****

**Faculty of Computing**

**RIPHAH INTERNATIONAL UNIVERSITY**

**Assignment 1**

|  |  |
| --- | --- |
| **Submitted by** | **CMS No.** |
| Hussnain Ahmad | 10746 |
|  |  |

Course Code – **Parallel and Distributed Computing**

BS-CS (7C)

instructor(s): M.USMAN

Date: 27/10/2021

**Assignment 1**

**Question1: -**

**Edge Computing and its architecture/Working.**

**Sol:-**

**Edge Computing:**

Edge computing is a type of computing that takes place at or near the edge of a network. The processing occurs either within or close to the device, so less data travels to the central server. Most operations happen in real-time near the source of data, which leads to:

* Improved response times.
* Better [bandwidth availability](https://phoenixnap.com/kb/linux-network-bandwidth-monitor-traffic).
* More reliable system insights.
* More comprehensive and quicker data analysis.
* [High availability](https://phoenixnap.com/blog/what-is-high-availability).

Edge computing also helps keep workloads up to date, ensure data privacy, and adhere to data protection laws such as [HIPAA](https://phoenixnap.com/blog/hipaa-compliance-checklist), [GDPR](https://phoenixnap.com/blog/gdpr-general-data-protection-regulation), and [PCI](https://phoenixnap.com/blog/pci-compliance-checklist). This processing model also enables further innovations with [artificial intelligence and machine learning](https://phoenixnap.com/blog/future-gpu-machine-learning-ai).

Edge devices collect and store data before sending information to an on-premises edge server. This server handles the following activities:

* Real-time data processing.
* Data visualization and analytics.
* Cashing and buffering.
* Data filtering.

The edge center sends the most complex processing requests (big data operations and business logic) to the [data center](https://phoenixnap.com/colocation) or the cloud. While the need for a [central dedicated server](https://phoenixnap.com/servers/dedicated) is still there, a business can set up slower, less expensive connections without risking latency due to local operations and pre-sorted data.

## **Edge Computing Architecture:**

* **Edge devices:** A special-purpose piece of equipment with limited computing capacity.
* **Edge node:** Any device, server, or gateway that performs edge computing.
* **Edge server:** A computer located in a facility close to the edge device. These machines run application workloads and shared services, so they need more computing power than edge devices.
* **Edge gateway:** An edge server that performs network functions such as tunneling, [firewall](https://phoenixnap.com/blog/types-of-firewalls) management, protocol translation, and wireless connections. A gateway can also host application workloads.
* **Cloud**: A [public or private cloud](https://phoenixnap.com/blog/public-vs-private-cloud) that acts as a repository for containerized workloads like applications and machine learning models. The cloud also hosts and runs apps that manage edge nodes.

Edge computing has three primary nodes: the device edge, local edge, and the cloud.

## **Edge Computing Working:**

Edge computing is all a matter of location. In traditional enterprise computing, data is produced at a client endpoint, such as a user's computer. That data is moved across a WAN such as the internet, through the corporate LAN, where the data is stored and worked upon by an enterprise application. Results of that work are then conveyed back to the client endpoint. This remains a proven and time-tested approach to client-server computing for most typical business applications.

**Question2: -**

**Fog computing and its architecture/working?**

**Sol:-**

**Fog Computing:**

Fog computing is defined as “an extremely [virtualized environment](https://www.sciencedirect.com/topics/computer-science/virtualized-environment) that delivers networking, storage, and compute resources between outdated CC information centers, usually, but not entirely situated at the network edge. A fog structure contains various edge nodes with few processing competences, which are frequently called fog nodes. These nodes of fog have less processing facilities and storage. In fog network, sometimes edge and many servers are called [cloudlets](https://www.sciencedirect.com/topics/computer-science/cloudlet) which take part in the shared computing surroundings, not outside the network edge. By using these devices of fog, the clients might obtain a real-time response for sensitive latency applications. Even though the phrase was initially devised by Cisco , various researchers and industries defined fog computing from many different perspectives.

