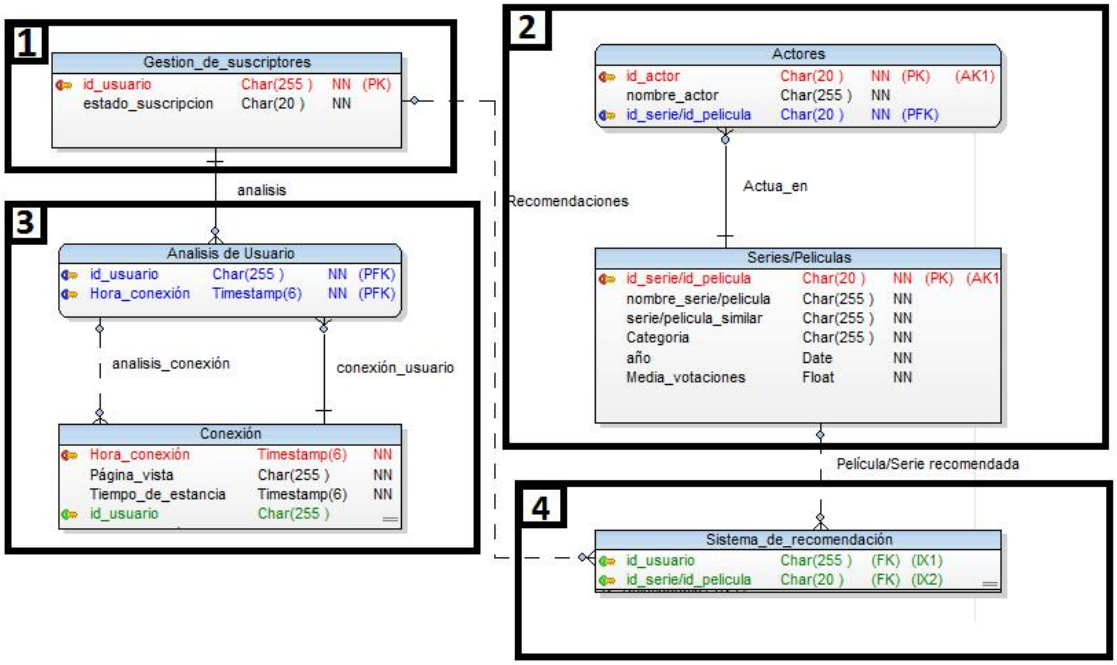
PROPUESTA VEEME

Máster en Business Analytics y Big Data
2016/2017

Mario Puente Anguio
Miguel David Monzón Fernández-Bermejo
Angello Rodríguez Gutiérrez

ÍNDICE

MODELO DE DATOS



1. Propuesta para la gestión de suscriptores: RIAK
2. Propuesta para el catálogo de películas y series: Neo4J
3. Propuesta para el análisis de usuarios: Cassandra
4. Propuesta para el sistema de recomendaciones: HBase

Nota: En vez de utilizar dos entidades, Series y Películas, se ha utilizado una única entidad porque ambas utilizan los mismos atributos y de esta manera se simplifica el modelo de datos. Sin embargo, también se acepta el modelo de datos con esas dos entidades por separado.

PROPUESTA DE BASES DE DATOS A UTILIZAR

Gestión de suscripciones

La única información a gestionar es conocer si un usuario tiene la suscripción activa o no, se propone un sistema Clave-Valor de manera que ofrezca la máxima disponibilidad.

Por ello proponemos utilizar la base de datos **RIAK** en la que la clave puede ser un código de usuario, y el valor True/False en función de si tiene la suscripción activa o no. Sabiendo que el número de usuarios activos se duplica cada mes, también se consigue la escalabilidad horizontal deseada.

Tampoco hace falta almacenar información muy compleja, por lo que no es necesario realizar búsqueda por atributos, únicamente es necesario obtener si esta la suscripción activa o no.

