

BASES DE DATOS DINÁMICAS Y ESTÁTICAS: UNA COMPARACIÓN ENTRE POSTGRES Y MySQL CASO UBER

Dada la demanda actual de los sistemas informáticos, la necesidad de tener bases de datos que sean óptimas (Buscando la eficacia encima de la eficiencia) es un paradigma a romper, dado que ello representa capacidad, adaptabilidad, fiabilidad, tiempo y finalmente costo para una organización. El cambio que describió la aplicación UBER en su artículo explicando las razones técnicas de su migración, es un claro ejemplo de la importancia de usar soportes que tengan la mencionada eficacia necesaria para ser competitivos en el mercado. Cabe aclarar que las bases de datos analizadas en el presente texto, son sin duda dinámicas ambas como lo son desde su concepto, pero se alude a la versatilidad –Término clave del escrito- cuando se diferencian una de otra.

Se le da el apelativo de base de datos estática a PostGres por las siguientes razones: Si bien recopila y tabula toda la información que entra a ella y permite arrojar un resultado cuando se solicita la búsqueda de determinado usuario, PostGres –Con base al artículo de UBER Engineering- tiene una deficiencia al momento de modificar un ingreso, puesto que la DB crea otra línea que respresenta una marca de texto conservando la obsoleta completamente (Manteniendo un registro de escritura anticipada –WAL-) entendiéndose a PostGres rígida en su necesidad de mantener ambas serie de datos al mismo nivel, y por ende su arquitectura es ineficaz. Esto hace que la mencionada data base, más su carácter inmutable posea los siguientes problemas:

- La hace propenso ante fallos inesperados en el sistema, puesto que la pérdida de información hace que no almacene correctamente los ingresos y mantenga la base de datos obsoleta debido al tamaño y relevancia de ambas.
- Ante una actualización de la base de datos, PostGres no será rápida puesto que al tener información que inclusive ya no es útil para el usuario, aumente el tiempo producto del recorrido que debe hacer por las líneas que son obsoletas tengan igual tamaño que las actualizadas.
- Dada su rigidez, el storage que debe tener producto de los datos obsoletos debe ser mayor (Alrededor de un 25% más según UBER Engineering), lo que implica tener más espacio de almacenamiento físico, haciendo así un aumento de costos y tiempo.
- La BD es proclive a que su información sea corrompida, puesto a que es susceptible a arrojar datos obsoletos. Si bien es menos complejo para un usuario con la experticia encontrar que el sistema tiene una salida errónea, no todos la tendrán y recibirán una respuesta equivocada. En el caso de hacer una limpieza de dichos errores, esta depuración –Considerando la extensión de los datos que tiene UBER en particular- sería una tarea laboriosa y por tanto ineficaz.

- Además de los problemas propios de su arquitectura y por su carácter inmutable, el cambio entre versiones de PostGres se debe hacer a nivel físico y su proceso es más laborioso, donde implica cerrar la BD y volverla abrir, implicando cierres parciales de la aplicación dirigida al usuario final y por tanto inconvenientes y malestar en el usuario final.

Considerando a MySQL como la base de datos dinámica, teniendo en cuenta dos características antagónicas a las de PostGres: Que es más versátil y es mutable, se mejora la eficacia en la manipulación de datos por las siguientes razones:

- El problema principal de PostGres en cuanto a la rigidez de su arquitectura no lo posee MySQL, debido a que su manejo del WAL es más compacto (A pesar de que esta última usa dos claves de registro respecto a la primera) y los hace en función de una marca de tiempo y no en una de escritura, es decir, lo hace en función del momento del ingreso de la línea y no de la variación del texto de la misma. Así, esta última soluciona los fallos inesperados en el sistema, la actualización tarda menos pues la depuración de la cantidad de líneas son más selectivas haciendo la tarea menos laboriosa, el storage posee la capacidad que requiere el sistema y reduce los costos asociados a datos obsoletos dado a que están más compactos (Solo se encuentran a modo de Back Up) y dado a que usa marcas de tiempo, hace que la información sea menos proclive a ser corrompida.
- MySQL tiene una compatibilidad mayor, por ende su proceso de cambio de versión es más ameno y no requiere un proceso tan laborioso como el de PostGres.
- En cuanto a la conexión y el almacenamiento intermedio, MySQL tiene una capacidad mayor pues posee tanto caché como búfers mayores que los de PostGres.

En términos generales, y dado tanto los aspectos de la versatilidad de la arquitectura de la base de datos, compatibilidad, almacenamiento y conexión, hacen de la migración de UBER algo exitoso, puesto que el dinamismo que requiere un sistema como el de esta empresa deba ser igual al de la demanda del usuario final. Posiblemente, a pesar de que ambas bases las usen con código abierto pueda ser de mayor precio la actual, pero la relación costo/beneficio también es superior, debido a que los costos operacionales y de almacenamiento deben ser menor. Por tanto, la apuesta a una base de datos que tenga más potencia y variedad a la hora de ser el soporte de un programa sea dos criterios sumamente importantes a considerar a la hora de seleccionarlo para una empresa.