

Bd NoSQL Clave Valor

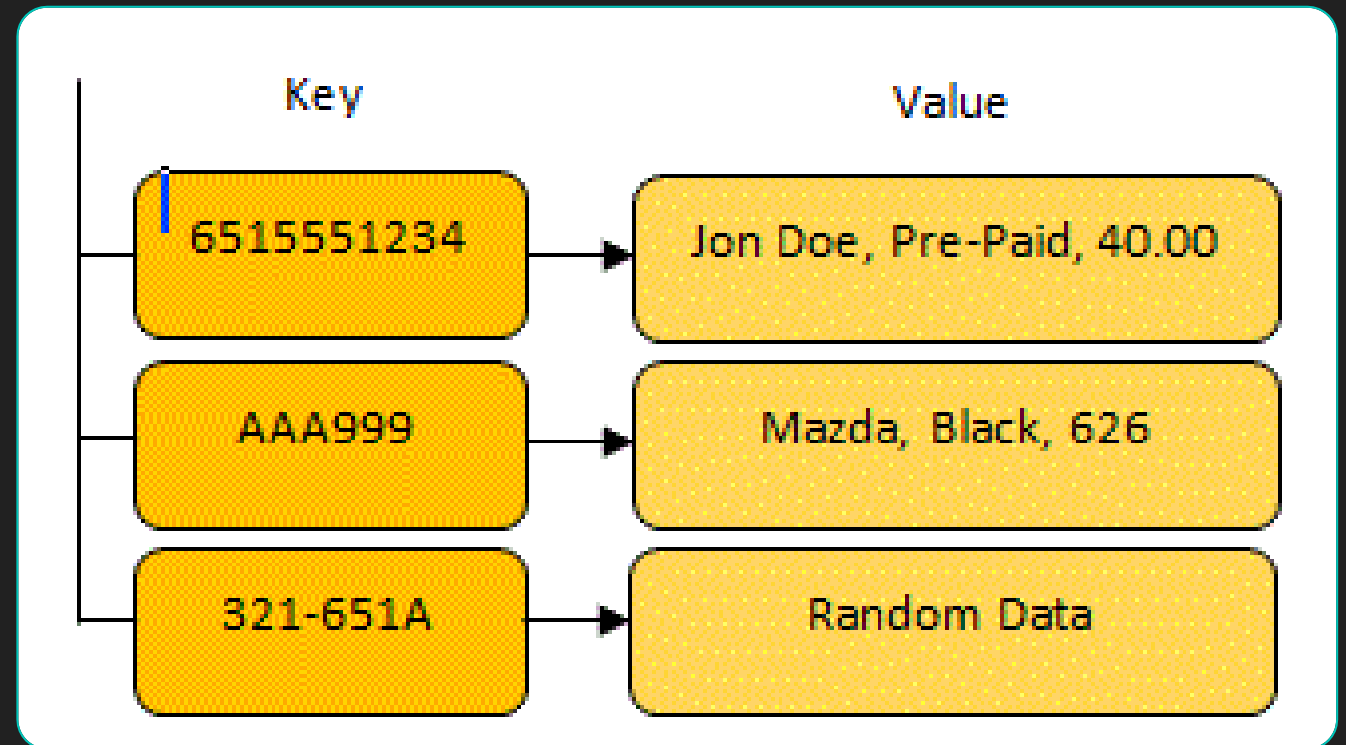
Kevin Moran
Bases2.fiusac@gmail.com

Agenda

- ✓ Modelo de datos
- ✓ Operaciones
 - ✓ Técnicas usadas
 - ✓ Cuando usar
 - ✓ Caso de estudio

