Modelos de agregación: Bases de datos orientadas a columnas

Máster en Business Analytics y Big Data

Bases de Datos No Convencionales

Contenidos de la sesión

- ¿Qué es Cassandra?
- Modelo de datos
- Operaciones CRUD: *hands-on*
- Recomendaciones de uso

¿Qué es Cassandra?

¿Qué es Cassandra?

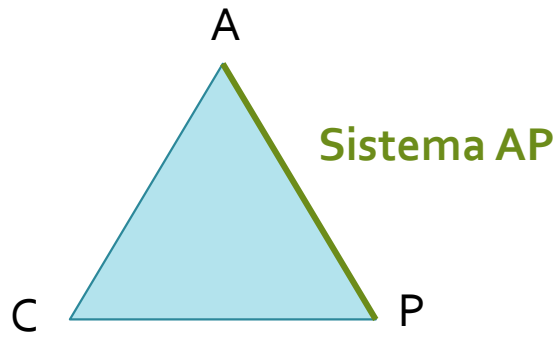
- Cassandra es un gestor de base de datos no SQL:
  - Originariamente creada por Facebook (2007), hoy en día es un producto open source mantenido por Apache.
    - Nace para solucionar problemas de almacenamiento en los buzones de entrada de Facebook.
  - Sin esquema.
  - Modelo de datos de agregación orientado a columnas.
    - Otros ejemplos: Amazon SimpleDB, Hbase.
    - Google BigTable (precursor).

¿Qué es Cassandra?

- Cassandra es un gestor de base de datos no SQL:
  - Tolerante a fallos.
    - Los datos se replican de forma automática en distintos nodos.
  - Descentralizada.
    - Un nodo del cluster de datos es idéntico a otros, evitándose cuellos de botella.
    - Alta disponibilidad.
    - Elástica, puede crecer horizontal y verticalmente, aumentando el rendimiento en lectura y escritura.
  - Consistencia final en el tiempo.
  - Proporciona drivers para múltiples lenguajes de programación:
    - .NET/C#, C++, Herlang, go, Java, Haskell, Perl, PHP, Python, R, Ruby, Scala, etc.

¿Qué es Cassandra?

- Cassandra garantiza que el sistema siempre esté disponible, aunque no tenga una consistencia estricta de los datos en presencia de particiones.
  - Puede configurarse para que se comporte de distinta forma, promocionando la consistencia a costa de la disponibilidad (y aumentando la latencia).

