



Ingeniería en Sistemas Computacionales

Bases de Datos No Relacionales

“NoSQL Success Stories”

Marco Ricardo Cordero Hernández

Tlaquepaque, Jal., 23 de enero de 2023

No es ningún secreto el hecho de que las bases de datos relacionales no siempre son la opción adecuada cuando se trata del manejo de la información orientada hacia un propósito específico. Cuando esto sucede, usualmente se recurre a soluciones emergentes y relativamente nuevas a comparación de la longevidad de los modelos entidad-relación de antaño. Dicho esto, a continuación se presentan una serie de casos de éxito en donde lo anterior ha tenido cabida.

Bases de datos orientadas a columnas (columnar databases)

Motor utilizado: Apache Druid

Compañía: Netflix

Según Ben Sykes (2020), Netflix, una compañía de streaming perteneciente a la lista de las “Fortune 500”, siempre se ha encontrado innovando y a la vanguardia en cuanto a integración de soluciones tecnológicas se refiere. Cuando la oferta de valor involucra ofrecer contenido digital con alta disponibilidad y definición, el análisis segregado de la información se vuelve sumamente relevante.

La forma en la que estos datos se recaba es mediante la recolección de *logs* de datos de usuarios a los cuales se les han proporcionado las actualizaciones más recientes de la plataforma, generando así alrededor de 2 millones de eventos por segundo. *Druid*, descrita como “una base de datos de alto rendimiento para análisis en tiempo real [...] excelente para visualización inmediata de datos, consultas ad-hoc, análisis operacional, y manejo de alta concurrencia”.

Con la utilización de *Druid*, Netflix es capaz de detectar picos de errores de manera inmediata con un alto nivel de personalización en su extracción e interpretación. La refinación de esto varía desde el tipo de dispositivo en donde el rendimiento decae más notoriamente, hasta la versión, región, momento, etc.

Bases de datos de documentos (document based databases)

Motor utilizado: MongoDB

Compañía: Pearson

Según un artículo encontrado en su propio portal (2017), *MongoDB*, quizás la más famosas de las bases de datos de documentos, fungió como piedra angular para el despliegue de una aplicación *mobile-first* para la compañía *Pearson*, líder en educación para múltiples niveles educativos, en la cual sería posible el fácil acceso a recursos didácticos con un enfoque dirigido hacia públicos en donde la forma de consumir este material es principalmente a través de dispositivos móviles.

La relevancia del uso de *MongoDB* recae en la agilidad y escalabilidad que este ofrece, así como un modelado orientado a objetos el cual puede ser almacenado fácilmente en archivos de texto plano sobre los cuales se pueden realizar operaciones de rápida lectura y consultas ad-hoc. Otro punto crítico en este caso es la necesidad de capturar el valor intrínseco de los datos sin estructura

predefinida, tales como interacciones sociales de punto a punto, las cuales se han vuelto un elemento fundamental en la educación contemporánea.

Para su momento de implementación, Pearson reportó transacciones exitosas de 120 millones de usuarios, con un pico de 50 millones de cuentas concurrentes por día.

Bases de datos orientadas a grafos (graph databases)

Motor utilizado: Neo4j

Compañía: Cisco

Finalmente, un caso presentado directamente por el grupo de Neo4j (2021) detalla cómo Cisco, dentro del top 100 de las Fortune 500, logró disminuir el tiempo en el que sus empleados realizaban búsquedas de documentos relevantes para ventas a potenciales clientes.

El problema del factor humano se hace presente nuevamente, en donde los representantes de ventas trabajando en Cisco requerían de la información contenida en archivos de la empresa, detallando información relevante para concretar posibles transacciones con prospectos para ventas. La cuestión es que, al contar con una base de archivos sumamente grande, los empleados perdían hasta una hora tratando de encontrar archivos específicos para las negociaciones. Neo4j detalla cómo Cisco transformó sus archivos hacia una asignación latente de Dirichlet para posteriormente alimentar los datos hacia el mismo motor mencionado previamente, creando así una ontología (forma de representación de un determinado conocimiento [Barrasa, 2020]). Esto ayudó a crear un sistema de indexación y etiquetado de archivos, que a su vez, dio paso a un motor de búsqueda basado en palabras clave y recomendaciones en tiempo real.

Referencias bibliográficas

Sykes, B. (2020). *How Netflix uses Druid for Real-time Insights to Ensure a High-Quality Experience*. Recuperado el 21 de enero del 2023 de <https://netflixtechblog.com/how-netflix-uses-druid-for-real-time-insights-to-ensure-a-high-quality-experience-19e1e8568d06>.

MongoDB (2017). *A Mobile-First, Cloud-First Stack at Pearson*. Recuperado el 22 de enero del 2023 de <https://www.mongodb.com/blog/post/mobile-first-cloud-first-stack-pearson?c=4fbcd947c6>.

Neo4j, Inc. (2021). Real-Time Graph Analysis of Documents Saves Company Over 4 Million Employee Hours. Recuperado el 22 de enero del 2023 de https://go.neo4j.com/rs/710-RRC-335/images/Neo4j-case-study-Cisco-EN-US.pdf?_gl=1*caapc8*_ga*MTA3MjI3Mjg0Mi4xNjY4NDgxNDgx*_ga_DL38Q8KGQC*MTY3NDM3MDUyMi4yLjEuMTY3NDM3MjEyOS4wLjAuMA..&_ga=2.81746449.763496034.1674370523-1072272842.1668481481.

Barrasa, J. (2020). *Ontologies in Neo4j: Semantics and Knowledge Graphs*. Recuperado el 22 de enero del 2023 de <https://neo4j.com/blog/ontologies-in-neo4j-semantics-and-knowledge-graphs/>.