ZooKeeper 是什么?

- ZooKeeper 个 分 个为分 供 件，分 以 于 Zookeeper / 分 / Master 举 分 分 列
- ZooKeeper 出 关 ， 供
- Zookeeper 保 了 下分 :

  (1)

  (2)

  (3)

  (4)

  (5) ( )
- 以 中 任 ， 上 册了 ， 个 也 zookeeper 于写 ， 些 会 其他 zookeeper 且 ， 会 ， zookeeper ， 会 但 写 会下
- zookeeper 中 个 ， 全 ， 个 个 ， 个 为 zxid（Zookeeper Transaction Id） 会 于 ，也 中会 个zookeeper zxid

ZooKeeper 提供了什么?

- 件
- 制

Zookeeper 文件

- Zookeeper 供 个 （ 为 znode） 与 件 不 ， 些 以 关 ， 件 中 件 以 不
- Zookeeper 为了保 低 ， 内 中 了 个 ， 使 Zookeeper 不 于 ， 个 上 为1M

Zookeeper 么保 主从 状 同步?

- Zookeeper 制， 个 制保 了 个 server 之 个 制 做 Zab Zab 两 ， 们分别

复模式

- ，Zab 入了 ， 举出 ，且 server 了 leader 以 ， 了 保 了 leader server 具

广播模式

- leader follower 了 ， 以 了， 入 候 个 server 入 ZooKeeper 中， 会 下 ， leader， leader 到 ， 也 与 ZooKeeper Broadcast ， 到 leader 了 leader 了 分 followers

四 型 数据 Znode

- (1) PERSISTENT- 久 删 ， 则 于 Zookeeper 上
- (2) EPHEMERAL-临 临 与 会 ， 会 （ 与zookeeper 不 会 ）， 么 个 创 临 会
- (3) PERSISTENT_SEQUENTIAL- 久 久 ， 了 ， 会 个
- (4) EPHEMERAL_SEQUENTIAL-临 临 ， 了 ， 会 个

Zookeeper Watcher 机制 – 数据变更

- Zookeeper 允 个 Znode 册 个 Watcher ， 些 事件 了 个 Watcher， 会 个事件 分 ， Watcher 事件 做出业 上
- 作 制:

  (1) 册 watcher

  (2) watcher

  (3) watcher

Watcher 特

1. ， 个 Watcher ，Zookeeper 会 其从 储中 减 了 ，不 于 ， 会不 事件 ， 于

2. 串 Watcher 个串
3. 3.1 Watcher , 会 了事件， 不会 事件 具体内
   3.2 册 Watcher 候， 不会 Watcher 体传 到
   ，仅仅 中使 boolean 了
4. watcher event watcher 事件从 server 到 client , 个
   ，不 之 socket 信， 于 其他 不
   刻 到事件， 于 Zookeeper 供了 ordering guarantee， 事件 ，
   会 znode 了 以 们使 Zookeeper 不 到
   Zookeeper 保 , 保
5. 册 watcher getData exists getChildren
6. watcher create delete setData
7. 个 到 个 上 ，watch 会 以任 会 事件 与 个
   候， 到 watch client , , 先前
   册 watch， 会 册 全 个 况下，watch 会
   丢 ： 于 个 创 znode exist watch， 创 了， 且
   上之前 删 了， 况下， 个 watch 事件 会 丢

客户 注册 Watcher 实现

- (1) getData()/getChildren()/exist()三个 API，传入 Watcher
- (2) request， Watcher 到 WatchRegistration
- (3) Packet , request
- (4) 到 , Watcher 册到 ZKWatcherManager 中
- (5) , 册

