Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

ingweb-tx-3

Ingeniería Inversa

Autor:	Toral Pallás, Héctor - 798095
Grado:	Ingeniería Informática
Curso:	2022-2023

31 de diciembre de 2022

Índice

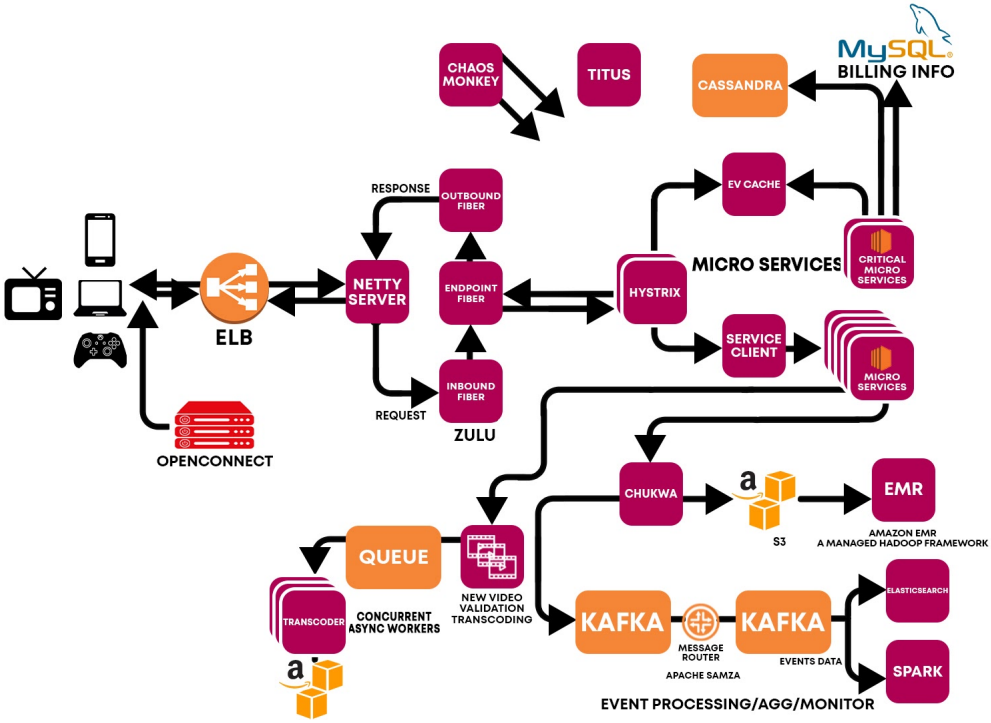
1. Introducción

Para la realización del informe se ha seleccionado Netflix; La empresa de entretenimiento y plataforma de streaming estadounidense que fue fundada el 29 de agosto de 1997 y un año después comenzó su actividad ofreciendo un servicio de alquiler de DVD a través del correo postal. En la actualidad, podemos ver como la forma en la que Netflix distribuía su producto ha ido cambiando a lo largo de los años.

Para hacernos una idea de cuales son algunos de los requisitos de la plataforma netflix.com dejo a continuación una tabla de requisitos y unos datos sobre la plataforma.

código	**Descripción**
RF-1	Crear cuentas, iniciar sesión y eliminar la cuenta.
RF-2	Suscribirse o darse de baja en diferentes planes.
RF-3	Permitir a los usuarios tener y manejar varias cuentas.
RF-4	Permitir a los usuarios ver vídeos.
RF-5	Permitir a los usuarios descargar y ver en modo offline.
RF-6	Permitir a los usuarios buscar y descubrir el vídeo a través de su título.
RF-7	Los desarrolladores de Netflix pueden subir vídeos desde el backend y mostrarlos en la plataforma.
RF-8	La plataforma mostrará las tendencias, los vídeos más populares y por categorías, lo que facilitará la elección a los usuarios.
RF-9	selección del idioma de los subtítulos para que los usuarios puedan ver cualquier vídeo aunque no hablen los idiomas.
RF-10	Agrupación de vídeos (series de TV, series dramáticas, series de películas y tratamiento de cada vídeo como independiente)
RF-11	Analytics también proporciona al usuario sugerencias o recomendaciones para el tipo similar de vídeos a los usuarios basados en el comportamiento del usuario.
RF-12	Sincronizar para el dispositivo diferente bajo la misma cuenta, lo que significa que los usuarios pueden utilizar un dispositivo diferente para seguir viendo el mismo episodio sin repetición.
RF-13	Reproducción 24/7.
RF-14	Fallback.
RNF-1	Los usuarios pueden ver vídeos en tiempo real sin retrasos ni problemas de latencia.
RNF-2	El sistema será muy fiable
RNF-3	Alta disponibilidad.
RNF-4	Escalabilidad.
RNF-5	Los datos de vídeo son duraderos y de fácil acceso.

