Important Instructions:

1) Open this MS-Word document and start writing answers below each respective question given on page 2.
2) Answers the question in the same sequence in which they appear.
3) Provide to the point and concrete answers. Some of the questions are open ended and therefore must be answered using your own opinion and thoughts but backed with logical reasons.
4) First read the questions and understand what is required of you before writing the answer.
5) Attempt the paper yourself and do not copy from your friends or the Internet. Students with exactly similar answers or copy paste from the Internet will not get any marks for their assignment.
6) You can contact me for help if you have any doubt in the above instructions or the assignment questions.
7) All questions must be attempted.
8) Do not forget to write your name, university ID, class and section information.
9) Rename you answer file with your university ID# before uploading to SIC.
10) When you are finished with writing your answers and are ready to submit your answer, convert it to PDF and upload it to SIC unzipped, before the deadline mentioned on SIC.

Re-Mid Semester Assignment
Course: - Distributed Computing

Deadline: - Mentioned on SIC **Marks: - 30**

Program: - MS (CS) **Dated: 14 June 2020**

Student Name: Iman **Student ID#: 13523**

Class and Section: MSCS

<u>Question1:</u> Discuss how the MMOG's as a Distributed System solves certain challenges due to its distributed architecture. (6)

<u>Question2:</u> Among the trends of Distributed Systems discussed in C1-Lec2, which trend in your opinion will be most dominant in the future and why? (6)

<u>Question3:</u> Among the challenges of Distributed Systems discussed in C1-Lec2, which problem in your opinion will accompany distributed systems into the future and why? (6)

<u>Question4:</u> The design of distributed systems can be described and discussed in three ways i.e Physical Model, Architectural Model and Fundamental Model. Describe the example of distributed system in Question1 with respect to these three models. (6)

<u>Question5:</u> For **the purpose of Inter Process Communication (IPC) in distributed systems, in what situation you will use UDP and TCP and why? (6)**

<u>Question1:</u> Discuss how the MMOG's as a Distributed System solves certain challenges due to its distributed architecture.

Answer;

Massively Multiplayer Online Games (MMOGs)

Massive multiplayer online games provide an immersive experience, and a large number of users interact with the persistent virtual world via the Internet.

The best examples of such games include EverQuest II of the Finnish company CCP Games and Sony's EVE Online.

The complexity of these worlds has greatly increased, and now includes complex game areas (such as EVE, online games are composed of a universe with more than 5,000 star systems) and many different social and economic systems.

The MMOG project is the main challenge of distributed system technology because it requires fast response time to maintain the user experience of the game.

Other challenges include spreading the event to many players in real time, and maintaining a consistent view of the shared world.

Many solutions for designing large-scale multiplayer online games have been proposed.

The largest online game, EVE Online, uses a client-server architecture, which keeps a single copy of the world's conditions on a centralized server and is accessed by client programs running on player consoles or other devices.

To support a large number of clients, the server itself is a complex entity, consisting of a cluster architecture consisting of hundreds of computer nodes.

The centralized architecture simplifies the management of the virtual world, and the single copy also reduces the consistency problem.

Then, the goal is to ensure a rapid response by optimizing network protocols and ensuring a rapid response to incoming events.

To support this, the load is divided by assigning each "star system" to a specific computer in the cluster. A heavily loaded star system has its own dedicated computer, while other star systems share a computer.

Incoming events are routed to the correct computer in the cluster to track the movement of players between galaxies.

Other MMOGs use a more distributed architecture, where the Universe is divided into (possibly very large) number of servers, and these servers can also be distributed by geographic location.

Massively multiplayer online games are online games with a large number of players (usually hundreds or thousands) on the same server.

Although some games are different, MMOG usually presents a huge and lasting open world.

These games are suitable for most network compatible platforms, including personal computers, video game consoles or smart phones and other mobile devices.

MMOGs can enable players to conduct large-scale cooperation and competition, and sometimes even interact meaningfully with people around the world.

<u>Question2:</u> Among the trends of Distributed Systems discussed in C1-Lec2, which trend in your opinion will be most dominant in the future and why?

Answer;

Distributed systems are undergoing an important change, which can be attributed to many impact trends:

The emergence of ubiquitous network technology;

The growing demand for multimedia services; multimedia distributed systems.

Pervasive networking and the modern Internet; The Internet is a large interconnected collection of many different types of computer networks (for example, WiFi, WiMAX, Bluetooth, and third-generation mobile phone networks). The Internet allows programs running anywhere to send messages to programs elsewhere. It enables users to use services such as the World Wide Web, e-mail and file transfers wherever they are. An Internet Service Provider (ISP) is a company that provides broadband and other types of connections to individual users and small organizations, thereby enabling them to access services anywhere on the Internet. For example, police and other security and law enforcement agencies may have at least some internal intranets isolated from the outside world (probably the most effective firewall—without any physical connection to the Internet).