服务 处理 Watcher 实现

1. Watcher 储 到 , 判 册 Watcher，
   ServerCnxn（ServerCnxn 代 个 , 了
   Watcher process , 以 个 Watcher ） 储 WatcherManager
   WatchTable watch2Paths 中
2. Watcher 以 到 setData() 事 NodeDataChanged 事件为例：
   2.1 WatchedEvent (SyncConnected) 事件 (NodeDataChanged) 以
   个 WatchedEvent
   2.2 Watcher 从 WatchTable 中 Watcher
   2.3 到； 上 册 Watcher
   2.4 到； 从 WatchTable Watch2Paths 中删 Watcher（从 以 出
   Watcher , 了）
3. process Watcher process 主 ServerCnxn TCP
   Watcher 事件

客户 回 Watcher

- SendThread 事件 ，交 EventThread Watcher
- Watcher 制 , , Watcher 了

ACL 权 控制机制

- UGO（User/Group/Others）

- 前 Linux/Unix 件 中使 ，也 使 制 件 制
- ACL（Access Control List） 制列
- **包括三个方 ：**
- （Scheme） （1）IP：从 IP 制 （2）Digest： ， 似于 username:password ，便于 分不 制 （3）World： 制 ， digest ， 个 “world:anyone” （4）Super：
- **授权对** 予 个 体，例 IP
- **权 Permission** （1）CREATE： 创 ，允 Znode 下创 （2）DELETE： 删 ，允 删 （3）READ： ，允 其 内 列 （4）WRITE： ，允 作 （5）ADMIN： ，允 ACL 关 作

Chroot 特

- 3.2.0 ， 了 Chroot ， 允 个 为 个 个 了 Chroot， 么 任何 作， 会 制 其 下
- Chroot， 个 于 Zookeeper ， 些 个 公 个 Zookeeper 下， 不 互

会 理

- 分 ： 似 会 中 ，以便于 Zookeeper 会 不 以
- 分 则： 个会 “下 ”（ExpirationTime）
- 公 ：

  ExpirationTime_ = currentTime + sessionTimeout

ExpirationTime = (ExpirationTime_ / ExpirationInrerval + 1) *

ExpirationInterval , ExpirationInterval Zookeeper 会 , tickTime

服务器

- Leader

  (1) 事 ，保 事

  (2) 内

- Follower

  (1) 事 , 事 Leader

  (2) 与事 Proposal

  (3) 与 Leader 举

- Observer

  (1) 3.0 以 入 个 , 不 事 上 事

  (2) 事 , 事 Leader

  (3) 不 与任何

Zookeeper 下 Server 工作状

- 具 ，分别 LOOKING FOLLOWING LEADING OBSERVING

  (1) LOOKING: Leader 于 , 会 为 前 中 Leader, 入 Leader 举

  (2) FOLLOWING: 前 Follower

  (3) LEADING: 前 Leader

  (4) OBSERVING: 前 Observer

数据同步

- 个 Leader 举之 ，Learner（Follower Observer ） Leader 册 Learner Leader 册 , 入
- : (以 传)
  - Learner Learder 册
  -
  -
- Zookeeper 分为 :

  (1) (DIFF)

  (2) 先 再 (TRUNC+DIFF)

