Writing Assignment-1 (Distributed Platforms)

What is a Distributed system?

A “Distributed” or “microservice” system is one in which the individual components of an app operate independently of each other and communicate their action by passing messages to one another from any system.

There are several benefits that distributed architecture provides over monolithic system.

- Data sharing
- Scalability
- Failure Handling
- Efficient

There are many platforms that are distributed in nature.

1. Distributed supply chain like Amazon

Amazon was one of the first big players to leverage microservices, that shifted from monolithic architectures to Amazon Web Services (AWS) in 2001.

The biggest online retailer in the world and a well-known cloud service provider, Amazon. The AWS CloudFront CDN hosts user instances, whereas Amazon S3 hosts static information. The automatic load balancer of Amazon directs incoming traffic to the Kubernetes cluster with Docker containers running microservices at Amazon ECS. Elasticache stores the information in a database (such as Aurora, RDS, or DynamaDB) as a cache.

Tech Stack: React, Java, MySQL, AngularJS, and Amazon S3 are some of the popular tools that Amazon uses.

2. Online banking system

A database management system used by banks to store financial data in a central location can be vulnerable to downtime. The central location may be inaccessible because of problems in communication infrastructure, natural disaster or malicious attack. By using a **distributed system**, it lets banks access the data or information they need at any time, regardless of the up-time status of a central system or a server. The main advantage of using a distributed database management system is that it allows banks to reroute their information requests around the inaccessible location to another site or server. This distributed system allows banks to access the data they require faster and more reliably than using a centralized database system.

When we implement a distributed database management system in online banking it allows each bank branch to have its own copy of the latest customer data. Having the latest data of each customer's account allows the bank to record and process each transaction locally, rather than sending it to a central server.

3. Spotify

Over 170 million people use Spotify, a large international music corporation.

Nginx serves as an elastic load balancer and API gateway, whereas Java is mostly used as a general development language. Users' billing and subscription information is stored in Postgres, while Kafka is used as an event-driven streaming pipeline.

4. Netflix

Users of the Netflix subscription-based streaming service can watch TV episodes and movies on a connected device. Most of Netflix's reliance is on Amazon Web Services (AWS), which is a crucial component of the Netflix backend ecosystem. It makes use of scalable databases DynamoDB and Cassandra, EC2 for computing instances, S3 for storage, and Spark for big data processing. Requests are handled via Netflix's internal content delivery network, Open-Connect.

Tech Stack: Python, Node.JS, Java, Kotlin, and Swift. Netflix owes its user-friendly interface to React and JS UI libraries.

5. Peer-to-peer file-sharing system like BitTorrent

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the network. A peer-to-peer network is designed around the notion of equal peer nodes simultaneously functioning as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client–server model where communication is usually to and from a central server. A typical example of a file transfer that uses the client-server model is the File Transfer Protocol (FTP) service in which the client and server programs are distinct: the clients initiate the transfer, and the servers satisfy these requests.

Submitted by:

Prashant Kumar

2022201058