Resumen #2

Bases de Datos II

Luis Diego Delgado Muñoz

Prof. Nereo Campos

Fecha de Entrega: 30/08/2022

Bigtable: A Distributed Storage System for Structured Data

Bigtable es un sistema de almacenamiento distribuido para administrar datos estructurados que está diseñado para escalar a un tamaño muy grande: petabytes de datos en miles de servidores básicos. Bigtable ha proporcionado con éxito una solución flexible y de alto rendimiento para productos de Google como: indexación web, Google Earth y Google Finance.

Introducción

Bigtable está diseñado para escalar de manera confiable a petabytes de datos y miles de máquinas. Bigtable ha logrado varios objetivos: amplia aplicabilidad, escalabilidad, alto rendimiento y alta disponibilidad. Bigtable es utilizado por más de sesenta productos y proyectos de Google, incluidos Google Analytics, Google Finance, Orkut, Búsqueda personalizada, Writely y Google Earth.

En muchos sentidos, Bigtable se parece a una base de datos, ya que comparte muchas estrategias de implementación con las bases de datos. Las bases de datos paralelas y las bases de datos de memoria principal han logrado escalabilidad y alto rendimiento, pero Bigtable proporciona una interfaz diferente a la de dichos sistemas. Bigtable no admite un modelo de datos relacional completo; en cambio, proporciona a los clientes un modelo de datos simple que admite el control dinámico sobre el diseño y la forma de los datos. Además, los datos se indexan mediante nombres de filas y columnas que pueden ser cadenas arbitrarias. Bigtable también trata los datos como cadenas no interpretadas, aunque los clientes suelen serializar varias formas de datos estructurados y semiestructurados en estas cadenas. Por último, los parámetros del esquema de Bigtable permiten a los clientes controlar de forma dinámica si entregan datos desde la memoria o desde el disco.

Data Model

Un Bigtable es un mapa ordenado multidimensional persistente, distribuido y disperso. El mapa está indexado por una clave de fila, una clave de columna y una marca de tiempo; cada valor en el mapa es una matriz de bytes no interpretada.

Supongamos que queremos conservar una copia de una gran colección de páginas web e información relacionada que podría ser utilizada por muchos proyectos diferentes; llamemos a esta tabla en particular Webtable. En Webtable, usaríamos las URL como

llaves de fila, varios aspectos de las páginas web como nombres de columna y almacenaríamos el contenido de las páginas web en la columna "contents:" debajo de las marcas de tiempo cuando se obtuvieron.

Filas

Bigtable mantiene los datos en orden lexicográfico por clave de fila. El rango de filas de una tabla se particiona dinámicamente. Cada rango de filas se denomina tableta, que es la unidad de distribución y equilibrio de carga. Como resultado, las lecturas de rangos de filas cortos son eficientes y generalmente requieren comunicación con solo una pequeña cantidad de máquinas.

Familias de columnas

Las llaves de columna se agrupan en conjuntos llamados familias de columnas, que forman la unidad básica de control de acceso. Todos los datos almacenados en una familia de columnas suelen ser del mismo tipo. Se debe crear una familia de columnas antes de que los datos se puedan almacenar bajo cualquier llave de columna en esa familia; una vez que se ha creado una familia, se puede utilizar cualquier clave de columna dentro de la familia.

Marcas de tiempo

Cada celda de Bigtable puede contener varias versiones de los mismos datos; estas versiones están indexadas por marca de tiempo. Las marcas de tiempo de Bigtable son números enteros de 64 bits. Pueden ser asignados por Bigtable, en cuyo caso representan "tiempo real" en microsegundos, o ser asignados explícitamente por aplicaciones cliente. Las aplicaciones que necesitan evitar colisiones deben generar marcas de tiempo únicas por sí mismas. Las diferentes versiones de una celda se almacenan en orden de marca de tiempo decreciente, de modo que las versiones más recientes se pueden leer primero.

API

La API de Bigtable proporciona funciones para crear y eliminar tablas y familias de columnas. También proporciona funciones para cambiar los metadatos de grupos, tablas y familias de columnas, como los derechos de control de acceso.

Bloques de construcción

Bigtable se basa en varias otras piezas de la infraestructura de Google. Bigtable usa el sistema de archivos de Google distribuido (GFS) para almacenar archivos de registro y datos. Un clúster de Bigtable generalmente opera en un grupo compartido de máquinas que ejecutan una amplia variedad de otras aplicaciones distribuidas, y los procesos de Bigtable a menudo comparten las mismas máquinas con procesos de otras aplicaciones. Bigtable depende de un sistema de administración de clústeres para programar trabajos, administrar recursos en máquinas compartidas, lidiar con fallas de máquinas y monitorear el estado de las máquinas. También, Bigtable se basa en un servicio de bloqueo distribuido persistente y de alta disponibilidad llamado Chubby. Un servicio Chubby consta de cinco réplicas activas, una de las cuales se elige para ser la maestra y atender activamente las solicitudes.