2. Arquitectura

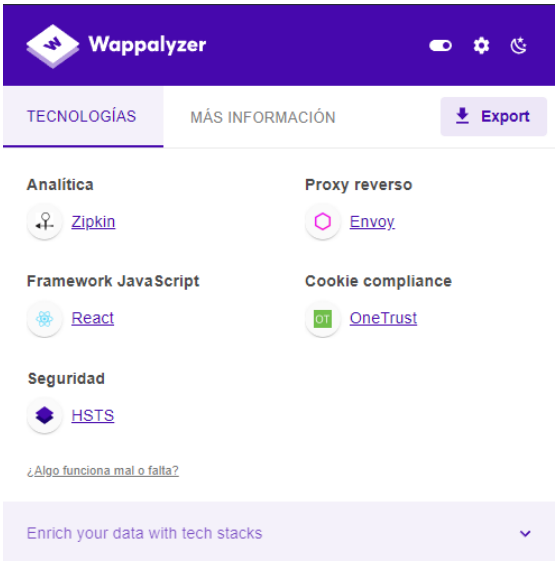


La arquitectura de Netflix representada en el diagrama mostrado en la parte superior se puede diferenciar en 4 partes.

La primera situada más a la izquierda, que engloba los dispositivos desde los cuales los clientes pueden acceder a los servicios de Netflix. La segunda, justo a su derecha que reflejaría el microservicio en particular al que se conectaría el usuario en un momento determinado y justo debajo la tercera encargada del tráfico y manejo de eventos.

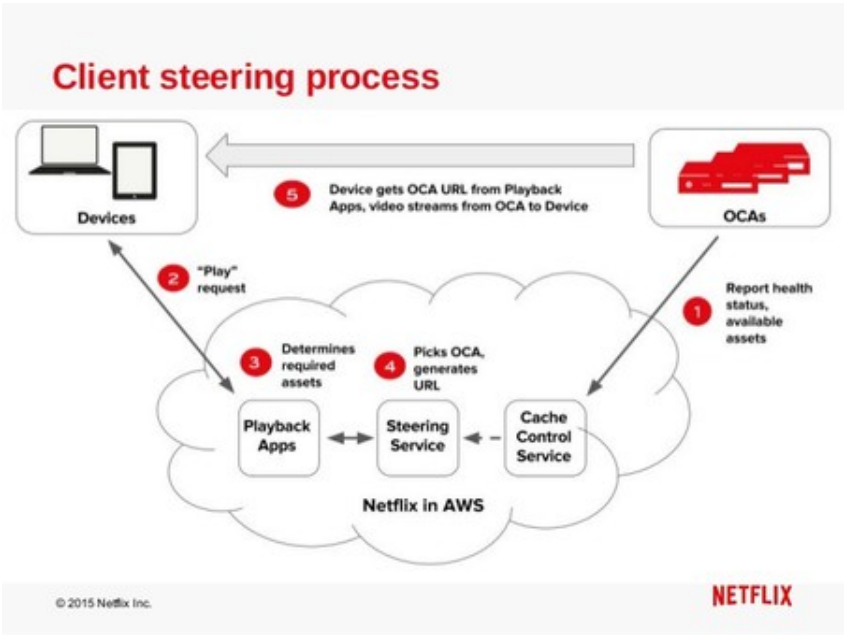
Por último, la cuarta parte podríamos decir que sería la parte de Chaos Monkey encargada de comprobar que los ingenieros desarrollan sus microservicios de forma resiliente.

3. Front-End



Para la capa de frontend, Netflix utiliza React ya que esta tecnología permite una primera carga más rápida a la vez que una gran modularidad y buen rendimiento en tiempo de ejecución.