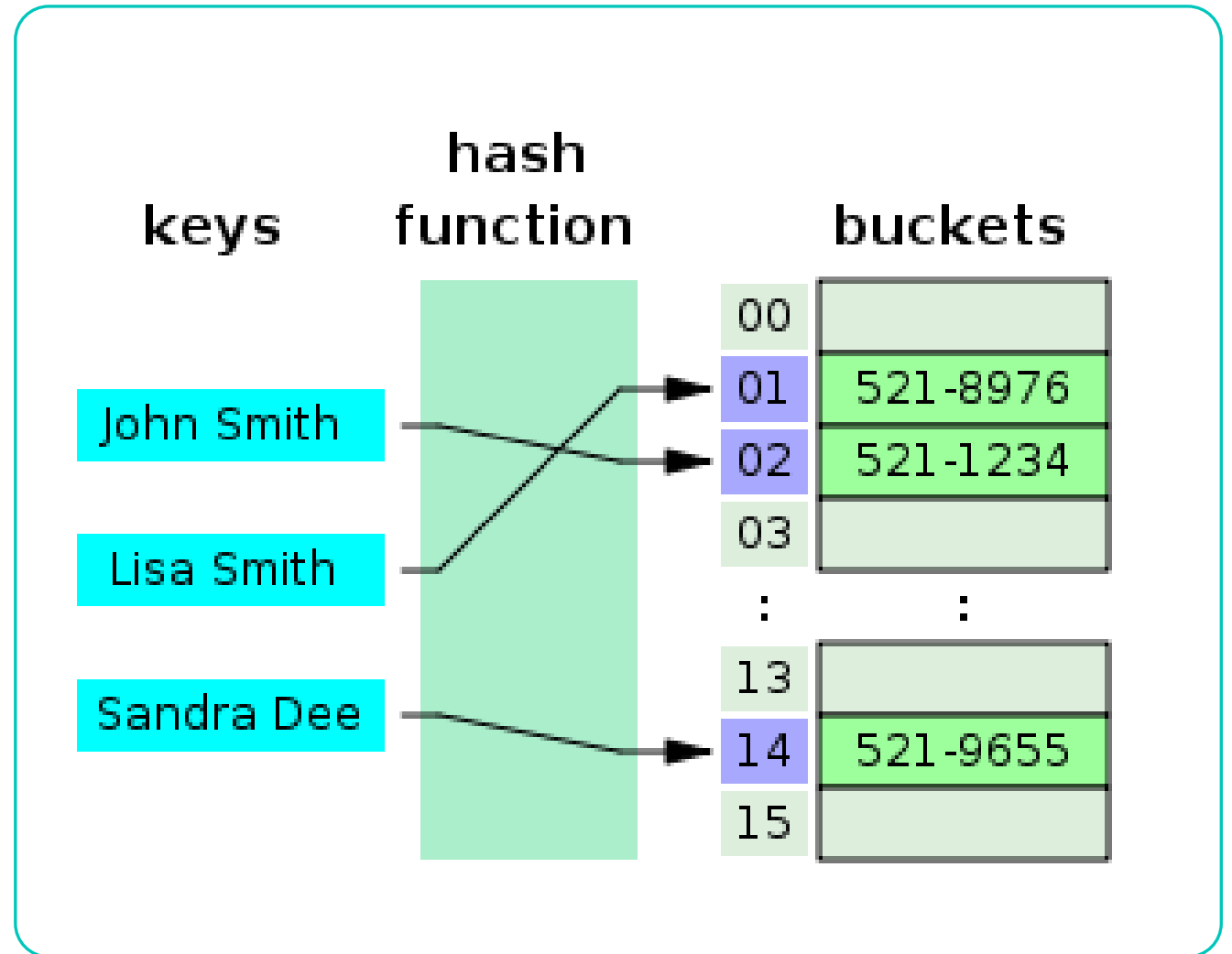
Modelo de datos

- Una **tabla hash, matriz asociativa, mapa hash, tabla de dispersión o tabla fragmentada**
 - Es una estructura de datos que asocia *llaves o claves* con *valores*.
 - La operación principal que soporta de manera eficiente es la *búsqueda*: permite el acceso a los elementos (teléfono y dirección, por ejemplo) almacenados a partir de una clave generada



