

Citus

José R.R. Viqueira

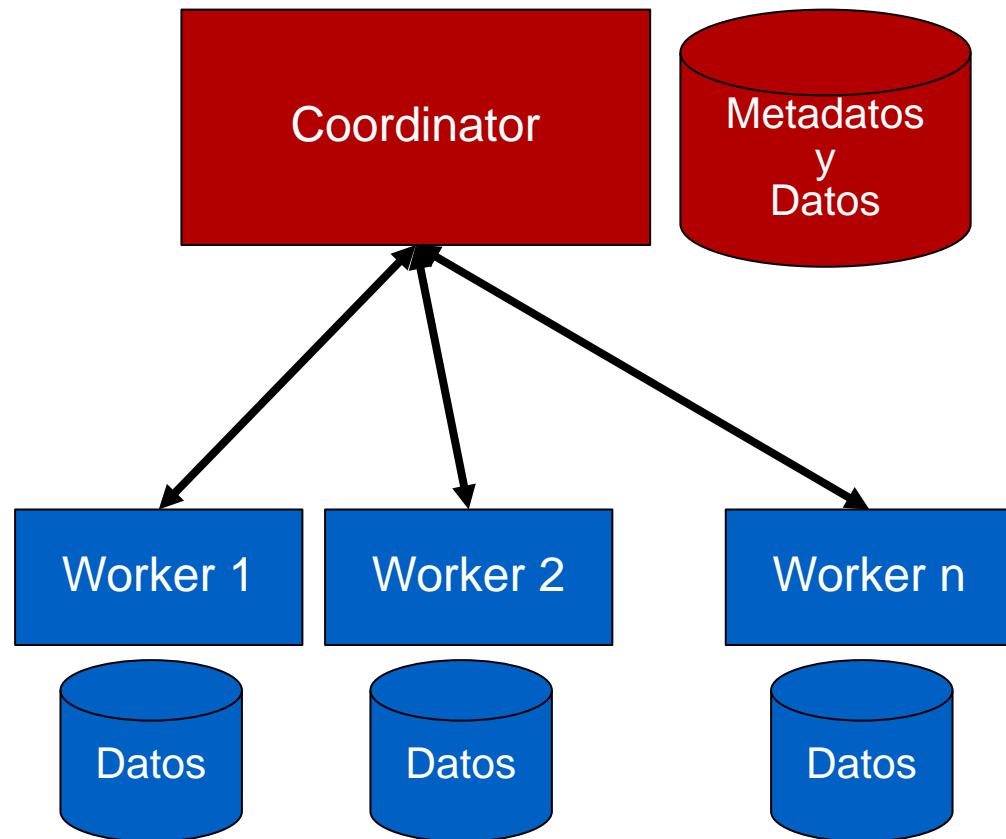
- **Arquitectura**
- **Tipos de tablas**
- **Sharding**
- **Co-localización**
- **Paralelismo**

■ Extensión de PostgreSQL

- ▷ Coordinación de varios servidores de datos básicos (commodity)
- ▷ Arquitectura "Shared Nothing"
- ▷ Escalabilidad horizontal

