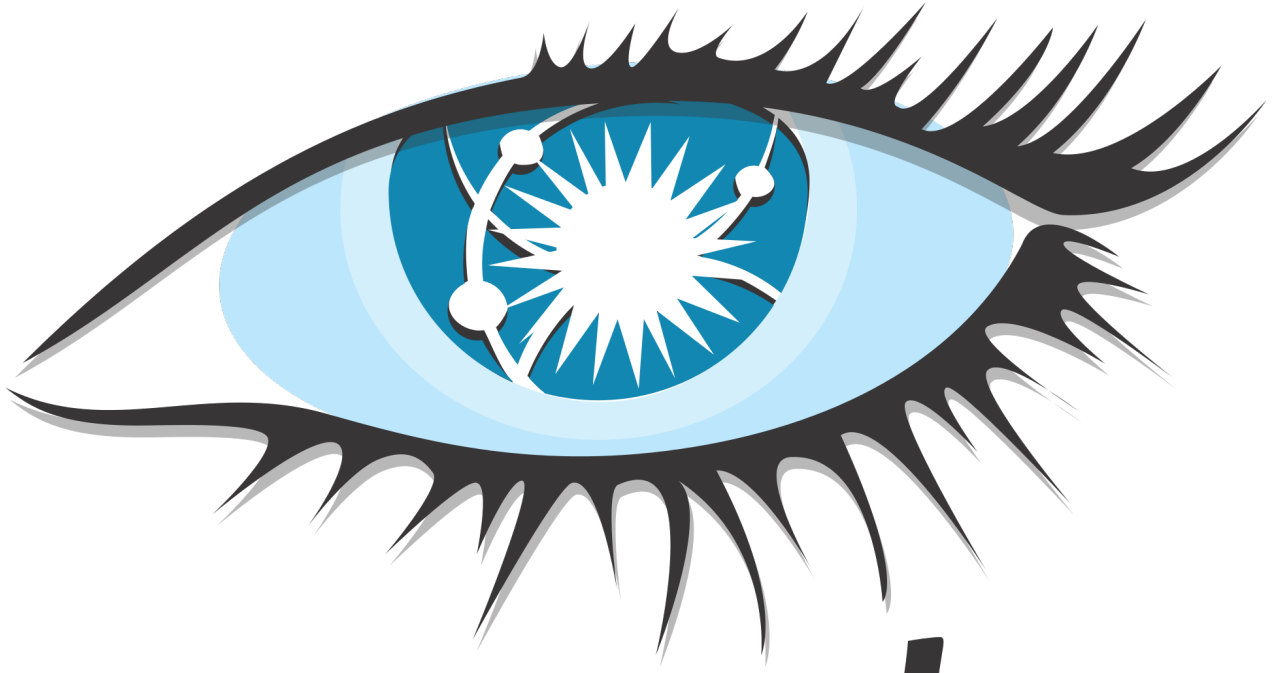


cassandra

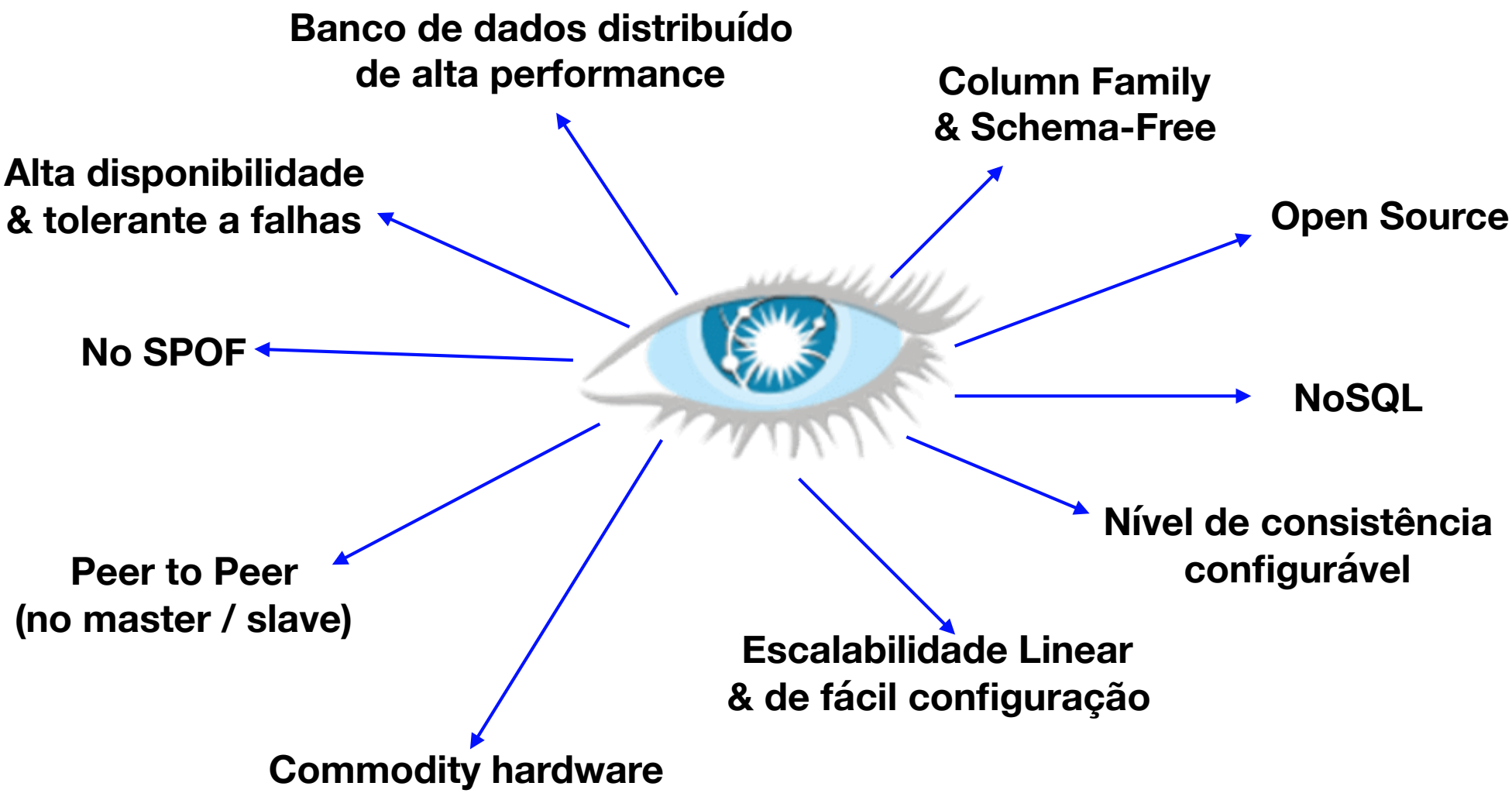
Seminário apresentado em 29/06/2017

Disciplina: Sistemas Distribuídos

Professora: Noemi Rodriguez

Aluno: Ricardo Dias

Visão Geral


Banco de dados distribuído
de alta performance
Column Family
& Schema-Free
Alta disponibilidade
& tolerante a falhas
Open Source
No SPOF
NoSQL
Nível de consistência
configurável
Peer to Peer
(no master / slave)
Escalabilidade Linear
& de fácil configuração
Commodity hardware

NoSQL

- 1998 - Carlo Strozzi
- 2009 - Eric Evans (Rackspace)

NoSQL

- Aproximadamente 225 database NoSQL no mercado

Tipos de Databases no NoSQL

- Wide Column Store / Column Families
    - Hadoop / HBase, **Cassandra**, Scylla, …
- Document Store
    - Elastic, MongoDB, Azure DocumentDB, CouchDB, …
- Key Value / Tuple Store
    - DynamoDB, Azure Table Storage, Riak, Redis, Berkeley DB, Oracle NOSQL Database, Voldemort, …
- Graph Databases
    - Neo4J, TITAN, HyperGraphDB, Onyx Database, OpenLink Virtuoso, GraphDB, ...

Tipos de bancos no SQL

- Multimodel Databases
    - ArangoDB, AlchemyDB, …
- Object Databases
    - Versant, Ninja Database Pro, ObjectDB, CoreObject, …
- Grid & Cloud Database Solutions
- XML Databases
- Multidimensional Databases
- …

http://nosql-database.org

Histórico de Cassandra

Since 2009

Usuários de Cassandra

Algumas das maiores implementações de produção

- Apple, com mais de 75.000 nós armazenando mais de 10 PB de dados
- Netflix (2.500 nós, 420 TB, mais de 1 trilhão de pedidos por dia)
- O mecanismo de pesquisa chinês Easou (270 nós, 300 TB, mais de 800 milhões pedidos por dia)
- eBay (mais de 100 nós, 250 TB)

