

Assignment 2

17000912

An example for a large distributed computing system will be a music distribution service (e.g. Spotify, Apple Music). The requirement for a distributed computing system for such a service is that it needs to provide its content to the entire globe; but the amount of information it needs to store, process, and aggregate can't be handled by a client service architecture. Since there is preference on the types of music in different regions, it's a logical choice to store and process music if the region is more interested in that type of music. A distributed computing system is the solution that fulfills such requirements to optimal resource usage and business advantages. To handle the large amount of information which will be used in a such service needs a horizontally scalable computing resources to scale with future user improvements. Also, to get the edge in the competition, such service should be able to provide continuous availability. If any of the users experience an unavailability of the service, it would be a huge impact on the business aspect when having many other competitors. Therefore, a distributed computing systems modularity and less coupling can maintain a minimum downtime and a smaller number of affected users in case of failure in one of the resources. A service like this has millions of users around the world, so it is important that the system is capable to handle billions of requests with minimum latency. Having a distributed system will offer the developers to replicate resources, load balance and scale geographically to support diverse user requirements.

But implementation of large distributed system will also add its own complications. One of the problems is keeping the consistency among the components of the distributed system. For example, adding or distributing new content have propagated through many computing resources and even between replications of same resource. This add more complexity during the both the implementation and the maintenance of the system. The reliance of the network adds inherent complications of networking to the system. There's always the risk of a network failure, data loss during transmission, latency of the network and many more issues. Also, a distributed system like this need have additional services and techniques to regularly monitor the status of the infrastructure in order to detect and resolve a failure at any point of the system. This added complexity on monitoring the service requires technical expertise, additional resources and reasonable amount of financial resource allocations. Another complication that comes with distributed systems is the security. Having the system as set of distributed computing resources that communicates with each other on a network adds set of new security concerns compared to a monolithic server. The communications between the resources has to authenticated, authorized and also encrypted if necessary. This additional properties and operations will affect in overheads and latency. Also, handling identities of users in a distributed system add more complications too. Identity information has to securely propagated and made consistent within the entire system to offer seamless user experience while minimizing the risk of unauthorized access. But in order to implement a large distributed system like this, proper infrastructure has to be implemented and maintained. This includes networking resources, distributed data centers, isolated backup, mirrors, and etc. Implementing and maintaining such secondary resources adds additional costs.