A broad spectrum of Fog computing is provided by Yi et al.. It is specified as “geographically shared computing framework with a pool of requirements that contains different universally linked heterogeneous computing devices at the network edge and not entirely flawlessly supported by services of cloud to collectively offer transmission, storage and elastic computation in remote surroundings to an enormous scale of users in closeness. A spectrum of Fog Computing is provided by Open Fog Consortium  as “system-level flat framework that divides storage, resources, computing services, and networking from every place along with the range from Cloud to Things.

**Fog Computing Working:**

Fog computing uses the concept of ‘fog nodes.’ These fog nodes are located closer to the data source and have higher processing and storage capabilities. Fog nodes can process the data far quicker than sending the request to the cloud for centralized processing.

The cloud is getting cluttered due to the enormous number of devices connecting to the internet. Since cloud computing is not viable in some cases, it has become necessary to use fog computing for IoT devices. It can handle the enormous data generated by these devices.

When implemented, fog-empowered devices locally analyze time-critical data that includes alarm status, device status, fault warnings, and so on. This minimizes latency and prevents major damage. Fog computing can effectively reduce the amount of bandwidth required, which in turn speeds up the communication with the cloud and various sensors.

**Fog Computing Architecture:**

Digging in deeper on what Fog Computing Architecture comprises, you will learn that it consists vastly of the physical and logical elements of the network, including the hardware and software. This establishes a large number of interconnecting devices.  Fog Computing Architecture contains multiple layers. These layers consist of many networking devices such as routers, set‐top boxes, proxy servers, base stations, etc. These levels are assigned various tasks to be conducted on the given data. Fog Computing Architecture is implemented in two predominant models the Hierarchical Architecture Model and the Layered Architecture Model. Let us learn more about these models.

**Question3: -**

**Cloud computing and its architecture/working?**

**Sol:-**

**Cloud Computing:**

[Cloud computing](https://www.simplilearn.com/tutorials/cloud-computing-tutorial/what-is-cloud-computing) refers to services like storage, databases, software, analytics, and other platforms that are accessible via the internet. It is any service that can be delivered without being physically close to the hardware. For example, Netflix uses cloud computing for its video streaming services. Another example is G Suite, which runs entirely on the cloud.

Simply put, Cloud Computing refers to the delivery of on-demand resources (such as a server, database, software, etc.) over the internet. It also gives the ability to build, design, and manage applications on the cloud platform.

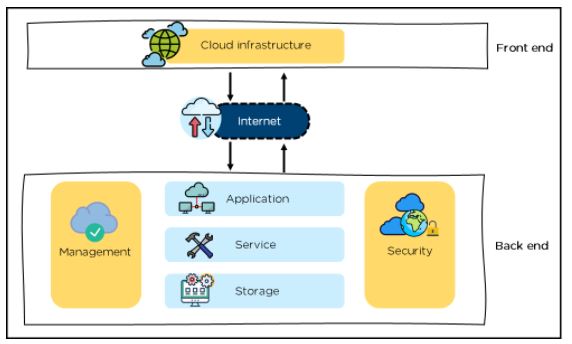
**Cloud Computing Service Providers:**

A few of the most popular cloud computing service providers include:

* [Microsoft Azure](https://www.simplilearn.com/azure-cloud-architect-certification-training-course)
* [Amazon Web Services (AWS)](https://www.simplilearn.com/tutorials/aws-tutorial/what-is-aws)
* Google Cloud
* Alibaba Cloud
* IBM Cloud
* Oracle
* Salesforce
* SAP
* Rackspace Cloud
* [VM Ware](https://www.simplilearn.com/tutorials/cloud-computing-tutorial/vmware-workstation)

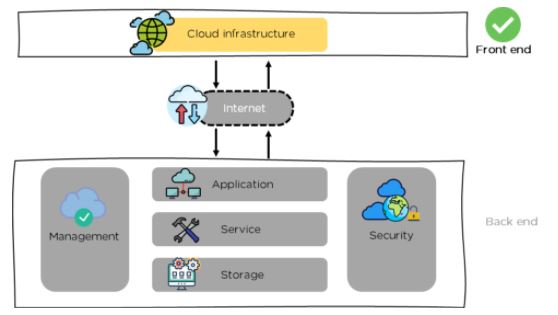
**Cloud Computing Architecture:**

Cloud Computing Architecture is divided into two parts, i.e., front-end and back-end. Front-end and back-end communicate via a network or internet. A diagrammatic representation of cloud computing architecture is shown below:



### **Front-End:**

* It provides applications and the interfaces that are required for the cloud-based service.
* It consists of client’s side applications, which are web browsers such as Google Chrome and Internet Explorer.
* Cloud infrastructure is the only component of the front-end. Let's understand it in detail.