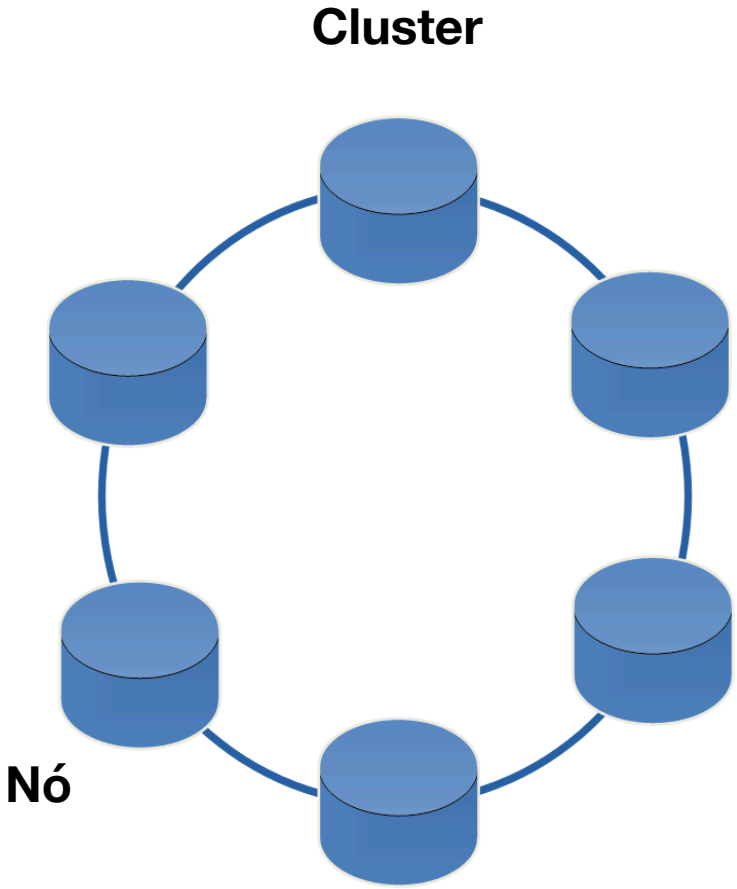
Arquitetura

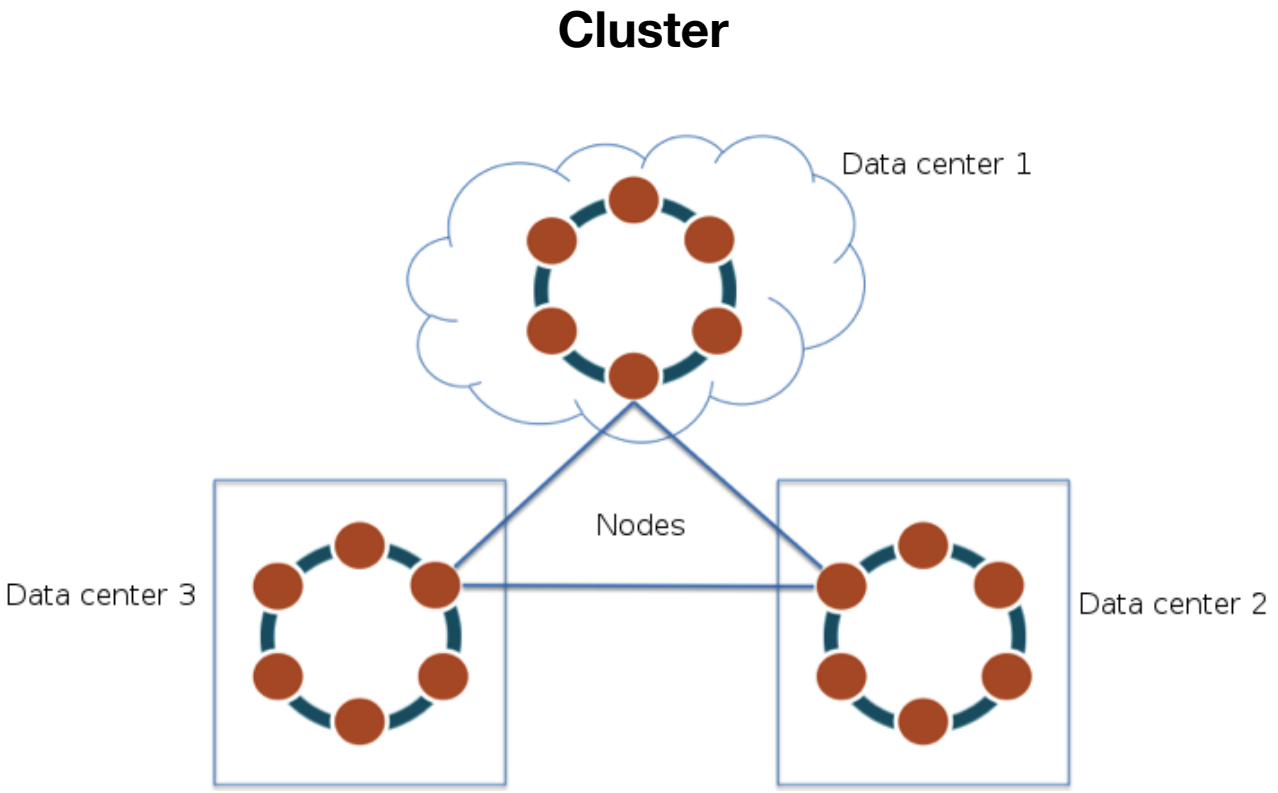
- Cassandra foi projetado para lidar com grande volume de dados e carga de trabalho
- Sua arquitetura baseia-se na compreensão de que falhas no sistema e no hardware podem ocorrer
- Cassandra aborda o problema das falhas empregando um sistema distribuído peer-to-peer em nós homogêneos, onde os dados são distribuídos entre todos os nós no cluster
- Cassandra não possui um único ponto de falha
- Cada nó freqüentemente troca informações de estado sobre si mesmo e outros nós, em todo o cluster, usando o protocolo de comunicação gossip

Arquitetura

- Nó
    - Onde os dados são armazenados. Componente básico da infraestrutura de Cassandra
- DataCenter
    - Coleção de nós relacionados. A replicação é definida por datacenter
- Cluster
    - Contém um ou mais datacenters

Arquitetura



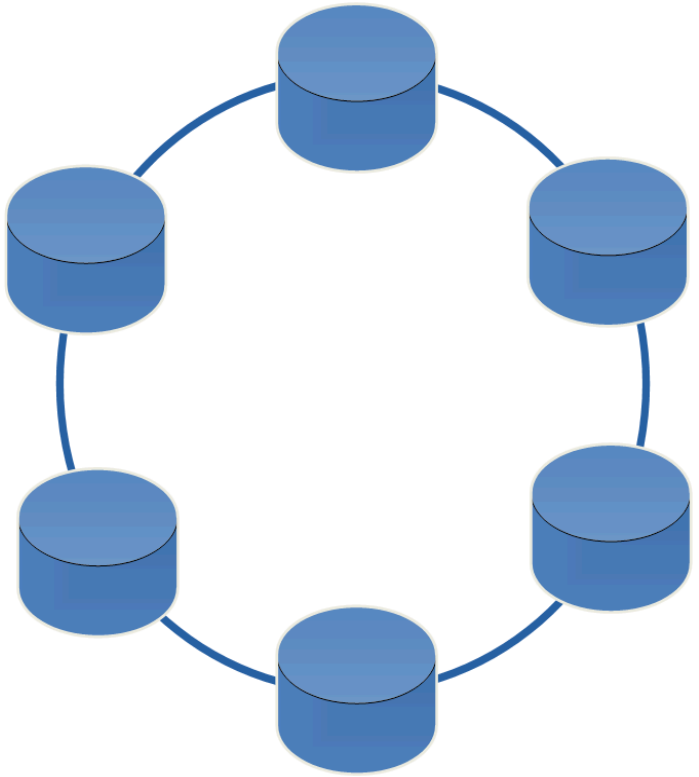


Arquitetura

- Arquivo de configuração (cluster name, etc)
- Seeds (configuração inicial para apresentação dos nós)
- Gossip
    - Apresentação dos nós
    - Servidores up, down ou com performance ruim
    - Testes periódicos nos nós que falharam para saber se os mesmos já estão de volta
- Nós podem ser adicionados ao cluster de forma dinâmica
- Remoção definitiva de um nó deve ser feito de forma explícita pelo utilitário nodetool