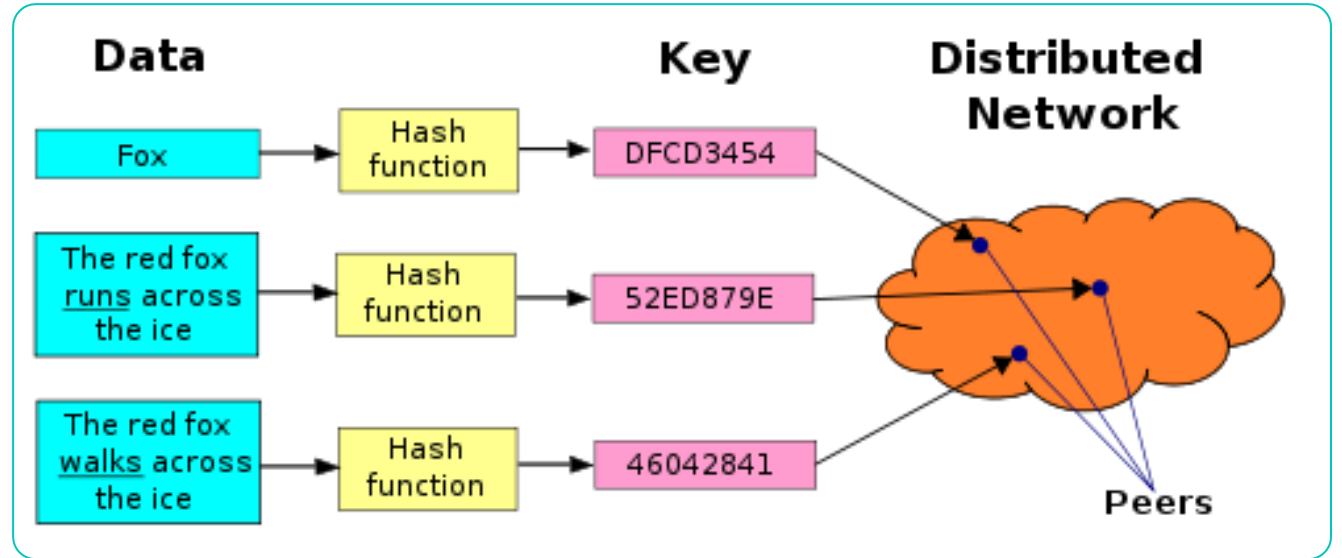
Distribución de los datos

- Para agilizar las búsquedas por la clave se utilizan normalmente funciones de asociación que son las que definen donde se almacenará el dato y luego permiten su rápido acceso

