Ejemplo de base de datos clave-valor: Riak

Máster en Business Analytics y Big Data

Bases de Datos No Convencionales

Basados en la documentación de la UOC. Autores: E. Rodríguez y J. Conesa

Contenidos de la sesión

- Introducción y características
- Modelo de datos
- Operaciones CRUD: *hands-on*
- Estrategias de distribución
- Recursos y enlaces

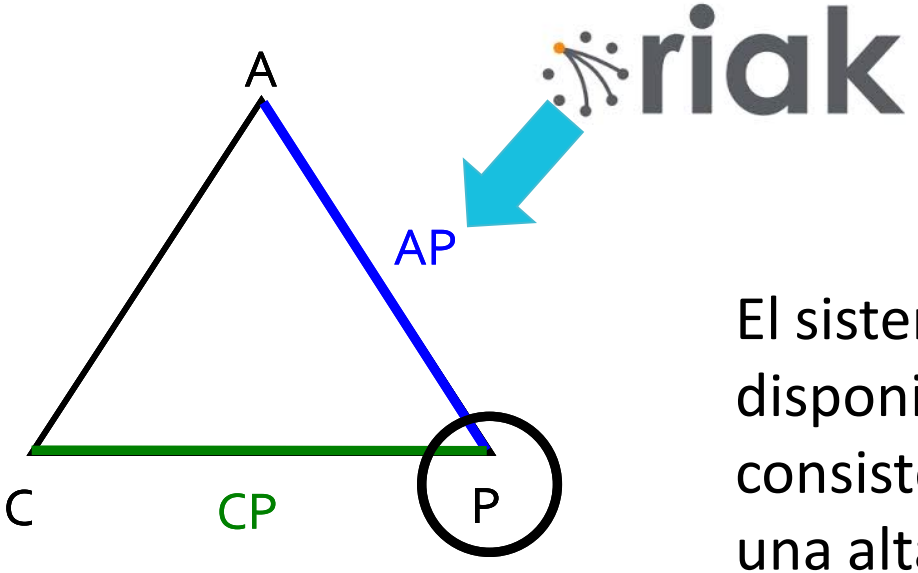
¿Qué es Riak?

- Creada por Basho Technologies y basada en Amazon Dynamo
- Sigue un modelo de clave-valor.
- Disponible en sistemas Linux y OS X
- Dispone de drivers para multitud de lenguajes de programación (C, C#, Java, Ruby, PHP, Python y Erlang, entre otros)
- Proporciona una API REST para consultar/modificar los datos de la base de datos.

Características de Riak

- Modelo de datos basado en objetos clave-valor:
  - Permite definir índices sobre los valores y los metadatos.
  - Permite definir enlaces entre objetos.
- Escrituras atómicas (a nivel de objeto):
  - Eventual consistency
  - Strong consistency a partir de la versión 2
  - No soporta transacciones de múltiples operaciones.
- Auto-sharding vía consistent hashing
- Gestión de réplicas vía quórums
- Soporta MapReduce.

Enfocado a proporcionar una alta tolerancia a fallos



riak

El sistema apuesta por la disponibilidad a costa de la consistencia, proporcionando una alta tolerancia a fallos pero permitiendo que algunas lecturas puedan devolver datos obsoletos.