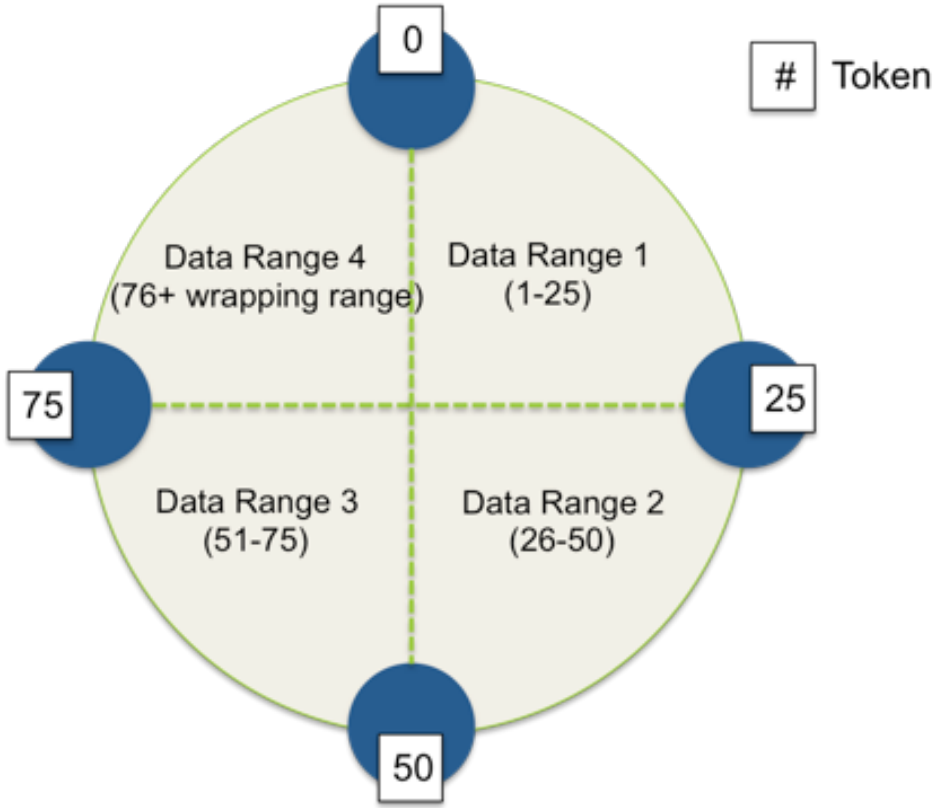
Arquitetura

- Sem master, slave e replica sets
- Sem servidor de configuração
- Todos os nós guardam dados e podem reponder queries (tanto escrita como leitura)



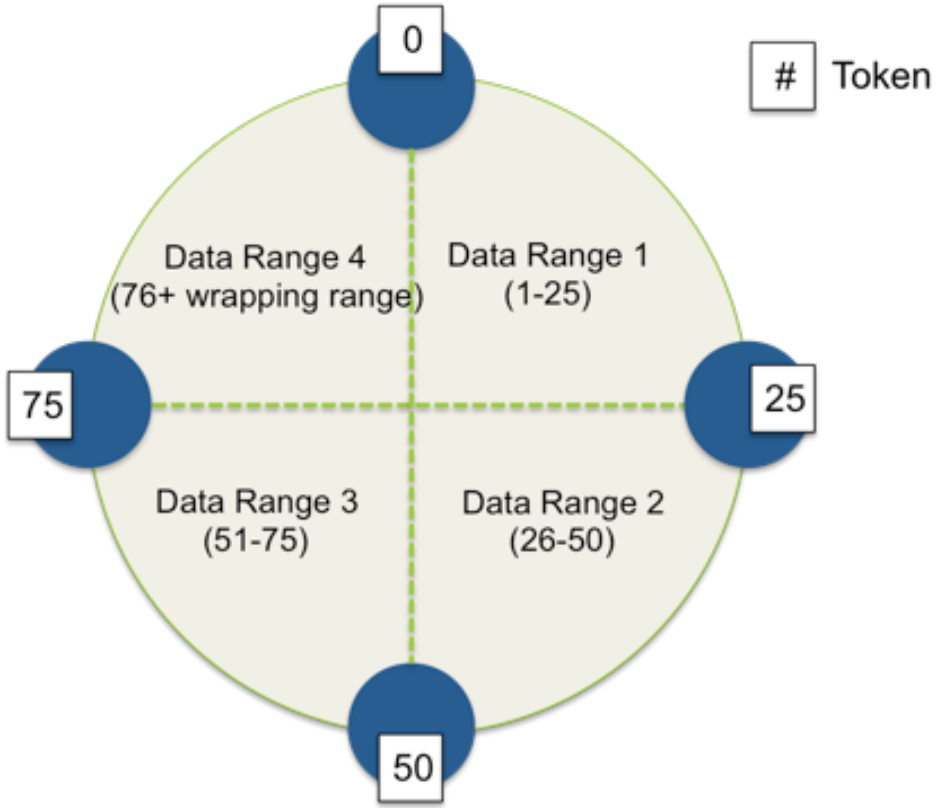
Particionamento

- O particionamento de dados determina como os dados são distribuídos através dos nós do cluster
- Tokens são atribuídos a cada nó
- Dados são particionados através dos nós do cluster (incluindo réplicas)
- A localização do dado no anel é determinada pela chave da partição e token correspondente



