io

- Web sockets are real time based communication.
- Rest follow the code in the video

THANKS!

You can find me at
hkhalid@staff.ncirl.ie & Teams
Discussion chat on teams