Se ha descartado la base de datos MongoDB**,** orientada a documentos, ya que no nos aporta esa disponibilidad que necesitamos y no necesitamos esa búsqueda especifica en atributos. La base de datos Neo4j no ha sido seleccionada en este caso debido al tema de escalabilidad, que potencia la escalabilidad vertical sobre la horizontal y pueden producirse caídas del sistema en casos de fallo. Cassandra no ha sido elegida puesto que no trabajamos con mucho tráfico de datos y tampoco se requieran muchas escrituras, estadísticas o análisis con los datos.

En cuanto al diseño conceptual, la estructura del agregado sería:

- Clave: clave de usuario.
- Valor: True/False

Catálogo de películas y series

Necesitamos una base datos que garantice la accesibilidad de todos los usuarios, conservando la integridad de datos y permita la disponibilidad por cada región, sin la exigencia de escalar horizontalmente.

Por lo tanto, la base de datos escogida es **Neo4j**, con el sistema de transacción ACID, puede ofrecer tanto la integridad, consistencia como la disponibilidad que se necesita y este tipo de datos se vinculan perfectamente a través de propiedades y relaciones.

El catálogo crece lentamente, por lo que la escalabilidad horizontal no es un requisito esencial. Es adecuado para representar las relaciones entre actores y películas y series, y se ahorra tiempo sobre tener que realizar SQL Joins.

El sistema solo tendrá metadatos sobre películas, series y actores. Películas y series en sí se almacenarán en un sistema de streaming separado.

Información a gestionar:

- Metadatos de películas y series: indicar películas/series similares, categoría, año de estreno y sinopsis.
- Metadatos de actores: indicar películas/series donde han actuado.

Por ejemplo, si se busca un actor se podrá obtener de forma rápida las películas/series en las que ha actuado, además de sus atributos y otras películas/series relacionadas con éstas.

Las demás bases de datos se han descartado por la carencia del sistema transaccional ACID y el tipo de escalabilidad.

Modelado del grafo:

- Película -[similar]-> Película.
- Película -[similar]-> Serie.
- Atributos de película: categoría, año de estreno y sinopsis.
- Actor -[actúa en] -> Película.
- Actor -[actúa en] -> Serie.

Análisis de usuarios

Para el análisis de usuarios es muy importante la escalabilidad horizontal debido a que estamos procesando información continuamente sobre todos sus movimientos y operaciones, produciéndose de esta manera mucho tráfico de datos.

La base de datos que usaremos será **Cassandra**, que debido a su sistema de supercolumnas y claves cuenta con un veloz tiempo de respuesta, organizando toda la información necesaria. Está diseñada para almacenar actualizaciones en la actividad del usuario como la hora de la conexión, página vista y tiempo que ha permanecido en ella y obtener resultados estadísticos de sus análisis de una manera más eficaz y eficiente que el resto de bases de datos.

Gracias a ello se podrá almacenar abundante información de conexiones de usuarios y poder aceder a esa información a través de la clave de usuario, mejor que, por ejemplo, _id de MongoDB.

Estructura del Agregado:

- Clave de usuario (Clave 1).
  - Hora de conexión (Clave 2).
    - Conexión.
    - Página vista.
    - Tiempo que ha permanecido en ella.

Sistema de recomendación

Para un sistema de recomendación se considera importante la consistencia de la información, para que ciertamente se le recomiende al usuario las películas adecuadas, además de que vea integridad en las votaciones que ha realizado.

Para conseguir una respuesta rápida de acceso, además de una escritura rápida (cada vez que el usuario vea una nueva película/serie, se recalcula el sistema de recomendación), manteniendo la consistencia, se propone utilizar **HBase**. Es una base de datos orientado a columnas, en la que prima la consistencia frente a la disponibilidad y escala bien horizontalmente.

Estructura del agregado:

- Clave de usuario (Clave 1):
  - Lista de votaciones realizadas.
  - Películas que ha visto.
  - Series que ha visto.
  - Recomendaciones personalizadas (calculada externamente).