TAREA 3

Tema: Apache Cassandra

Materia: Big Data

Integrantes:

- Francisco Pintos
- Lucas Candia
- Matías Sánchez

San Lorenzo – Paraguay

2023

Introducción

Apache Cassandra es una base de datos NoSQL altamente escalable y distribuida diseñada para manejar grandes cantidades de datos en muchos servidores de commodities, proporcionando alta disponibilidad sin un solo punto de fallo. Originalmente desarrollada por Facebook para su función de búsqueda de bandeja de entrada, se lanzó como un proyecto de código abierto en 2008 y se convirtió en un proyecto de alto nivel de Apache en 2010.. Cassandra está diseñada para manejar grandes cantidades de datos a través de muchos servidores de hardware básicos, proporcionando alta disponibilidad sin un único punto de fallo.

¿Qué es Apache Cassandra?

Es una base de datos distribuida NoSQL. Por diseño, las bases de datos NoSQL son livianas, de código abierto, no relacionales y ampliamente distribuidas. Entre sus puntos fuertes se encuentran la escalabilidad horizontal, las arquitecturas distribuidas y un enfoque flexible para la definición de esquemas

1. Escalabilidad: Cassandra puede manejar grandes cantidades de datos en muchas máquinas físicas. A medida que aumenta la demanda, Cassandra puede expandirse linealmente simplemente añadiendo más máquinas al cluster.

2. Alta disponibilidad: Cassandra ofrece una alta disponibilidad a través de su arquitectura de distribución peer-to-peer. No hay un único punto de fallo ya que todos los nodos son iguales y cada uno puede servir cualquier solicitud.

3. Tolerancia a fallos: Cassandra es altamente resistente a los fallos. En caso de que un nodo se caiga, las solicitudes pueden ser servidas por otros nodos en el cluster. Además, los datos se replican en varios nodos para asegurar que los datos no se pierdan en caso de fallo.

4. Consistencia eventual: A diferencia de las bases de datos tradicionales que garantizan la consistencia inmediata, Cassandra ofrece consistencia eventual. Esto significa que los datos pueden no ser consistentes en todos los nodos al mismo tiempo, pero se volverán consistentes con el tiempo.

Modelo de Datos

El modelo de datos de Cassandra es una extensión del modelo BigTable y se organiza alrededor del concepto de columnas y familias de columnas. Las columnas son

pares de clave-valor, y una familia de columnas es un contenedor de un conjunto de columnas. Una familia de columnas en Cassandra es similar a una tabla en una base de datos relacional. Sin embargo, a diferencia de las tablas en una base de datos relacional, todas las filas en una familia de columnas de Cassandra no necesitan tener el mismo conjunto de columnas.

Además, Cassandra introduce el concepto de espacios de nombres, que es similar a las bases de datos en los sistemas de gestión de bases de datos relacionales. Un espacio de nombres es un contenedor de familias de columnas.

Replicación de Datos

La replicación es una característica fundamental en Apache Cassandra y se utiliza para garantizar la disponibilidad y la resistencia a los fallos. En un clúster de Cassandra, los datos se replican en varios nodos para garantizar que la información esté disponible incluso si algunos nodos fallan. Los usuarios pueden configurar la cantidad de réplicas y cómo se distribuyen.

Lenguaje de Consulta de Cassandra (CQL)

CQL, o Cassandra Query Language, es un lenguaje de consulta que se utiliza para interactuar con la base de datos de Cassandra. CQL es similar a SQL en términos de sintaxis y uso, lo que facilita su aprendizaje y uso para las personas familiarizadas con SQL. Sin embargo, debido a las diferencias entre los modelos de datos relacional y NoSQL, hay algunas diferencias clave en cómo se utilizan SQL y CQL.

Consistencia de Datos

Apache Cassandra utiliza un modelo de consistencia eventual, que es una estrategia para manejar la consistencia en un sistema distribuido. En un sistema con consistencia eventual, los cambios realizados en un nodo pueden tardar un tiempo en replicarse en otros nodos. Esto puede dar lugar a una visión temporalmente inconsistente de los datos en diferentes nodos, pero garantiza que, finalmente, todos los nodos tendrán la misma versión de los datos.

