Bigtable

Que es Bigtable

Bigtable está diseñado para escalar de forma fiable a petabytes de datos y miles de máquinas. Bigtable es utilizado por más de sesenta productos y proyectos de Google, incluyendo Google Analytics, Google Finance, Orkut, Búsqueda personalizada, Writely y Google Earth.

Objetivos de Bigtable

- Amplia aplicabilidad, escalabilidad, alto rendimiento y alta disponibilidad.

Data model

Un Bigtable es un mapa ordenado multidimensional disperso, distribuido y persistente. Es indexado por una clave de fila, una colunma y una marca de tiempo:

```
(row:string, column:string, time:int64) → String
```

Se estableció que el modelo de datos después de examinar una variedad de usos potenciales de un sistema a una Bigtable y como ejemplo concreto se impulsó alguna de las decisiones de diseño que se quisieron para mantener una copia de una gran colección de páginas web e información que podría ser relacionada y utilizada en muchos proyectos diferentes.

ROWS

Son cadenas arbitrarias (actualmente de hasta 64KB de tamaño, aunque 10-100 bytes es un tamaño típico para la mayoría de nuestros usuarios). Cada lectura o escritura de datos bajo una sola clave de fila es atómica, una decisión de diseño que facilita a los clientes razonar sobre el comportamiento del sistema en presencia de actualizaciones simultáneas a la misma fila. Bigtable mantiene los datos en orden lexicográfico por clave de fila. El rango de fila de una tabla se divide dinámicamente. Cada rango de fila se llama *tablet*.

Column Families

Las claves 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. Una clave de columna se nombra usando la siguiente sintaxis: ***family:qualifier***. Los nombres de las columnas deben ser imprimibles, pero los calificadores pueden ser cadenas arbitrarias.

Timestamps

Cada celda en un Bigtable puede contener múltiples versiones de los mismos datos; estas versiones están indexadas por marca de tiempo. Las marcas de tiempo grandes son enteros de 64 bits. Pueden ser asignados por Bigtable, en cuyo caso representan "tiempo real" en microsegundos.

API

proporciona funciones para crear y eliminar tablas y familias de columnas. También proporciona funciones para cambiar los metadatos de clúster, tabla y familia de columnas, como los derechos de control de acceso. Las aplicaciones cliente pueden escribir o eliminar valores en Bigtable, buscar valores de filas individuales o iterar sobre un subconjunto de datos en una tabla.

Building Blocks

Bigtable se basa en varias otras piezas de la infraestructura de Google. Bigtable utiliza el sistema de archivos distribuido de Google para almacenar registros y archivos de datos. Un clúster Bigtable normalmente opera en un grupo compartido de máquinas que ejecutan una amplia variedad de otras aplicaciones distribuidas, y los procesos Bigtable a menudo comparten las mismas máquinas con procesos de otras aplicaciones.

Implementation

Tablet location: Utilizamos una jerarquía de tres niveles análoga a la de un árbol B+ para almacenar la información de ubicación de la *tablet*. El primer nivel es un archivo almacenado en Chubby que contiene la ubicación de la tableta raíz. **Tablet assignment:** Cada tableta se asigna a un servidor de tabletas 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, incluyendo qué tabletas no están asignadas. **Tablet serving:** El estado persistente de una tableta se almacena en GFS, como se ilustra en la Figura 5. Las actualizaciones se comprometen a un registro de confirmación que almacena los registros de rehacer. **Compactions:** A medida que se ejecutan las operaciones de escritura, el tamaño del memtable aumenta.

Refinements

Locality groups: 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. **Compression:** Los clientes pueden controlar si las SSTables para un grupo local están comprimidas o no, y si es así, qué formato de compresión se utiliza. **Bloom filters:** Como se describe en la Sección 5.3, una operación de lectura tiene que leer de todas las SSTables que componen el estado de una tableta. **Commit-log implementation:** Si mantuviéramos el registro de confirmación para cada tableta en un archivo de registro separado, un gran número de archivos se escribirían simultáneamente en GFS. **Speeding up tablet recovery:** Si el maestro mueve una tableta de un servidor de tableta a otro, el servidor de tableta de origen primero hace una compactación menor en esa tableta. **Exploiting immutability:** Además de las caché de SSTable, varias otras partes del sistema Bigtable se han simplificado por el hecho de que todas las SSTables que generamos son inmutables.

Performance Evaluation

Se evaluaron elementos como el rendimiento y la escalabilidad de la Bigtable. Para el rendimiento de una tablet-server se obtuvieron lecturas aleatorias lentas. En la escalibilidad, el rendimiento agregado aumenta drásticamente, en más de un factor de cien, a medida que aumentamos el número de servidores de tabletas en el sistema de 1 a 500. Este comportamiento ocurre porque el cuello de botella en el rendimiento de este punto de referencia es la CPU del servidor de tableta individual. Sin embargo, el rendimiento no aumenta linealmente.