

OrientDB

Greta Augat Abib

Pontos Abordados

Banco de Dados Orientados a Grafos

OrientDB

Edições e Licenças

Conceitos e Estruturas Básicas do OrientDB

Arquitetura Distribuída

Demo

Banco de Dados orientados a Grafos

Aplicações onde interconectividade de dados é importante (ex: redes sociais)

Consultas de forma eficiente, junções são eliminadas

Boa performance para aplicações altamente focadas no relacionamento dos dados, levando a uma modelagem direta.

.

OrientDB



Sistema de gerenciamento de banco de dados **NoSQL** de software livre

Escrito em **Java**

Banco de dados de múltiplos modelos , suportando:

- Document
- Graph
- Key/Value
- Real Object

Relacionamentos são gerenciados como em bancos de dados de grafos com conexões diretas entre registros

Edições e Licenças

OrientDB Community Edition: gratuito para qualquer uso (licença do **Apache 2**). Software de código aberto construído pela comunidade. Recursos como dimensionamento horizontal, tolerância a falhas, armazenamento em cluster, fragmentação e replicação não são desativados no OrientDB Community Edition.

<https://orientdb.org/>

OrientDB Enterprise Edition é a extensão comercial do OrientDB Community Edition criada para lidar com casos de uso mais robustos e exigentes. O OrientDB Enterprise Edition inclui recursos adicionais, como um gerenciador de perfis de consulta, configuração de clustering distribuído, gravação de métricas, um monitor ao vivo, Teleporter (uma ferramenta de migração) e alertas configuráveis.

Em 30 de janeiro de 2018, foi adquirido pela SAP.

<https://orientdb.com/>

Conceitos e Estruturas Básicas do OrientDB

Record: menor unidade que pode ser carregada e armazenada no banco de dados.

Pode ser armazenado em quatro tipos:

- Document
- Record Bytes
- Vertex
- Edge

RecordID (RID) : #<cluster>:<position>.

<cluster> número de identificação do cluster e <position> posição do registro no cluster

Conceitos e Estruturas Básicas do OrientDB

Documents: tipo de registro mais flexível disponível no OrientDB.

São definidos por classes de esquema com restrição definida, porém também suporta o modo sem esquema.

Os documentos podem ser facilmente manipulados por exportação e importação no formato JSON.

```
{
  "id"      : "1201",
  "name"    : "Jay",
  "job"     : "Developer",
  "creations" : [
    {
      "name"      : "Amiga",
      "company"   : "Commodore Inc."
    },
    {
      "name"      : "Amiga 500",
      "company"   : "Commodore Inc."
    }
  ]
}
```

Conceitos e Estruturas Básicas do OrientDB

RecordBytes

Mesmo que BLOB no RDBMS. Armazena registro juntamente com dados binários.

Vertex

Em DB orientados a grafos, a unidade mais básica de dados é o nó (node), no OrientDB é chamada de Vertex.

Edge

Conecta um vertex a outro. Podem ser bidirecionais ou podem apenas conectar dois vértices.

Conceitos e Estruturas Básicas do OrientDB

Class

Modelo extraído de POO.

Os dados são armazenados na forma de coleção, enquanto os dados do modelo de banco de dados relacional são armazenados em tabelas.

Podem ser sem esquema, repletas de esquema ou mistas.

Podem herdar de outras classes, criando árvores de classes. Cada classe tem seu próprio cluster ou clusters (criados por padrão, se nenhum for definido).

Conceitos e Estruturas Básicas do OrientDB

Cluster

Local onde um grupo de registros é armazenado.

Usado para armazenar registros, documentos ou vértices.

Por padrão, é criado 1 cluster por classe

Todos os registros de uma classe são armazenados no mesmo cluster com o mesmo nome da classe.

