



ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

- SISTEMAS DISTRIBUÍDOS-

APACHE ZOOKEEPER

Grupo 02

JUNHO de 2015

CARLOS SERRA	36907
ADRIANO SOUSA	38205
MARTA NASCIMENTO	38222

SISTEMAS DISTRIBUÍDOS

PROBLEMAS?

- Dois eventos ocorridos no sistema podem ser conhecidos, em sítios diferentes, segundo uma ordem de ocorrência diferente.
- Pedidos simultâneos envolvendo alterações de estado partilhado.
- Consistência.

SINCRONIZAÇÃO DISTRIBUÍDA

O QUE É?

- coordenação e sincronização de tomada de decisões num ambiente em que os múltiplos processos se executam em processadores diferentes, interligados por um sistema de comunicações que apresenta atrasos de transmissão não desprezáveis;

SINCRONIZAÇÃO DISTRIBUÍDA

SOLUÇÕES?

- Algoritmo de Christian
- Algoritmo de Berkeley
- Algoritmo de Lamport
- Algoritmo Bully
- Exclusão Mútua
- Token Ring

...



PROBLEMAS:

- Falta de segurança
- Falhas de comunicação
- Conhecimento de todos os tipos de algoritmos de sincronização existentes para saber aplicar o mais correto.



APACHE ZOOKEEPER

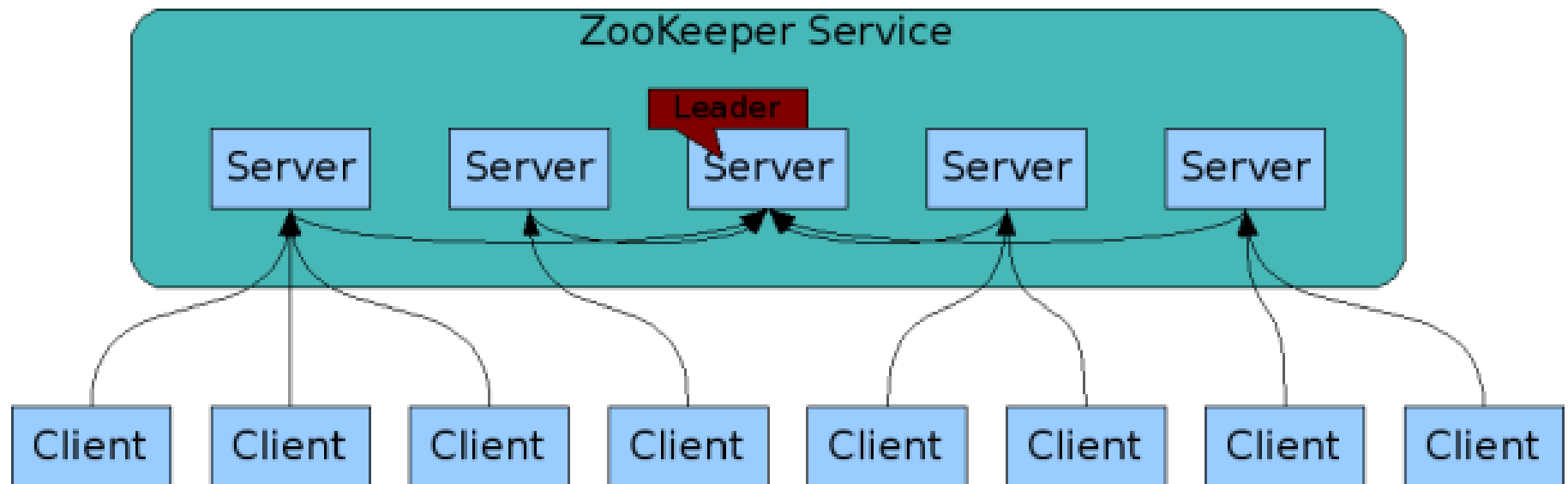


O QUE É?

