



APACHE

ZooKeeper™

SSD Poject ZooKeeper

Elia Calisesi

Luca Mengozzi



APACHE

ZooKeeper™

INTRODUZIUONE

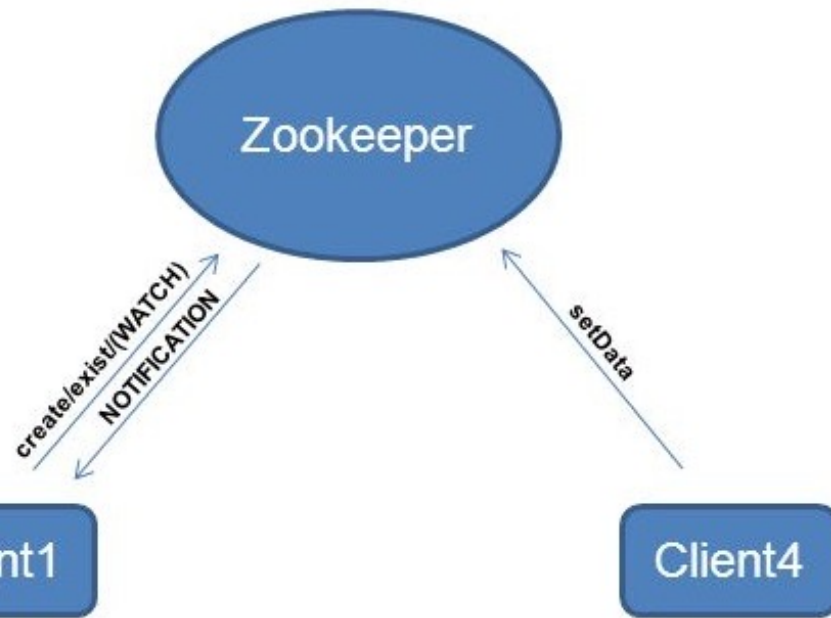
Zookeeper è :

- ◇ Progetto Apache open-source
- ◇ Sistema Distribuito
- ◇ Middleware

Componenti

ZooKeeper è composto da :