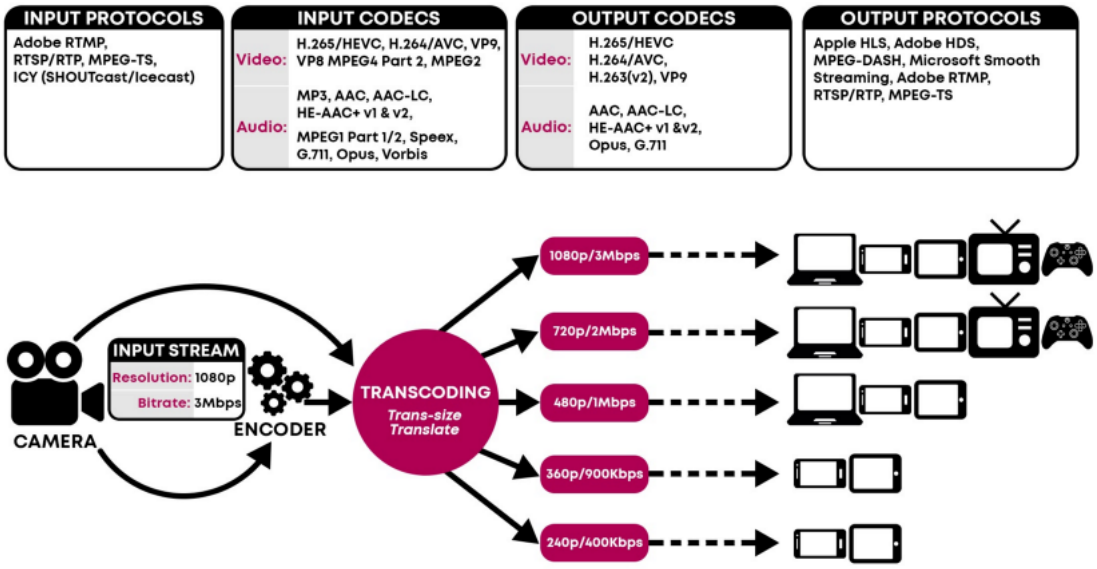
Además, para distribuir su plataforma hemos visto que en la parte de arquitectura Netflix utiliza una red de CDNs, la que permite distribuir el contenido de forma global permitiendo a los usuarios el accederlo con una menor latencia. Además, también sirve de cache de tal manera que si el usuario no encuentra el vídeo puede obtenerlo de ahí. Sin embargo, Netflix no utiliza una red de CDNs convencional sino que desarrollo una solución con OpenConnect. Como el objetivo de Netflix era reducir el tiempo de carga para el usuario, construyó lo que llaman como Netflix and Chill de tal manera que para evitar tener todas las películas en todos los servidores del mundo lo que hicieron fue que mediante técnicas de machine learning preveer donde se iba a consumir cada contenido para distribuirlo de la mejor forma posible para que así tener disponible los recursos necesarios en los momentos indicados, permitiendo así ahorrar tiempo de carga y espacio.



4. Servicios Web

Como hemos comentado en la sección de arquitectura, Netflix opta por un diseño basado en microservicios entre los que podemos distinguir:

- Servicio de usuarios y autenticación
- Servicio de gestión de suscripciones
- Servicio de vídeos: Guarda metadatos en los RDBMS, para un acceso rápido este servicio implementa una cache de escritura utilizando una caché en memoria como Redis o Memcached.
- Servicio TransCoder: Se encargaría de subir el mismo vídeo en diferentes formatos a la vez que actualizaría los metadatos del vídeo para permitir búsquedas full text y a la vez que deja el recurso marcado como disponible para su consumo.
- Servicio de búsqueda global: Encargado de hacer los rankings y la busqueda de los recursos.



Para escalar dichos servicios utilizan distintas técnicas como el escalado horizontal para crear nuevas instancias cuando estas sean necesarias debido a la escasez de recursos del servidor como podría ser el consumo excesivo de CPU o RAM y otras técnicas que se comentarán en apartados siguientes.

5. Búsqueda y Persistencia

Para la capa de búsqueda Netflix utiliza en gran parte elastic search llegando a tener entre 700-800 nodos en producción repartidos entre 100 clusters de Elasticsearch superando los límites cuando se trata de extraer información en tiempo real a gran escala. Lo usan principalmente para visualizar datos, soporte a clientes y detección de errores.

Para la parte de persistencia, utilizan distintas bases de datos tanto SQL como No-SQL entre las que están MySQL y Cassandra.

