**Research/Study logs**

1. **Blockchain**

Blockchain is the shared and decentralized ledger behind digital currency, a structure and technology that makes the information to be stored and processed in blocks securely based on cryptography. It’s like a database that is widely accepted through a big community rather than a trust or central authority like a bank or a government, they controlled currency. Each “block” has its unique number for transaction records that are hashed and encoded, and “chain” links any two of the blocks with a hash function together. After the creation of blocks, the blocks can be confirmed by a distributed network of devices. They get links to existing entry, form a chain, so a chain of blocks is created. Often time, the blocks are created simultaneously, creating a temporary fork. Blockchain are secure by design, it uses peer to peer protocols. Even though it was developed for Bitcoin by Satoshi Nakamoto in 2008, the use of blockchain has spread out to most of the cryptocurrencies such as Ethereum, Litecoin, Neo etc...

1. **Decentralization**

As I have mentioned in the blockchain section, a main function or characteristic of blockchain is decentralization. It’s one of the terms that are popular in the cryptoecnomic space. It usually means that no need of an independent third party that is a central authority to govern. But when we are talking about it, there actually have three different types of decentralization that people don’t actually know. There are architectural decentralization (numbers of physical computers a system is made of), political decentralization (numbers of individuals or organizations control the system), logical decentralization (numbers of interface and data structures that the system presents). I saw some good examples about which belongs to what. For example, BitTorrent is logically decentralized, because the downloading and uploading resources is between peer-to-peer, but the software is controlled and managed by one company. Blockchain actually has all traits of those three types of decentralization. Because blockchain has no central point of failure, no one is controlling, and is peer-to-peer.

1. **Distributed Computing**

Distributed computing uses many devices to do the calculation together. Some fields of studies like aeronautics, astronomy and math need to very complicated computations and it can take days to finish the calculation using a single computer. But distributed computing can help with the situation by shorten the time and decrease the workloads to each machine. Distributed computing can divide the job into many pieces and reform them together in the end. A master node in distributed computing takes the charge to perform the dividing task and assign them to worker nodes. Worker nodes will work on the job that is given to them and send the finished job to another node which waits for all the works, combines them to a result and output it.

1. **Cloud Computing**

Cloud computing, an implementation, is more like a term compare to distributed computing. Cloud computing is a service that charges money depends on the users’ need. Mostly are the resources like using time, storage, network bandwidth, memory, services and CPU numbers. It’s very convenient for users to use, they only need to pay a small amount of money but is able to use a powerful hardware cloud-based computing. Last quarter I took this web framework class at DePaul, our professor said that you don’t need a fancy computer nowadays, you actually only need a keyboard, a mouse, a monitor and some device to connect to the internet, you can then perform complicated computation or build services just use the cloud computing. He introduced us to use services like Amazon AWS, Digital Ocean and pythonanywhere to build our own website. All those services are awesome, and I only need to do tiny things to keep my website up and running.

1. **Cloud Storage**

Cloud Storage is the location for store the data generated by the cloud computing. Cloud storage services are often owned and managed by hosting companies. Those companies normally have big warehouses to keep multiple physical hard drives and servers. The cloud system provides uniform interface, stable connection and reliable data transfer services to customers and organizations. Users only need to pay little for the storage space as they actually use but are able to access or upload resources to the storage anytime and anywhere with internet connection. Some services that are designed for everyday use and well-known of are Apple Cloud, Google Drive, and BaiDuYun. They are free or fairly cheap but have strong impacts and help me a lot nowadays in real life.

1. **LAN**

LAN abbreviated as **Local Area Network**, a computer network, it can operate in fairly small place such as in school, home or company. Many devices can connect to this, like multiple computers, mobile phones, copy and print machine and other devices. Each of the device mentioned above has its own IP address in the network, it can send and receive data from other devices in the same LAN. Like the company I have worked before, we all have access to a same hard drive. The connection is strong and reliable, with very low cost. Different LANs can communicate with other LANs.

1. **WAN**

WAN is abbreviated as **Wide Area Network.** Unlike LAN that mentioned above, WAN covers a lot more, bigger space like cities, states or countries. Like LAN, WAN is also a computer network and has the very same components. WANs actually can be formed with LANs. But WANs are more likely connect to switchers, and because of its wide range, it costs more to send and receive data from one device to other devices. Also, it is harder to maintain a WAN.