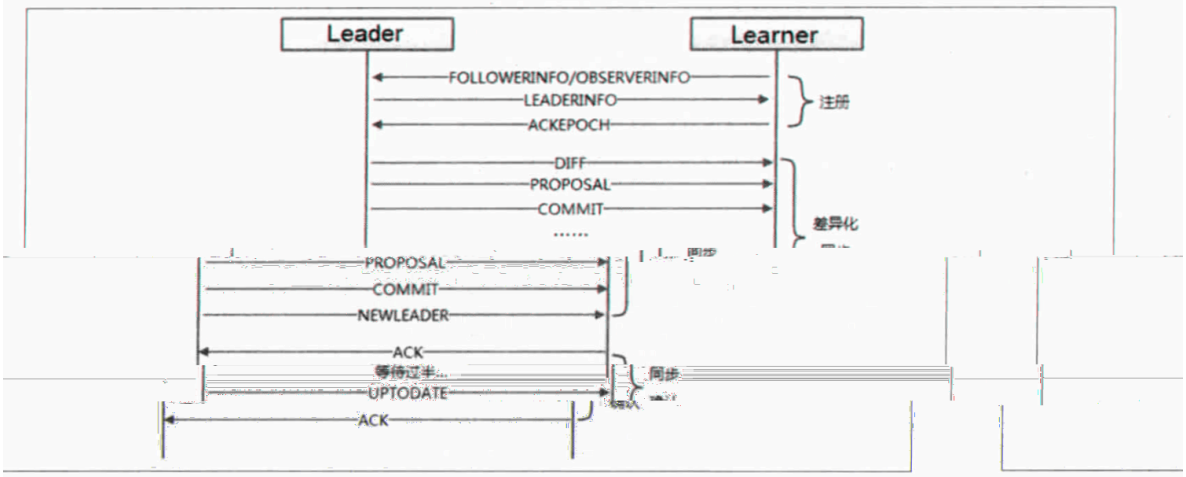
  (3) 仅 (TRUNC)

  (4) 全 (SNAP)

- 前，Leader 会 初 :
  - peerLastZxid：从learner 册 ACKEPOCH 中 lastZxid (Learner ZXID)
  - minCommittedLog：Leader Proposal 列committedLog中 ZXID

    - maxCommittedLog：Leader Proposal 列committedLog中 ZXID
- （DIFF ）：peerLastZxid介于minCommittedLog maxCommittedLog 之

整个直接差异化同步过程中涉及的 Leader 和 Learner 之间的数据包通信如图 7-50 所示。



差异化同步方式中 Leader 和 Learner 之间的数据通信 图 7-50. 直接差

- 先 再 （TRUNC+DIFF ）： Leader 个Learner 了 事 ， 么 Learner 事 -- 到Leader 上 ， 也 于peerLastZxid ZXID
- 仅 （TRUNC ）：peerLastZxid 于 maxCommittedLog
- 全 （SNAP ）：peerLastZxid 于 minCommittedLog 二：Leader 上 Proposal 列且peerLastZxid不 于lastProcessZxid

zookeeper 是如何保 事务 序 ？

- zookeeper 了全 事 Id ， proposal （ ） 出 候 上了 zxid，zxid 上 个 64 位 ， 32 位 epoch （ ； 元; 世; 代) leader ， leader 产 出 ，epoch会 ，低 32 位 产 proposal 候，会依 两 ， 先会 其他 server 出事 ， 且 ， 么 会

分布式 中为什么会有 Master主 ？

- 分 中， 些业 中 ，其他 以共享 个 ， 以 减 ， ，于 leader 举

zk 机如何处理?

- Zookeeper 也 ， 不 于 3 个 Zookeeper 也 保 个 ，其他 会 供
- 个 Follower ， 2 供 ， 为 Zookeeper 上 个副 ， 不会丢 ；
- 个 Leader ，Zookeeper 会 举出 Leader
- ZK 制 ， 供 ZK ， 剩 不到 作，

所以

- 3 个 cluster 以 1 个 (leader 以 到 2 >1.5)
- 2 个 cluster 不 任何 1 个 了(leader 以 到 1 <=1)

zookeeper 均 和 nginx 均 区别

- zk 以 ，nginx ，其他 写 件；但 nginx zk ， 业

Zookeeper 有哪几 几 模式?

- Zookeeper 三 ：
  1. ： 上 ；
  2. ： ；
  3. 伪 ： 个 Zookeeper 例

最少 几台机器， 则是 样 ？ 中有 3 台服务器，其中 个 机， 个时 Zookeeper 可以使 吗?

- 则为 2N+1 ，N>0， 3 以 使 ，以 使

支持动 添加机器吗?

