

Zookeeper

1/2 ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them, which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed. 1/2

Zookeeper

<http://zookeeper.apache.org/>

<https://github.com/apache/zookeeper>

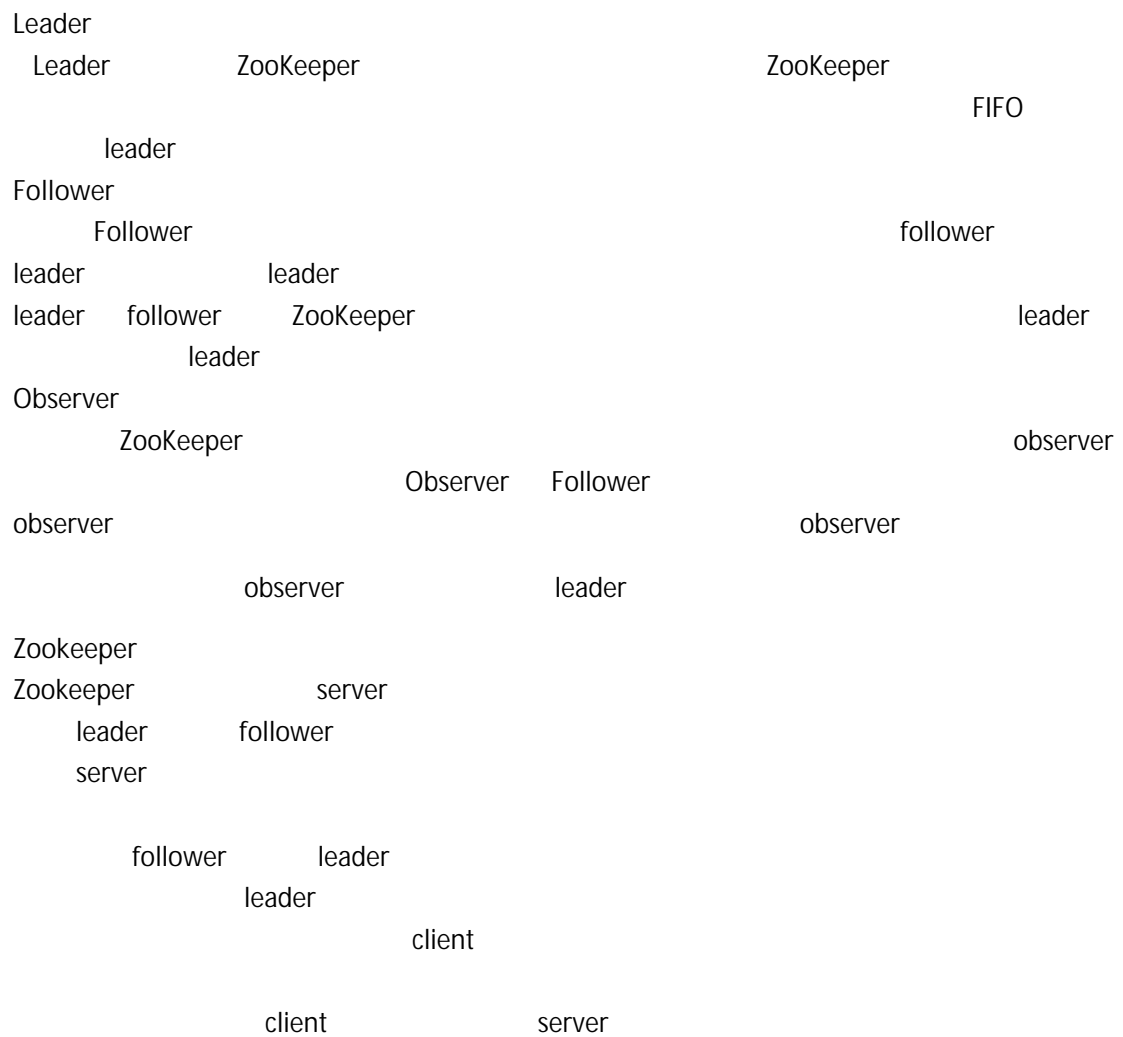
Zookeeper

- QQ 245553999

Zoo.cfg

tickTime	2000

syncLimit	Leader follower initLimt*ticktime
initLimt	zk initLimt*ticktime
dataDir	
dataLogDir	
clientPort	
Server.A=B:C:D	A: B IP C Leader follower D leader



- QQ 245553999

client

Session

TCP

org.apache.zookeeper.ClientCnxn

Zookeeper

Znode

Znode

Stat

czxid	zxid
mzxid	zxid
ctime	.
mtime	.
dataVersion	.
cversion	
aclVersion	ACL() .
ephemeralOwner	ephemeral , ephemeralOwner session id. ephemeral , ephemeralOwner 0. ephemeral

- QQ 245553999

dataLength	.
numChildren	.

2

PERSISTENT

PERSISTENT_SEQUENTIAL

EPHEMERAL session

EPHEMERAL_SEQUENTIAL

znode

znode

ACL(Access Control List)

org.apache.zookeeper.ZooDefs

ACL schemes

world

auth (cli addauth digest user:pwd

)

digest :

ip lp

ACL

CREATE:

READ

WRITE:

DELETE:

ADMIN:

Watcher:

KeeperState	EventType			
SyncConnected 3	None -1			
	NodeCreated 1	Watcher		Create
	NodeDeleted 2	Watcher		Delete/znode
	NodeDataChanged 3	Watcher		setData/znode
	NodeChildChanged 4	Wather		Create/child
Disconnected 0	None -1	ZooKeeper		
Expired -112	None -1		SessionExpiredException	
AuthFailed 4	None -1	1 schema 2 SASL	AuthFailedException	

The diagram illustrates the ZAB (ZooKeeper Atomic Broadcast) protocol flow. It shows the interaction between various components and the sequence of operations:

- ZAB** (ZooKeeper Atomic Broadcast) is the overall protocol.
- Zookeeper** is the central component.
- Zab** (ZooKeeper Atomic Broadcast) is the protocol used by the Zookeeper.
- leader** and **server** are the roles of the Zookeeper nodes.
- follower** is the role of the Zookeeper nodes that are not the leader.
- zk** (ZooKeeper) is the client interface.
- proposal** is the data being broadcast.
- epoch** is the current state of the Zookeeper.
- zk** (ZooKeeper) is the client interface.
- id (zxid)** is the unique identifier for the proposal.
- 32** and **64** are the sizes of the proposal and the zxid, respectively.
- leader** and **server** are the roles of the Zookeeper nodes.
- follower** is the role of the Zookeeper nodes that are not the leader.
- zk** (ZooKeeper) is the client interface.

- QQ 245553999

Leader

LOOKING, FOLLOWING, LEADING, OBSERVING