1. **WLAN**

WLAN is abbreviated as **Wireless Local Area Network**. Like LAN, WLAN is a local computer network, and it allows computers or wireless devices connect to it use wireless signals. The biggest advantages of using WLAN is that mobile devices are free to be carried away within certain range and still has connection to the internet. WLAN is widely used these days because the easiness of installment and use. Like in most of the homes, you only need a router connect to the cables, and it can provide WIFI signals for the whole family members to find, connect and use the internet. Each device will be assigned to a unique IP address, but it that device shouts off the WIFI connection, the router recycles the address and save it for future new connection. Same device may have different IP address each time it connects to the WLAN.

1. **VPN**

VPN is abbreviated as **Virtual Private Network**, an encrypted network built upon public network. VPN is very popular in mainland China as the government set a wall through the internet, but many people have their need to use websites and services like Gmail, Facebook, YouTube etc., they need the VPN to escape the wall. The VPN allows remote access by encrypting and transforming data, so that Chinese people who has the access to the internet, can use VPN to break the wall.

1. **UDP**

UDP is abbreviated as **User Datagram Protocol** which is a connectionless protocol and it has an eight bytes header. This UDP is a lightweight protocol so it is fast, and because it does not care about the recovery of error and it does not have a fixed network infrastructure. The packets transfers use this protocol always find their paths to reach the finish line all by themselves. It has some disadvantages: it does not guarantee the packets will hundred percent reach to the end because they are likely to be attacked and get lost; it does not use handshake procedure, so it has no idea where the packets are going to or receiving from.

1. **TCP**

TCP is abbreviated as **Transmission Control Protocol** which is very different from the UDP and it has a twenty bytes header. It is a heavyweight protocol, not as fast as UDP, but it cares about the detection and recovery of errors and has a fixed network infrastructure. TCP requires at least three packets to make the connection between client and server work. It is connection oriented, so all the packets are sent out in order, follow a unique path to the target. By using handshake protocol, this makes it easy to monitor where the packets from or going.

1. **DNS**

**DNS** is abbreviated for **Domain Name System**. It is used as a Domain Name and IP address to map a distributed database on the world wide web. DNS enables users to have more convenient access to the Internet, and do not need to remember the complicated IP address that can only read by the machine. Each IP address has a host name which is formed by strings and decimal points. Users just need to remember the intuitive sense of the host name instead of each IP device’s IP address. By using the domain name, the process of getting the corresponding device’s IP address is called the host name resolution. Domain Name System uses the User Datagram Protocol and it has the port number 53.

1. **Network Encryption**

**Network Encryption,** also can be called as the network layer or level of encryption of encryption. It is a process of network security, applied in network transport layer encryption services. The network layer usually locates at the OSI third layer and forth layer, it mainly focuses on the connection between the two endpoints and the routing. By using the existing network services and software applications, network encryption is transparent for users, and independent of the other encryption processes. Data encryption can be only used in the transmission, but not encrypted in the sender and receiver end. Network encryption is encrypted with Internet protocol (IPsec), it combines a set of Internet engineering task force (IETF) to generate a unique communication across the IP network architecture. Nowadays, many companies such as Cisco, Motorola and Oracle offers Network Encryption services and products.

1. **Multithreading**

**Multithreading** is a technology term refers to the software or hardware implements multiple threads concurrently. Normally, computers have the capability of multithreading with hardware support more than one thread can be executed at the same time, so it can improve the performance. With the system includes chip-level multithreading, symmetric multiprocessor, simultaneous multithreading processor and multi-core processors. In a program, every single program fragment was called "threads", use these to program at the same time is called "multithreading". Multithreading is used for complete various jobs synchronized, not for improve the operation efficient, but for improve the efficiency of resource use to improve the efficiency of the system. Threads are required to complete many different jobs at the same time. A simple example for multithreading is like a carriage of train, and the process is the train. If carriage leaves the train, it is impossible for it to run. In the same way, the train could not only have one carriage to be more efficient.

