

1.

Filtrar el campo de palabra de la tabla post, de manera que solo se permita publicar palabras relevantes, eliminando artículos, pronombres y signos de puntuación. Esto mientras se busca una mejor solución.

Recomendaría un BigTable o un AWS ya que estos permiten almacenar grandes cantidades de información. Además, BigTable nos permite tener los datos con timestamps de las cuales se ordenan de manera descendiente, por lo que podemos localizar más fácil las palabras recientes. Por otro lado, AWS nos brinda almacenes para distintos tipos de datos, entre ellos el streaming data que provienen de aplicaciones, como en este caso. Para la base de datos, recomendaría Elasticsearch. Esta base de datos tendría un data node frozen para guardar la información sin gastar mucho los recursos junto con un data node hot para guardar la información reciente.

Es muy conveniente mover la base de datos a un Cloud Provider, en especial si es cold o frozen ya que permite almacenar los datos sin utilizar muchos recursos. Al ser posts en una red social, es muy probable que con el tiempo varios de ellos no sean accedidos por nadie, por lo que un cold o frozen serían ideales en esta situación.

Por medio del char filter y token filter que se le aplican al texto, en este caso a la palabra del post. También, se puede validar la palabra con los stop words para filtrar aquellas palabras que no se quieran permitir de publicar.

2.

Los índices nos permiten realizar búsquedas más rápidas sobre las tablas de la base de datos, lo que permite tiempos de respuesta más rápidos. Cada índice ocupa memoria, por lo que hay que seleccionar adecuadamente los campos que tendrán un índice. Estos campos pueden ser campos que no cambien frecuentemente. Entre más índices hayan o más grande sea el índice, peor será el rendimiento por la cantidad de memoria que se usará.

Si se crearan muchos índices sin importar el hardware, el rendimiento podría ser similar a que si no hubieran índices ya que si se crean índices para casi todos los campos, las búsquedas serían prácticamente lo mismo que si no se tuvieran los índices.

3.

El disco es el componente más lento comparado con la memoria, cache o CPU, por lo que las consultas en las que se necesiten acceder al disco serán más lentas. La memoria almacena datos temporales, por lo que las consultas en las que se necesiten acceder a la memoria serán rápidas. El cache almacena datos de alta velocidad, por lo que las consultas en las que se necesiten acceder al cache serán las más rápidas. El CPU se encarga de ejecutar las consultas. Entre más núcleos tenga el CPU, más threads podrán atender las consultas, por lo que los tiempos de respuesta serán más rápidas. La red es el encargado de comunicar los almacenamientos durante algunos procesos de replicación. Esta comunicación puede tardar, por lo que si el almacenamiento es de consistencia fuerte, el tiempo de respuesta será muy lento, mientras que si es de

consistencia eventual, las consultas pueden no estar actualizadas. Cuando se usa el sistema operativo, la información del CPU se mueve a la memoria. Este proceso, llamado context switch, requiere de bastante tiempo, aumentando el tiempo de respuesta de la base de datos. Los programas de usuario al utilizar también los recursos de la computadora, reducen los recursos disponibles para la base de datos. Esto aumenta los tiempos de respuesta, ya sea porque la memoria está siendo utilizada al completo, por lo que necesitará acceder al disco, o el CPU está realizando procesos de los programas de usuario.

4.

La observabilidad nos muestra la información necesaria, generalmente mediante gráficas, para decidir las configuraciones y los recursos necesarios para lograr la escalabilidad adecuada. Otra métrica importante para lograr la escalabilidad, es la cantidad de usuarios activos que se presentan ya que nos permite saber cuántos recursos se necesitarán para soportar ese aumento de usuarios activos. También, el uso que le dan a la aplicación ya que podemos preparar más a los servicios que presenten mayor actividad.