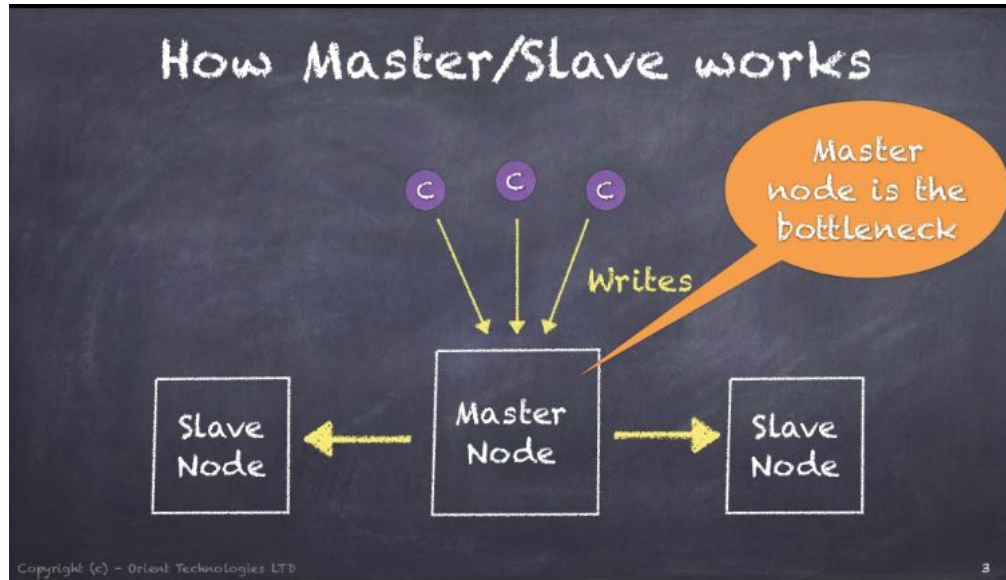
Possível criar até 32.767 (2^{15-1}) clusters em um banco de dados.

A classe CREATE é um comando usado para criar um cluster com nome específico.

Arquitetura Distribuída

Em 2012, o modelo de Master/Slave era utilizado para replicação dos dados.

Problema: Gargalo (bottleneck) no nó mestre (master).



Arquitetura Distribuída

Nova arquitetura distribuída:

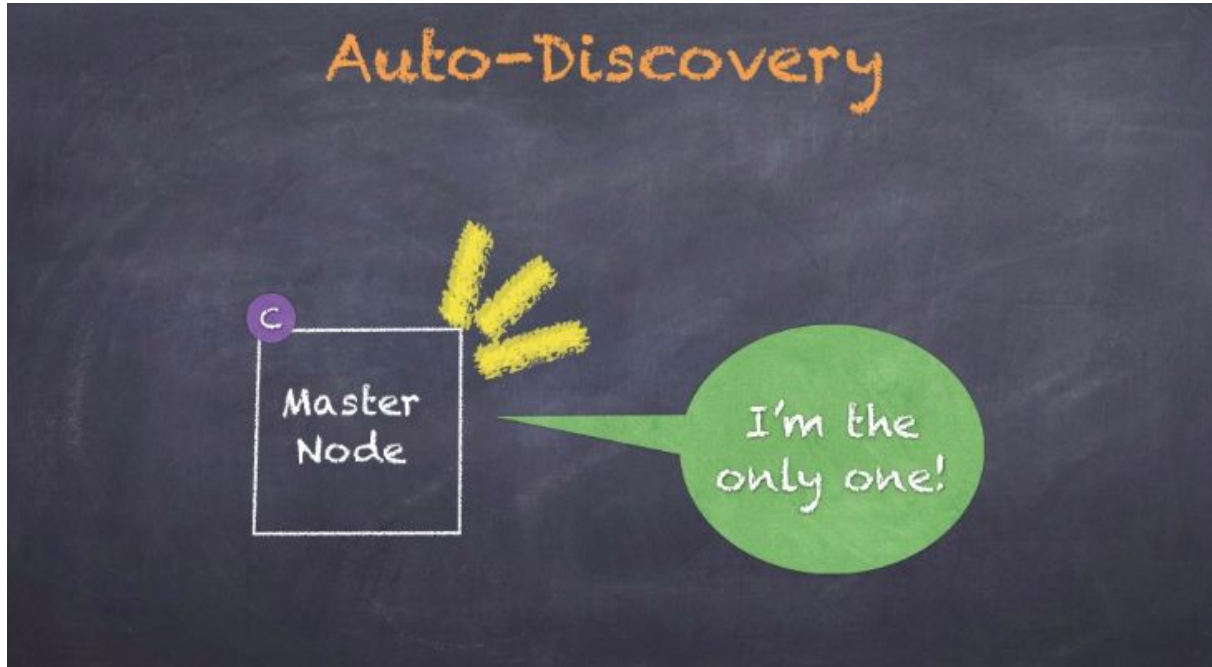
- Multi-mestre (Multi-Master): todos os nós devem aceitar escritas
- Sharding: divide dados em várias partições
- Melhor failover
- Configuração simplificada com detecção automática (Auto-Discovery)