1. **Virtualization**

**Virtualization** is like a resource management technology, is to abstract, convert and breakdown the barriers between various entities, such as servers, memory, network and storage. It shall be cutting between the entity structure, so the users can do better than the original configuration to apply these resources. Normally, the virtual interface environment uses the same resources as the original one. The term often means virtualization of resources including data storage and computational power. Virtualization technology is a set of solutions. The complete situation requires the support of the motherboard chipset, CPU, BIOS, and software, such as the VMM software or some other operating systems themselves. Even if the CPU only supports virtualization technology, it will have better performance than the system that doesn't support virtualization technology with the VMM software.

1. **Virtual Machine**

**Virtual Machine** is a term means a complete computer system running in a fully isolated environment with full hardware system capabilities that is simulated by software. The virtual system often time generates a new virtual image of the actual windows system. It has the same functions as any OS that it is designed for. After entering the virtual system, operations are performed in the new unique virtual system. The software’s data can be independently installed and saved. With its own independent desktop, it does not affect anything on the real system, and it can switch between existing systems and virtual images easily. I have tried some well-known virtual machine software includes Virtual Box, VMware, and Virtual PC myself, these software can virtualize multiple computers on different kinds of systems. I have used Windows on my MacBook, and Ubuntu on my desktop, and I think the VMM software can have a bigger impact in the industry world.

1. **Socket**

**Socket** is one end of the two network programs exchange data in a two-way communication connection. To establish this kind of connection, you mast at least have one pair of the sockets. Socket is essentially an API. Socket is very similar to a telephone socket. Taking a national telephone network as an example, the two sides of the telephone call are equivalent to two processes that communicate with each other, the area code is its network address, the exchange of one unit in the area is equivalent to one host, and the number assigned to each user by the host is the internal number corresponds to the Socket number. Before any user calls, he must first use a telephone, which is like to applying for a Socket; at the same time, he must know the number of the other user, which means that the other use has a fixed Socket. Then dialing the call to the other party is equivalent to sending a connection request (if the other party is not in the same area, it must dial the other area code, which is equivalent to giving the network address). If the other party is present and idle (equivalent to another host of the communication being powered on and can accept the connection request), pick up the telephone and the two people can formally talk, which is equivalent to a successful connection. The process of a call between two people is a process in which a phone sends a signal to the phone and the other receives a signal from the phone, which is equivalent to sending data to the socket and receiving data from the socket. After the call is over, suspending the phone by one user is equivalent to closing the socket and revoking the connection.

1. **Dead Lock**

A **deadlock** means two or more processes are in the process of execution because of competing resources or due to communication with each other. If there is no external force, they will not be able to proceed. At this time, the system is deadlocked, or the system has been deadlocked. These processes that are waiting for each other are called deadlock processes. In a computer system, if the system's resource allocation strategy is not developed appropriately, it is more common that a program written by a programmer has errors, which can result in the process being deadlocked due to improper competition resources. After a deadlock has occurred in the system, the occurrence of a deadlock should be detected in a timely manner and appropriate measures should be taken to lift the deadlock to avoid bigger trouble.

1. **Streams**

**Streams** is an operating environment that allows the use of multiple communication protocols on a single network. It applies to UNIX and Novell NetWare environments, but it also applies to other environments. Applications running in the streams environment can easily use any of the communication protocols that supports. Streams is a modular system in which the protocol stack can be added or removed as required. It provides developers with a set of tools for implementing communication protocols in the form of modules. At a lower level, interoperability provides multi-protocol support, so users can access many different types of systems. For example, if the TCP/IP and SPX/IPX stacks are installed on a single computer, users who uses this computer can access a NetWare server and a UNIX server.

1. **Serialization**

**Serialization** is the process of converting the state of data or an object, so it can be transmitted or stored. During the serialization, objects will write their current state to either a persistent or temporary storage. And you can recreate the object by de-serializing the state of object that are preserved in the storage. Serialization will help other code to see or change the object instance data that cannot be seen or modified without serialization. For example, code that execute serialization need special permissions like the security permission that specifies the serialization formatter flag. Under the default policy, if you copy and paste code from the Internet it does not grant this privilege; but the code on the local machine can granted the privilege.