CIFF

Universidad de Alcalá

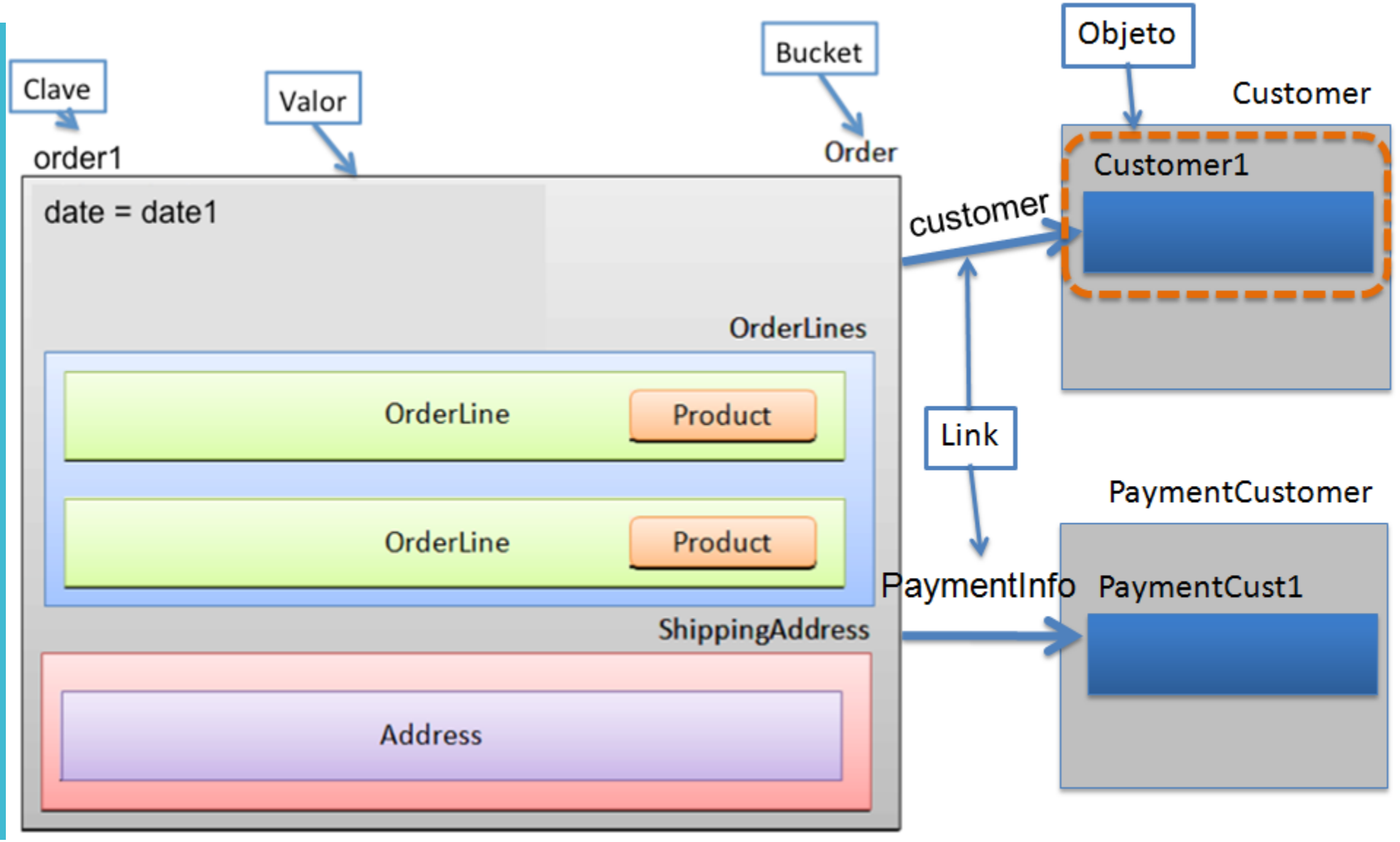
Modelo de Datos

Modelo de datos

Modelo Relacional	Riak
Base de datos	Base de datos
Tabla, vista	Bucket
Fila	Objeto = <clave, valor,…>
Columna	No tiene un equivalente directo ≈ metadato
Clave primaria	clave
Clave foránea	*Link*
Combinación (*join*)	≈ *Link Walking*
Trigger	≈ *Hook*

CIFF
Universidad de Alcalá

Modelo de datos: ejemplo



Índices

- Riak permite definir índices sobre los valores de los objetos: sólo disponible para valores en formato textual, XML, JSON y Erlang.
- Riak permite definir índices secundarios:
  - Los datos del valor son "opacos" y no siguen un esquema.
  - Los índices se definen sobre los metadatos.
  - Los valores indexados pueden no tener correspondencia con los datos del objeto.
  - El nombre del metadato indica su tipo:
    - *x-riak-index-<name>-bin*
    - *x-riak-index-<name>-int*

Operaciones CRUD

Operaciones básicas

- Las operaciones de creación, recuperación, actualización y borrado (CRUD):
  - Se realizan en el ámbito de una bucket.
  - Se realizan a nivel de objeto.

