

## **Schema-Agnostic Indexing with Azure DocumentDB**

Azure DocumentDB es un servicio de base de datos distribuida que administra documentos Json.

### **Capacidades:**

- Basado en JSON y lenguaje JavaScript, tiene un enfoque de “desajuste de impedancia” entre lenguajes lo cual provoca:
- Se pueden hacer consultas relacionales y jerárquicas. Este tipo de lenguaje de consulta es de tipo SQL.
- Su motor de BD automáticamente indexa los documentos y no necesita ningún esquema o índice secundario. Se ejecuta de manera transaccional la lógica de la aplicación, se ejecuta directamente dentro del motor de la base de datos del DocumentDB.
- Su coherencia es bien definida y ajustable.
- La administración de máquinas y recursos se abstraen de los usuarios. Se puede escalar el rendimiento y almacenamiento

Una colección de DocumentDB es un contenedor independiente del esquema de documentos generados por el usuario. Cada partición tiene una imagen única del sistema para los recursos que administra. En donde se puede interactuar con la partición con URI y estas son altamente disponibles gracias a las réplicas.

Cada réplica aloja una instancia de DocumentDB , que gestiona los recursos y el índice asociado, estas réplicas se conservan entre las instancias del motor de BD dentro del conjunto de réplicas.

### **Objetivos de diseño para la indexación**

- Los documentos se indexan automáticamente.
- Se pueden hacer configuraciones de almacenamiento y rendimiento.
- Las consultas son jerárquicas y relacionales son eficientes.
- Las consultas deben ser consistentes ante un volumen sostenido de escritura de documentos.
- Las actualizaciones de índice deben realizarse dentro del presupuesto de recursos.

### **INDEXACIÓN AGNÓTICA DE ESQUEMA**

Para que el motor de BD sea independiente del esquema:

- El uso de JSON por su simplicidad y su falta de especificación de esquema. Permite que los documentos dentro de una colección de Documentos DB varíen en el esquema. Operen dentro de la gramática de JSON lo cual permite indexar documentos automáticamente sin necesidad de esquemas o índices secundarios.
- **Documentos como árboles:** La representación de documentos JSON como árboles a su vez normaliza tanto la estructura como los valores de instancia en los documentos.
- **Índice como documento:** Al ser la indexación automática, esa indexación se va para un árbol de documentos y cada actualización de estos se conduce a una actualización de la estructura del índice.
- **Consultas de DocumentosDB:** Operan directamente contra la representación del árbol. Se puede referir a propiedades en documentos JSON en cualquier

profundidad.

### ORGANIZACIÓN DEL ÍNDICE LÓGICO

El índice es una unión de todos los documentos y también se representa como un árbol. Cada nodo contiene una lista de ID de documentos que hacen match con la etiqueta. Esta organización se puede interpretar como un conjunto de tuplas clave-valor que se llama entrada de índice. Se puede dirigir un camino como términos que representan una ruta única con valores de posición y etiqueta.

Se predeterminan tres segmentos pues los JSON en muchos casos tienen: raíz, clave y valor y además gracias a esos tres segmentos se pueden distinguir varias rutas entre sí pero manteniendo su tamaño reducido (beneficiando el costo y rendimiento).

### PERSONALIZACIÓN DEL ÍNDICE

- Se puede configurar los valores predeterminados, por ejemplo:
- Incluir o excluir documentos y sus rutas de índice.
- Configurar varios tipos de índice y configurar su actualización.

### ORGANIZACIÓN DEL ÍNDICE FÍSICO

**La estructura de datos de escritura:** El rendimiento de una actualización del índice tiene que ver con la tasa de llegada de las rutas indexadas. Las actualizaciones no deben asumir ninguna localidad de ruta entre los documentos entrantes, para los documentos de una colección se realiza dentro del presupuesto de CPU, memoria e IOPS. Debe tener la menor amplificación de escritura posible. Debería incurrir en una amplificación de lectura mínima.