1. Introduction
   1. What is distributed system?

A group of machines presents to the user a unified system. The system uses common resources, and can be dynamically assigned tasks, decentralized those resources through computer networks to achieve the exchange of information.

* 1. Design goals
     1. The most desirable aspect of a distributed system is that it allows users to access resources remotely, and they can share these resources with other users. Like files, data, and computer resources are all good examples. Because resource sharing is much more cost-effective than asking everyone to buy and maintain.
     2. No matter what architecture of the distributed system is and how complicated the system is, it is transparent to the user at the other end. Transparency has different forms like: concurrency, migration, access, location, relocation, failure and replication.
     3. The openness of a distributed system is essential because people want to use that to connect to more outside systems. The systems are set to follow standard rules, so they are able to speak the same languages to each other and Interface Definition Language aka IDL is the term that defines this. Also, the capability to take out or receiving fraction parts is another important feature.
     4. To be large enough for the whole world’s users to use is another goal that the systems creators would consider. Size, geographical and administrative scalability are 3 main question that developers need to worry about.
     5. To design a successful distributed system do need a lot of time and design, there may be many barriers to stop the creators. They actually need to think more realistically and do not make any naïve guess.
  2. Distributed systems Types
     1. Distributed systems can be divided into two subcategories. In cluster computing, the underlying hardware is composed of workstations that run the same operating system. But grid computing is very different, it has different operating systems, hardware, software and deployment network technology.
     2. Distributed systems used a lot of network applications in the organization’s use. In order to solve the problem of poor reading and writing among these applications, middleware solutions are used.

According to the degree of integration, there are two different systems: Combine all requests into one larger request and solve it with **distributed transaction**. It is characterized by either running all requests or not running at all. The other is splitting the request into separate database components and transaction processing components: **enterprise application integration** which is used by large enterprises.

* + 1. Because of the increase in mobile and embedded devices, distributed systems are not always as stable as mentioned before. An important feature of **pervasive systems** is that no one really manages to control it. Nowadays, smart home systems like the Alexa, Google smart home, electronic health care systems like the Apple watch and fitbit and sensor networks are good examples for pervasive systems.
  1. Distributed systems are made up of many computers working together, but it feels like a coherent system. They can integrate many different programs from many different systems into a single system. There are many different types of distributed systems. Because of the computing methods, information processing and pervasive are divided into different types.

1. Architectures

Distributed systems are mainly composed of software architecture. The chapter 2 focus on these different kinds of architecture, where the final instance is called system architecture.

* 1. The architecture style is defined based on how the **components** (provide replaceable interfaces) are connected, how data is exchanged and how they are grouped together. Passes the information in the components and help the cooperation mechanism is the connector. Because of the configuration and application of different components and connectors, many different architecture styles can be separated.
     1. **Layered architectures** are just components make up different layers.
     2. Each object corresponds to a component forms **object-based architecture**.
  2. One important purpose of **middleware** is to provide transparency. Many middleware use object-based architecture. Putting the right middleware into the system can make the design of the application easier, but it may not be the most preferred in the developer's mind. So, people have developed a variety of different middleware to meet different needs.
     1. Wrappers or adaptors work just like every day used electric adaptor, to help different interface can be plugged in together.
     2. Interceptors act as a software structure that can interrupt control flow to allow other code to run. In different environments, it can be divided into request-level and message-level interceptor.
     3. Because the operating environment of distributed programs may be very different, the middleware designer decided to develop modifiable middleware to handle the difference between similar power consumption and hardware.
  3. There are many different types of system architectures, such as the most common centralized and decentralized architectures and two hybrid structures: client-server, peer to peer, edge-server, and so on.
  4. Some examples for this chapter.
  5. Distributed systems can be organized in many ways. According to the composition of different components, different software architectures and system architectures can be distinguished.

1. **Processes**

This chapter discusses in detail how various processes play their most important role in distributed systems. The concept of process is originally from the operating system, it is used to define the program in execution. In the operating system environment, process management and scheduling are important, but for distributed systems, it has other important implications. The migration of processes between different machines is a particularly important issue for distributed systems and will be discussed in this chapter.