Arquitetura Distribuída

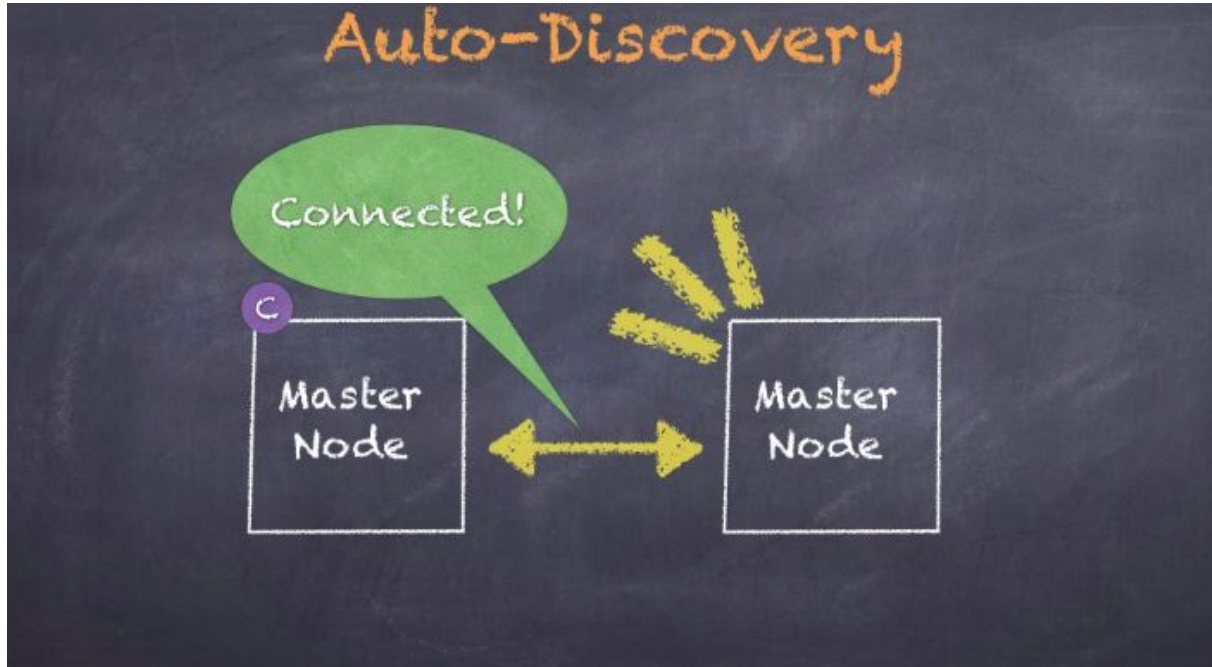
Regras utilizadas provenientes do **Hazelcast Open Source** - <https://hazelcast.org/>

- Auto descoberta - Auto discovering (Multicast / TCP-IP / Amazon)
- Filas para solicitações e respostas
- Armazenar metadados em mapas distribuídos
- Bloqueios Distribuídos (Distributed Locks)