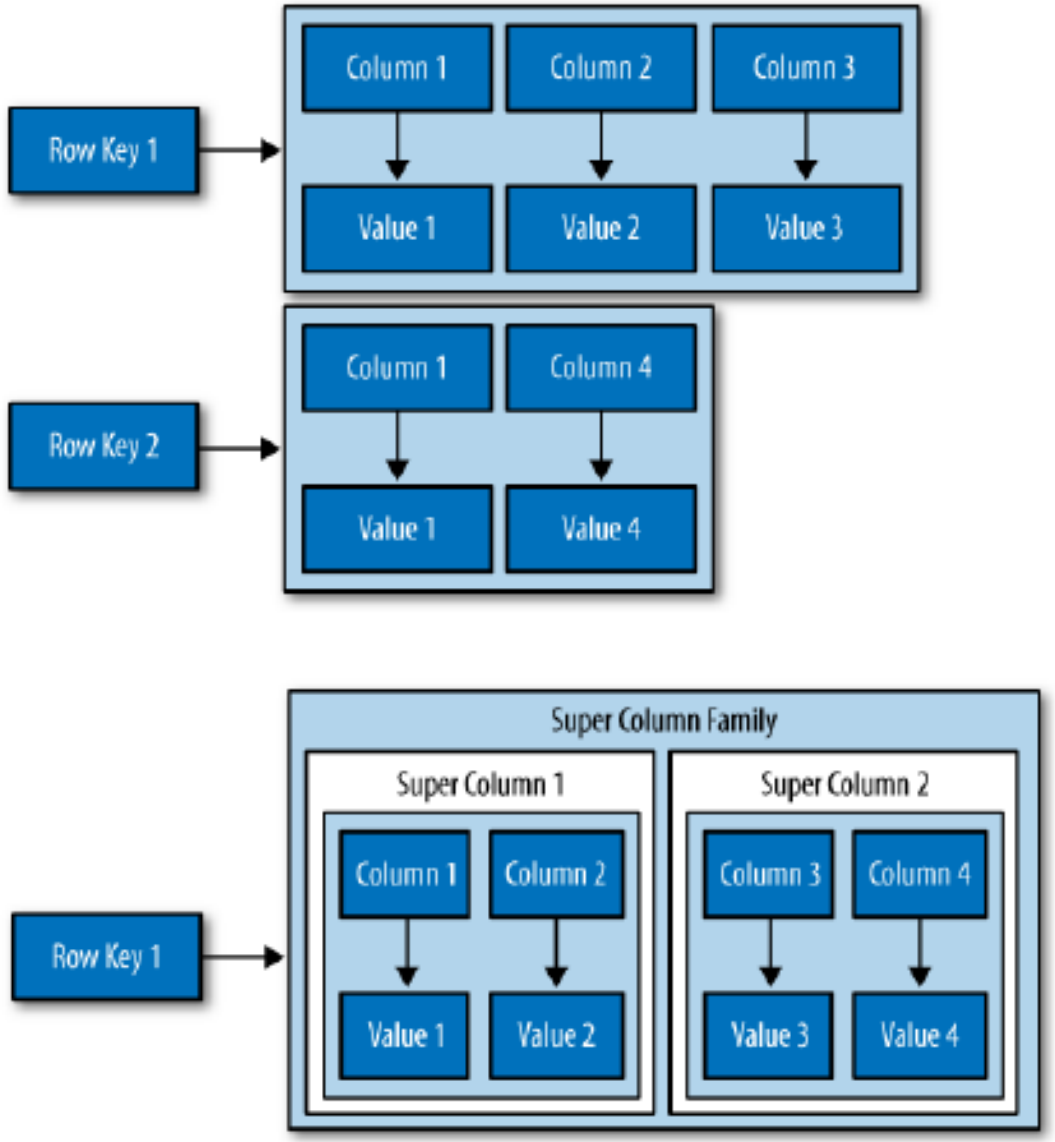


CIFF

Universidad de Alcalá

Modelo de Datos

Modelo de datos

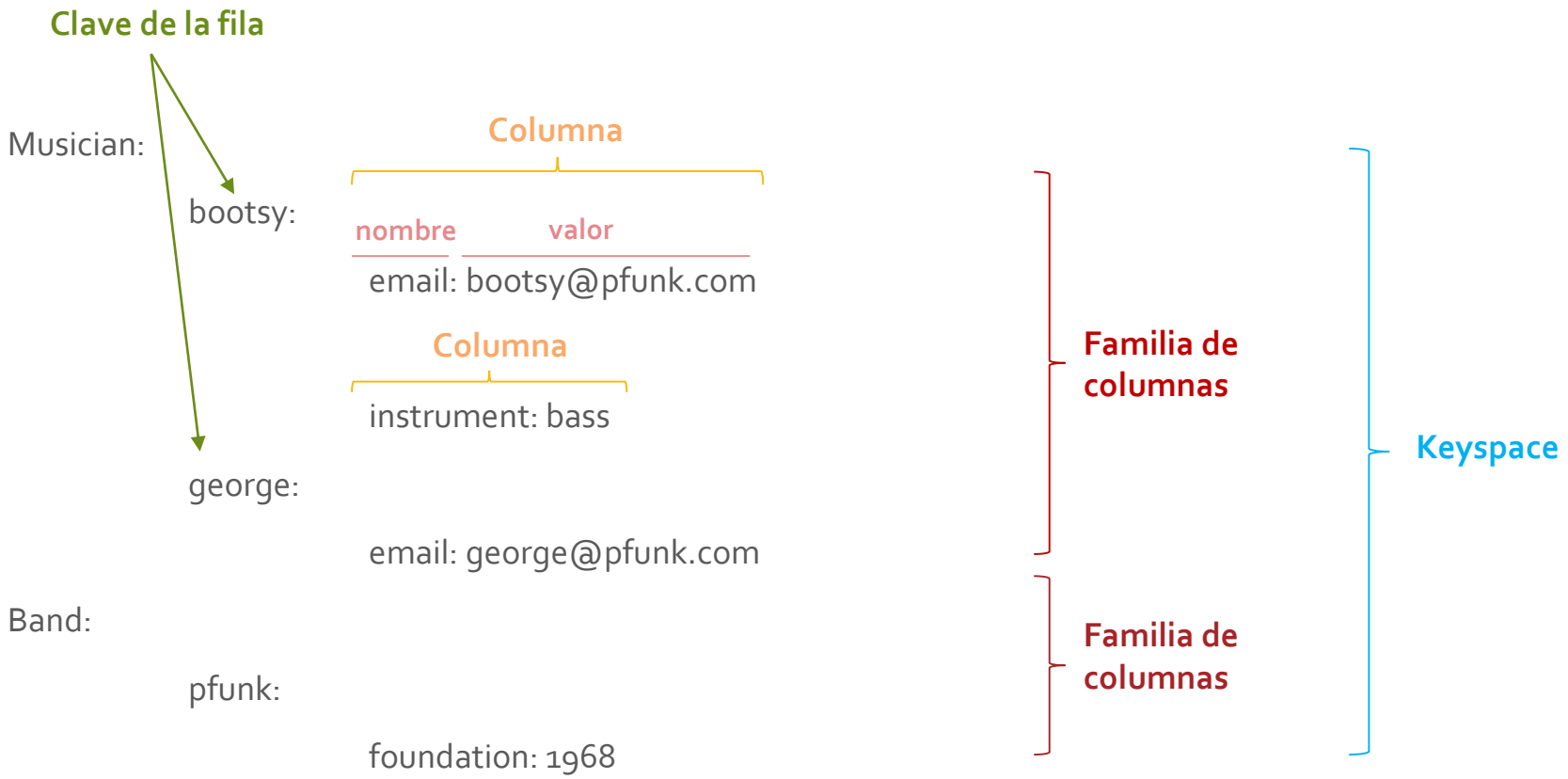


Modelo de datos

Modelo relacional
Base de datos
Tabla
Fila
Columna

Cassandra
Keyspace
Familia de columnas
Fila
Columna (clave-valor)

Modelo de datos



Modelo de datos

- Cada familia de columnas se almacena en un fichero separado.
- Se puede añadir columnas a una familia en cualquier momento.
- Las columnas se pueden agrupar en super-columnas.
  - Se crean familias de supercolumnas.
- El modelo de datos de la aplicación debe estar guiado por las consultas.
- El concepto de agregación en super-columnas optimiza el tiempo de recuperación.
  - Cualquier acceso a las columnas de dentro de una super-columna deserializa todas las columnas de la super-columna.
  - No conveniente si la super-columna contiene muchas columnas y solo se necesita accede a una pequeña porción de la misma.

Modelo de datos

```
Hotel {
            key: AZC_043
            {           name: Cambria Suites Hayden,
                        phone: 480-444-4444,
                        address: 400 N. Hayden Rd.,
                        city: Scottsdale,
                        state: AZ,
                        zip: 85255
            }
            key: AZS_011
            {           name: Clarion Scottsdale Peak,
                        phone: 480-333-3333,
                        address: 3000 N. Scottsdale Rd,
                        city: Scottsdale,
                        state: AZ,
                        zip: 85255}
            key: NYN_042
            {           name: Waldorf Hotel,
                        phone: 212-555-5555,
                        address: 301 Park Ave,
                        city: New York,
                        state: NY,
                        zip: 10019}
}

PointOfInterest{
SCkey: Cambria Suites Hayden
{
            key: Phoenix Zoo
            {
                        phone: 480-555-9999,
                        desc: They have animals here.
            },
            key: Spring Training
            {
                        phone: 623-333-3333,
                        desc: Fun for baseball fans.
            },
},
SCkey: Waldorf-Astoria
{
            key: Central Park
            {
                        desc: Walk around. It's pretty.
            },
            key: Empire State Building
            {
                        phone: 212-777-7777,
                        desc: Great view from the 102nd floor.
            }
}
}
```