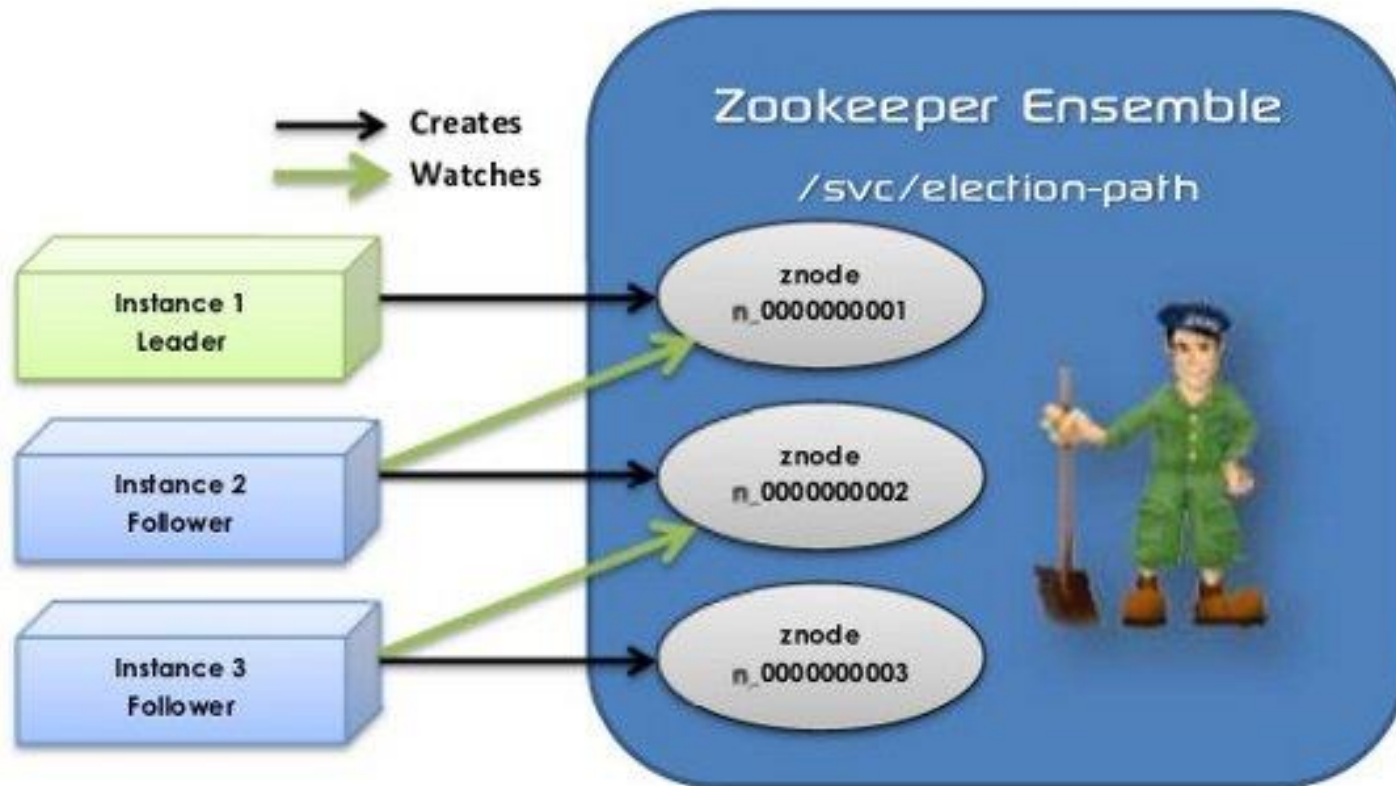
- Projeto Apache Open-Source.
- Implementado em JAVA e C.
- Serviço distribuído e de alta confiança.
- Serviço de coordenação de aplicações distribuídas permitindo implementar consensos, gestão de grupos, eleições.
- Serviço centralizado de manutenção de informação de configuração.
- Proporciona sincronização distribuída.



COMO FUNCIONA?