En Cassandra, la consistencia se configura en tiempo de ejecución en lugar de en tiempo de escritura. Esto significa que los desarrolladores pueden escoger el nivel de consistencia que mejor se ajuste a las necesidades de su aplicación. Los niveles de consistencia varían desde 'ONE', donde solo se requiere que un nodo responda para considerar que la operación fue exitosa, hasta 'ALL', donde todos los nodos en el clúster deben responder.

Escalabilidad y Rendimiento

Apache Cassandra está diseñada para escalar linealmente, lo que significa que se puede aumentar la capacidad de procesamiento simplemente añadiendo más nodos al clúster. Esto permite a Cassandra manejar eficientemente grandes volúmenes de datos y mantener un rendimiento constante, incluso cuando se añaden o se eliminan nodos.

El rendimiento de Cassandra es predecible, lo que significa que se puede anticipar cómo se comportará el sistema a medida que crece. Esto es una gran ventaja para las organizaciones que necesitan planificar su crecimiento y garantizar que su infraestructura pueda manejar la carga de trabajo prevista.

Objetivo de la herramienta. ¿Cuál es el problema resuelto por ella?

Las bases de datos NoSQL permiten una organización y un análisis rápidos y ad hoc de tipos de datos dispares y de gran volumen. Eso se ha vuelto más importante en los últimos años, con la llegada de Big Data y la necesidad de escalar rápidamente las bases de datos en la nube. Cassandra se encuentra entre las bases de datos NoSQL que han abordado las limitaciones de las tecnologías de gestión de datos anteriores, como las bases de datos SQL. Apache Cassandra está diseñada para manejar grandes volúmenes de datos y ofrecer una alta disponibilidad y rendimiento. Su objetivo principal es resolver los problemas relacionados con el almacenamiento y acceso eficiente de grandes conjuntos de datos estructurados y no estructurados en entornos de alta demanda.

¿Con qué otras herramientas de Big Data puede integrarse?

Cassandra se integra con otras herramientas de Big Data, como:

- Apache Hadoop: Cassandra se puede usar como fuente o destino de datos para aplicaciones basadas en Hadoop, como Apache Hive o Apache Pig.

- Apache Spark: Cassandra se integra con Spark, lo que permite realizar análisis y consultas en tiempo real sobre los datos almacenados en Cassandra.

- Apache Kafka: Cassandra puede recibir datos de Kafka para su posterior almacenamiento y procesamiento.

Arquitectura. ¿Qué componentes tiene?

Apache Cassandra adopta una arquitectura de igual a igual (peer-to-peer), lo que significa que todos los nodos en un clúster de Cassandra son iguales. No existe un nodo maestro. Cada nodo puede aceptar lecturas y escrituras, independientemente de los demás nodos en el clúster. Esta característica elimina el riesgo de un único punto de fallo y permite una alta disponibilidad y resistencia a fallos.

Los nodos en un clúster de Cassandra se organizan en un anillo o círculo, y cada nodo es responsable de una cierta porción de los datos. Esta estructura es una implementación del algoritmo de consistente hashing, lo que permite una distribución equitativa de los datos y simplifica el proceso de añadir y eliminar nodos.

Algunos conceptos clave en Cassandra

- **Nodos:** Cassandra está diseñado para funcionar en un clúster de nodos distribuidos en múltiples máquinas. Cada nodo es responsable de un rango de datos y se comunica con otros nodos para garantizar la coherencia y la replicación de los datos.
- **Anillo de particionamiento:** Los datos se distribuyen en el clúster utilizando un algoritmo de particionamiento basado en hash, lo que permite un balance de carga eficiente y una alta escalabilidad.
- **Replicación:** Cassandra ofrece un modelo de replicación configurable que permite distribuir copias de los datos en varios nodos, proporcionando redundancia y tolerancia a fallos.

- **Consistencia:** Cassandra ofrece diferentes niveles de consistencia, desde consistencia fuerte hasta consistencia eventual, permitiendo ajustar el equilibrio entre rendimiento y coherencia de los datos.
- **Columna:** La unidad más pequeña de datos en Cassandra. Cada columna consta de un nombre de columna, un valor de columna y una marca de tiempo que indica cuándo se modificó por última vez.
- **Familia de Columnas:** Una familia de columnas es un contenedor de una colección de filas, donde cada fila contiene columnas ordenadas.