CIFF
Universidad de Alcalá

Operaciones básicas en modo consola: CQL

Operaciones básicas

(ver fichero Cassandra.docs para la sesión hands-on)

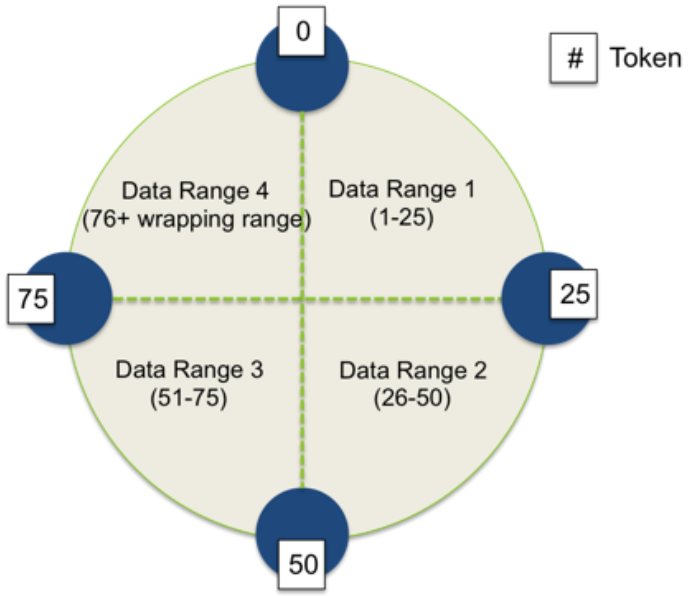
CIFF

Universidad de Alcalá

Distribución de datos

Fragmentación de Datos

- La distribución inicial de una fila se hace en función de su clave (row key).
- Cada nodo tiene asignado un valor (token).
  - Dentro del rango de valores de la clave de fila (o de la función hash que utilice).
- El rango de la clave de fila se divide entre el número de nodos del cluster.
- Cada nodo es responsable del rango de datos que va desde el token del nodo anterior + 1 hasta su token.
- El primer nodo para ubicar una réplica es el primero encontrado siguiendo las agujas del reloj que tenga un valor de token mayor que la clave de fila.



Nodos=4
Rango clave= 0..100

CIFF
Universidad de Alcalá

Fragmentación de Datos

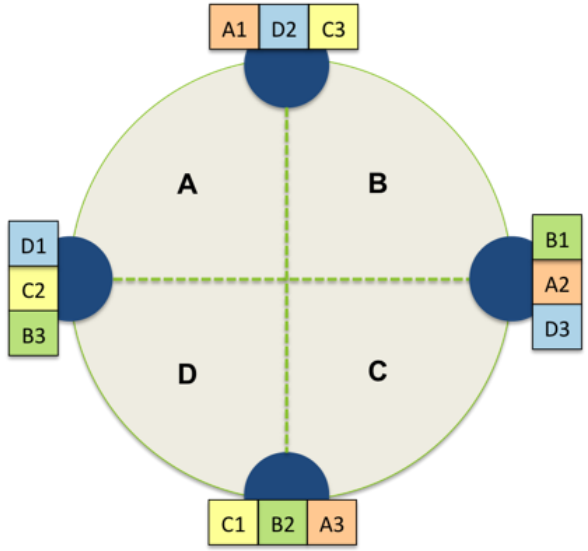
- Dos tipos de fragmentadores (partitioners):
  - RandomPartitioner
    - Utiliza una función hash para convertir la clave de fila.
    - Cada nodo tiene un token en el rango de la función hash.
  - ByteOrderedPartitioner
    - Utiliza el valor hexadecimal de la clave de fila.
    - Cada nodo tiene un token en ese rango rango hexadecimal.
    - ¡Muy poco recomendado!
      - Puede generar nodos muy cargados si hay escrituras secuenciales
      - El balanceo de los datos entre nodos es costoso y manual.
      - En el caso de tener varias familias de columnas, si el balanceo se hace para un tipo de clave, se generarán "puntos calientes" para las otras.
      - La ventaja de la consulta de valores consecutivos se pueden paliar con la definición de índices.