Mobile Computing & Ubiquitous Computing:

Device miniaturization and technological advancements in wireless networks have increasingly led to the integration of small portable computing devices into distributed systems. These devices include: laptop computers.

Mobile computing is human-computer interaction, and it is expected that the computer will be transported during normal use so that data, voice, and video can be transmitted. Mobile computing involves mobile communications, mobile hardware and mobile software. Communication issues include self-organizing networks and infrastructure networks, as well as communication attributes, protocols, data formats, and specific technologies. Mobile software handles the characteristics and requirements of mobile applications.

Portable devices such as smartphones, GPS-compatible devices, pagers (PDAs) and digital cameras. Wearable devices, such as smart watches that function like PDAs.

Built-in appliances in washing machines, high-fidelity audio systems, automobiles and refrigerators.

In mobile computing, users who are far from the "home" intranet (intranet at work or at home) can always access resources through the devices they carry. They can continue to access the Internet (home intranet); more and more, regulations require users to use nearby resources, such as printers or even sockets, while traveling.

Pervasive computing or ubiquitous computing is the use of many small and inexpensive computing devices that exist in users' physical environments (including homes, offices, and even natural environments).

Pervasive computing (also called ubiquitous computing) is a growing trend of integrated computing power.

The term "ubiquitous" is intended to indicate that small computing devices will eventually become so common among everyday objects that they are hardly noticed. Ubiquitous overlaps with mobile computing because mobile users can basically benefit from computers anywhere. But they are usually different. Ubiquitous computing can benefit users while remaining in unique environments such as homes or hospitals. Similarly, even if it involves only traditional and cautious computers and peripherals (such as laptops and printers), mobile computing has advantages. These two Internet are connected to the rest of the Internet. Users can access three forms of wireless connection.

It connects to the rest of the host intranet through a gateway or access point. The user also has a mobile (cellular) phone connected to the Internet. The phone provides access to Web and other Internet services, limited only by the content displayed on its small screen, and can also provide location information through the built-in GPS function.

Question3: **Among the challenges of Distributed Systems discussed in C1-Lec2, which problem in your opinion will accompany distributed systems into the future and why?**

Answer;

Heterogeneity

Internet users access services on heterogeneous computers and network collections. The heterogeneity of the Internet is obscured by the use of Internet protocols.

These differences must be considered so that programs written in different languages can communicate with each other. Unless they use common standards, programs written by different developers cannot communicate with each other.

Openness; Openness of the distributed system mainly depends on the extent to which new resource sharing services can be added and provided to various client programs.

Only by publishing the specifications and documents of the main software interfaces can the opening be realized. Another advantage of open systems is that they are independent of various vendors.

Security; The information resources available and maintained in the distributed system are of great value to its users.

Following two security challenges :

Denial of service attacks: Users may wish to interrupt service for any reason. This can be achieved by bombarding the service with a large number of unnecessary requests so that serious users cannot use it.

Mobile code security: Consider someone receiving an executable program as an email attachment. It seems to show an interesting image, but in fact it can access local resources.

Transparency; The transparency is the concealment of the separation of components in a distributed system by users and application programmers, so the system is regarded as a whole, not a collection of independent components. To illustrate the transparency of access, consider a graphical user interface with folders. The interface is the same regardless of whether the files in the folder are local or remote.

Scalability; From small intranets to the Internet, distributed systems can operate effectively on many different scales. If the system is still effective with a significant increase in the number of resources and users, the system is described as scalable. The number of computers and servers on the Internet has greatly increased.

Concurrency; In a distributed system, multiple clients can try to access shared resources at the same time. The process of managing shared resources can only accept one client request at a time. But this method limits the process. As a result, services and applications often allow multiple client requests to be processed simultaneously.

Quality of service; The main non-functional attributes of the system that affect the quality of service of customers and users are reliability, security and performance. The performance aspects of quality of service were initially defined in terms of responsiveness and throughput, but have been redefined in terms of the ability to meet punctuality guarantees.

Question4: **The design of distributed systems can be described and discussed in three ways i.e Physical Model, Architectural Model and Fundamental Model. Describe the example of distributed system in Question1 with respect to these three models.**

Answer;

Physical Model

The physical model is a representation of the underlying hardware elements of the distributed system, and it is different from the specific details of the computer and network technology used. The physical model is the most definite way to describe the system. They capture the hardware components of the system from the perspective of computers (and other devices, such as mobile phones) and their interconnected networks.

Baseline physical model; A Distributed System is a system in which hardware or software components located on a networked computer communicate and coordinate their actions only by transmitting messages. This leads to the smallest physical model of a distributed system, which is a set of scalable computer nodes interconnected by a computer network for the required delivery of messages. In addition to the basic model, there are three generations of distributed systems.