- **Espacio de Nombres:** En Cassandra, un espacio de nombres, o keyspace, es un contenedor de alto nivel que contiene familias de columnas.
- **Clúster:** Un conjunto de muchos nodos. Cada cluster puede tener varios nodos, que pueden estar en varias ubicaciones geográficas.

Todos los modelos de licenciamiento soportado. Costos. Comparación entre cada tipo de licencia.

En cuanto al licenciamiento, Apache Cassandra se distribuye bajo la licencia Apache 2.0, que es una licencia de código abierto y gratuita. Los costos asociados con Cassandra generalmente se relacionan con el hardware, la infraestructura y el soporte técnico necesarios para implementar y mantener un clúster de Cassandra.

Nivel de madurez de la herramienta: años en el mercado. Actividad reciente en la comunidad, frecuencia de publicación de actualizaciones, última actualización. Clientes que usan la herramienta.

Apache Cassandra fue inicialmente desarrollado por Facebook en 2008 y posteriormente donado a la Apache Software Foundation en 2009. Desde entonces, ha experimentado un crecimiento significativo y ha sido adoptado por numerosas organizaciones en todo el mundo. La comunidad de Cassandra es activa y hay una frecuencia constante de actualizaciones y nuevas versiones, lo que indica un nivel de actividad y desarrollo continuo en la herramienta.

- Clientes destacados:

Algunos de los clientes que utilizan Apache Cassandra incluyen empresas como Netflix, Apple, Spotify, Uber y eBay, entre otros. Estas organizaciones aprovechan las características de escalabilidad, rendimiento y disponibilidad de Cassandra para gestionar y analizar grandes volúmenes de datos en tiempo real.

Casos de Éxito de Apache Cassandra

Numerosas empresas de renombre han adoptado Apache Cassandra para sus aplicaciones, incluyendo Apple, Netflix, Uber y Facebook. Estas empresas manejan enormes cantidades de datos distribuidos y necesitan una base de datos que pueda manejar este volumen de datos de manera eficiente.

Apple, por ejemplo, utiliza Cassandra para su servicio Apple Music. Con más de 60 millones de canciones y millones de usuarios en todo el mundo, Apple necesitaba una base de datos que pudiera manejar un gran volumen de datos y proporcionar un rendimiento constante y predecible. Cassandra ha demostrado ser capaz de satisfacer estas necesidades.

Aplicaciones posibles

- Aplicaciones de análisis de datos en tiempo real.
- Sistemas de gestión de contenido y publicación.
- Sistemas de seguimiento y análisis de registros (logs).
- Aplicaciones de IoT y recolección de datos en tiempo real.
- Plataformas de recomendación y personalización.
- Sistemas de mensajería y chat en tiempo real.

Mejores Prácticas y Consejos para Utilizar Apache Cassandra

Al igual que con cualquier tecnología, hay varias mejores prácticas y consejos que pueden ayudar a maximizar el rendimiento y la eficiencia de Apache Cassandra.

- Diseñar el modelo de datos en función de las consultas de la aplicación: En Cassandra, es importante diseñar el modelo de datos en función de las consultas

que se ejecutarán. A diferencia de las bases de datos relacionales, donde el diseño del modelo de datos puede ser independiente de las consultas, en Cassandra, el diseño del modelo de datos debe estar estrechamente vinculado a las consultas para garantizar un rendimiento óptimo.

- Monitorear y ajustar la configuración del clúster de Cassandra: Es importante monitorear regularmente el clúster de Cassandra y ajustar su configuración según sea necesario para garantizar un rendimiento óptimo. Esto incluye el ajuste de los parámetros de JVM, la configuración de la memoria y la configuración del sistema operativo.

- Implementar medidas de seguridad adecuadas: Como cualquier sistema de gestión de bases de datos, es crucial asegurar que se implementen medidas de seguridad adecuadas en Cassandra. Esto incluye autenticación, autorización y cifrado de datos.