Replicación de Datos

- Se almacenan las réplicas a nivel de fila (row key).
- En número total de réplicas de una fila se conoce como *replication factor*
  - Un factor de replicación 2 implica que hay dos copias de cada fila, cada réplica en un nodo diferente.
- No se sigue un modelo master-slave
  - Cada réplica es igualmente importante.
- Hay dos tipos de estrategia de replicación:
  - SimpleStrategy
  - NetworkTopologyStrategy

Replicación de Datos

- SimpleStrategy :
  - Recomendada cuando solo hay un centro de datos.
  - Ubica la primera réplica en el nodo indicado por el partitioner.
  - A partir de ahí sitúa las demás copias en nodos consecutivos siguiendo el sentido de las agujas del reloj.
    - Con independencia del data center o del rack

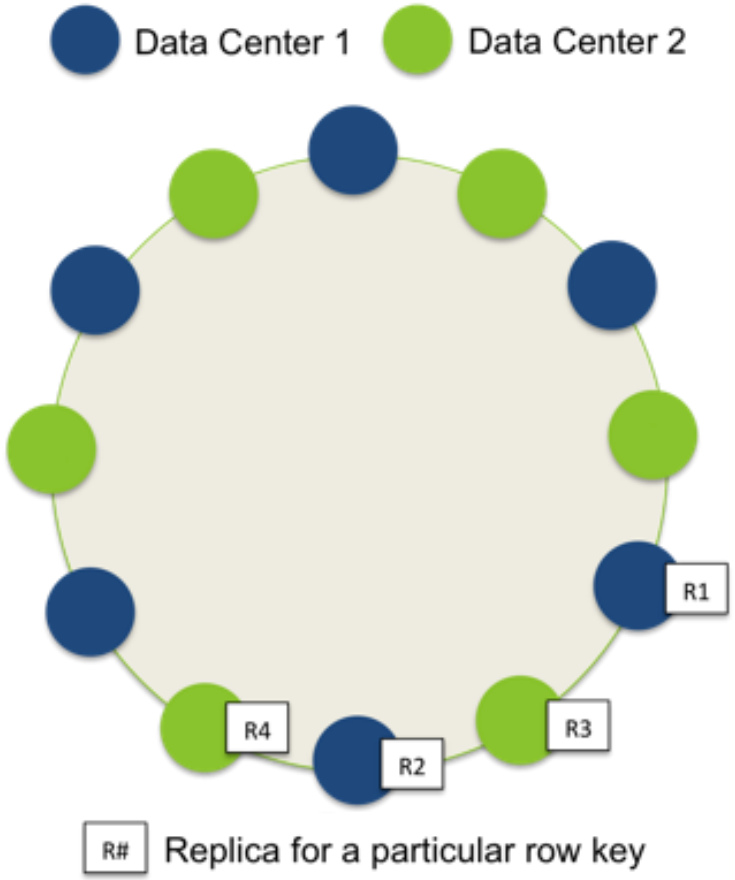


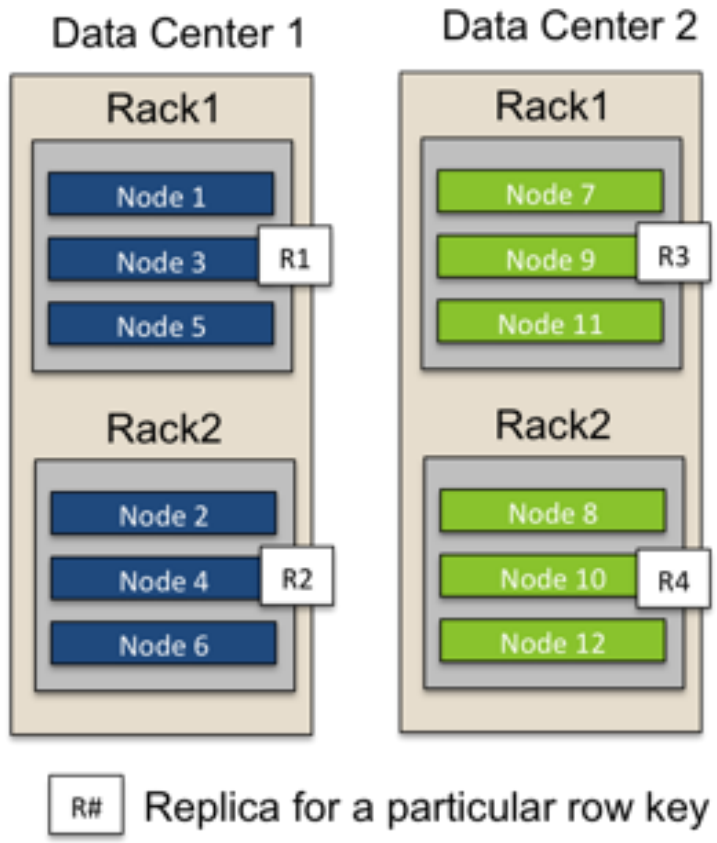
Replication factor=3
Data center=1
Nodos=4

Replicación de Datos

- NetworkTopologyStrategy:
  - Recomendada cuando se tiene el cluster distribuido en varios data center.
  - Hay que especificar cuántas réplicas se ubicarán en cada data center.
  - Ubica la primera réplica en el nodo indicado por el partitioner.
  - A partir de ahí sitúa las demás copias en nodos de racks diferentes siguiendo el sentido de las agujas del reloj.

Replicación de Datos





Replication factor=4
Racks= 2 por data center

Data centers=2
Nodos=3 por rack

CIFF
Universidad de Alcalá

CIFF

Universidad de Alcalá

