

Resumen 4

Fecha de entrega: 21/09/22 7:25 pm

- Max Richard Lee Chung - 2019185076

Introduction

Subsistema de índices DocumentDB. El subsistema de índices ocupa soportar índices automáticos de documentos sin tener que usar esquemas o índices secundarios, lenguajes de consultas DocumentDB, consultas consistentes a tiempo real para procesar documentos pesados y multiservicios en recursos extremadamente frugales en donde se mantenga un rendimiento alto y efectivo.

Está basado en el modelo de datos JSON y lenguaje de JavaScript en el mismo motor de bases de datos. Capacidades:

- Soporta consultas jerárquicas y relacionales.
- El motor está optimizado para consultas de documentos de escrituras grandes. De forma predeterminada, el motor de la base le proporciona a todos los datos automáticamente un índice.
- Ejecución transaccional por medio de procedimientos almacenados y disparadores (triggers).
- Ofrece la opción de escoger o personalizar el nivel de consistencia.
- Todas las máquinas y recursos son abstractos para todos los usuarios.
- Permite que la habilidad de escalar elásticamente el rendimiento y almacenamiento con costos eficientes.

Un usuario de DocumentDB inicia con una cuenta proporcionada. Una cuenta maneja una o más bases de datos de DocumentDB. A su vez, gestiona los usuarios, permisos y colecciones. Una colección es un contenedor independiente del esquema de datos arbitrarios generados por usuarios, además gestiona los procedimientos almacenados, disparadores, funciones definidas de usuarios y archivos adjuntos. Todos las cuentas, bases de datos, usuarios, colecciones, entre otros son recursos representados como un documento JSON. Los desarrolladores pueden interactuar con los recursos en HTTP usando la sintaxis estándar del CRUD, consultas y procedimientos almacenados. Se pueden crear particiones de los nuevos recursos para tener una escalabilidad elástica. Cada partición tiene un único sistema de imagen y una réplica set.

El servicio se encuentra desplegado en múltiples regiones Azure. El servicio se despliega y gestiona en un clústers en máquinas locales, cada una con su memoria dedicada. Luego de ser desplegado, se manifiesta como una red de máquinas (federaciones) el cual se extiende en una o más clústers. Las réplicas correspondientes a las particiones de recursos se colocan y equilibran la carga entre las máquinas de la federación. El motor de la base de datos DocumentDB, a su vez, consta de componentes que incluyen la máquina de estado replicada (RSM) para la coordinación, el tiempo de ejecución del lenguaje JavaScript, el procesador de consultas y los subsistemas de almacenamiento e indexación responsables del almacenamiento transaccional y la indexación de documentos.

Schema agnostic indexing

El esquema del documento describe la estructura y el tipo de sistema del documento independiente de la instancia del documento. La simplicidad de la gramática de JSON es una de las razones de su adopción a pesar de la poca especificación del esquema. Esto

permite procesar documentos sin asumir el tipo ya que cada documento puede variar. El sistema opera a nivel de la gramática JSON.

La técnica que ayuda a desdibujar el límite entre el esquema de los documentos JSON y sus valores de instancia es representar los documentos como árboles. Esto permite normalizar la estructura y los valores instanciados en un concepto unificador de una estructura de ruta codificada dinámicamente. Cada etiqueta es un nodo del árbol y el contenido de cada etiqueta es un nodo hijo.

Uno de los requisitos principales de la indexación automática de documentos es garantizar que el costo de indexar y consultar un documento con una estructura profundamente anidada sea el mismo que el de un documento JSON plano que consta de pares clave-valor de solo un nivel de profundidad. Por lo tanto, una representación de ruta normalizada es la base sobre la cual se construyen tanto la indexación automática como los subsistemas de consulta. Hay dos asignaciones posibles de documentos y rutas: mapeo de índice hacia adelante (id, ruta) y mapeo de índice invertido (ruta, id). Dado que el lenguaje de consulta DocumentDB opera sobre las rutas de los árboles de documentos, el índice invertido es una representación muy eficiente. El índice del árbol crece dependiendo de cuántos documentos nuevos sean añadidos.

