System Design of Netflix

Sanket Saxena · Follow
6 min read · Jun 20, 2023

Netflix is a global entertainment giant with over 200 million subscribers. To comprehend the architecture of a system supporting such a large user base, we'll start with some back-of-the-envelope calculations.

Assume at peak times, about 25% of subscribers, approximately 50 million users, are actively streaming. If we consider each stream to consume about 3GB for a 2-hour movie, and this data is evenly distributed over the movie's duration, we calculate:

- Data streamed per second: (50 million streams * 3GB/stream) / (2 hours/stream * 3600 seconds/hour) ≈ 10.42 GB/s

Due to Netflix's worldwide presence, data must be replicated across various geographical locations to ensure redundancy and low latency. If we assume an average replication factor of 3, the overall data output requirement increases to about 31.25 GB/s.

These calculations underline the massive scale of Netflix's operations, providing context for the subsequent system design discussions.

Open Connect (OC) — Netflix's Content Delivery Network (CDN)

To efficiently handle this huge volume of data, Netflix uses its own Content Delivery Network (CDN) known as Open Connect (OC). The OC includes original servers and edge servers. Edge servers are strategically placed at Internet Service Providers (ISP) to enable swift data delivery.

Question: What advantages does Netflix gain by hosting edge servers at ISPs?
Answer: Hosting edge servers at ISPs allows Netflix to deliver content directly to users without traversing through the broader internet, minimizing latency and reducing costs. It also ensures better scalability during peak traffic periods.

Load Balancer (LB)

To ensure no single server gets overwhelmed, Netflix employs a two-tier load-balancing mechanism. The first tier performs round-robin routing across multiple zones, and the second tier distributes the load among services within a zone.

Question: Why is the two-tier load-balancing approach beneficial for a service like Netflix?
Answer: A two-tier load balancing system ensures a more evenly distributed load, leading to improved reliability and robustness. It helps in preventing any single server or zone from becoming a bottleneck, thereby enhancing overall performance.

Video Preprocessing — Transcoding

To accommodate a wide array of devices and network conditions, Netflix transcodes each video into around 1200 different versions. This process ensures that every user gets the best possible video quality their device and network can support.

Question: Why does Netflix need so many versions of each video?
Answer: Creating multiple versions of each video ensures that users can

stream content optimally across different devices and network conditions. It ensures the best possible viewing experience whether the user is on a high-speed broadband connection on a 4K TV or on a slower mobile network on a smartphone.

Zuul — Edge Gateway Service

Zuul, functioning as an edge gateway service, is instrumental in managing all inbound and outbound traffic of the Netflix platform. Utilizing the powerful asynchronous network library Netty, Zuul dynamically routes, monitors, and secures the traffic, offering critical insights and control over interactions between Netflix and its clients.

Question: What consequences would Netflix face without a system like Zuul in place?
Answer: Without Zuul or a similar gateway service, managing the myriad of requests, the security concerns, and the complex routing for a platform as large as Netflix would be significantly more challenging. A system like Zuul offers centralized control over these operations, enhancing the overall system resilience and maintainability.

Hystrix — Latency and Fault Tolerance Library

Hystrix serves as a shield against latency and fault in distributed systems. It prevents any failure from cascading across multiple systems and brings the failing services down swiftly and gracefully, ensuring uninterrupted service delivery.

Question: What strategies could be used to determine the thresholds for triggering a fail-fast in Hystrix?
Answer: The thresholds can be adaptive based on the system's current load and performance. They can also be statistically calculated based on the past request processing times — average and standard deviation could be a good starting point.

starting point.

EVCache — Memcached-based Cache Layer

EVCache, a Memcached-based caching solution, plays a key role in Netflix's high-speed data access strategy. It is implemented on SSDs and is lauded for its low-latency read and write operations. EVCache reduces the load on primary data storage by holding frequently requested data, thereby enhancing the performance of the Netflix service.

Question: Why does EVCache use SSDs? What are the benefits?
Answer: SSDs provide faster data access compared to traditional hard drives, making them an excellent choice for caching solutions. SSDs' low latency and high throughput enable EVCache to rapidly serve frequently requested data, thus significantly improving the performance.

Data Storage — MySQL and Cassandra

Netflix employs a combination of SQL and NoSQL databases, each serving a unique purpose. MySQL, with its strong consistency and reliability, serves as the source of truth for most business data. The data is replicated across multiple data centers (cross-DC) to ensure high availability and disaster recovery.

On the other hand, Netflix uses Apache Cassandra for use-cases demanding heavy read and write operations. Cassandra's distributed architecture and high write performance make it an excellent choice for maintaining users' viewing history.

Question: Why would Netflix use both SQL (MySQL) and NoSQL (Cassandra) databases in their system design?
Answer: SQL and NoSQL databases serve different needs. MySQL, an SQL database, provides strong consistency and is great for transactional data, making it ideal for business operations. Cassandra, a NoSQL database, excels in scenarios that require high write performance and scalability, perfect for

in scenarios that require high write performance and scalability, perfect for storing and processing high-volume data like user viewing history.