Architectural Model

The architectural model describes the system in terms of calculation and communication tasks performed by the computing elements of the system. The computing element is a single computer or a collection of it supported by an appropriate network interconnection.

System structure based on individually specified components and their interrelationships. The purpose is to ensure that the structure meets current and possible future requirements. The main consideration is to make the system reliable, manageable, adaptable and profitable.

Architectural elements

Communicating entities: From a system perspective, the entities communicating in a distributed system are usually the processes associated with the appropriate interprocess communication paradigm.

Web services

Web services represent the third important example of distributed system development.

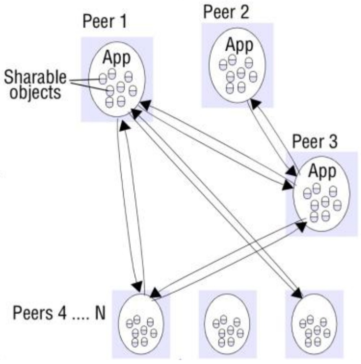
Web services are essentially integrated into the World Wide Web, using Web standards to represent and discover services.

Web services support the use of XML-based message exchange through the Internet protocol to interact directly with other software agents.

Roles and responsibilities

In a distributed system, processes (or objects, components, or services) interact with each other to perform useful activities, such as chat sessions.

- ❖ **Client-server;** Historically, it is the most important and still the most used. Client processes interact with various server processes in potentially independent hosts to access the shared resources they manage. For example, a web server is usually a client of a local file server that manages files storing web pages.
- ❖ **Peer-to-peer;** All the processes involved in the task play similar roles and collaborate with peers, and there is no difference between the client and server processes or the computer running them**.** Specifically, all participating processes execute the same program and provide the same set of interfaces. In a peer-to-peer system, the network and computing resources owned by the service user can also be used to support the service.



Peer-to-peer architecture

Architectural patterns; The architectural model builds on the most original architectural elements discussed above and provides a cyclic composite structure that has proven effective in a given situation. These are not necessarily complete solutions, but partial overviews. When used in conjunction with other models, these overviews can lead designers to provide solutions for a given problem area.

Layering; A complex system is divided into several layers, and a given layer uses the services provided by the next layer. Distributed services can be provided by one or more server processes, interacting with each other and interacting with client processes to maintain a system-wide consistent view of service resources. Due to the complexity of distributed systems, it is often useful to organize these services in layers.

Fundamental Model

The basic model should contain only the basic elements that must be considered in order to understand and reason about certain aspects of system behavior. The basic model uses abstract views to examine all aspects of the distributed system. Three important aspects of distributed systems: interaction model, failure model and security model.

- **Interaction model;** A distributed system consists of many processes, interacting in a complex way. For example: multiple server processes can cooperate with each other to provide services, such as the domain name system. There are two important factors that affect the interaction process in a distributed system:
    1. Communication performance is usually a limiting feature.
    2. It is impossible to maintain a single concept of global time.

In a distributed system, two processes can have different time values. In fact, computer clocks originate from perfect weather, and more importantly, their drift rates are different from each other.

- **Failure model;** In a distributed system, processes and communication channels may fail. The failure model defines how failures occur in order to provide an understanding of the impact of failures.
    1. Omission failures: When the process or communication channel cannot perform the operation that should be performed.
    2. Arbitrary failures: You cannot check for any process failure by looking at whether the process responds to the call, because it may respond arbitrarily. The communication channel may suffer any failure; for example, the content of the message may be destroyed, the non-existent message may be delivered, or the actual message may be delivered more than once. For example, checksums are used to detect damaged messages, while message sequence numbers can be used to detect non-existent and duplicate messages.
    3. Timing failures: Timing failures are applicable to synchronous distributed systems that set time limits on process execution time, message delivery time, and clock drift rate. Real-time operating systems are designed to provide synchronization guarantees, but their design is more complex and may require redundant hardware.
- **Security model;** The security of the distributed system can be achieved by protecting the processes and channels used for interaction and protecting the objects they encapsulate from unauthorized access. For example, some objects may contain private data from users, such as their mailboxes, while other objects may contain shared data.
    1. Protecting objects: The server is responsible for verifying the identity of the principal behind each call, and is responsible for verifying that the server has sufficient access rights to perform the requested operation on the specific objects of the call, and reject those that are not.
    2. Threats to processes: Processes designed to handle incoming requests may receive messages from any other process in the distributed system, and may not necessarily determine the identity of the sender. Without reliable sender identity information, the server will not be able to tell whether to perform the operation or reject the operation.