Para la administración de los títulos de las películas, la facturación se requieren propiedades ACID es por eso que utilizan para este caso MySQL como base de datos relacional. Además, Netflix tiene una implementacion master-master de MySQL construida usando InnoDB sobre instancias de EC2 en la infraesrtuctura de Amazon. La configuración sigue el "Synchronous replication protocol". La replicación de los datos se realiza de forma síncrona, lo que indica que existe una relación master-slave entre los nodos, y cualquier operación de escritura en el nodo principal se considerará realizada solo si los datos están sincronizados por los nodos locales y remotos para garantizar una alta disponibilidad. Las consultas de lectura no son manejadas por el nodo principal sino por réplicas, y solo las consultas de escritura son manejadas por la base de datos maestra. En caso de conmutación por error, el nodo secundario ocupará el nodo maestro y manejará bien la consulta de escritura.

Por otro lado utilizan Cassandra para guardar el historial de usuario. Utilizan esta debido a que les permite manejar grandes volúmenes de lectura de forma eficiente y optimiza la latencia para operaciones de lectura costosas. Algunas caracteristicas sobre el uso de casandra en Netflix son:

- Más de 50 clústeres de Cassandra
- Más de 500 nodos
- Más de 30 TB de copias de seguridad diarias
- El clúster más grande tiene 72 nodos.
- 1 clúster de más de 250 000 escrituras/s

6. Infraestructura

La infraestructura de Netflix esta construida sobre la de Amazon AWS en su totalidad afirmado por ellos en este recurso en el minuto 32,14.

Ahí explican durante la ronda de preguntas que ellos no tienen un problema si una región de AWS se cae, debido a que tienen un sistema probado ante fallos de este estilo en el que transfieren el tráfico a otras regiones. Además, la filosofía del equipo de desarrollo se basa en hacer sus sistemas robustos frente a caídas inesperadas de instancias en producción esto lo consiguen gracias a Chaos Monkey una herramienta de código abierto desarrollada por ellos para generar esta serie de fallos en sus maquina en producción.

7. Otros

Escalabilidad

Como se ha mencionado anteriormente, Netflix utiliza diversas técnicas así como el escalado horizontal, replicación de la base de datos con un patrón de master-master en el que las lecturas se realizan de las replicas y las escrituras en los principales, sharding de la base de datos distribuyendo así sus datos en multiples servidores, sharding de cache creando así una cache distribuida sobre varias instancias de Redis y Sharding en las búsquedas mediante el soporte nativo que ofrece Elastic Search.

Seguridad

Netflix entre otros utiliza https para encriptar el tráfico entre cliente-servidor. Además cada petición posterior al inicio de sesión va firmada por un token en la cabecera http que garantiza que las solicitudes son legítimas.

Resistencia

Replicar servidores de bases de datos en una configuración maestro-esclavo de tal manera que si uno de los nodos se cae, los demás toman el relevo y siguen funcionando como es debido.

Balanceo de carga

El balanceador de carga se encarga de enrutar el tráfico a los servicios frontales. Elastic Load Balancing, ELB realiza un esquema de equilibrio de carga de 2 niveles en el que la carga se equilibra primero en las regiones y después en las instancias (servidores).

El primer nivel consiste en un equilibrio mediante Round Robin básico basado en DNS. Cuando la solicitud llega al primer equilibrio de carga, se equilibra a través de una de las zonas que su ELB está configurado para utilizar.

El segundo nivel es una matriz de instancias de balanceador de carga y realiza la técnica de distribución Round Robin para repartir las solicitudes entre las instancias que están detrás de él en la misma zona.

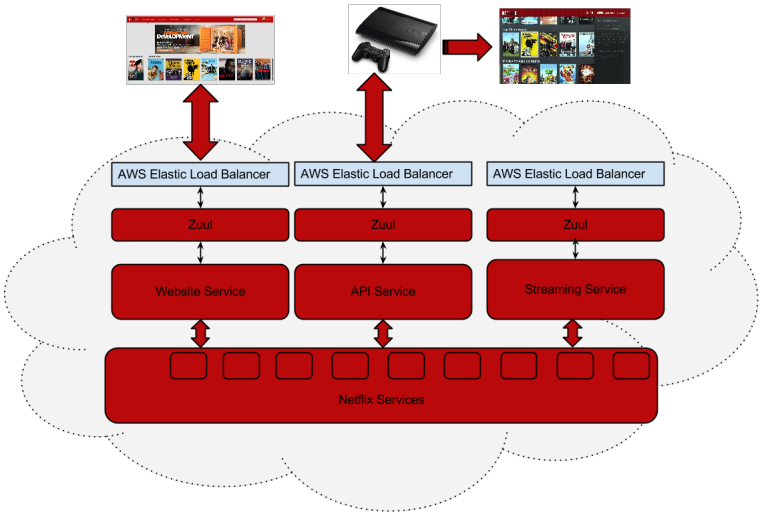
A parte de estos dos niveles, es el propio balanceador de carga que se encarga de detectar que máquinas son las que están todavía activas para desconectar del balanceo aquellas que no lo están de tal manera que no haya fallos por solicitudes no atendidas.