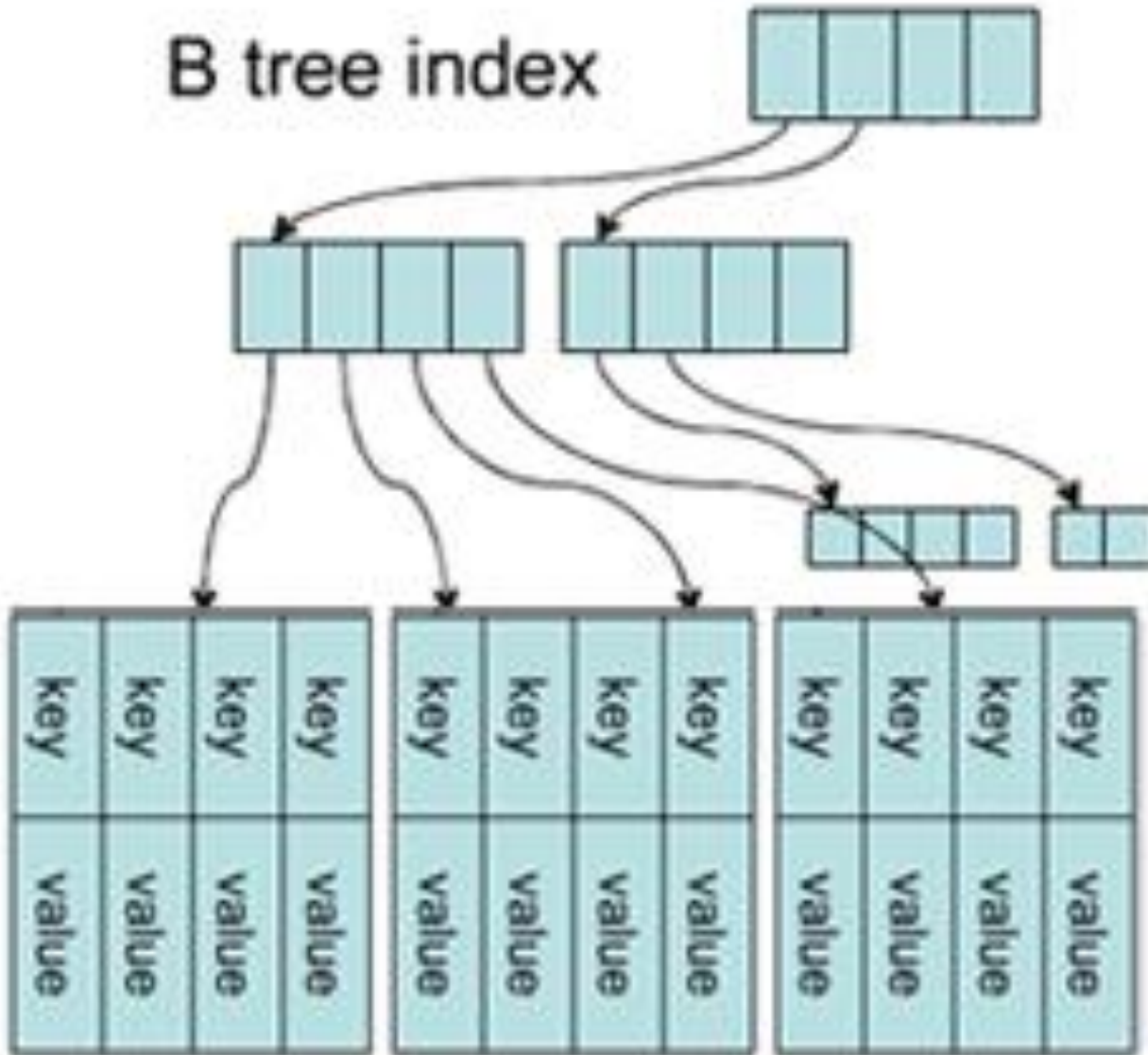


Distribución de los datos

- Pero en sistemas distribuidos donde los nodos pueden entrar y salir de manera dinámica este esquema es complejo de manejar
- En estos casos se propone el uso de la técnica como “Hashing Consistente” que requiere el diseño de Tablas hashing distribuidas



B tree index



Almacenamiento

- Si cada nodo tiene un conjunto de claves aleatorias, internamente puede almacenarse de distintas maneras:
 - Tabla hash
 - arboles B+ (Tokyo Cabinet)
- Tipos de registros
 - $\langle \text{key}, \text{value} \rangle$
 - Tupla $\langle \text{key}, v_1, \dots, v_n \rangle$

Operaciones

- Operaciones simples
 - Set(key, value) o put (key, value)
 - Get (key) : value
 - Delete (key)
- Otras operaciones
 - Operaciones sobre listas, conjuntos, mapas, entre otros
 - Push, Insert, remove, count, leng, etc.
 - mapreduce(keyList, mapFunc, reduceFunc)
 - Depende de la base de datos

Tipo	Base de datos
Clave-Valor en memoria	Coherence, eXtreme Scale, Redis, GigaSpaces, GemFire, Hazelcast, Infinispan, JBoss Cache, Memcached, Repcached, Terracotta, Velocity
Clave –Valor persistente	Flare, Keyspace, RAMCloud, SchemaFree, Hyperdex, Aerospike
Clave –Valor (Consistencia Eventual)	DovetailDB, Dynamo, Riak, Dynomite, MotionDb, Voldemort, SubRecord
Clave –Valor (Ordenada)	Actord, FoundationDB, Lightcloud, Luxio, MemcacheDB, NMDB, Scalaris, TokyoTyrant

Tipos

Técnicas usadas

- Voldemort, Riak, Redis, Scalaris, Tokyo Cabinet, Memcached/Membrain/Membase
 - Hashing Consistente(DHT)
 - Solo índices primarios: búsqueda por clave
 - No manejan índices secundarios
 - Transacciones: actualización simple o multiple.
 - Concurrencia: bloqueos o MVCC

