

## Por qué los ingenieros de Uber cambiaron de Postgres a MySQL

Inicialmente cuando se empieza con el proyecto de Uber se implementa un aplicación back-end basada en python la cual usaba postgres para la persistencia de los datos, pero a medida que la aplicación crece, en usuarios y funcionalidades se fueron presentando algunas limitaciones con respecto a Postgres, una arquitectura pobre al momento de programar, una replicación de datos ineficiente, problemas con tablas corruptas, mal soporte para manejo de la consistencia de los datos además de dificultades al actualizar el aplicativo a nuevas versiones.

Se debe tener en cuenta que para una base de datos relacional debe ejecutar ciertas tareas claves: capacidad de insertar, actualizar y eliminar, capacidad para hacer cambios en los esquemas, además de la capacidad de manejar la concurrencia en múltiples versiones, es cuando se observa que con postgres los índices, tanto primarios como secundarios apuntan directamente a los desplazamientos de espacio en disco de la tupla, es cuando la tupla cambia que todos los índices deben actualizarse.

Postgres también tiene problemas de replicación y la manera como maneja los respaldos encaso que el sistema colapse, ya que puede hacer tedioso el manejo de los datos y su relación con el espacio en disco reservados para su almacenamiento. Además el diseño final era ineficiente y presentaba dificultades para los datos de uber.

El primer problema con Postgres fue que para cada pequeño cambio, hace necesario ejecutar una considerable cantidad de sentencias necesarias para no afectar nuestro modelo de datos. la replicación de Postgres aún puede causar una cantidad abrumadora de datos para una base de datos que usa muchos índices. La compra de cables de red de gran ancho de banda es costoso, e incluso en los casos en que el dinero no es un problema, simplemente no es posible obtener un enlace de red entre países con el mismo ancho de banda que una interconexión local., lo cual también representó un problema para uber que tiene un crecimiento exponencial en su servicio.

Aparte también presenta problemas con la corrupción de datos y en conclusión no estaban seguros sobre la cantidad de réplicas con datos dañados y si el problema podría haber afectado al maestro. Aunque el problema afectaba a pocas filas de las tablas, se identifica que este problema afecta directamente la capa física, y esto podría terminar afectando todos los índices de la base de datos. Finalmente se revolió la inconsistencia de las réplicas, respaldándolas todas a una nueva instancia del maestro, pero esto era un proceso laborioso y solo se tenía la capacidad suficiente para sacar algunas réplicas del grupo de equilibrio de carga a la vez. Otro grande problema son las actualizaciones de Postgres, gracias a que la replicación de datos funciona directamente sobre la capa física, no es posible replicar la información cuando se tiene diferentes versiones. Este problema se mitigó primero apagando la base de datos, luego clonando la base de datos maestro, luego se ejecutó nuevamente el maestro, se borran todas las réplicas y se restablece nuevamente, se actualiza nuevamente y se ejecutan las réplicas nuevamente.

Ahora viendo las desventajas presentadas con el motor de bases de datos Postgres, se tuvieron los siguientes aspectos en cuenta que se acoplaban a las necesidades de Uber y que llevó al uso de MySQL, ya que está se encontró más favorable para los usuarios.

MySQL utiliza una capa adicional de direccionamiento lo cual permite que los registros de índice secundario que apuntan a los registros de índice primario, y el índice principal en sí mismo contienen las ubicaciones de archivo en disco. Si un desplazamiento de fila cambia, solo el índice principal debe actualizarse. Lo cual hace la labor mucho más sencilla. En MySQL la replicación se vuelve más compacta y sencilla de implementar, las cuales resultan ser más aplicables y eficientes. De la única manera que falle una réplica en MySQL es en caso que alguna sentencia no se ejecute correctamente aunque no sucede con frecuencia, esto puede causar pérdida de datos, pero no afectará el funcionamiento de la base de datos en general.

Otras ventajas de diseño que tiene MySQL es que la arquitectura de MySQL es mucho mejor que postgres. En cuanto al almacenamiento del buffer, hace posible ejecutar un diseño LRU aleatorio, lo q hace posible detectar patrones de acceso patológico que apagarían el LRU y evitar que causen demasiado daño.

Por último una de las grandes ventajas es el manejo de las conexiones con MySQL, ya que implementa conexiones concurrentes que en definición consumen mucha menos memoria y esto ayuda a trabajar más fluidamente en la herramienta, y manejar más conexiones abiertas simultáneamente trabajando con normalidad.

En conclusión Postgres le fue de gran utilidad a Uber en un principio cuando no había o manejaban grandes volúmenes de información, en la actualidad los datos son demasiado robustos y se identificó que la mejor manera de manejar dichos datos es por medio de MySQL.