- Planificar la capacidad del clúster y las estrategias de crecimiento*: A medida que las necesidades de almacenamiento y rendimiento evolucionan, es vital planificar la capacidad del clúster y desarrollar estrategias de crecimiento. Es posible que se necesite agregar más nodos al clúster, y tener una estrategia clara de cómo y cuándo hacerlo puede ayudar a garantizar un crecimiento suave y un rendimiento constante.

- Utilizar estrategias de compactación adecuadas: Cassandra utiliza un proceso llamado compactación para fusionar las tablas de SSTable, eliminar las entradas duplicadas y liberar espacio en disco. Hay varias estrategias de compactación disponibles en Cassandra, y es importante seleccionar y utilizar la estrategia de compactación adecuada para optimizar el uso del espacio en disco y mejorar el rendimiento de lectura.

Limitaciones de Apache Cassandra y Cuándo No se Recomienda Su Uso

A pesar de las numerosas ventajas que ofrece, Apache Cassandra no es la solución adecuada para todas las aplicaciones. Hay varias limitaciones y situaciones en las que no se recomendaría su uso.

- Complejidad de modelado de datos: A diferencia de las bases de datos relacionales, que permiten un modelado de datos bastante flexible, Cassandra requiere que se conozcan las consultas de la aplicación antes de comenzar a modelar los datos. Esto puede dificultar el modelado de datos, especialmente para aplicaciones con consultas de datos complejas o variables.

- No es adecuada para aplicaciones que necesitan transacciones ACID: Aunque Cassandra ofrece atomicidad y aislamiento a nivel de fila, no soporta transacciones ACID en múltiples particiones. Por lo tanto, no sería la mejor opción para aplicaciones que requieran este tipo de transacciones.

- No soportan operaciones JOIN: Cassandra no soporta operaciones JOIN, que son comunes en las bases de datos relacionales. Esto significa que las aplicaciones que dependen de estas operaciones necesitarán realizarlas en la capa de aplicación, lo cual puede ser menos eficiente.

- Consistencia eventual: Aunque la consistencia eventual puede ser una ventaja en ciertos casos, también puede ser una desventaja en aplicaciones que requieren consistencia inmediata. En un sistema de consistencia eventual, puede haber un período de tiempo durante el cual las réplicas no están sincronizadas, lo que puede no ser aceptable para algunas aplicaciones.

Aspectos de seguridad en Cassandra

Cassandra ofrece varias características de seguridad, como la autenticación y autorización de usuarios, y el cifrado para la protección de datos tanto en reposo como en tránsito. El control de acceso basado en roles (RBAC) se utiliza para definir los permisos de los usuarios, y la auditoría se puede habilitar para registrar las actividades de los usuarios.

Integración y soporte de lenguajes

Cassandra proporciona soporte para una variedad de lenguajes de programación a través de controladores, incluyendo Java, Python, C++, C#, Node.js, Ruby, y más. Esto permite a los desarrolladores trabajar en el lenguaje de su elección.

Rendimiento de Cassandra

Cassandra es conocida por su rendimiento y capacidad para manejar grandes volúmenes de datos. El rendimiento es predecible y se mantiene incluso a medida que la carga o el volumen de datos aumentan. Para mantener un rendimiento óptimo, Cassandra proporciona varias técnicas de optimización, como la compactación para reducir el uso del espacio en disco, y la caché de filas para acelerar las lecturas.

Conclusión

En resumen, Apache Cassandra es una base de datos NoSQL altamente escalable y distribuida que ofrece alta disponibilidad y rendimiento predecible, lo que la hace adecuada para manejar grandes volúmenes de datos en tiempo real. Aunque Cassandra puede requerir una curva de aprendizaje para los desarrolladores que están acostumbrados a las bases de datos SQL tradicionales, sus ventajas en términos de escalabilidad, rendimiento y disponibilidad a menudo superan este desafío. En última instancia, la elección de usar Cassandra depende de las necesidades específicas del caso de uso.

Fuentes bibliográficas:

- https://cassandra.apache.org/_/cassandra-basics.html
- https://es.wikipedia.org/wiki/Apache_Cassandra
- https://refactorizando.com/cassandra-que-es-cuando-usarla/
- https://es.slideshare.net/zanorte/introduccion-a-cassandra