Particionamento

- Fatores envolvidos na distribuição dos dados: número de nós, distribuição dos nós em racks e o número de datacenters (topologia do cluster)
- MurmurHash (particionamento default)
- Os dados de todas as tabelas são distribuídos uniformemente
- Cada nó recebe aproximadamente a mesma quantidade de dados



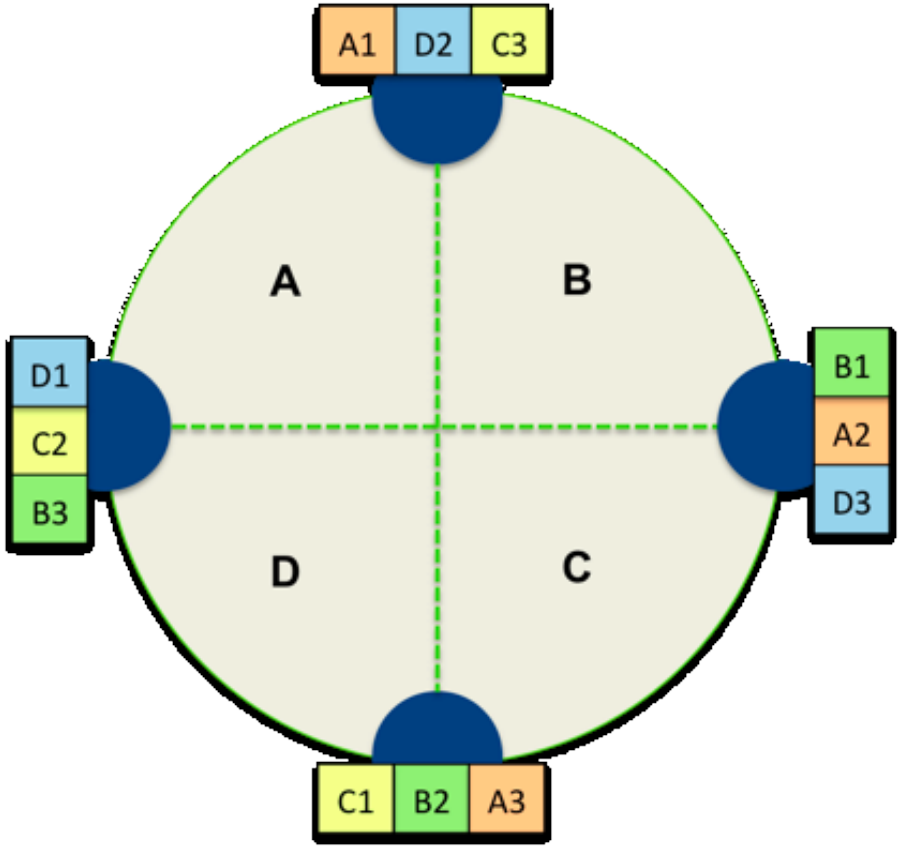
Replicação

- Replicação é o processo de armazenar cópias de dados em vários nós para garantir a confiança e tolerância a falha
- Cassandra armazena cópias, chamada replicas, de cada linha baseada na row key
- O número total de replicas através do cluster é referenciado como fator de replicação
- Todas as réplicas tem a mesma importância. Não existe réplica master ou primária
- O fator de replicação é definido para cada DataCenter

Estratégias de Replicação

- **SimpleStrategy**

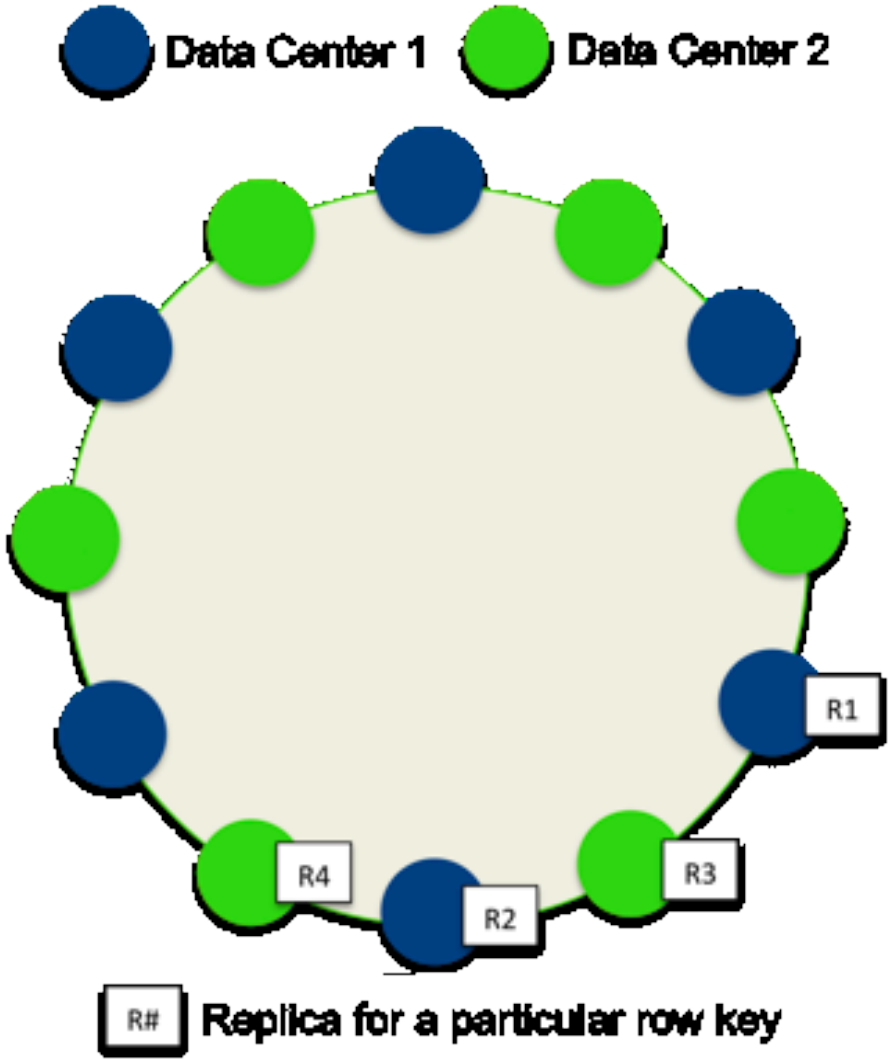
  Usada para um cluster simples em um único datacenter