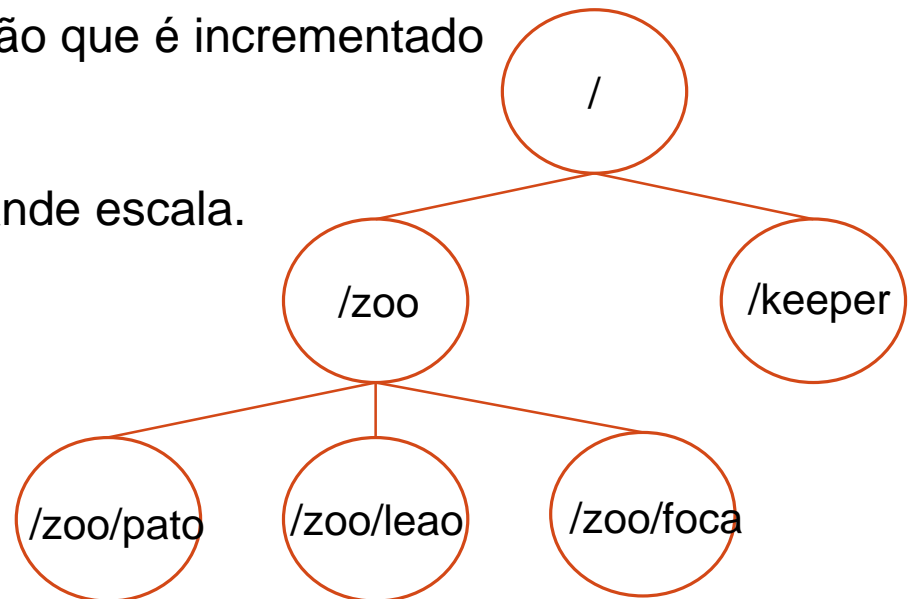


ELEIÇÃO DO LÍDER



MODELO DE DADOS

- Contém um espaço de nomes hierárquico
- Cada nó é designado por ZNode e contém dados.
- Cada ZNode pode conter nós filhos.
- É representado por um caminho (absoluto e separado por barras).
- Cada nó tem um número de versão que é incrementado sempre que este nó é alterado.
- Não funciona para objetos de grande escala.



TIPOS DE ZNODE

- Persistente
- Efémoro
- Sequencial

WATCHES (NOTIFICAÇÕES)

- Clientes registam-se num ou vários ZNodes.
- Quando existe alteração de um ZNode, esse notifica todos os clientes que se registaram nele.
- 1 registo → 1 notificação.
- Notificações ordenadas por registo (FIFO).

WATCHES

(REGISTO EM EVENTOS)

- Evento de criação.
- Evento de alteração de estado.
- Evento de eliminação.
- Evento sobre estado dos nós filhos.

API

- Fornece uma interface de programação muito simples de utilizar
- Operações:
 - Create
 - Delete
 - Exists
 - Get data
 - Set data
 - Get children
 - Sync

API

- Síncrona

```
Stat exists(String path, Watcher watcher)
```

```
exists("/zoo/pato/CONFIGS", null)
```

- Assíncrona:

```
Stat exists(String path, Watcher watcher, AsyncCallback.StatCallback cb, Object ctx)
```