(ver fichero "Guia práctica de Riak" para la sesión hands-on)

Operaciones básicas de consulta

- GET /riak/BUCKET/KEY
  - curl –v –X GET http://localhost:8098/riak/order/order1
  - curl –v –X GET http://localhost:8098/riak/customer/customer1

Operaciones avanzadas de consulta: Link Walking

- GET /riak/BUCKET/KEY [bucket],[tag],[keep]
  - curl -v –X GET http://localhost:8091/riak/order/order1 _, customer, _
  - curl -v –X GET http://localhost:8091/riak/order/order1 PaymentCustomer, _, _
  - curl -v –X GET http://localhost:8091/riak/person/person1 Person, boss, 1

¿Cómo funcionan las operaciones de escritura?

- Transacciones a nivel de objeto
- Gestión de concurrencia basada en timestamping:
  - Todo el mundo puede leer/actualizar datos en todo momento.
  - Uso de vector clocks para la gestión de concurrencia

Operaciones básicas de escritura (creación / modificación)

- [PUT|POST] /riak/BUCKET/KEY
- [PUT|POST] /riak/BUCKET/
  - curl -v –X PUT http://localhost:8091/riak/order/order2 \

    -H "Content-Type: application/json" \

    -H "Link: </riak/customer/cust1>; riaktag=\"customer\"" \

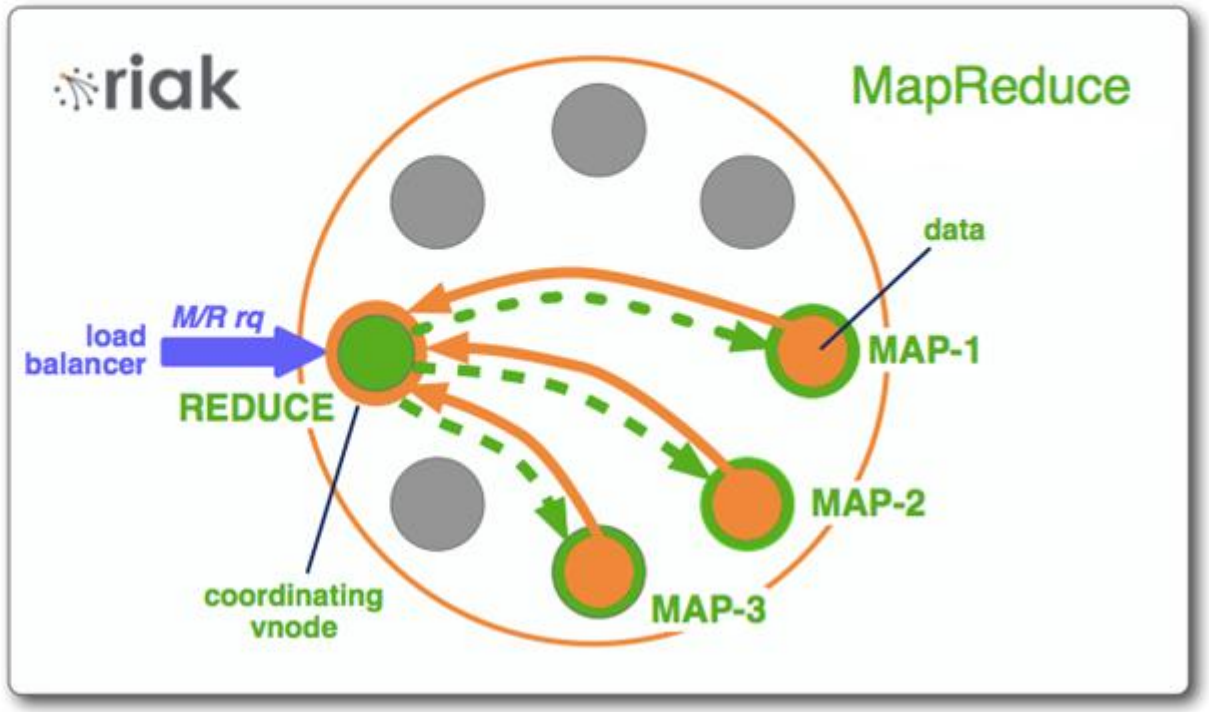
    -d '{"date":"31/07/2014", ...}'

