

Arquitectura

Apache Cassandra - Bases de datos II

Alberto Díaz Álvarez (<alberto.díaz@upm.es>)

Departamento de Sistemas Informáticos

Escuela Técnica superior de Ingeniería de Sistemas Informáticos

License CC BY-NC-SA 4.0

Un poquito de arquitectura

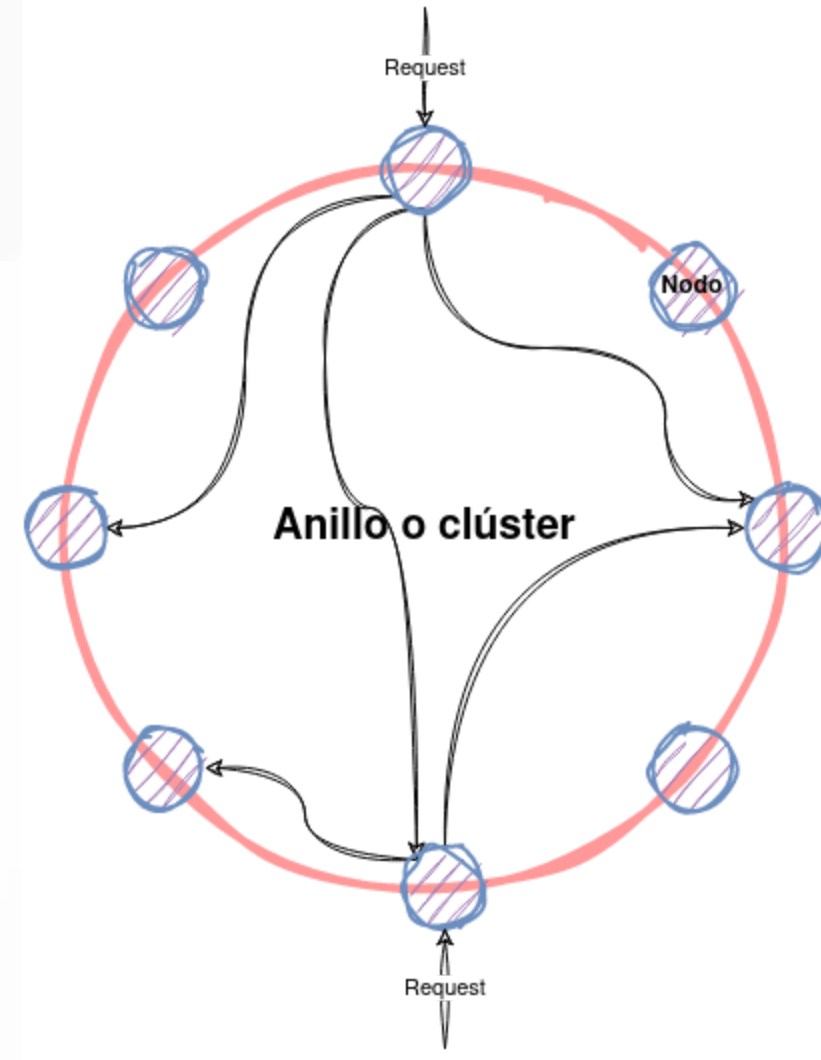
Topología de Apache Cassandra

Arquitectura de sistema distribuido

- Aun así, puede instalarse en una única máquina (o contenedor)

Nodo: Una instancia de Cassandra

- Entidad más pequeña de un **clúster**
- La escalabilidad horizontal surge de añadir más nodos al clúster
- Todos tienen la misma jerarquía
- Contienen una réplicas para diferentes rangos de datos



Comunicación entre nodos

Cassandra utiliza un mecanismo de comunicación denominado *gossip* (cotilleo)

- Comunicación interna para permitir la comunicación dentro de un clúster
- Informa a cada nodo del estado del resto (hasta tres nodos por segundo)
- De esta manera, cada nodo conoce a otros nodos
- Este protocolo ayuda a la descentralización y a la tolerancia a fallo

Los mensajes tienen un formato específico que incluye un número de versión

- La comunicación es muy eficiente
- Permite que cada nodo construya rápidamente una visión general del clúster
 - Nodos caídos, qué tokens se asignan a cada nodo, etcétera

Componentes de un nodo (I)

Algunos de los componentes más importantes de un nodo son los siguientes:

- **Memtable:** Estructura **en memoria** donde se almacenan las escrituras
 - Suele haber una por "tabla" (es un concepto similar al relacional)
 - Eventualmente se vuelcan a disco pasando a ser SSTables
- **SSTable:** Archivo **inmutable** utilizado para la persistencia de datos en disco
 - Según se van volcando a disco, se van compactando en una sola
 - Cada una se compone de varios archivos, algunos de los cuales son:
 - `Data.db`: Los datos reales
 - `Index.db`: El índice de los datos
- **CommitLog:** Archivo de registro de los cambios en un nodo
 - Toda escritura en una Memtable pasa antes por el CommitLog