**Árbol BW para DocumentosDB:** Este árbol utiliza actualizaciones en memoria sin bloqueo y almacenamiento estructurado de registros para la persistencia. Aprovecha los procesadores multinúcleo con jerarquía de memoria/caché de varios niveles y los SSD basados en memoria flash con lecturas aleatorias rápidas. Este árbol ofrece escrituras rápidas y admite actualizaciones incrementales para que se utilice el ancho de banda de escritura de almacenamiento completo para las actualizaciones de índice.

**Alta concurrencia en el árbol:** Este funciona sin bloqueos, lo que permite mucha concurrencia. La modificación de una página se realiza agregando un registro delta en la parte superior de la parte existente de la página y esto requiere que la ubicación inicial de la página cambie después de cada actualización. Las referencias a páginas utilizan un nivel de indirección por medio de la tabla de mapeo y estas proporcionan la traducción del ID de la página lógica a la ubicación de la página física.

La organización del almacenamiento optimizado del árbol: El almacenamiento está organizado de forma estructurada por registros y sus páginas se vacían de manera incremental.

### ACTUALIZACIONES DEL ÍNDICE

**Indexación coherente:** Se realiza de manera sincrónica con las escrituras de documentos entrantes, durante la actualización del índice. **Indexación perezosa:** Se realiza en segundo plano de forma asincrónica con las escrituras entrantes.

**Análisis de documentos:** Toma el contenido del documento correspondiente a una marca de tiempo lógica cuando se actualizó por última vez. La política de indexación produce un conjunto de caminos. Este analizador proporciona operadores básicos para agregar dos instancias de documentos y además estos operadores proporcionan la base para el mantenimiento de índices.

**Actualizaciones de índices eficientes:** Se puede actualizar parcialmente cualquier registro sin acceder al valor existente de la clave y sin requerir ninguna coordinación entre múltiples llamantes.

**Actualizaciones de índice diferido con mapa de bits de invalidación:** Es un mapa de bits que representa los identificadores de documentos correspondientes a las imágenes de documentos eliminados y reemplazados, además un recuento que representa el número de veces que se ha registrado la actualización del documento.

**Replicación y recuperación de índices:** El modelo de gestión lo que hace es dividir los presupuestos de los recursos entre la réplica principal y secundaria de manera no uniforme. La principal se lleva la mayor parte de la escritura y responsabilidades y las secundarias atienden las lecturas y consultas. Con respecto a la recuperación de índices, el árbol expone una API que permite que la capa superior indique que todas las actualizaciones de índice deben estabilizarse

#### **GOBERNANZA DE LOS RECURSOS DEL ÍNDICE**

Para ofrecer un modelo normalizado para la contabilidad, la asignación y consumo de los recursos del sistema, etc. Se hace por medio de una moneda basada en una tasa abstracta llamada: Unidad de solicitud que encapsula una porción de CPU, memoria e IOPS.

**Recursos de la CPU:** Todos los subsistemas dentro de DocumentDB están diseñados para ser completamente asíncronos y escritos para nunca bloquear un subproceso, lo que a su vez permite que la cantidad de subprocesos en el grupo de subprocesos permanezca baja.

**Recursos de memoria:** El límite de memoria para Bw-Tree se mantiene intercambiando páginas residentes en caché de memoria cada vez que se detecta presión de memoria. La instancia de motor de BD y sus componentes, operan dentro de un presupuesto de memoria.

**Recursos de IOPS de almacenamiento:** El árbol Bw mantiene un promedio móvil de uso de IOPS a lo largo del tiempo. Justo antes de emitir una E/S, verifica si la E/S conduciría a una violación del presupuesto de IOPS, si lo hace se retrasa y luego lo vuelve a intentar.

#### **SISTEMAS COMERCIALES RELACIONADOS NOSQL, EXISTEN DOS TIPOS:**

- Sistemas de almacenamiento NoSQL no orientados a documentos que están diseñados y operados como servicios en la nube de múltiples usuarios con SLA.
- Bases de datos de documentos, diseñadas como sistemas de inquilino único que se ejecutan en las instalaciones o en máquinas virtuales dedicadas en la nube.

Con respecto a DocumentDB no cabe en ninguno de los dos anteriores, pues proporciona un motor de base de datos gobernado por recursos y agnóstico del esquema para proporcionar indexación automática. Tasa sostenidas y rápidas de actualización que permiten las consultas.