Operaciones básicas de eliminación

- DELETE /riak/BUCKET/KEY
  - curl -i -X DELETE http://localhost:8091/riak/order/order2

Otras operaciones

- Gestión de *buckets*
- Búsquedas por índice
- MapReduce



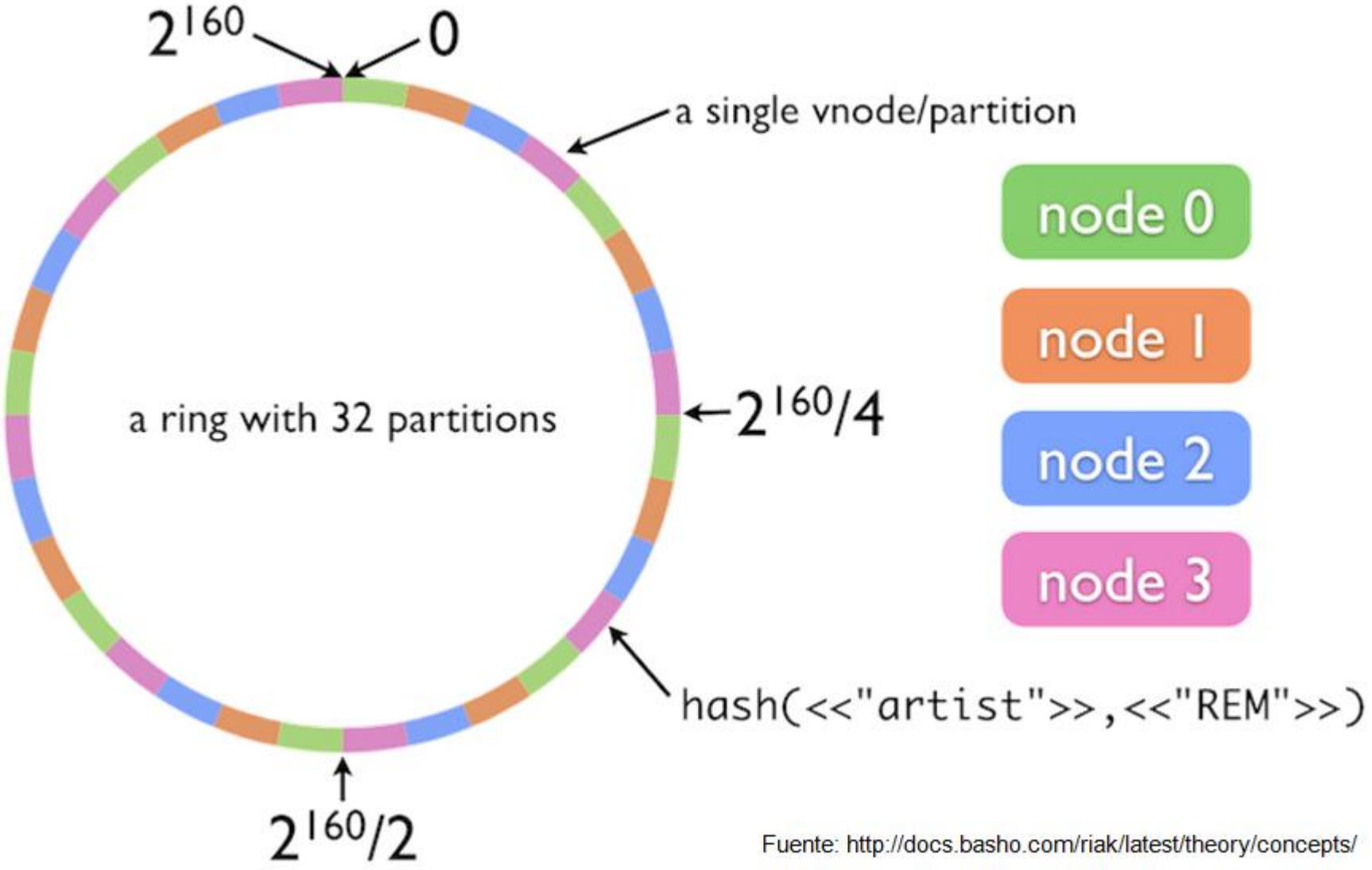
Fuente: http://docs.basho.com/riak/latest/dev/using/mapreduce/

