# **Ensayo**

# **¿Por qué la ingeniería de UBER ENGINEERING cambió de Postgres a MySQL?**

Actualmente la necesidad de compañías y sitios web requieren almacenar y administrar grandes volúmenes de datos de la forma más eficiente posible. Muchos optan por soluciones comerciales, parte de éstos consideran confiables el uso e implementación de software libre optando como solución a PostGreSQL o MySQL.

UBER ENGINEERING fue una de las empresas que inicialmente optaron por operar con Postgres, sin embargo, debido al crecimiento de sus operaciones decidieron migrar a MySQL, lo cual los llevo a enfrentar innumerables retos propios de una migración de esta magnitud.

De acuerdo a lo documentado por ello, en nuestra opinión consideramos que la problemática que llevó a UBER ENGINEERING a migrar del motor de base de datos Postgres a MySQL tuvieron que ver con las operaciones de escrituras lentas, operaciones en los índices a muy alto costo computacional y problemas con los updates de Postgres.

Debido a que Postgres se ve obligado que reconstruir sus índices debido al UUID interno (Universally Unique IDentifier) que le asigna, y en combinación con replicación tiene que hacer múltiples escrituras en disco, lo cual tiende a convertirse en un problema si se tienen varios datacenters repartidos, ya que no les daba el ancho de banda para enviar tanta información. La replicación en MySQL por otro lado, es sumamente sencilla y no requiere de tanta información.

El otro punto en contra para Postgres era que estaban operando sobre la versión (9.2) la cual es bastante desactualizada y que de acuerdo a lo documentado dichas falencias fueron mejoradas en la actualización de versión a 9.5

Se ha demostrado que tanto Postgres como MySQL cuentan con características a favor y en contra, de las cuales listaremos las más relevantes:

**Características a favor de Postgres**

* Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
* Implementa el uso de rollback’s, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.
* Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

**Características en contra de Postgres**

* Consume gran cantidad de recursos.
* Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
* Es de 2 a 3 veces más lento que MySQL.

**Características a favor de MySQL**

* Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
* Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
* Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
* Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
* El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).

**Características en contra de MySQL**

* Carece de soporte para transacciones, rollback’s y subconsultas.
* El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
* No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.