- ◊ Znode
- ◊ Watches

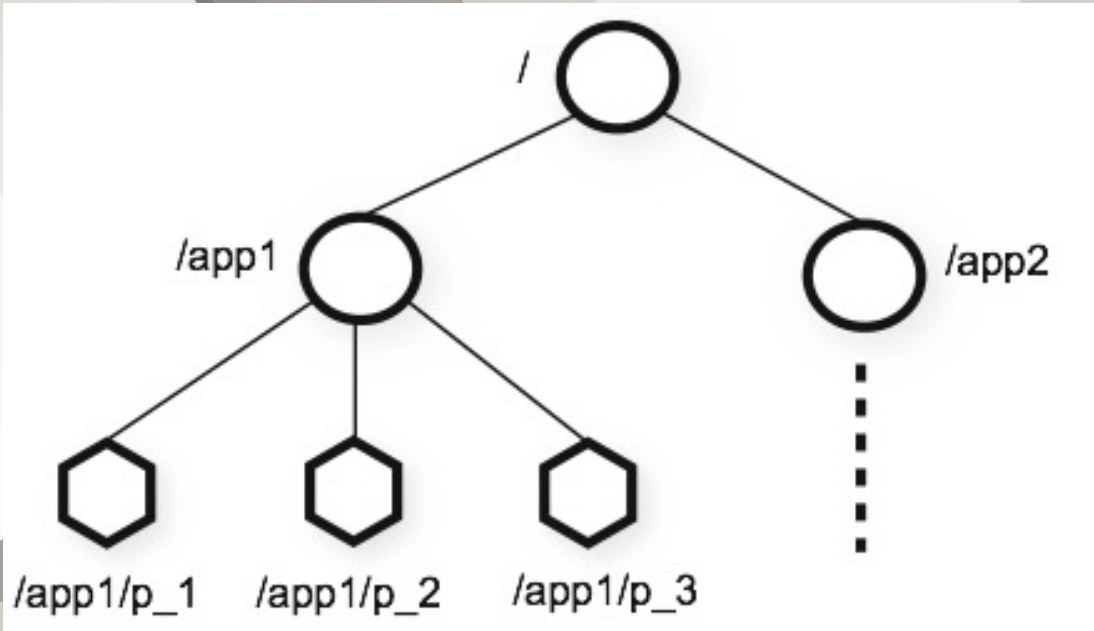


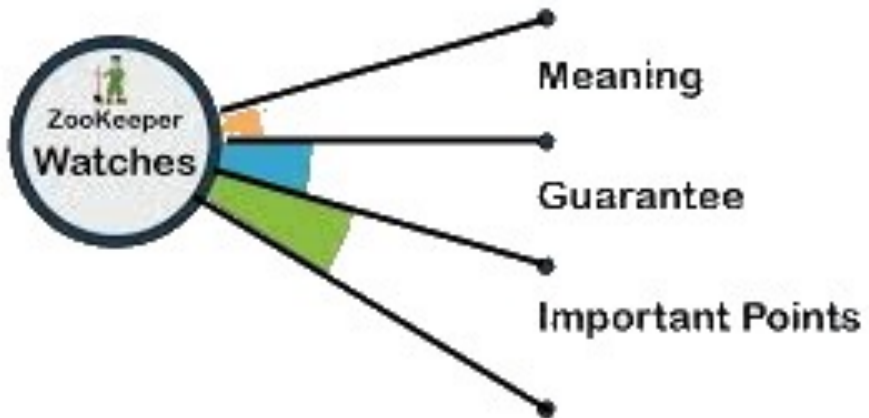
Znode

È composto da Timestamp e numero di versione

Operazioni:

- ◇ Creazione
- ◇ Ottenimento dati
- ◇ Inserimento dati
- ◇ Creazione di un figlio
- ◇ Verifica stato nodo
- ◇ Rimozione