CIFF

Universidad de Alcalá

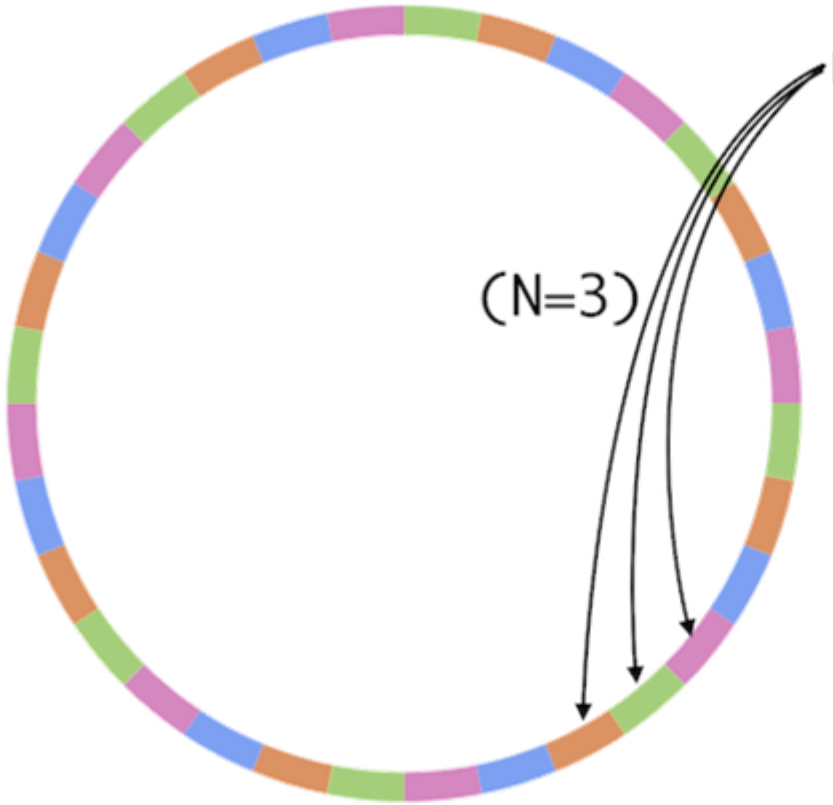
Distribución de datos

Arquitectura de distribución



Fuente: http://docs.basho.com/riak/latest/theory/concepts/

Replicación de datos



Fuente: http://docs.basho.com/riak/latest/theory/concepts/

- *N_val* indica el número de réplicas.
- Gestión de concurrencia vía *vector clocks*
- Gestión de réplicas vía quórums: parámetros *R*, *W* y *DW*