Logical index organization

La representación lógica del índice se puede ver como un conjunto ordenado de tuplas clave-valor, cada una de las cuales se denomina entrada de índice. Un término representa una ruta única en el árbol de índices. La dirección de la ruta tiene compensaciones asociadas que incluyen costo de almacenamiento, medido por la cantidad de rutas generadas para la estructura del índice, costo de mantenimiento de la indexación: recursos consumidos para el mantenimiento del índice correspondiente a un lote de escrituras de documentos, costo de las consultas de búsqueda. La información de ruta de codificación se separa en tres segmentos porque principalmente los archivos JSON se separan en la raíz, llave y valor, además para distinguir los caminos y mantener el costo de almacenamiento bajo. Los esquema de codificación de ruta de avance parcial implica el análisis del documento desde la raíz y la selección de tres nodos de sufijo sucesivamente para producir un camino distinto que consta de exactamente tres segmentos (este esquema es utilizado para hacer rango e indexación espacial) y la codificación de un segmento se realiza de forma diferente para etiquetas numéricas y no numéricas. El esquema de codificación de ruta inversa parcial es similar al anterior, en el que selecciona tres nodos de sufijo sucesivamente para producir un camino distinto que consta de exactamente tres segmentos.

Los mapas de bits como listas de publicaciones capturan los id de todos los documentos que contienen dicho término. Existen dos métodos para mantener captruas dinámicas, rápidas y compactas:

- **Particionar una lista de publicaciones:** Se realiza una partición de las listas de publicaciones en entradas de publicaciones y permite una mejor visualización y cálculo de las páginas de un árbol B+. Una entrada de publicaciones es un conjunto ordenado de una o más palabras de publicaciones de documentos dentro de un rango de identificación de documento específico.
- **Codificación dinámica de entradas de publicación:** Dependiendo de la distribución, las palabras de publicación dentro de una entrada de publicación se codifican dinámicamente mediante un conjunto de esquemas de codificación que incluyen varios esquemas de codificación de mapa de bits.

La personalización de los índices se puede realizar siguiendo los siguientes aspectos:

- Incluir/Excluir documentos y rutas hacia/desde el índice
- Configuración de varios tipos de índice (hash, rango, espacial o texto)
- Configuración de los modos de actualización de índices (consistente, perezoso o consistencia eventual y ninguno o sin índices)

Physical index organization (in-memory and on-disk)

El mantenimiento debe de seguir las siguientes restricciones:

- El rendimiento de la actualización del índice debe ser una función de la tasa de llegada de las rutas indexables.
- La actualización del índice no puede asumir ninguna localidad de ruta entre los documentos entrantes.
- La actualización del índice para los documentos de una colección se debe realizar dentro del presupuesto de CPU, memoria e IOPS asignado por colección de DocumentDB.
- Cada actualización de índice debe tener la menor amplificación de escritura posible.
- Cada actualización de índice debe incurrir en una amplificación de lectura mínima.

The Bw-Tree for DocumentDB

El BwTree utiliza actualizaciones de memoria sin asegurar y almacenamiento estructurado de registros para la persistencia. Aprovecha dos tendencias en el hardware moderno: procesadores multinúcleo con jerarquía de memoria/caché de varios niveles y SSD basados en memoria flash con lecturas aleatorias rápidas. Un seguro de escritura no permite lecturas o escrituras simultáneas en la página, por lo tanto, subprocesos que necesitan realizar operaciones conflictivas en el bloque de página. Todas las referencias de página usan un nivel de indirección a través de la tabla de mapeo.

Index Updates

- Consistente: La actualización de índice se haría como una actualización de modificación de lectura.
- Perezoso: El mantenimiento de índices con el modo de indexación perezosa se realiza en segundo plano, de forma asíncrona con las escrituras entrantes.

Index Replication and Recovery

- Replicación: La réplica principal que recibe las escrituras analiza el documento y genera los términos. Lo aplica a su instancia de base de datos y envía el flujo que contiene los términos a los secundarios. Cada secundario aplica los términos a su instancia de base de datos local. Usa método failover.
- Recuperación: Primero, se debe de recuperar el árbol bw. Segunda, el subsistema de indexación de la base de datos vuelve a indexar los documentos con actualizaciones y los inserta en el árbol.

Related Commercial systems

Actualmente hay dos tipos comerciales NoSQL, los cuales son sistemas de almacenamiento NoSQL no orientados a documentos y bases dedatos de documentos.

Referencia

- Microsoft Corporation. Schema-Agnostic Indexing with Azure DocumentDB.