Recomendaciones de uso

¿Cuándo usar Cassandra?

- ¿Cuándo usar Cassandra?
  - En grandes desarrollos con mucho tráfico de datos y gran número de accesos simultáneos.
    - Las características de Cassandra están diseñadas para hacer instalaciones distribuidas.
  - Cuando se requieran muchas escrituras, estadísticas o análisis.
    - Diseñada en origen para almacenar actualizaciones en la actividad del usuario, uso de redes sociales, recomendaciones y revisiones y aplicación de estadísticas.
  - Cuando se tienen aplicaciones en evolución.
    - Las ausencia de esquema facilita futuras modificaciones.

¿Quién está usando Cassandra?

- Twitter: Almacena los tweets, realiza estadísticas en tiempo real, etc.
- Facebook: Aún es uso para la búsqueda en los buzones de entrada.
  - Almacena 150TB de datos en más de 100 máquinas.
- Bee.tv: Personalización de la televisión en streaming tanto en web como en dispositivos móviles..
- eBay, Instagram, Github, Cisco y otras muchas.

Lecturas

- Lakshman and Malik (2009) "A Decentralized Structured Storage System" *http://www.cs.cornell.edu/projects/ladis2009/papers/lakshman-ladis2009.pdf* (artículo sobre los orígenes de Cassandra en Facebook)
- DATASATAX CORP. (2013) Introduction to Apache Cassandra http://www.odbms.org/wp-content/uploads/2014/06/WP-IntroToCassandra.pdf
- Cassandra Query Language (CQL) http://www.datastax.com/documentation/cassandra/2.1/cassandra/cql.html