Sloppy vs *strict* quórums

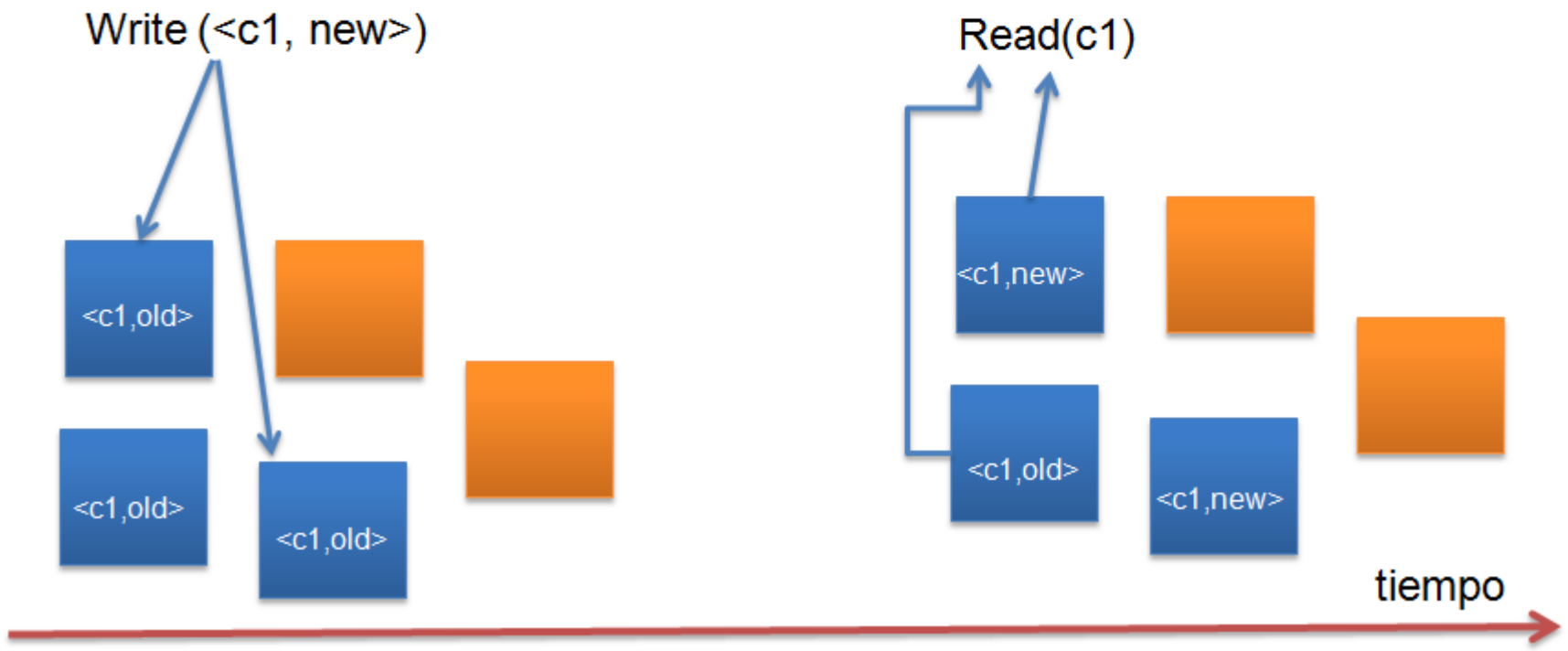
- NR indica el número de réplicas y N indica el número de servidores (nodos), siendo N>= NR.
- R y W: número de servidores que deben responder a una operación para que se considere válida
- Strict quórums: las respuestas deben ser de servidores que almacenen las réplicas.
  - (R+W)>NR y W>NR/2 garantiza consistencia fuerte.
- Sloppy quórums: las respuestas pueden ser de cualquier servidor.
  - (R+W)>NR y W>NR/2 NO garantiza consistencia fuerte.

