CS 6650 – Building Scalable Distributed Systems – Homework 1

Name: Sheetal Turaga
Email: Turaga.s@northeastern.edu
Student ID: 001385799

Question-1: 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.

ANSWER:

Parallel Computing refers to performing multiple tasks at the same time with the idea to do more in less time. This is achieved by running multiple processors simultaneously one single machine to get the task done. The machines that support parallel computing have shared memory or distributed memory that are used for executing the task. Parallel computing offers reliability because the failure of one processor, in a multiprocessor machine, does not hinder the functionality of the other processors. Adversely, if the machine has a single processor, then this could add latency to the computing process while adding more processors to counter the latency effect, would be expensive. Parallel Computing today is used to run models and simulate real-world scenarios such as Internet of Things or Super Computers.

Distributed computing, on the other hand, refers to splitting the task between multiple machines or computers geographically distant and connected via a network. All the machines connected coordinate and communicate by sending messages through the network. Every computer in the network has its own memory that is used for the executing of the task. Distributed computing offers scalability and efficiency by keeping resources shareable. A good illustration of this type of computing – YouTube platform. The ability of this application/platform to accommodate the growing amount of content while maintaining high throughput when resources are added, provide high quality service for its users and be available to users across the world proves that it is a distributed system. Here are a few characteristics elaborated based on the Design considerations:

No Global Clock: Youtube's presence across the globe with millions of users is possible through the network it has developed for content delivery. Since Youtube is asynchronous and given the different time zones that it's available in, it does not have a global clock.

Geographical Distribution, Independence, and heterogeneity: Youtube uses a content distribution network to make content available for its users requesting data from its servers

spread across several geographical locations to avoid any network latency while being cost-effective.

Fail-Over mechanism: Reference: For user data, Youtube has master-slave replicas which are systematically updated, database sharding and have user data in data centers across several geographical zones, periodically backed up as a fail-safe in case of any disasters. For videos, they are stored on hard-drives in large google data centers considering the petabyte-scale of data that needs to be dealt with.

Security: While the content is available to anyone who has access to the website, on the user front, there is an option to login securely to one's account for content viewing, moderation, and personalization. On the backend, Youtube uses encryption to secure user data and content

---

Question-2: In your own words, explain what the Tannenbaum textbook's Layer cake cut diagram is about, providing 2 examples each for the 5 architectural models shown

What is port? List some well-known ports and explain the applications associated with them

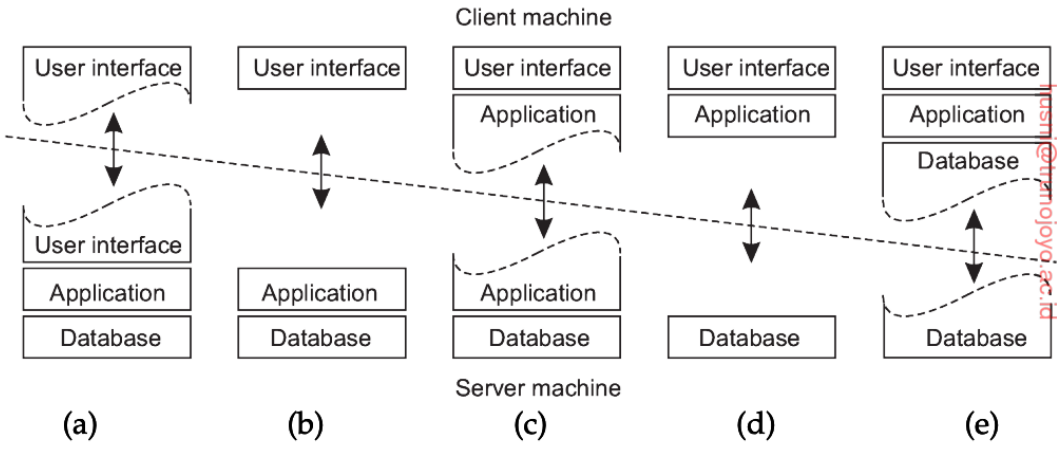


Figure 2.16: Client-server organizations in a two-tiered architecture.