* 1. **Threads**

Although the process is the most basic component of a distributed system, but the source it comes from, operating system, is a very different architecture, it makes more sense to divide each process into several threads. Using threads can make distributed applications become easier to build and achieve better performance.

* + 1. To execute the program, the operating system creates a number of virtual processors, each of which runs a program. In order to ensure correspondence, the operating system has a process table to store the corresponding information. A process is a running program, typically a program running on a virtual processor.
    2. Like a process, a thread executes its program code independently. But the difference between them is that threads do not sacrifice performance in order to achieve high levels of concurrent transparency. So threading systems generally only maintain the minimum amount of information needed to allow multiple threads to share the CPU. In many cases, multithreading can improve performance, but the development of multithreaded applications requires more effort.
    3. Threads are generally implemented through thread packages, which have operations for creating and destroying threads, as well as operations for synchronizing variables such as mutex variables and condition variables. There are generally two ways to implement it: construct a thread library that executes entirely in user mode; the kernel manages and schedules threads.
    4. In a distributed system, threads can be conveniently expressed as a form of maintaining multiple logical links simultaneously. When using a multi-threaded client, establishing a connection with different server replicas allows for simultaneous data transfer, and is much shorter in time. In this case, using threads is the best way.
  1. **Virtualization**

Only a single processor but feels more than one processor mechanism can be extended to other resources, leading to the so-called resource virtualization.

* + 1. Virtualization can help by migrating the underlying interface of the old software to the new platform, or let each application run on its own virtual machine, so that the types of platforms and machines can be relatively reduced.
    2. There are many ways to implement virtualization in practice, such as building a runtime system and providing a set of abstract instruction sets to execute programs or providing a system that makes it a completely shielded hardware but provides the same Instruction Set.
  1. **Clients**
     1. The main task of the client's machine is to allow individual users and remote servers to have relevant communication and interaction. There are two ways to support this interaction: one is that for each remote service, the client has a separate part to contact the service through the network; the other is to provide a user-friendly interface for direct access to the remote service. Thin-client computing is another example.
     2. Client-side software does not just include the client interface. Most of the time, some of the processing in the client-server application is performed on the client side. In addition to the client interface and other application-related software, the client-side software also contains components for obtaining distributed transparency.
  2. **Servers**
     1. The server is provided for the client to implement a specific service process. Each server first waits for a request from the client, then processes that request and finally waits for the next request.
        1. **Iterative sever** processes the request and return the result to the requesting client by itself.
        2. **Concurrent server** does not process the request itself, it sends the request to a separate thread or process, and it immediately returns and waits for the next request.
        3. The main different between **stateless** and **stateful server** is whether it save the client’s information or not. Stateless server does not, but stateful server does.
     2. Object server for objects to live. Object is formed by its state data and the executing code for its method.
     3. Apache Web server host 50% of all web sites and is a good server example that balances the mechanisms and policies.
     4. To outsiders, server clusters should behave like a computer. However, if it involves how to manage this pile of things, it is not the same as a single machine. The most primitive approach is to extend the method of how to manage a single computer to the server cluster, allowing administrators to remotely log in to the cluster and manage it. However, if there are many nodes, this method will not work. In fact, if the cluster's size is very large, it must be managed through experience and different methods.
  3. **Code migration**
     1. The code migration in a distributed system is done through the form of process migration: all the processes that are still running are completely moved to another machine.
     2. In the code migration, we need to change the reference to the resource without affecting the resource binding of the process. The specific way to change a reference depends on whether resources and code can be migrated together to the target.
  4. In a distributed system, the process is a basic part. They form the basis for communication between different machines.

1. Communication

The core of a distributed system is the communication between processes, so sharing the mechanism of information exchange between machines in a distributed system becomes a top priority. Communication is basically passed through the low-level messages provided by the underlying network. This chapter focuses on everything about communication.

* 1. **Foundations**
     1. Because there is no shared memory, all communications in a distributed system are based on low-level information exchange. People have made a lot of agreement to ensure the smooth and consistent communication between the two parties. There are many **layered protocols** for developers to use.
     2. Transient, persistent, synchronous and asynchronous communications are some different **types of communication**.
  2. **Remote procedure call**