**CLOUD COMPUTING**

**ASSIGNMENT**

1. Explain High-performance computing and High-throughput computing.

Answer:

High Performance Computing:

High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.

It is the use of super computers and parallel processing techniques to solve complex problems. It focuses on developing parallel processing algorithms to enhance the performance of the HPC to solve research-based problems. It is used mostly in the following avenues:

Simulation

Bio-Sciences

Weather Models

Geographical Data, etc.

High Throughput Computing:

In contrast to HPC, high throughput computing does not aim to optimize a single application but several users and applications. In this way, many applications share a computing infrastructure at the same time – in this way the overall throughput of several applications is supposed to be maximized

High Throughput Computing uses many computing resources to complete a computing task.

2. What are the design objectives to achieve High-performance computing and High-throughput computing?

Answer:

The design objectives of HPC are:

to achieve the solution of the problem given by means of parallel computing and high-performance metrics.

to use the resources optimally and achieve the output.

Objectives of HTC:  
to achieve maximum throughput for the process to be completed in minimum time.

To avoid latency and give the result as fast as possible.

3. What are the applications of High-performance computing and High-throughput computing?

Answer:

Applications of HPC and HTC:  
Research purpose

Media and Entertainment

Weather Forecasting, etc.

4. Define Centralized computing, Parallel computing, Distributed computing, Cloud Computing, Ubiquitous Computing, Internet Computing and Utility computing

Answer:

Centralized Computing:

Centralized computing is a type of computing architecture where all or most of the processing/computing is performed on a central server.

Parallel Computing:

Parallel computing is a type of computation in which many calculations or the execution of processes are carried out simultaneously. Large problems can often be divided into smaller ones, which can then be solved at the same time.

Distributed Computing:

 A distributed system is a system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another.

Cloud Computing:

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet.

Ubiquitous Computing:

Ubiquitous computing (or "ubicomp") is a concept in software engineering and computer science where computing is made to appear anytime and everywhere. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format.

Internet Computing:

Internet computing is the only architecture that supports all information flows and processes over the Internet — providing access to all applications.

Utility Computing:

Utility computing, or The Computer Utility, is a service provisioning model in which a service provider makes computing resources and infrastructure management available to the customer as needed, and charges them for specific usage rather than a flat rate.

5. Explain Service Oriented Architecture

Answer:

Service-oriented architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. The basic principles of service-oriented architecture are independent of vendors, products and technologies. A service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently, such as retrieving a credit card statement online.

Its architecture consists of three parts:

Service provider:

It creates a web service and provides its information to the service registry. Each provider debates upon a lot of hows and whys like which service to expose, which to give more importance: security or easy availability, what price to offer the service for and many more*.* The provider also has to decide what category the service should be listed in for a given broker service and what sort of trading partner agreements are required to use the service.

Service broker, service registry or service repository:

Its main functionality is to make the information regarding the web service available to any potential requester. Whoever implements the broker decides the scope of the broker. Public brokers are available anywhere and everywhere but private brokers are only available to a limited amount of public. UDDI was an early, no longer actively supported attempt to provide Web services discovery.

Service requester/consumer:

It locates entries in the broker registry using various find operations and then binds to the service provider in order to invoke one of its web services. Whichever service the service-consumers need, they have to take it into the brokers, bind it with respective service and then use it. They can access multiple services if the service provides multiple services.