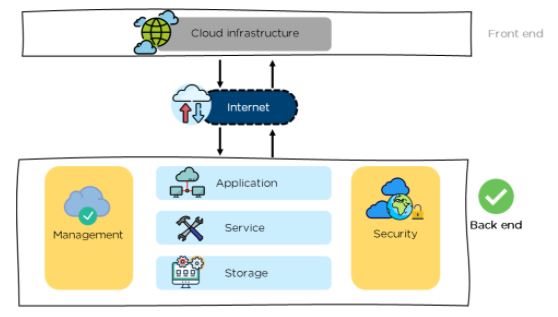
* Cloud infrastructure consists of hardware and software components such as data storage, server, virtualization software, etc.
* It also provides a Graphical User Interface to the end-users to perform respective tasks.

Moving ahead, let’s understand what the back-end is.

### **Back-End:**

It is responsible for monitoring all the programs that run the application on the front-end

It has a large number of data storage systems and servers. The back-end is an important and huge part of the whole cloud computing architecture, as shown below:



The components of the back-end cloud architecture are mentioned below. Let's understand them in detail one by one.

**Cloud Computing Working:**

Cloud computing is an application-based software infrastructure that stores data on remote serves, which can be accessed through the internet. To understand how cloud computing works, it can be divided into front-end and backend.

The front end enables a user to access data stored in the cloud using an internet browser or a [cloud computing software](https://www.hcltech.com/services/cloud-computing). However, the primary component of cloud computing – responsible for securely storing data and information – is the backend. It comprises servers, computers, databases, and central servers.

The central server facilitates operations by following a set of rules known as protocols. It uses a software, middleware, to ensure seamless connectivity between devices/computers linked via cloud computing. Cloud computing service providers usually maintain multiple copies of the data to mitigate instances of security threats, data loss, data breach, etc.

**Question4: -**

**Briefly explain Parallel & Distributed computing (Architectures/Models & Working)?**

**Sol:-**

## **Parallel Computing:**

Parallel computing refers to the process of breaking down larger problems into smaller, independent, often similar parts that can be executed simultaneously by multiple processors communicating via shared memory, the results of which are combined upon completion as part of an overall algorithm. The primary goal of parallel computing is to increase available computation power for faster application processing and problem solving.

Parallel computing infrastructure is typically housed within a single datacenter where several processors are installed in a server rack; computation requests are distributed in small chunks by the application server that are then executed simultaneously on each server.

There are generally four types of parallel computing, available from both proprietary and open source parallel computing vendors -- bit-level parallelism, instruction-level parallelism, task parallelism, or superword-level parallelism:

* **Bit-level parallelism:** increases processor word size, which reduces the quantity of instructions the processor must execute in order to perform an operation on variables greater than the length of the word.
* **Instruction-level parallelism:** the hardware approach works upon dynamic parallelism, in which the processor decides at run-time which instructions to execute in parallel; the software approach works upon static parallelism, in which the compiler decides which instructions to execute in parallel
* **Task parallelism:** a form of parallelization of computer code across multiple processors that runs several different tasks at the same time on the same data
* **Super word-level parallelism:** a vectorization technique that can exploit parallelism of inline code

## **Parallel Computer Architecture:**

Parallel computer architecture exists in a wide variety of parallel computers, classified according to the level at which the hardware supports parallelism. Parallel computer architecture and programming techniques work together to effectively utilize these machines. The classes of parallel computer architectures include:

* **Multi-core computing:** A multi-core processor is a computer processor integrated circuit with two or more separate processing cores, each of which executes program instructions in parallel. Cores are integrated onto multiple dies in a single chip package or onto a single integrated circuit die, and may implement architectures such as multithreading, superscalar, vector, or VLIW. Multi-core architectures are categorized as either homogeneous, which includes only identical cores, or heterogeneous, which includes cores that are not identical.
* **Symmetric multiprocessing:** multiprocessor computer hardware and software architecture in which two or more independent, homogeneous processors are controlled by a single operating system instance that treats all processors equally, and is connected to a single, shared main memory with full access to all common resources and devices. Each processor has a private cache memory, may be connected using on-chip mesh networks, and can work on any task no matter where the data for that task is located in memory.
* **Distributed computing**: Distributed system components are located on different networked computers that coordinate their actions by communicating via pure HTTP, RPC-like connectors, and message queues. Significant characteristics of distributed systems include independent failure of components and concurrency of components. Distributed programming is typically categorized as client–server, three-tier, n-tier, or peer-to-peer architectures. There is much overlap in distributed and parallel computing and the terms are sometimes used interchangeably. ‍
* **Massively parallel computing:** refers to the use of numerous computers or computer processors to simultaneously execute a set of computations in parallel. One approach involves the grouping of several processors in a tightly structured, centralized computer cluster. Another approach is grid computing, in which many widely distributed computers work together and communicate via the Internet to solve a particular problem.

Other parallel computer architectures include specialized parallel computers, cluster computing, grid computing, vector processors, application-specific integrated circuits, general-purpose computing on graphics processing units ([GPGPU](https://www.omnisci.com/technical-glossary/gpgpu)), and reconfigurable computing with field-programmable gate arrays. Main memory in any parallel computer structure is either distributed memory or shared memory.

## **Distributed Computing:**

Distributed computing is a computing concept that, in its most general sense, refers to multiple computer systems working on a single problem. In distributed computing, a single problem is divided into many parts, and each part is solved by different computers. As long as the computers are networked, they can communicate with each other to solve the problem. If done properly, the computers perform like a single entity.

The ultimate goal of distributed computing is to maximize performance by connecting users and IT resources in a cost-effective, transparent and reliable manner. It also ensures fault tolerance and enables resource accessibility in the event that one of the components fails.

## **Distributed System Architecture:**

* Distributed systems must have a network that connects all components (machines, hardware, or software) together so they can transfer messages to communicate with each other.
* That network could be connected with an IP address or use cables or even on a circuit board.
* The messages passed between machines contain forms of data that the systems want to share like databases, objects, and files.
* The way the messages are communicated reliably whether it’s sent, received, acknowledged or how a node retries on failure is an important feature of a distributed system.
* Distributed systems were created out of necessity as services and applications needed to scale and new machines needed to be added and managed. In the design of distributed systems, the major trade-off to consider is complexity vs performance.

## **Types of Distributed System Architectures:**

Distributed applications and processes typically use one of four architecture types below:

**Client-server:**

In the early days, distributed systems architecture consisted of a server as a shared resource like a printer, database, or a web server. It had multiple clients (for example, users behind computers) that decide when to use the shared resource, how to use and display it, change data, and send it back to the server. Code repositories like git is a good example where the intelligence is placed on the developers committing the changes to the code.

Today, distributed systems architecture has evolved with web applicationsinto:

* **Three-tier:** In this architecture, the clients no longer need to be intelligent and can rely on a middle tier to do the processing and decision making. Most of the first web applications fall under this category. The middle tier could be called an agent that receives requests from clients, that could be stateless, processes the data and then forwards it on to the servers.
* **Multi-tier:** Enterprise web services first created n-tier or multi-tier systems architectures. This popularized the application servers that contain the business logic and interacts both with the data tiers and presentation tiers.
* **Peer-to-peer:** There are no centralized or special machine that does the heavy lifting and intelligent work in this architecture. All the decision making and responsibilities are split up amongst the machines involved and each could take on client or server roles. Blockchain is a good example of this.

In parallel computing, all processors may have access to a [shared memory](https://en.wikipedia.org/wiki/Shared_memory_architecture) to exchange information between processors.[[19]](https://en.wikipedia.org/wiki/Distributed_computing#cite_note-19)

In distributed computing, each processor has its own private memory ([distributed memory](https://en.wikipedia.org/wiki/Distributed_memory)). Information is exchanged by passing messages between the processors.