Question5: For **the purpose of Inter Process Communication (IPC) in distributed systems, in what situation you will use UDP and TCP and why?**

Inter Process Communication (IPC)

Inter process communication (IPC) is used to exchange data between multiple threads in one or more processes or programs. The process can run on one or more computers connected through the network. The complete form of IPC is inter process communication. Inter-process communication (IPC) is a set of programming interfaces that allows programmers to coordinate activities

Sockets

UDP and TCP use socket abstraction. The user datagram protocol (UDP) does not retransmit lost packets. The socket comes from BSD UNIX, but it also exists in most other versions of UNIX, including Linux as well as Windows and Macintosh OS. Processes can use the same socket to send and receive messages. Each computer has a large number (216) of possible port numbers for the local message receiving process.

UDP

UDP is a datagram-oriented protocol. Indeed, open the connection, keep the connection and end the connection without overloading. UDP has a fixed-length header of 8 bytes. UDP is very light. UDP supports broadcasting.

Datagrams sent by UDP can be transmitted without confirmation or retry. If it fails, the message may not arrive. To send or receive messages, the process must first create a socket bound to the local host's Internet address and local port.

Message size: The receiving process must specify a byte array of a specific size in which to receive messages. If the message is too large for the form, it will be truncated on arrival. The maximum packet length allowed by the basic IP protocol is 216 bytes, including headers and messages.

Blocking: Sockets usually provide non-blocking transmission and blocking reception for datagram communication (in some implementations, non-blocking reception is an option). You can collect messages in the queue by suspending or calling receive on this socket in the future. If no process has bound the socket to the target port, the message will be deleted at the target.

Timeouts: A server that is waiting to accept its client's request forever can use a permanently blocked reception. However, in some programs, it is not appropriate to wait indefinitely for the process of calling the receiving operation in the case where the sending process may crash or the expected message may be lost. In order to meet these requirements, a failure time can be set on the socket.

Receive from any: The receive method does not specify the source of the message. Instead, the call to receive will get a message pointing to its socket from any source. The datagram socket can be connected to a remote port and a specific Internet address, in which case the socket can only send and receive messages from that address.

Use of UDP ; For some applications, it is acceptable to use services that may cause occasional omissions. For example, DNS and IP are implemented through UDP. For example, a domain name system for searching DNS names on the Internet is implemented on UDP.UDP datagrams are sometimes an attractive choice because they do not suffer from the overheads associated with guaranteed message delivery. UDP datagrams are sometimes an attractive option because they are not affected by the overhead costs associated with guaranteed messaging.

There are three main sources of overhead:

1. Status information needs to be stored in the source and destination.
2. Transmit other messages.
3. Latency for the sender.

TCP

TCP is relatively slower than UDP. It is possible to retransmit lost packets in TCP, but it is not possible in UDP. TCP has variable-length header (20-80) bytes. TCP is heavy. TCP does not support broadcasting. HTTP, HTTP, FTP, SMTP and Telnet use TCP.

Message sizes: The application can choose how much data to write or read in the stream. It can handle very small data sets. The basic implementation of TCP streaming determines how much data to collect before transmitting the data as one or more IP packets. Upon arrival, the data will be transferred to the application as required. If needed, the application can force the data to be sent immediately.

Lost messages: The TCP protocol uses a confirmation scheme. As an example of a simple solution (not used in TCP), the sending end keeps a record of each IP packet sent, while the receiving end confirms receipt of all arriving messages. If the sender does not receive confirmation within the delay, it will resend the message.

Flow control: The TCP protocol attempts to match the speed of the read and write processes in the stream. If the writer is too fast for the reader, it will be blocked until the reader consumes enough data.

Use of TCP; TCP is a connection-oriented protocol. Many frequently used services run through TCP connections with reserved port numbers. These include the following;

HTTP: HTTP stands for Hypertext Transfer Protocol. These are all rules for transferring hypertext files or web pages between the network and computers. Hypertext Transfer Protocol (HTTP) is a protocol implemented using TCP, used to control World Wide Web (WWW) communication. Since 1990, it has been the foundation of Internet (i.e. Internet) data communication.

FTP: FTP stands for File Transfer Protocol. File transfer protocol (FTP) is a common protocol for exchanging files through the Internet. FTP is an open protocol standard widely used to transmit and receive large files. FTP uses ports for communication and also uses encryption to protect the information received and sent.

SMTP: Stands for "Simple Mail Transfer Protocol". This is a protocol for sending e-mail via the Internet. Your mail client (such as Outlook for Mac OS X Mail, Eudora) uses SMTP to send mail to the mail server, and the mail server uses SMTP to relay mail to the appropriate receiving mail server. A protocol for sending email between servers. Most e-mail systems use SMTP to send mail from one server to another through the Internet, and then e-mail clients can use POP (Internet Protocol) to retrieve mail. Post Office) or IMAP (Internet Mail Access Protocol).