- 其 了，Zookeeper 不 两 ：
- 全 ：关 Zookeeper ，修 之 不 之前 会
- 个 ： 则下， 不 个 供

- 3.5

Zookeeper 对 watch 听 是永久 吗? 为什么不是永久 ？

- 不 ： 个 Watch 事件 个 ， 了 Watch 了 候，则 个 了 Watch ，以便 们
- 为什么不 久 ，举个例 ， ， 况下， 到 ，
- getData(“/ A”,true)， A 了 删 ， 会 到 watch 事件，但 之 A 了 ， watch 事件， 不再

- 中， 况下， 们 不 ，

Zookeeper java 客户 有哪 ？

- java ：zk zkclient Apache Curator

chubby 是什么，和 zookeeper 比你 么 ？

- chubby google ， 全 paxos ，不 zookeeper chubby ，使 zab ，paxos

几个 zookeeper 常　　命令

- 　　令：ls get set create delete

ZAB 和 Paxos　法　　与区别?

- 　　：

  (1) 两　　个　似于 Leader　　，　其　　个 Follower

  (2) Leader　　会　　Follower 做出　　，　会　个　　交

  (3) ZAB　　中，　个 Proposal 中　　个 epoch 值　代　　前　Leader　　，Paxos 中　为 Ballot

- 不　　：

  ZAB　　分　　主　　(Zookeeper)，Paxos　　分

Zookeeper　典型应　场景

- Zookeeper　　个典　　/　　分　　与　　，　人　以使　　分
- 　　Zookeeper 中丰　　交　使　，　Watcher 事件　　制，　以　　便　　列分　　中　会　　，　：

  (1)　　/

  (2)

  (3)

  (4) 分　　/

  (5)

  (6) Master　举

  (7) 分

  (8) 分　　列

1 数据发布/

介

- 　　/　　，　　中　，　　义　　供

- 　　(　信　)
- 　　(　信　)　　中

模式

- Push
- Pull

数据(　信　)特

(1)　　(2)　内　　会　　(3)　中　　共享,

- 　：　列　信　　关　　信

基于 Zookeeper 实现方式

- · 储: (信) 储到 Zookeeper 上 个
- · : 初 从 Zookeeper , 上 册 个 Watcher
- · : , Zookeeper , Zookeeper会 到 , 到

2 均

- zk
- , 利 zk 创 个全 , 个 以作为 个 , 中 , 供 , 个

分布式 和协

- 于 : 作人 制 个 , zk 些 册了 个 watcher
- 于 况 : 个 作 个 下创 个临 作 , 以 作 全 况

zk 命名服务（文件 ）

- , 利 zk 创 个全 , , 个 以作为 个 , 中 , 供 , 个

zk 理（文件 机制）

- 分 不 上, 信 zk znode 下, , 也 znode , 以 zk 中 个 内 , 利 watcher 个 , 从

Zookeeper 理（文件 机制）

- 乎两 : 出 入 举 master
- 于 , 下创 临 ,
- , 与 zookeeper , 其 创 临 删 , 其他 到 : 个兄 删 , 于 , 人 : 上 了
- 入也 似, 到 : 兄 入, highcount 了, 于 二 , 们 下, 创 临 , 作为 master

Zookeeper 分布式 （文件 机制）

- 了 zookeeper 件 , 以分为两 , 个 保 , 个 制
- 于 , 们 zookeeper 上 个 znode 作 , createznode 创 /distribute_lock , 创 个 也 了 删 创 distribute_lock 出
- 于 二 , /distribute_lock 先 , 下 创 临 , master , , 删 , 依 便

Zookeeper 列 理（文件 机制）

- 两 列:

  (1) 列, 个 列 , 个 列 , 则 到
  (2) 列 FIFO 入 出 作

- ， 下创 临 ， 们
- 二 ， 分 中 制 ，入列 ，出列 下创 PERSISTENT_SEQUENTIAL ，创 Watcher 列， 列删 列 以 下Zookeeper znode 于 储，znode 储 列中 内 ，SEQUENTIAL 列 ， 出 于创 久 ， 以不 列 丢