Estratégias de Replicação

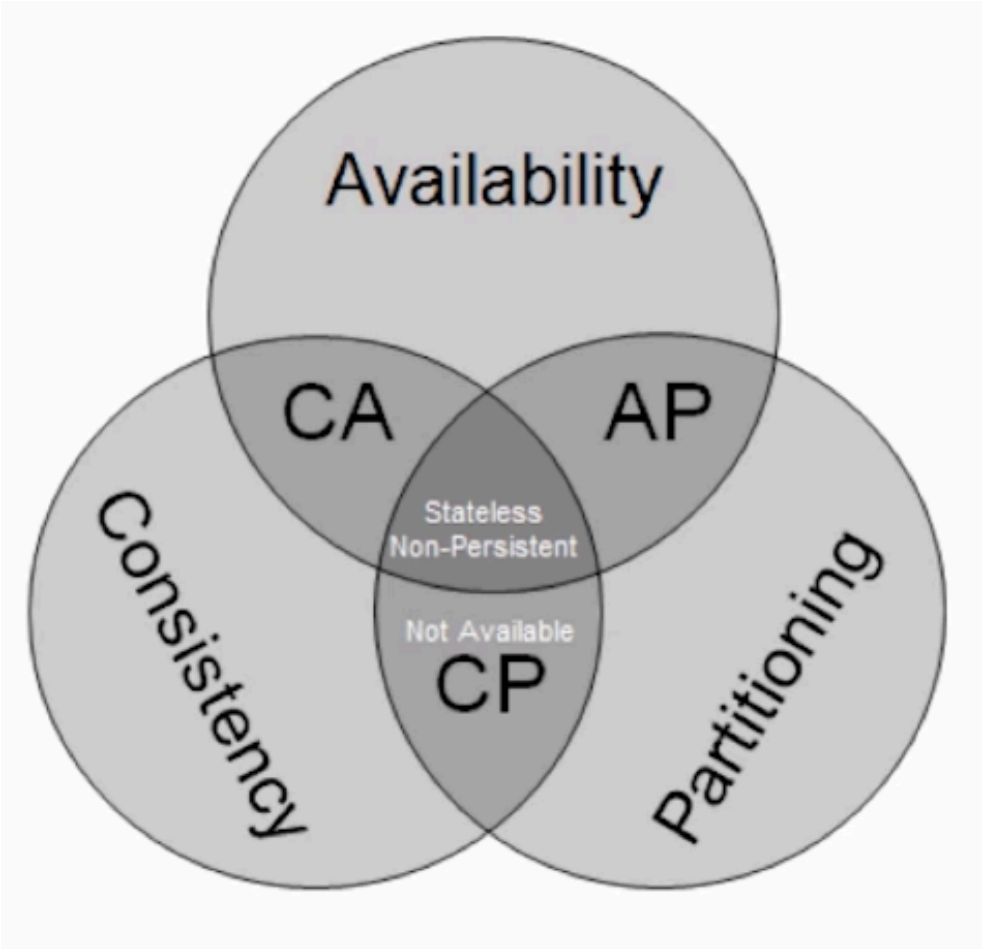
- **NetworkTopologyStrategy**

  Quando você tem ou planeja ter um cluster distribuído em vários datacenters



CAP x Cassandra

- Impossível ser consistente e ter alta disponibilidade durante uma partição de rede
- Latência entre datacenters também tornam a consistência mais difícil
- Cassandra escolhe Disponibilidade & Tolerância ao Particionamento sobre Consistência: AP



Nível de consistência

- É definido por query
- ANY, ONE (TWO, THREE, etc), QUORUM, ALL
- Escrita: quantos nós responderam ok
- Leitura: recupera dados de N nós e retorna o mais recente

Requisições Cliente

- Todos os nós no Cassandra são semelhantes
- Uma requisição cliente de leitura ou escrita pode ser feita em qualquer nó do cluster
- Quando um cliente conecta com um nó e executa uma requisição de leitura ou escrita, este nó serve como coordenador para esta operação particular
- O coordenador atua como um proxy entre a aplicação cliente e os nós (ou réplicas) que possuem os dados sendo requisitados