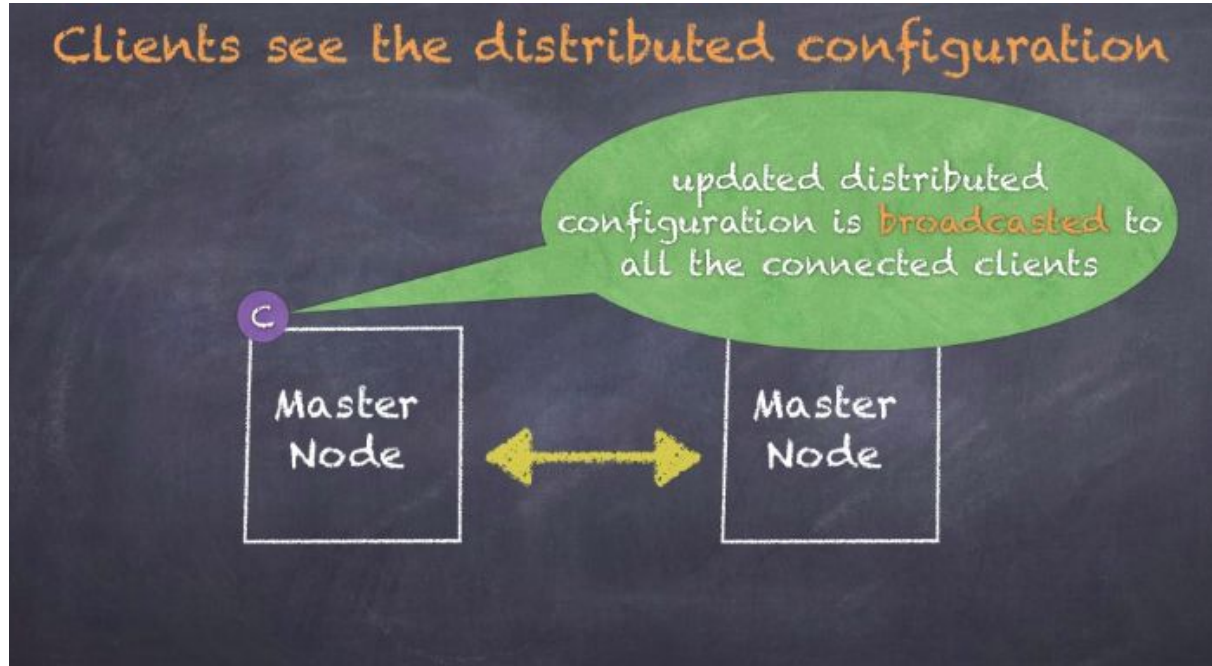
Arquitetura Distribuída



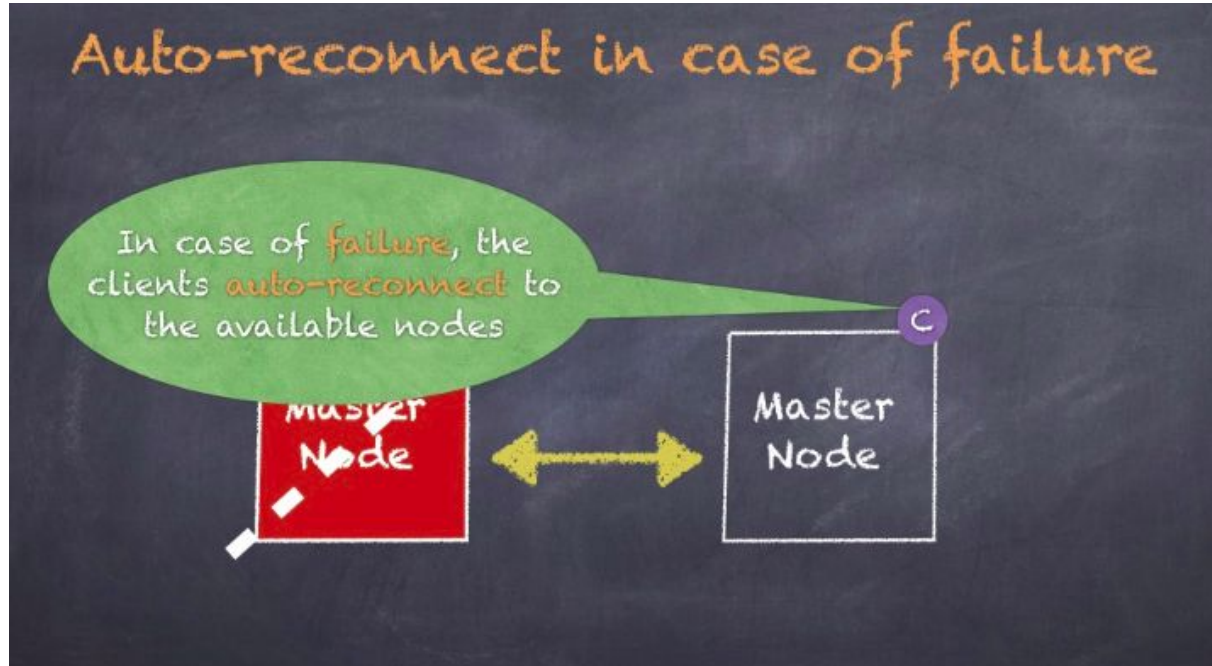
Arquitetura Distribuída