Kafka and Chukwa for Event Logging

Netflix generates about 500 billion events per day from various sources, including view activity, error logs, and performance events. Kafka, a highly scalable and fault-tolerant messaging system, serves as the first point of contact for these events.

Chukwa, a data collection system built on top of Hadoop Distributed File System (HDFS), transports the events to a centralized store like AWS S3 for long-term storage and analysis.

Question: Why does Netflix use both Kafka and Chukwa for event logging?
Answer: Kafka and Chukwa serve complementary roles in event logging. Kafka provides a real-time, fault-tolerant, and highly scalable messaging system ideal for ingesting a massive stream of events. Chukwa collects these events and reliably stores them in a centralized data store like S3 for subsequent batch processing and long-term storage.

Elasticsearch for Real-time Search and Analysis

Elasticsearch, with around 150 clusters and 3500 instances, provides real-time search and analysis capabilities to Netflix. It enables Netflix to derive actionable insights from a sea of data quickly.

Question: How does Elasticsearch contribute to improving the user experience on Netflix?
Answer: Elasticsearch helps Netflix analyze user behavior and preferences in real-time. These insights enable Netflix to provide personalized content recommendations, enhance search functionality, and proactively identify and resolve potential issues, thereby enhancing the overall user experience.

Spark for Recommendation Systems and More

Netflix makes extensive use of Apache Spark for a variety of tasks, but its most well-known use is for powering Netflix's recommendation engine. This engine uses both collaborative filtering and content-based filtering to curate a personalized set of recommended movies and shows for each user. The process is complex and computationally intensive, making Spark's distributed computing capabilities a perfect fit.

The recommendation engine also takes care of choosing the right artwork or thumbnail for each movie or show. The system generates multiple thumbnails and uses advanced machine learning techniques to select the one that is most likely to attract a particular user based on their viewing history.

Question: How does Netflix's recommendation system impact user experience and engagement?
Answer: Netflix's recommendation system plays a significant role in enhancing user engagement by providing personalized content recommendations. It understands user preferences and suggests content that aligns with their interests. This increases the likelihood of users finding content they enjoy, thereby increasing overall user satisfaction and retention.

AWS Services — Route53, SNS, S3, and more

Finally, it's important to note that Netflix extensively uses various Amazon Web Services (AWS) to supplement its own infrastructure. AWS Route 53 is used for DNS, Amazon Simple Notification Service (SNS) is used for push notifications, and Amazon S3 is used for storing massive amounts of data, including video files and user viewing history.

Question: Why would Netflix choose to use AWS services rather than creating their own solutions?

Answer: Leveraging AWS services allows Netflix to focus on their core business — delivering excellent streaming experiences — rather than investing time and resources into developing, deploying, and maintaining such infrastructural components. AWS services are reliable, highly scalable, and offer a broad set of capabilities that can be tailored to Netflix's needs.

In conclusion, Netflix's system design serves as an outstanding example of how various technologies and architectural patterns can be combined to build a highly scalable and reliable platform. The system's design ensures that millions of users worldwide can enjoy streaming content with minimal interruptions or delays. This design also allows Netflix to efficiently store and analyze vast amounts of data, enabling them to continually improve and personalize their service.

Netfl System Design Interview

Written by Sanket Saxena

Follow

137 Followers

I love writing about software engineering and all the cool things I learn along the way.

More from Sanket Saxena

Sanket Saxena

System Design of Collaborative Editing Tool

The widespread shift towards remote and collaborative work necessitates advanced

Jun 17, 2023 4

Sanket Saxena

Database—Part Three

Distributed Locking: A Detailed Overview

Jul 10, 2023 2 1

Sanket Saxena

Rate limiting, API gateway & Load balancing

Rate Limiting

Jul 5, 2023 6

Sanket Saxena

Concurrency in Python

This article delves into advanced techniques and best practices for leveraging Python's

Jun 15 4

See all from Sanket Saxena

Recommended from Medium

Santosh P.

System Design Of Netflix

Netflix is a subscription-based streaming service that allows our members to watch TV

Jul 5 1

Love Sharm in ByteByteGo System Design Allianc

System Design Blueprint: The Ultimate Guide

Developing a robust, scalable, and efficient system can be daunting. However,

Sep 18, 2023 8.7K 59

Lists

Natural Language Processing

1733 stories · 1315 saves

Alexander Nguyen in Level Up Coding

Talha Şahin

The resume that got a software engineer a $300,000 job at Google.

1-page. Well-formatted.

Jun 1 23K 452

High-Level System Architecture of Booking.com

Take an in-depth look at the possible high-level architecture of Booking.com.

Jan 10 6K 47

Vishal Lokam

I just spent 5+ hours reading Netflix engineering blogs, so you

My learning while exploring the Netflix tech blogs

Jun 17 108 1

Kelechi Divine

How to Design a photo-sharing system

Table of contents

Jul 2

See more recommendations