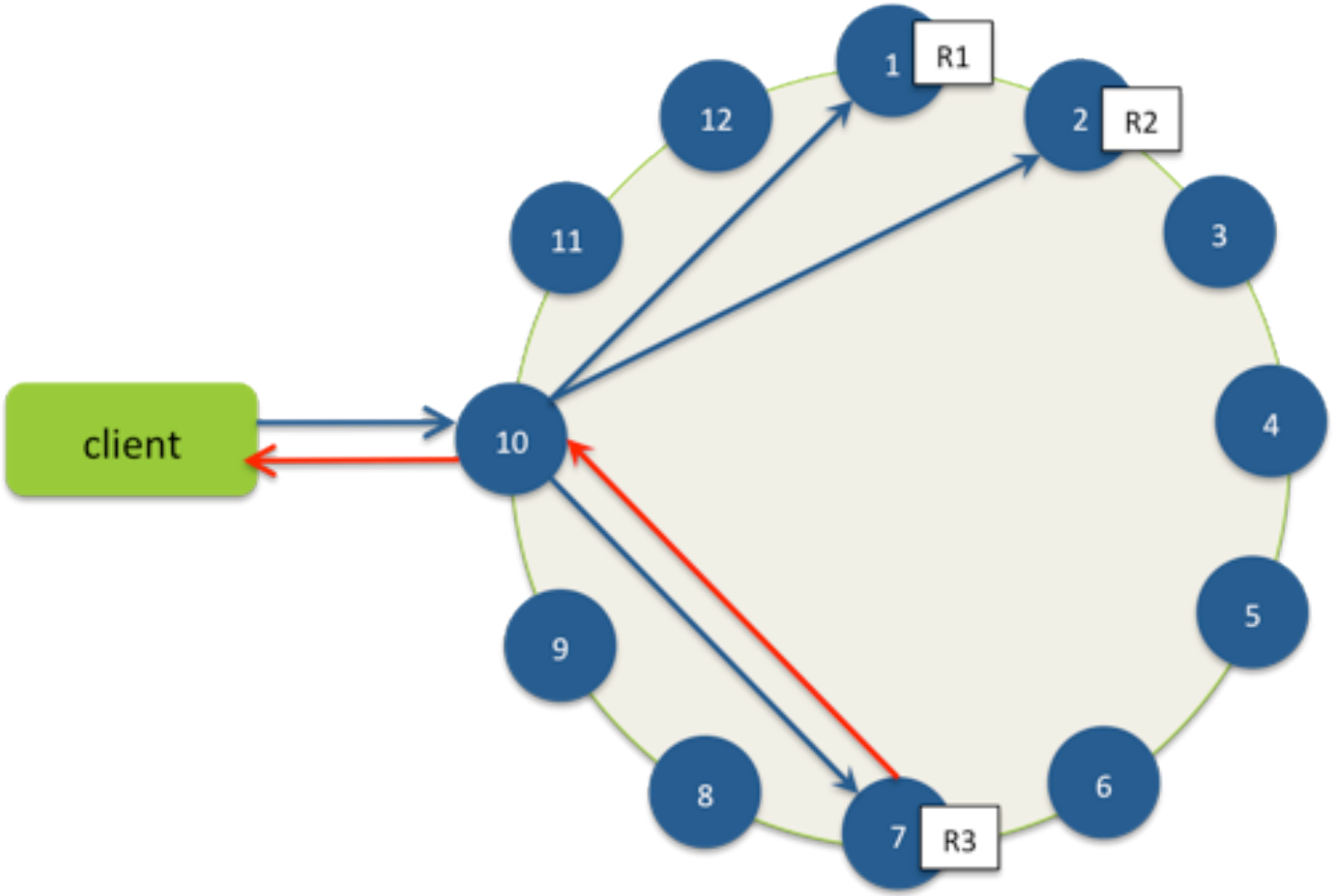
Requisições de Escrita

- Qualquer nó pode receber uma requisição de escrita (coordenador)
- O coordenador envia a ordem de gravação para todas as réplicas que possuem o dado que está sendo escrito
- Contanto que todos os nós estejam acessíveis, eles irão receber a requisição de escrita independente do nível de consistência
- O nível de consistência de escrita determina quantas réplicas precisam responder com sucesso para que a escrita seja considerada bem sucedida