Redundancia geográfica

Como se menciona en una de las secciones anteriores, Netflix utiliza esta técnica para preveer posibles perdidas de servicio debido a la caída de una región entera del proveedor de su infraestructura (Amazon), de tal manera que así garantiza que si una región cae, el contenido podrá seguir sirviendose desde otro centro de datos.

ZUUL

Zuul es una librería generada por Netflix para tener un solo punto de acceso (gateway) a todo los componentes que forman parte de sus sistemas. Por medio de este pueden gestionar el acceso a diferentes microservicios en vez de que cada uno tenga un punto de acceso único.



Zuul a su vez sirve para realizar pruebas de estrés redirigiendo mas tráfico hacia ciertas máquinas para medir en ellas ciertas características para así comprobar por ejemplo si una compilación completa tiene o no las características de rendimiento necesarias para pasarla a producción. Además es parte fundamental del proyecto de resistencia multirregional del ELB, que denominan Isthmus para enrutar las solicitudes de la región de la nube de la costa oeste a la costa este para ayudarles a tener redundancia multi-región en sus ELBs para los dominios críticos.

Hystrix

La librería Hystrix ayuda a controlar las interacciones entre estos servicios distribuidos añadiendo tolerancia a la latencia y lógica de tolerancia a fallos. Hystrix hace esto aislando los puntos de acceso entre los servicios, el sistema remoto y las librerías de terceros. La librería ayuda en:

- Detener los fallos en cascada en un sistema distribuido complejo.
- Controlar la latencia y los fallos de las dependencias a las que se accede (normalmente a través de la red) mediante bibliotecas cliente de terceros.
- Fallar rápido y recuperarse rápidamente.
- Fallback y gracefully degrade cuando sea posible.
- Permitir la supervisión, las alertas y el control operativo casi en tiempo real.
- Caché de peticiones consciente de la concurrencia. Agrupación automatizada mediante colapso de solicitudes.

8. Conclusión

Tras haber realizado el informe, se puede observar como el conjunto de decisiones tomadas por el equipo de desarrollo de Netflix tienen bastante sentido, empezando por su arquitectura. Como hemos podido observar en el diagrama simplificado tienen todo colocado en distintos microservicios los cuales interactúan con los clientes y entre ellos mismos además, la librería Zuul les permite una política de control de acceso bastante granulada ya que permite filtrar el trafico que debe ir a cada uno de estos microservicios.

Por el lado del frontend, se ha podido observar el problema que tiene Netflix para almacenar todos sus recursos (vídeos), esos que los clientes van a consumir desde distintos dispositivos y como lo gestionan mediante algoritmos de machine learning para distribuirlos momentos antes de que un usuario lo necesite para que así hacer que la experiencia del usuario sea mucho mejor a la vez que ahorran en espacio teniendo todas las versiones de cada recurso en cada servidor.

En cuanto a la capa de servicios han optado por un tipo de escalado bastante común como es la replicación de máquinas bajo demanda usando servicios de Amazon como el EC2 Auto Scaling.

Por último, en cuanto al tema de persistencia, hemos podido observar como Netflix no utiliza una misma bbdd para todos sus microservicios sino que escoge aquella que más le conviene según el caso de uso así como usar Redis para implementar un sistema de cache distribuido. También podemos observar como han decidido el utilizar soluciones de sharding nativas como la que ofrece Elastic Search.

9. Anexo

En el anexo se exponen algunos de los enlaces visitados a la hora de redactar el informe:

- La Tecnología Global de Netflix. BettaTech
- System Design Netflix. A Complete Architecture
- Netflix System Architecture
- Cloud Architecture. Netflix TechBlog
- Netflix's Viewing Data
- Announcing Zuul: Edge Service in the Cloud
- arRESTful Development: How Netflix Uses Elasticsearch to Better Understand
- Netflix/zuul
- Configuración de una replicación Master-Master en MySQL