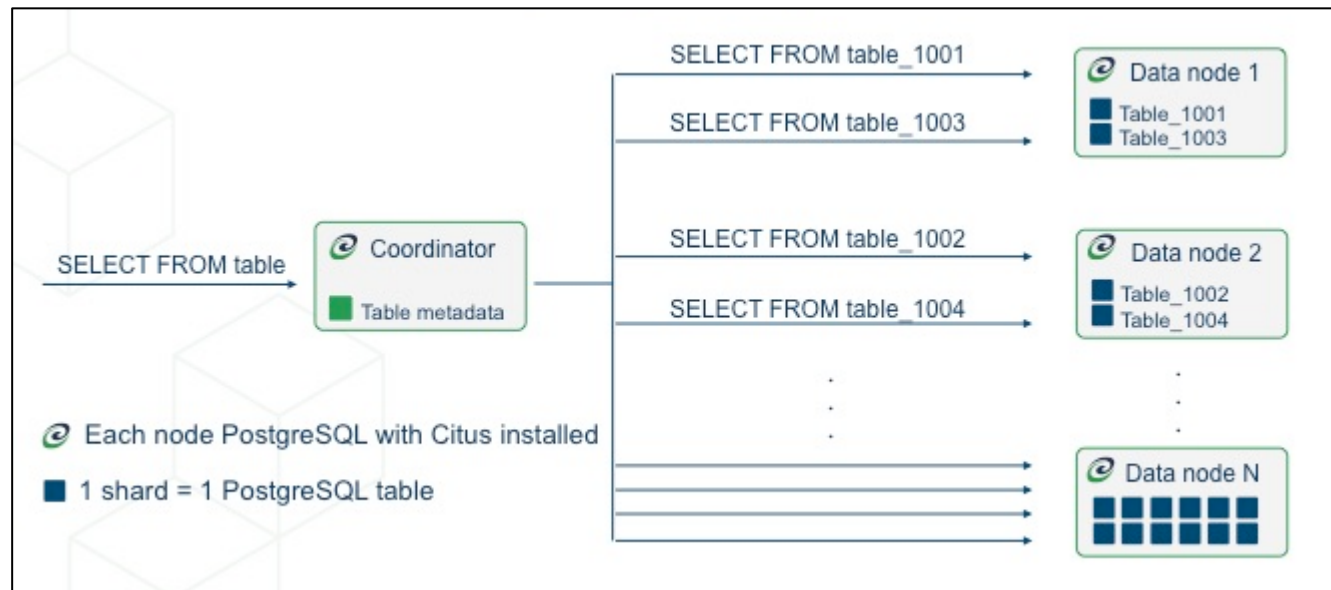
■ Tipos de nodos

- ▷ 1 Nodo **Coordinator**
 - Punto de acceso al cluster
 - Utiliza los workers para resolver las consultas
 - Almacena tablas de metadatos
- ▷ Varios nodos **Workers**



■ Tablas distribuidas

- ▷ Particionadas horizontalmente entre los nodos worker
 - _ Cada pedazo se llama shard
 - _ Las consultas SQL e instrucciones DDL se distribuyen desde el coordinador a los workers
- ▷ El particionamiento horizontal se hace usando una columna de la tabla.
 - _ La **columna de distribución** se especifica cuando se particiona la tabla.
 - _ Importante elegir bien la columna para tener un buen rendimiento
 - https://docs.citusdata.com/en/v10.0/sharding/data_modeling.html#distributed-data-modeling





■ Tablas de referencia

- ▷ Tabla de tamaño no muy grande. Datos relevantes para todos los workers.
 - _ Enumeraciones, dimensiones de tamaño reducido.
 - Género de la película, País, Idioma, etc.
- ▷ Una sola partición horizontal replicada en todos los workers, para evitar tener que moverla entre máquinas en tiempo de consulta.
- ▷ Se utiliza protocolo de compromiso en dos fases en las transacciones
 - _ Se asegura la consistencia a pesar de estar replicada
- ▷ Más información en la siguiente URL
 - _ https://docs.citusdata.com/en/v10.0/develop/reference_ddl.html#reference-tables

■ Tablas Locales

- ▷ Tablas creadas en el coordinador que no se distribuyen
- ▷ Tablas pequeñas sobre las que no se hacen JOINS
 - _ Ejemplo: Tabla de usuarios utilizada solo para autenticación

Arquitectura

Tipos de tablas

Sharding



Co-localización

Paralelismo

- Tabla de metadatos **pg_dist_shard**
 - ▷ Una fila por cada shard: rango de valores del resultado de un hash
- Colocación de los shard en los workers
 - ▷ Localización de cada shard también en tablas de metadatos
 - ▷ Número de shards por cada tabla es configurable.
 - _ Solución de compromiso
 - ▷ Replicación de los shards (dos opciones)
 - _ Replicación CITUS
 - Crea shards adicionales de back-up y se encarga de mantenerlos
 - Mejor en aplicaciones de tipo "append-only"
 - _ Replicación streaming
 - Replicación directamente proporcionada por PostgreSQL
 - Replica el nodo completo en otro servidor.
 - Es transparente para citus.
 - Mejor en aplicaciones con muchas transacciones (OLTP).
 - ▷ Protocolo de compromiso
 - _ Una fase o dos fases (una fase por defecto).
 - SET citus.multi_shard_commit_protocol = '2pc';

Arquitectura

Tipos de tablas

Sharding

Co-localización



Paralelismo

- Objetivo: Colocar tuplas relacionadas entre si en el mismo worker
 - ▷ Evitar movimientos de datos entre workers para realizar JOINS
- Como: Utilizar una clave común para particionar
 - ▷ Se indica al particionar que la tabla estará co-localizada con otra.
- Más detalles en la siguiente URL
 - ▷ https://docs.citusdata.com/en/v10.0/sharding/data_modeling.html#colocation

Arquitectura

Tipos de tablas

Sharding

Co-localización

Paralelismo



- Objetivo: Ejecutar partes de la consulta en workers distintos en paralelo
 - ▷ Resultado parcial de cada worker enviado al coordinador, que combina todo el resultado para enviar al cliente
 - ▷ Los beneficios se notan de verdad cuando el resultado tiene un volumen bajo respecto a los datos de entrada
- Más detalles en la siguiente URL
 - ▷ https://docs.citusdata.com/en/v10.0/develop/reference_processing.html#citus-query-processing

Citus

<https://docs.citusdata.com/en/v10.1/>

José R.R. Viqueira