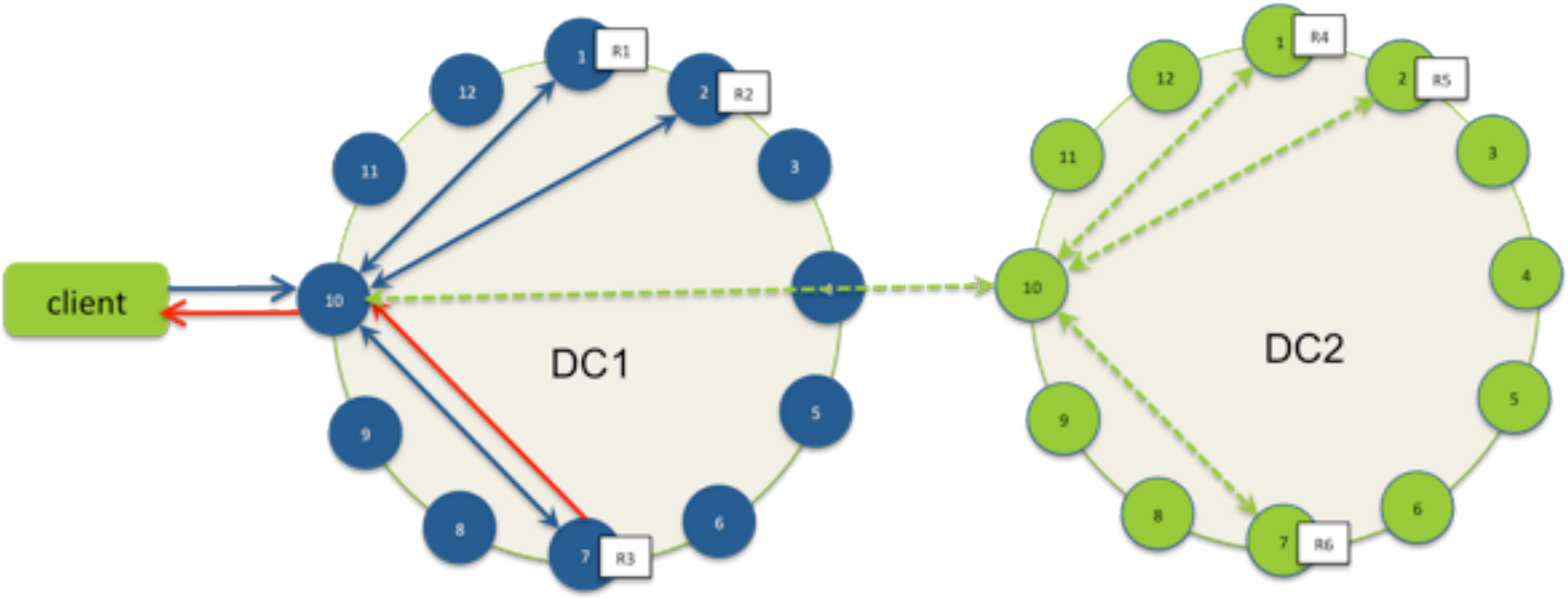
Replicação da Escrita

- Cada operação de escrita inclui um timestamp
- Deletes são um caso especial de escrita, chamado "tombstone"
- Os dados são sempre replicados para cada réplica
- Se o nó está indisponível, dados não recebidos são reapresentados quando o nó voltar a estar disponível

Requisições de Escrita



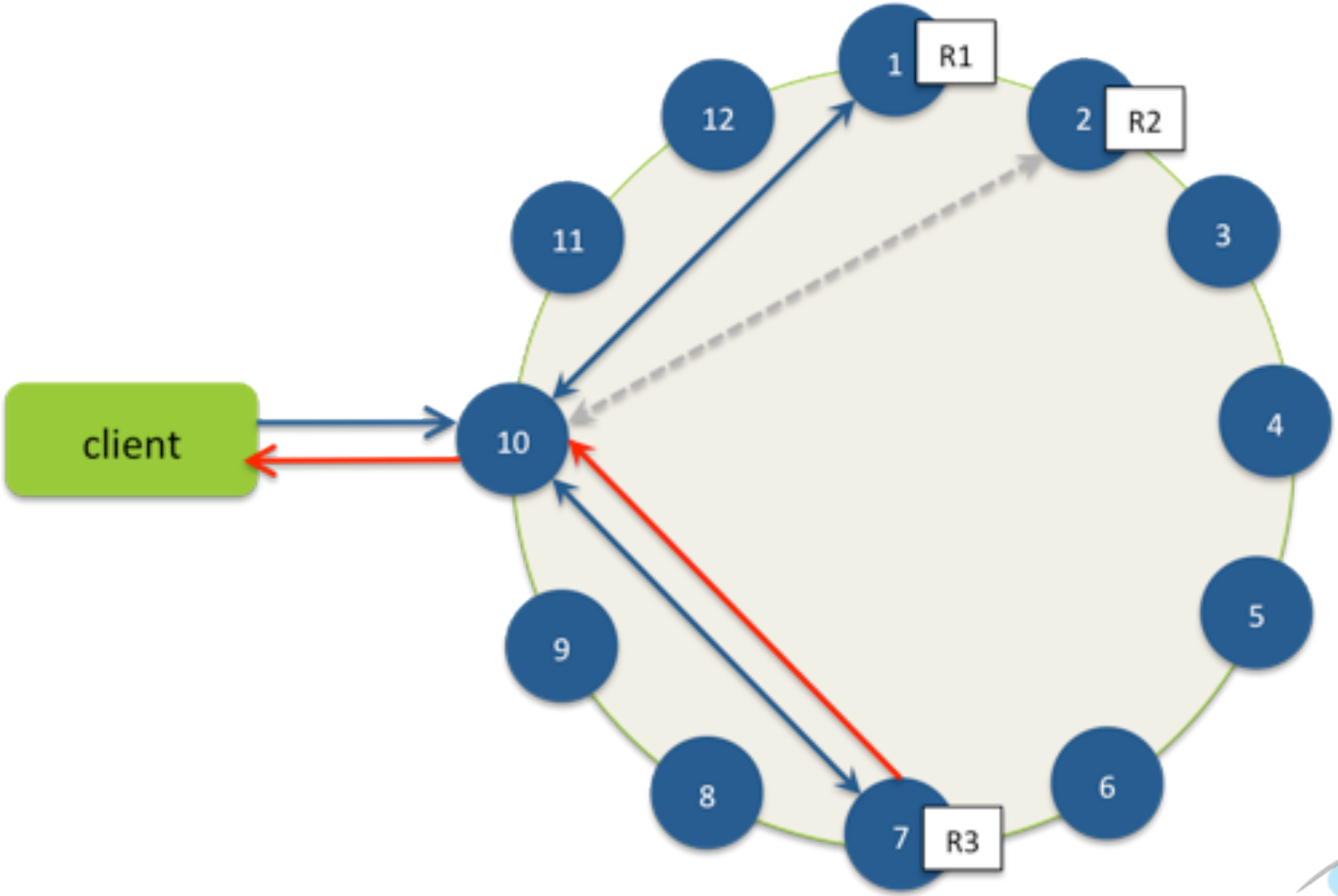
Requisições de Escrita em Múltiplos Datacenters



Requisições de Leitura

- Qualquer nó pode ser consultado. Atua como coordenador
- Coordenador contata os nós que possuem a chave requisitada
- Nível de consistência determina o número de réplicas contatados pela solicitação de leitura
- As linhas de diferentes réplicas são comparadas em memória para verificar a consistência
- A linha com o dado mais recente é enviada de volta ao cliente
- Solicitações de read-repair são enviadas em background para todas as réplicas adicionais

Requisições de Leitura



Referências

- http://cassandra.apache.org/
- http://docs.datastax.com/en/cassandra/3.0/index.html