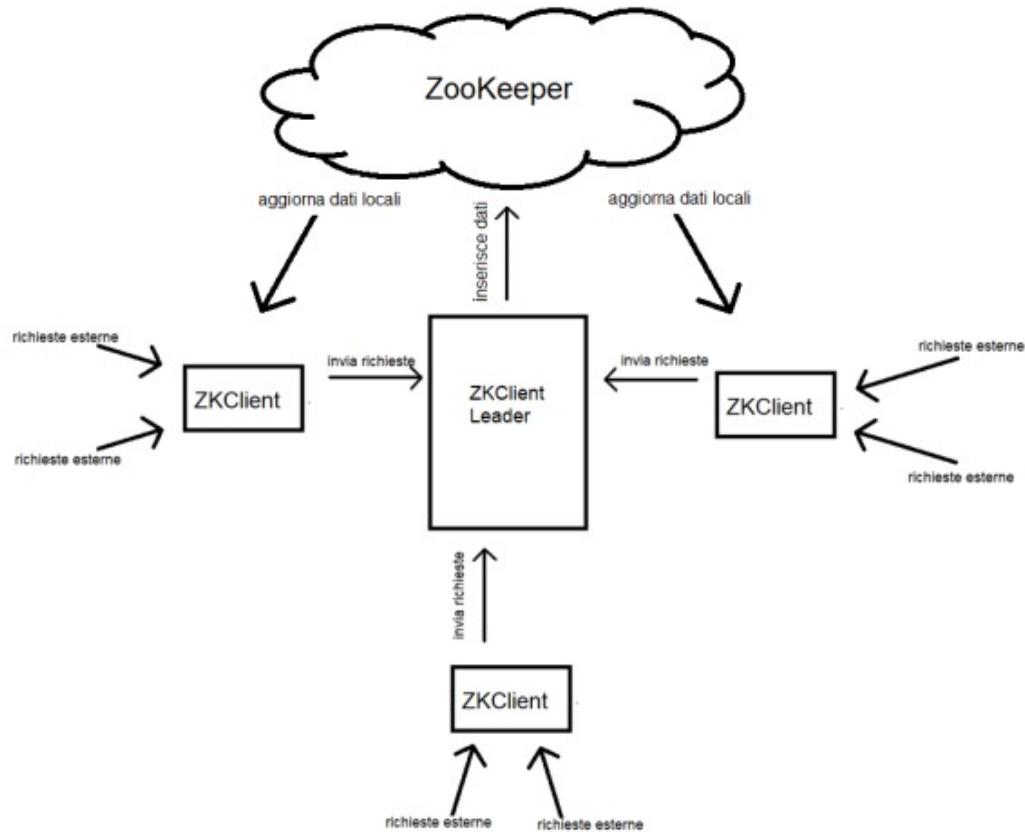
Watches

Viene associato a uno Znode per intercettare i cambi di stato.

Possono essere:

- ◊ One time trigger
- ◊ Risposta al client
- ◊ Watch su dati e nodi

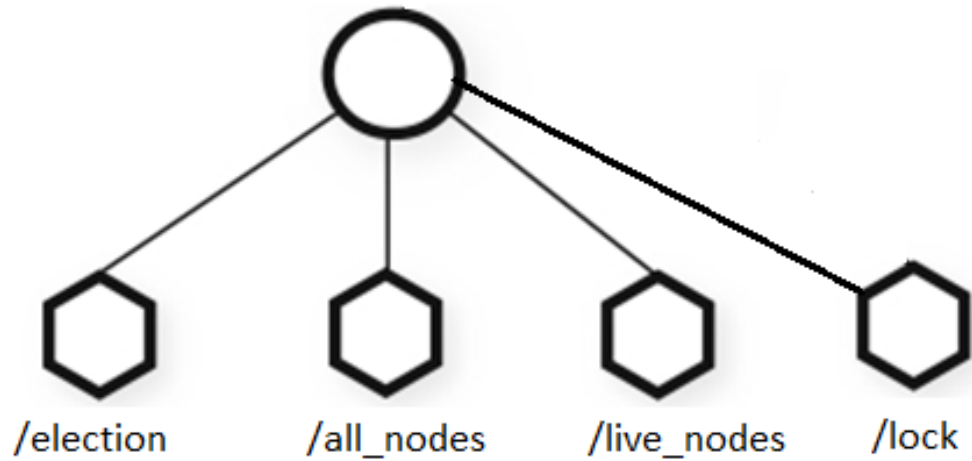
Architettura



- ◇ Server ZooKeeper
- ◇ Client ZooKeeper, suddivisi a loro volta:
 - ◇ Leader
 - ◇ Slave

Composizione servizio:

- ◇ All nodes,
- ◇ Live nodes,
- ◇ Election,
- ◇ Lock



Require: *watch*

```
path ← process(event)
if path = '/election' then
    setwatch('/election')
else if path = '/lock' then
    setwatch('/lock')
end if
if path = '/election' then
    insertWatch('/election', PERSISTENT)
    getLeader()
    if amILeader then
        iAmLeader ← true
    end if
else if path = '/lock' then
    if AmINextServed then
        iAmNext ← true
    else
        insertWatch('/lock', PERSISTENT)
    end if
end if
```

```
if exists/lock then
    res ← get(nodesInLock)
    order(res)
    first ← getFirst(res)
    data ← getData(first)
end if
```

```
if exists/election then
    res ← get(nodeInElection)
    order(res)
    first ← getFirst(res)
    data ← getData(first)
end if
```

Algoritmi:

- ◈ Gestione watch
- ◈ Gestione lock
- ◈ Gestione election



Testing

JUnit

- ◊ Verifica leader
- ◊ Verifica dati
- ◊ Verifica client attivi



APACHE

ZooKeeper™

Conclusioni

- ◊ API semplici
- ◊ Pensato su più nodi
- ◊ Elezione leader più complesso se distribuito
- ◊ Utilizzato ebay, Yahoo!, RockSpace e Solr