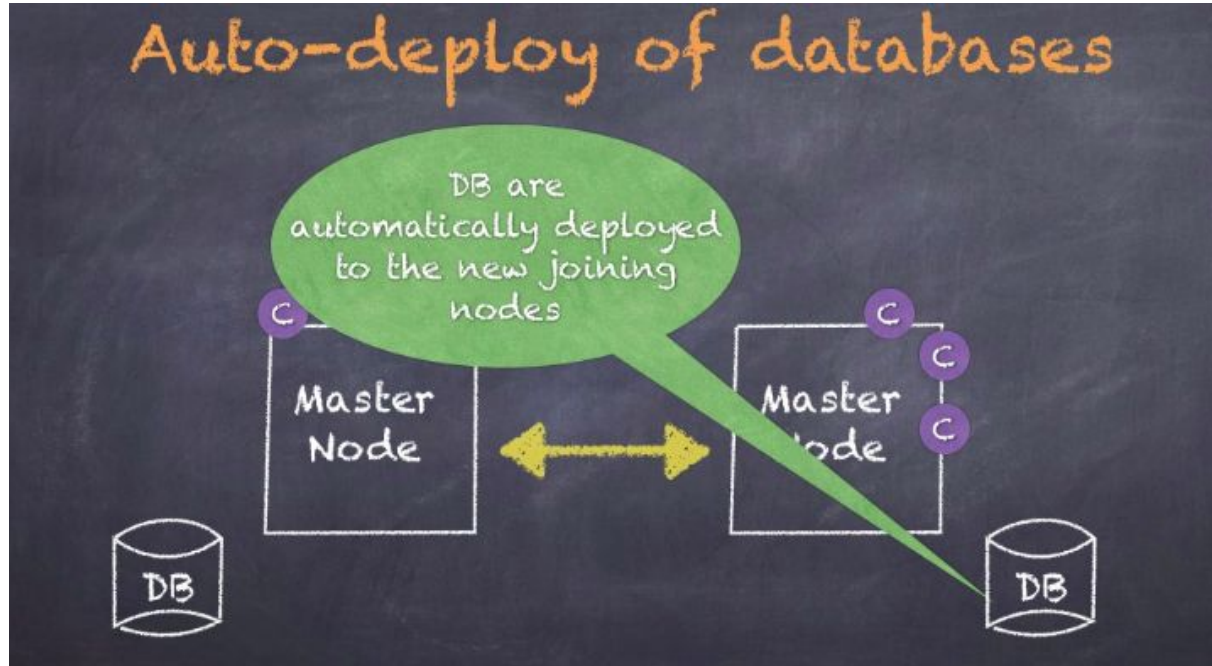
Arquitetura Distribuída



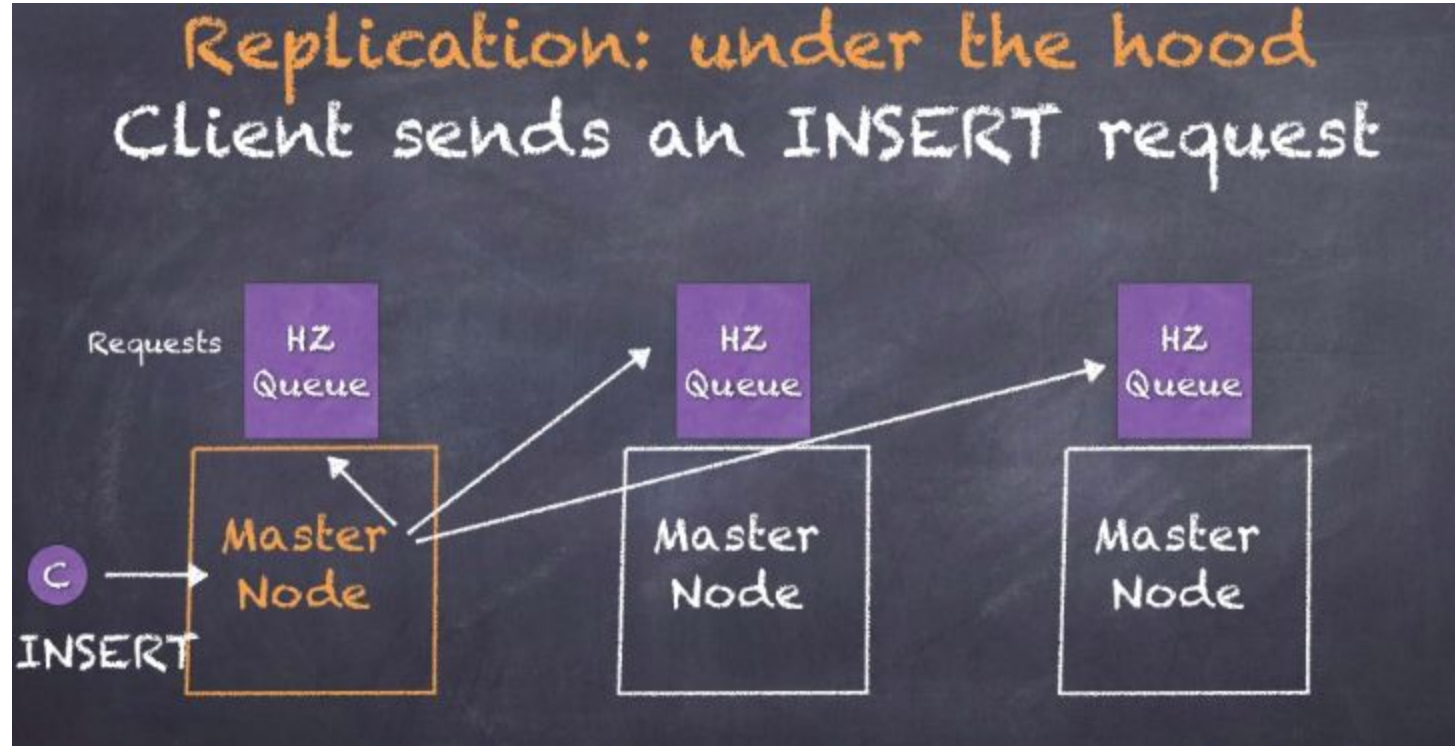