ANSWER:

Part I: Tannenbaum's Layer cake cut diagram is a depiction of the number of ways a distributed system can be organized to serve in the best interest of a business. The idea behind separating the layers to run on two different physical pieces of hardware, namely the client and the server machines, is to improve the performance of the application and help with scalability.

The communication is directly between the client and server with the business logic scattered around the two tiers based on the way the architecture is organized. There are two schools of thought associated with this distribution of the business logic. They are known as thick client and

thin client. The Thin client refers to a thought that client software should be small and minimal while most of the processing must take place on the server. This type of client can be a moderately equipped machine with good network connection. On the other hand, the thick client, the client machine, or software does most of the heavy lifting while the server machine performs the simple tasks such as retrieving or storing data.

a) ATM machine, Parking Ticket Machine
b) Zoom, Teams, Maps
c) MS Word, Excel sheets, Desktop games
d) Banking or investment applications Robinhood, Betterment
e) AWS, Any Website – Amazon.com

Part II: Port

A Port is a communication access point with a unique identifier number associated with specific process or a type of network service. It is managed by a machine's operating system and are a part of the transport layer. They play key role in identifying and differentiating between the traffic that being sent through the network, for instance, an email being sent from one app on the sender's computer reaching the inbox of the same app on the receiver's side.

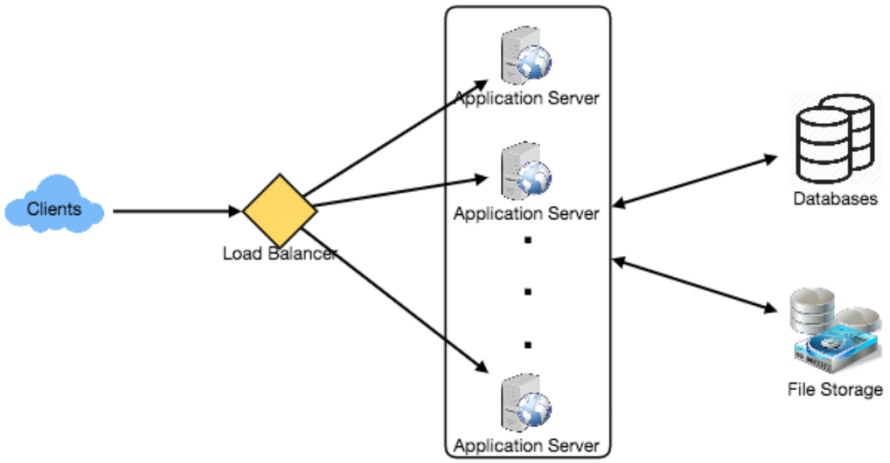
The port helps machines understand how to manage the data they receive to transmit. They are standardized and reserved for each transport control protocol or User datagram protocol which dictates the port of a packet being transmitted using TCP/UDP headers. Some of the well-known ports are:

- Port 53 for DNS
- Port 80 for HTTP
- Port 21 for FTP

Question-3: 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.

ANSWER:

Twitter is one of the biggest real-time event channels that are available today for users that are spread across the world. In the context of Twitter, the following properties ensure that the user of Twitter has a seamless experience and more pleasing.



A key goal of a distributed system is Transparency – which is to ensure that the user has no idea of how the underlying application programming. For example: For an individual who's writing a tweet, it is one seamless system that takes the input (tweet) and then displays it in the feed once he/she posts it. Whereas, in the background, the input is being added to a database with some metadata associated with so that we can fetch it when the user/writer attempts to retrieve it. The same concept of Transparency is extended to the other types of transparencies that are adhered by the programmer so that the application is portrayed as a single system rather multiple systems working to present the information requested by the user

Access Transparency: The user's experience of searching for specific any tweet or photo shouldn't be any different, whether that piece is accessed from cache or a remote server

Location Transparency: The physical location of where the data resources are stored is irrelevant to the user and therefore, should not be visible/accessible to the user

Migration Transparency: With the numerous write requests (tweets posted, for example) that flow into the application, the user does not need to know how the load balancers in the application is managing and distributing the inflow of traffic to ensure the application is running