Más allá del despliegue local

Un clúster de Cassandra puede ser un despliegue en un mismo centro de datos

- En una máquina o en varias máquinas repartidas a lo largo de la misma red

Sin embargo, soporta el despliegue en múltiples centros de datos

- A efectos del sistema cliente, siempre se ve una única instancia de Cassandra
- Internamente, Cassandra se encarga de la replicación de datos entre CDC
- Y de la comunicación entre nodos, independientemente de su localización

Modelo de datos

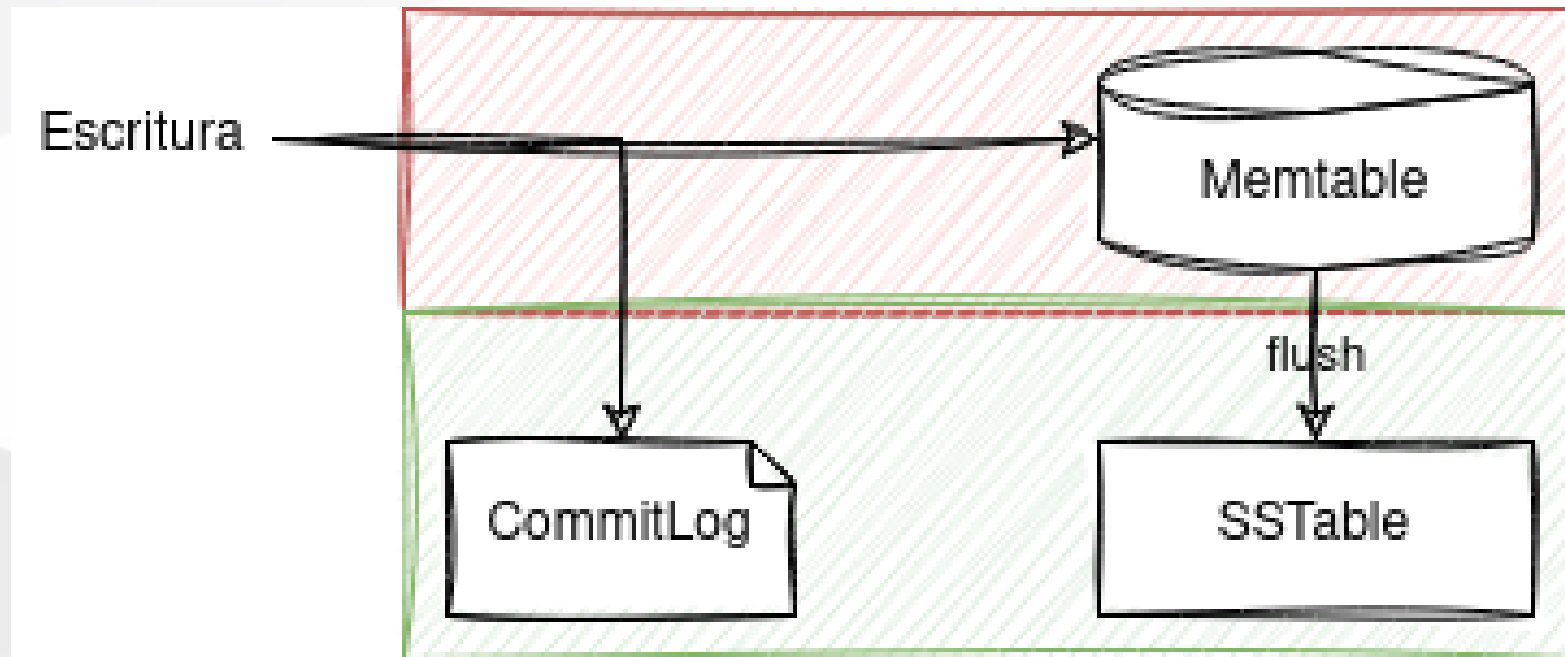
Replicación de datos

Operaciones de lectura/escritura

Proceso de escritura a nivel de nodo

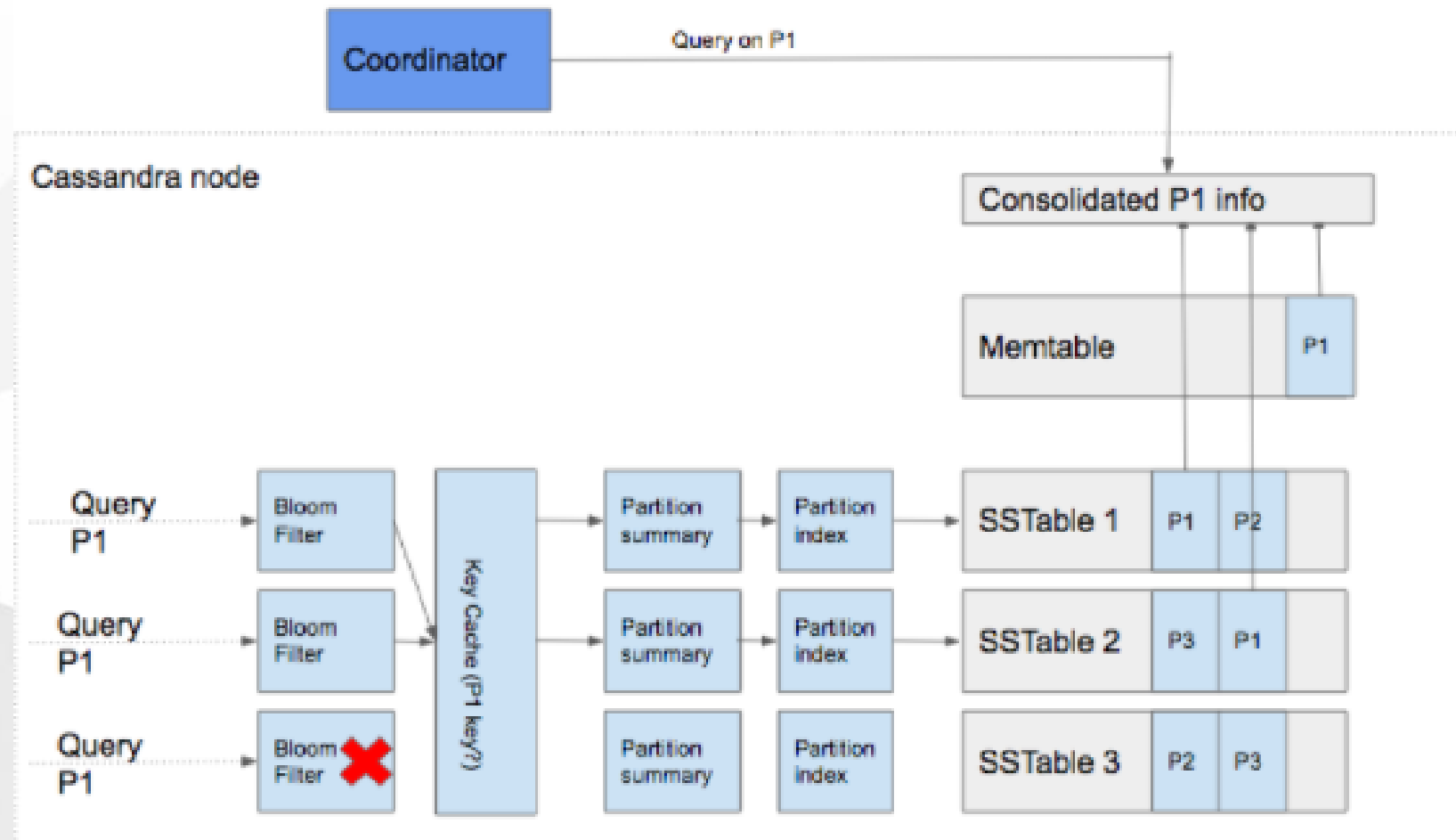
Cassandra procesa los datos en varias etapas durante la escritura:

1. Registro de la operación en el `CommitLog`
2. Escritura de datos en la `Memtable`
3. Descarga de datos de la `Memtable` a las `SSTables`



Proceso de lectura a nivel de nodo

La lectura en Apache Cassandra es más compleja que la escritura



Proceso de escritura a nivel de nodo (II)

- Los datos en disco pueden estar fragmentados en varias SSTables
- Necesita consolidar los datos existentes en Memtables y SSTables
- La lectura necesita identificar la SSTable que más probablemente contenga información sobre las particiones que estamos consultando
- Esta selección se realiza mediante la información del BloomFilter.

Proceso de escritura a nivel de nodo (y III)

Los pasos para la lectura son los siguientes:

1. Comprobación de la Memtable
2. Comprobación del BloomFilter
3. Comprobación en caché de claves de partición (si está activa)
4. Si la partición no está en la caché, se comprueba el resumen (*summary*) de la partición
5. Se accede al índice de la partición
6. Se localizan los datos en el disco
7. Se obtienen los datos de la SSTable
8. Antes de devolverlos, los datos se consolidan a partir de la Memtable y la SSTable

Vamos, que es un proceso algo largo y tedioso

Compresión de datos

Consistencia de datos ajustable

Gossip Protocol

Deteccción de fallos

HintedHandoffs

Filtros de Bloom

Gracias