Strict quórums

NR = 3, N=5, R=2, W=2

R+W>NR y W>NR/2

→ *Strong Consistency*

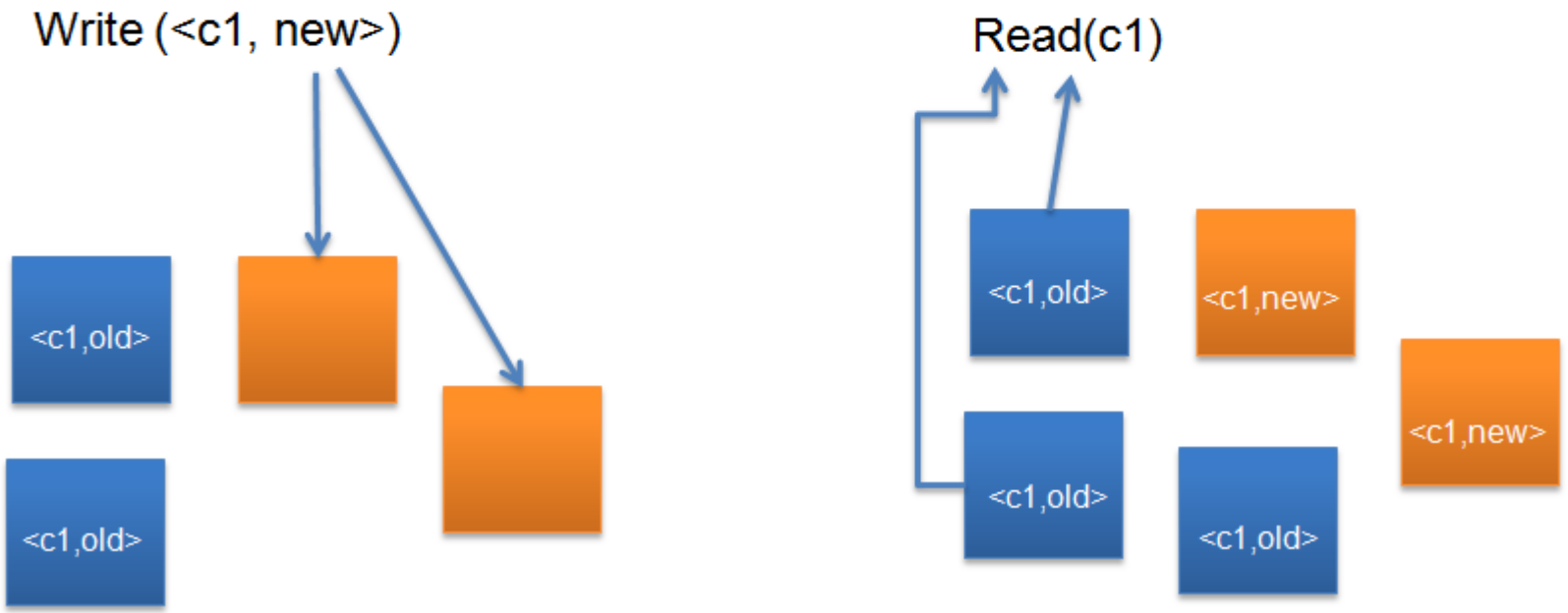


Sloopy quórums

NR = 3, N=5, R=2, W=2

R+W>NR y W>NR/2

→ *Eventual consistency*



CIFF

Universidad de Alcalá

Recursos y enlaces

Lecturas

- Página oficial de Riak: http://basho.com/riak/
- Documentación oficial: http://docs.basho.com/riak/2.0.opre11/
- Little Riak Book http://littleriakbook.com/
- E. Redmond, J. Wilson (2012). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. Pragmatic Bookshelf.
- P.J. Sadalage & M. Fowler. (2013). NoSQL Distilled. A brief Guide to the Emerging World of Polyglot Persistence, Pearson Education.