Zookeeper 有哪 功 ？

1. ：；
2. 主 举：主 了之 以从 主，主 举 个 举 ，使 Zookeeper 以 个 ；
3. 分 ：Zookeeper 供两 ： 共享 个 使 ， 共享 共享， 写互 ， 以 个 ， 使 写 也 个 使 Zookeeper 以 分 制
4. ： 分 中， 使 ， ， 供 信

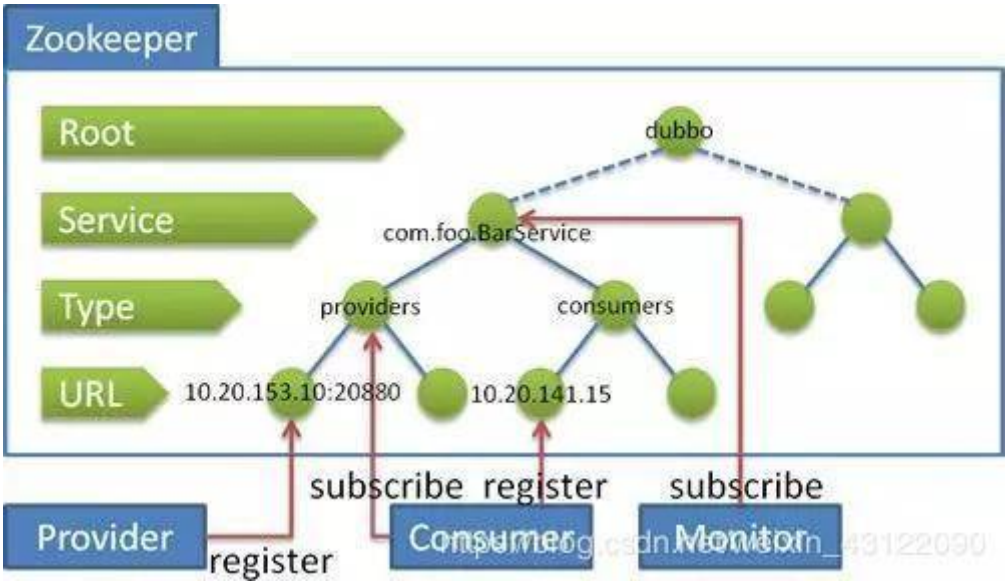
下 Zookeeper 机制?

- client 会 个 znode 个 watcher 事件， znode ， 些 client 会 到 zk ， client 以 znode 做出业 上

Zookeeper 和 Dubbo 关 ？

- Zookeeper 作 ： zookeeper 册 ， 个 个 供 ， ip 关 也 以 关 业 代 中 ，但 供 ， 不 代 会 供 zookeeper 制 以 ip 关 从列 中删 于 ， ， 不 代 况 zookeeper 册 ， 供 了 了
- dubbo： 中 具， 业 到 仓 入 供 ，dubbo 供 个 决 个 dubbo 个 ， 于你 上 什么 全 决于你 ， 像 个 ， 你 你 个 中 个分 册中 ，储 元 ，你 以 zk，也 以 别 ， zk
- zookeeper dubbo 关 ： Dubbo 册中 ， 以 不 储 介 册中 供 ， ZooKeeper，Memcached，Redis

  入了 ZooKeeper 作为 储 介，也 ZooKeeper 先 ， 册中 ， 到 候 分 ， 为了分 ， 个 ZooKeeper Web 以 到 ； ， 不 ， 之 ，ZooKeeper 具 ； ， 于 全 列 ， 供 候， ZooKeeper 上 /dubbo/${serviceName}/providers 下写入 URL ， 个 作 了 其他 Mast 举，分


Zookeeper
Root
Service
Type
URL
dubbo
com.foo.BarService
providers
consumers
10.20.153.10:20880
10.20.141.15
subscribe
register
subscribe
Provider
Consumer
Monitor
register