Arquitetura Distribuída



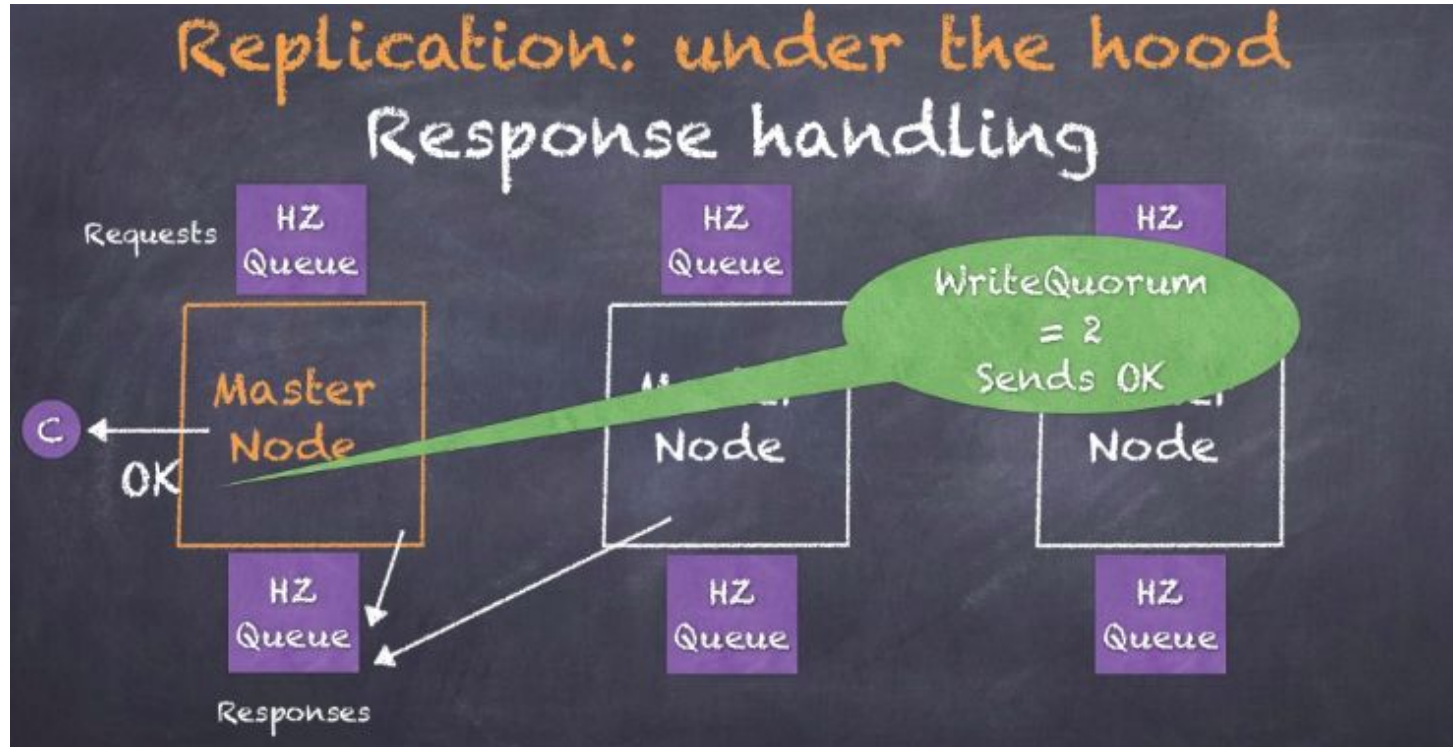
Arquitetura Distribuída



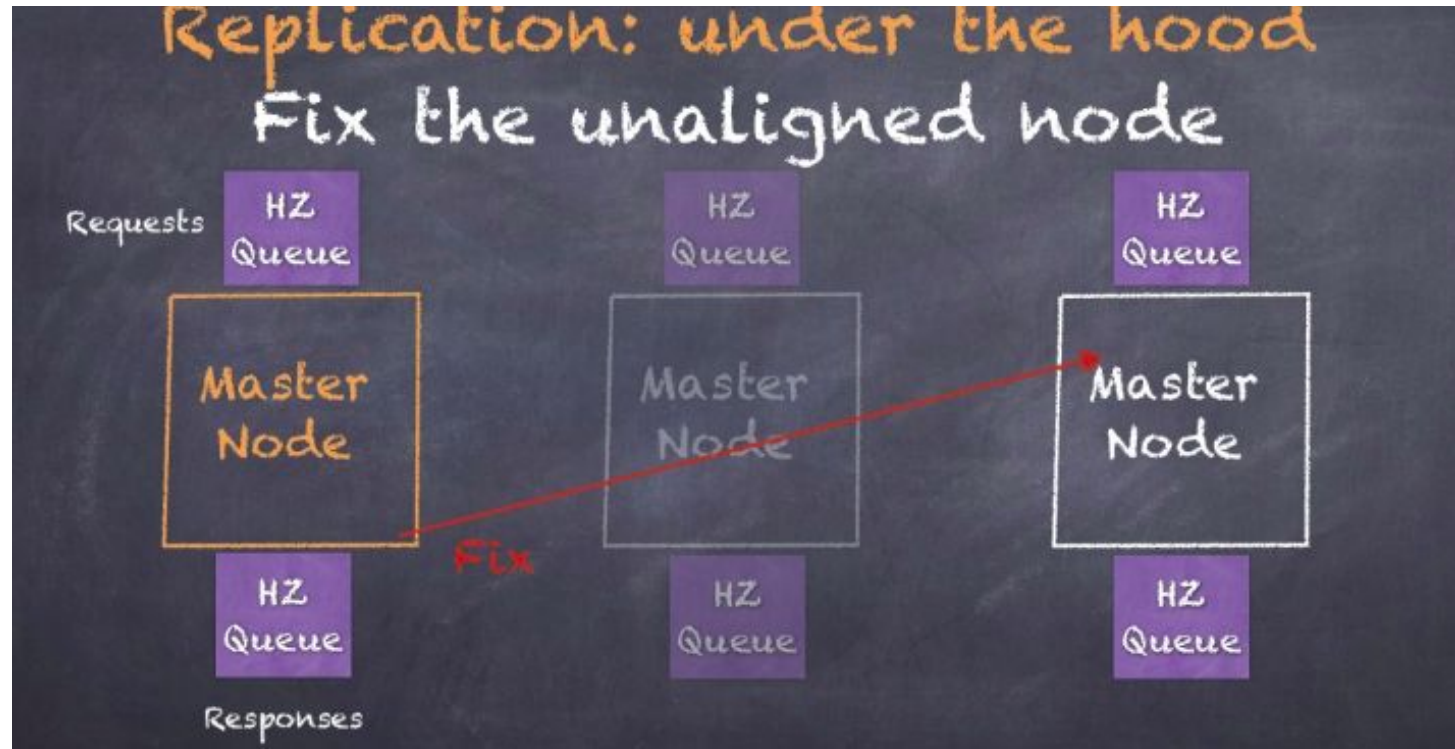
Arquitetura Distribuída



Arquitetura Distribuída



Arquitetura Distribuída



DEMO

<https://github.com/gretaaugat/orientdb>