Implementación

La implementación de Bigtable tiene tres componentes principales: una biblioteca que está vinculada a cada cliente, un servidor maestro y muchos servidores de tabletas. Los servidores de tabletas se pueden agregar (o eliminar) dinámicamente de un clúster para adaptarse a los cambios en las cargas de trabajo. El maestro es responsable de asignar tabletas a servidores de tabletas, detectar la adición y el vencimiento de servidores de tabletas, equilibrar la carga del servidor de tabletas y la recolección de basura de archivos en GFS. Además, maneja los cambios de esquema, como la creación de familias de tablas y columnas.

Ubicación de las tabletas

El primer nivel es un archivo almacenado en Chubby que contiene la ubicación de la tableta raíz. La tableta raíz contiene la ubicación de todas las tabletas en una tabla especial de METADATOS. Cada tableta METADATOS contiene la ubicación de un conjunto de tabletas de usuario. La tableta raíz es solo la primera tableta en la tabla de METADATOS, pero se trata de manera especial (nunca se divide) para garantizar que la jerarquía de ubicación de la tableta no tenga más de tres niveles.

Asignación de tabletas

Cada tableta se asigna a un servidor de tableta a la vez. El maestro realiza un seguimiento del conjunto de servidores de tabletas en vivo y la asignación actual de tabletas a servidores de tabletas, incluidas las tabletas que no están asignadas. Cuando una tableta no está asignada y hay disponible un servidor de tabletas con suficiente espacio para la tableta, el maestro asigna la tableta enviando una solicitud de carga de tabletas al servidor de tabletas. El maestro es responsable de detectar cuándo un servidor de tabletas ya no está atendiendo sus tabletas y de reasignar esas tabletas lo antes posible. Para detectar cuándo un servidor de tabletas ya no está atendiendo a sus tabletas, el maestro pregunta periódicamente a cada servidor de tabletas por el estado de su bloqueo.

Refinamientos

- Los clientes pueden agrupar varias familias de columnas en un grupo de localidad. Se genera una SSTable separada para cada grupo de localidad en cada tableta.
- Los clientes pueden controlar si se comprimen o no las SSTables para un grupo de localidad y, de ser así, qué formato de compresión se utiliza. El formato de compresión especificado por el usuario se aplica a cada bloque SSTable. Se pierde algo de espacio al comprimir cada bloque por separado, nos beneficiamos porque se pueden leer pequeñas porciones de una SSTable sin descomprimir todo el archivo.
- Para mejorar el rendimiento de lectura, los servidores de tabletas utilizan dos niveles de almacenamiento en caché. Scan Cache es un caché de nivel superior que almacena en caché los pares clave-valor devueltos por la interfaz SSTable al código del servidor de la tableta. Block Cache es un caché de nivel inferior que almacena en caché bloques SSTables que se leyeron desde GFS.
- Se reduce la cantidad de accesos al permitir que los clientes especifiquen que se deben crear filtros Bloom para SSTables en un grupo de localidad particular.

Un filtro Bloom nos permite preguntar si una SSTable podría contener algún dato para un par específico de fila/columna.

- Se agregan mutaciones a un solo registro de confirmación por servidor de tableta, mezclando mutaciones para diferentes tabletas en el mismo archivo de registro físico. El uso de un registro proporciona importantes ventajas de rendimiento durante el funcionamiento normal, pero complica la recuperación. Cuando un servidor de tabletas muere, las tabletas que sirvió se moverán a una gran cantidad de otros servidores de tabletas: cada servidor normalmente carga una pequeña cantidad de las tabletas del servidor original.
- Si el maestro mueve una tableta de un servidor de tabletas a otro, el servidor de tabletas de origen primero realiza una compactación menor en esa tableta. Esta compactación reduce el tiempo de recuperación al reducir la cantidad de estado sin compactar en el registro de confirmación del servidor de la tableta.

Aplicaciones Reales

Para agosto de 2006, habían 388 clústeres de Bigtable que no son de prueba ejecutándose en varios clústeres de máquinas de Google, con un total combinado de alrededor de 24 500 servidores de tabletas. La Tabla 1 muestra una distribución aproximada de servidores de tabletas por clúster. Muchos de estos clústeres se utilizan con fines de desarrollo y, por lo tanto, están inactivos durante períodos significativos. Un grupo de 14 clústeres ocupados con 8069 servidores de tabletas en total vio un volumen agregado de más de 1,2 millones de solicitudes por segundo, con un tráfico RPC entrante de aproximadamente 741 MB/s y un tráfico RPC saliente de aproximadamente 16 GB/s. La Tabla 2 proporciona algunos datos sobre algunas de las tablas actualmente en uso. Algunas tablas almacenan datos que se entregan a los usuarios, mientras que otras almacenan datos para el procesamiento por lotes; las tablas varían ampliamente en tamaño total, tamaño de celda promedio, porcentaje de datos servidos desde la memoria y complejidad del esquema de la tabla. En el resto de esta sección, describimos brevemente cómo tres equipos de productos usan Bigtable. Si estos datos son de hace aproximadamente 26 años, no cabe duda que estas tecnologías están cada vez más presentes en nuestra vida cotidiana y en las aplicaciones de nuestro uso diario.