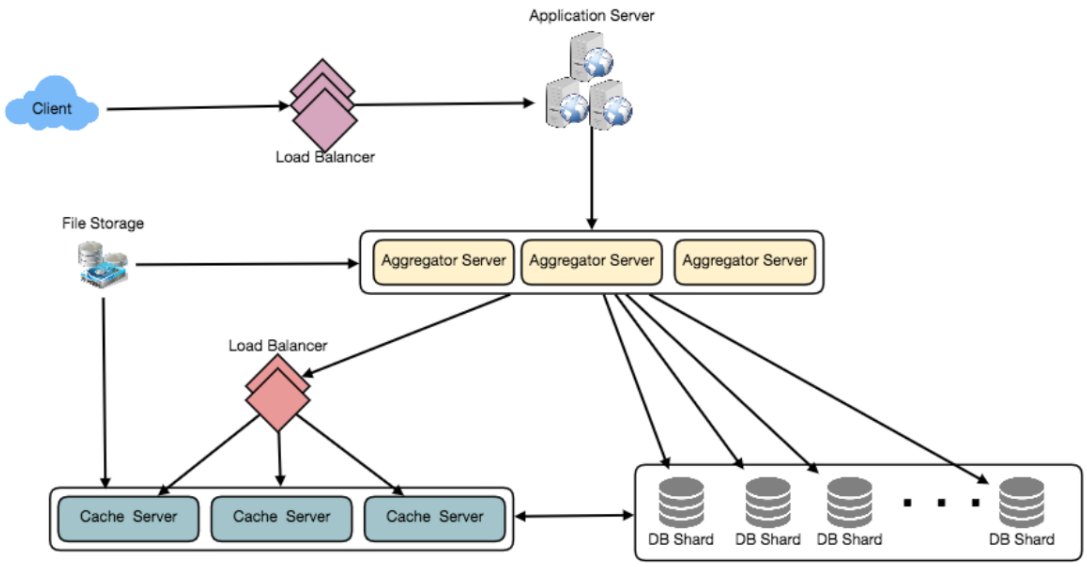
Replication Transparency: Being a social networking service, there is an intricate connection between users and therefore, has huge amounts of data that is basically active in real-time. Irrespective how many datasets the resources are spread across, the user should be able to access it as one single data set

Failure Transparency: Given the number of systems/services/components involved in building Twitter, this property allows to plan for ways to manage faults/failure without interrupting the application from completing the tasks while concealing it from the user

Concurrency Transparency: The ability of system to have 200 million active users being able to access different kinds of information on the application, speaks for the ability of the system to be simultaneously accessed by these active users

Scalability: In the context of Twitter, scalability amounts to the growing number of users which leads to increase the amount data that needs to be stored and accessed in real-time. The thousands of tweets being posted every minute of every day and read requests in real-time, scalability becomes critical to ensure a good user experience without compromising content delivery performance

Dependability:
Performance: In a distributed system, evaluating performance is a good pointer to how the system will function with respect to the non-functional requirements. Based on the twitter design, elements like load balancers placed at different stages in the system (as shown in the image below) can help moderate and distribute requests between client-app server or app-servers-databases, which in turn avoids overloading that could lead to latency in response. Storing data in multiple databases also improves performances because it reduces redundancy



Question-4: 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

ANSWER:

Given that a server process has a shared object, my argument in support of allowing concurrent executions by the server is that this means more throughput by the server. But in contrast, it can also create an interference between these concurrent operations. For instance, consider bank transactions being mapped to a customer's account that holds a certain amount $100. If we have two threads running concurrently, doing their own transaction as the user requested, then they will be recording the value of the amount in their respective threads. The likelihood of two threads having different values for these amount at any given time is quite high. A way to prevent such interferences is to make sure the concurrent executions are synchronized with the help of lock, mutex or semaphores and the data object is not corrupted.

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

ANSWER:

Servers might transfer resources to other servers in the network to keep them updated of any changes that made on one server and those changes might be required by the others. This also helps balance the processing load between the servers. Multicast of requests to all servers would overload the network. It would also end up working the servers that do not have the resources needed to fulfil the request made.

Question - 5:

The goal of this assignment is to implement a TCP client and server. You can use Java. Your TCP or UDP client/server will communicate over the network and exchange data.

Code folder zipped and attached