

- I. *Study PACKT Dist Sys in Java book pages 15 to 19. Based on this, explain how Parallel Computing and Distributed Computing are different? Is the Youtube platform a parallel system or distributed system? Explain how each Design consideration is relevant to the Youtube platform. [15 points]*

Based on the readings, Parallel Computing and Distributed Computing differs in the following ways:

The main difference is that Parallel Computing has multiple processors that communicate with each other using a shared memory space whereas multiple processors in Distributed Computing communicate through a communication network with no shared memory space. As the name suggests, Parallel Computing, multiple processors performs tasks simultaneously, and in Distributed Computing, a single task could be completed by multiple autonomous computers in different geographical locations. Processes running in Distributed Computing requires less dependency than Parallel Computing.

Youtube platform is a large system with a variety of functionalities, such as video streaming, uploading videos, sharing videos, search, leave comments, subscribe and etc. It's hard to say that it's only utilizing only one computing system, I think it's a mix of both Parallel Computing and Distributed Computing. Consider the characteristics of both computing, for example, Youtube platform has millions of daily active users in different locations around the globe using their own devices. Youtube platform should be highly scalable, reliable, easier to share resources and data to meet users' expectation. In this case, Distributed Computing should be more suitable. For video transcoding, using parallel process instead of sequential makes more sense since the process is time-consuming and expensive.

- II. *Study Chapter 13 Java Socket Programming from this book Object-Oriented Programming with Java: Essentials and Applications Authors Buyya, Selvi and Chu [2009] Tata McGraw Hill [PDF posted in Lecture 1 Folder]*

13.22 Explain the TCP/IP stacks. Briefly explain the TCP and UDP protocols and the difference between the two protocols. [15 points]

TCP stands for Transmission Control Protocol, IP stands for Internet Protocol. They define the rules that computer communicate with each other through the internet. Example applications use TCP are HTTP, HTTPS. Every computer connected to the internet has a unique, 4-byte IP address, that maps to name like www.google.com. UDP stands for User Datagram Protocol.

The difference between TCP and UDP are: TCP is more reliable that it tracks data to make sure it's delivered whereas UDP has no guarantee on that. TCP can also detect errors and it will retransmit lost and duplicate data. UDP uses lower bandwidth than TCP and is connectioless.

13.23 What is port? List some well-known ports and explain the applications associated with them. [5 points]

Port is an access channel that associated with a specific process or service on computer. Well-known ports are: 80 for World Wide Web HTTP(www), 118 for SQL Service, 443 HTTPS etc. Insecure web browsers is associated with port 80, secure web application is associated with port 443, SQL service is associated with 118.

III. Study Dist Sys Design Goals.pdf. Consider the architecture of Twitter. What do the Goals and Transparencies described in this paper mean in the context of Twitter? Why are they important? Explain with a diagram.[15 points]

The goals described in this paper are Transparency, Scalability, Dependability, Performance and Flexibility. In the context of Twitter, scalability means be able to handle more users and more tweets without experiencing significant loss of performance and becoming more difficult to manage. It is important because as number of users (around the globe) are growing at Twitter at a rapid rate, we don't want to sacrifice their experience using Twitter and increase administrative complexity. Twitter has to be scalable to ensure that.

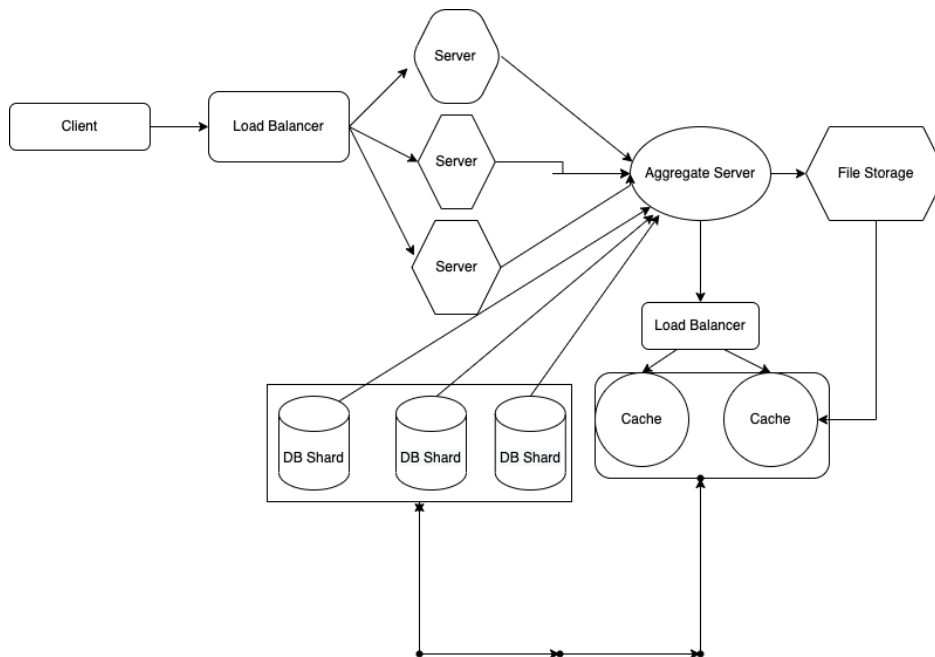
Dependability requires the system needs to be consistent, secure and fault tolerant. As Twitter has around 400 million users, it's especially important to have a dependable system to ensure users' experience. In that paper, to improve fault tolerant, a secondary database serve is added to address read-heavy property.

Performance is also important in design of Twitter, therefore in that paper, loading balancing, cache, data sharding using Tweet ID and tweet creation time is added to ensure a better performance.

Flexibility means to provide tailored services, which has properties such as openness, extensibility and interoperability. In that paper, it's included in the extended requirements where other functionalities are addressed, those can be built on top of the existing system. Transparency means separation of the components of a distributed system and application programmer/user. In the context of Twitter, I could clearly see location, access, replication and failure and concurrency transparency implied. Users/application programmers need not be aware of the details about data/resources locating in different geographical locations, replica's consistency

issues etc. they are all accessible in the same way. If individual component fails, user/application programmers should not be aware of that happens on the surface level. These has the advantage that when user/application programmer are developing, they can treat the data/resource as if it's not being distributed, and it's extendable meaning no need to make modification each time.

Diagram below:



IV. From Chapter 1 Coulouris Book

1.12 A server process maintains a shared information object such as the BLOB object of Exercise 1.7. Give arguments for and against allowing the client requests to be executed concurrently by the server. In the case that they are executed concurrently, give an example of possible 'interference' that can occur between the operations of different clients. Suggest how such interference may be prevented. Pg. 22 [15 points]

Arguments for allowing the client requests to be executed concurrently by the server: there will be more throughput in the server

Against allowing the client requests to be executed concurrently by the server: example to see there could be problems interference between concurrent operations: Client one's thread reads value of variable A, and client two's thread also reads the value of variable A. If client one is adding 10 and storing the result in A, whereas client two is subtracting 10 and storing the result

in A. The result can be problematic. In this example, we could prevent it by using synchronized operations in Java to overcome the interference.

1.13 A service is implemented by several servers. Explain why resources might be transferred between them. Would it be satisfactory for clients to multicast all requests to the group of servers as a way of achieving mobility transparency for clients? Pg. 23 [10 points]

It's because if one server will update a resource that other servers need, so the server does the update will send changes to other servers. If clients multicast all requests to the group of servers, the load will be heavy and servers will do unnecessary work to filter out request that they cannot fulfill.