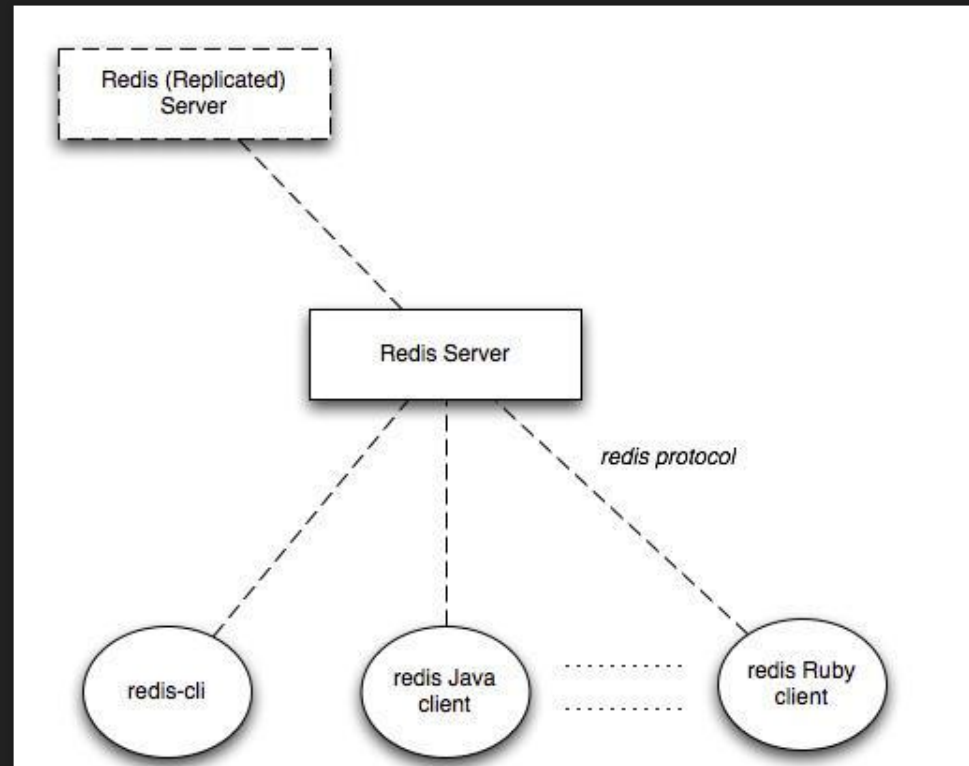
Caso de estudio - Redis

- Es una almacén de clave valor de código abierto
 - Primera versión: abril 2009
 - Escrito en C
 - Multiplataforma (menos Windows)
 - Licencia BSD
 - Sitio web: <http://redis.io>
- Maneja varios tipos de datos indexados por la clave como:
 - Cadena de caracteres (string)
 - hashes,
 - listas,
 - conjuntos y conjuntos ordenados (set, sorted set)

Caso de estudio - Redis

- Trabaja en memoria principal lo que disminuye los tiempos de respuesta
 - Cada segundo Redis puede manejar mas de 100.000 operaciones de Lectura/Escritura
- Es posible tener persistencia
 - volcar el conjunto de datos en el disco de vez en cuando
 - o añadiendo cada comando en un registro log.
- Particionamiento
 - Partición del lado del cliente: el cliente selecciona el nodo donde almacenar la clave
 - Partición asistida por proxy: Se configura el esquema de partición en un proxy y este se encarga de transmitir los datos a la instancia correspondiente
 - Enrutamiento de consultas: la consulta puede dirigirse a cualquier nodo y este se asegurará que llegue al nodo correcto
 - No implementa Hash Consistente (aún)

Caso de estudio – Redis Arquitectura



Caso de estudio – Redis

- Seguridad
 - Diseñado para ser accedido por los clientes de confianza
 - Si el acceso no es de confianza debe estar mediado por una capa de la aplicación de auditoria.
 - Aunque no implementa control de acceso ofrece una capa de autenticación que se activa en el archivo redis.conf
 - Redis no implementa cifrado de datos
 - Es muy difícil la inyección de código por no tener código de escape

Caso de estudio - Redis

- Tiene API's para diferentes lenguajes
 - PHP,
 - Java,
 - Python,
 - Ruby,
 - Perl,
 - Scala,
 - C, C++, Objective C, Node.js,
 - Muchos más...

Caso de estudio - Redis

- Algunos usos de Redis
 - Mensajería
 - Mostrar últimos artículos listados en su página principal.
 - Supresión y filtrado.
 - Tablas de clasificación y problemas relacionados
 - Colas
 - Análisis en tiempo real de lo que sucede, estadísticas, anti spam.
 - Caching
 - Muchos más ...

The Twitter logo, featuring the word "twitter" in a lowercase, blue, sans-serif font.The Instagram logo, featuring the word "Instagram" in a black, cursive script font.The ask.fm logo, featuring the text "ask.fm" in a white, lowercase, sans-serif font on a dark blue rectangular background.The Stack Overflow logo, featuring a stylized orange and yellow sunburst icon to the left of the text "stackoverflow" in a black, lowercase, sans-serif font.The Blizzard Entertainment logo, featuring the word "BLIZZARD" in a large, blue, stylized, blocky font with a registered trademark symbol, and the word "ENTERTAINMENT" in a smaller, blue, sans-serif font below it.

Caso de estudio – Redis
¿Quién lo usa?

Cuando usar almacenes clave-valor

Cuando la capacidad de almacenar y recuperar grandes volúmenes de datos es importante

No es importante almacenar las relaciones entre los elementos

La data es no estructurada

Util para el desarrollo rápido de prototipos

Manejo del crecimiento rápido de la lista de elementos: twitter, registros históricos de internet, blogs

No se requiere la implementación de validaciones lógicas o restricciones de los datos en la BD

Limitaciones de clave-valor

- Falta de estándares para el manejo de datos
- Lenguajes de consulta no son de alto nivel
- Herramientas para su uso y configuración
- Consultas y ordenamiento solamente por clave primaria



Lenguajes de consulta no son de alto nivel