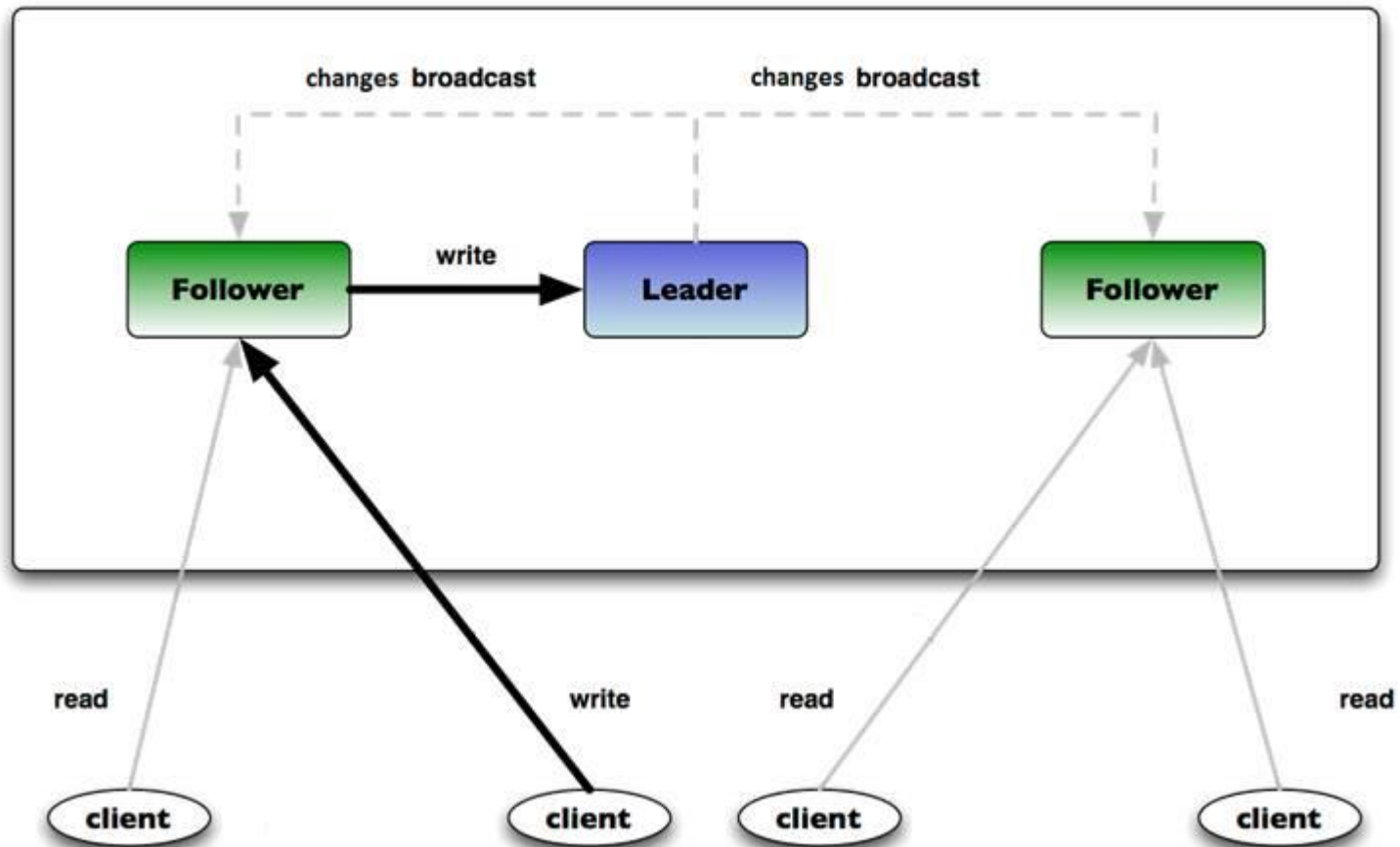
```
exists("/zoo/pato/CONFIGS", null, new StatCallback(){  
    @Override  
    public void processResult(int rc, String path, Object ctx, Stat atat){  
        //process result when called back later  
    }  
}, null);
```

COMO É QUE O ZOOKEEPER GARANTE A
SINCRONIZAÇÃO DISTRIBUÍDA?

ZAB – ZooKeeper Atomic Broadcast

COMO É QUE O ZOOKEEPER GARANTE A SINCRONIZAÇÃO DISTRIBUÍDA?

ZAB



APLICAÇÕES



YAHOO!



rackspace[®]
the open cloud company



REFERÊNCIAS

- <http://zookeeper.apache.org/>
- <http://zookeeper.apache.org/doc/r3.1.2/api/org/apache/zookeeper/ZooKeeper.html>
- http://zookeeper.apache.org/doc/trunk/zookeeperProgrammers.html#ch_zkWatches
- <https://www.youtube.com/watch?v=2jzpkdYHwSQ>

OBRIGADO!