**主题概要**

1. zookeeper概要、背景及作用
2. 部署与常规配置
3. 节点类型

一、zookeeper概要、背景及作用

zookeeper产生背景：

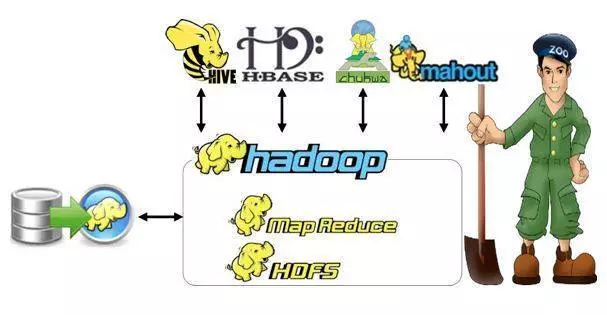
项目从单体到分布式转变之后，将会产生多个节点之间协同的问题。如：

1. 每天的定时任务由谁哪个节点来执行？
2. RPC调用时的服务发现?
3. 如何保证并发请求的幂等
4. ....

这些问题可以统一归纳为多节点协调问题，如果靠节点自身进行协调这是非常不可靠的，性能上也不可取。必须由一个独立的服务做协调工作，它必须可靠，而且保证性能。

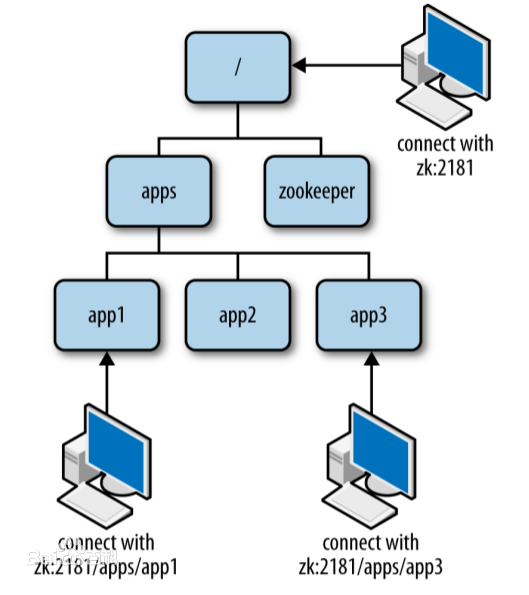
zookeeper概要：

ZooKeeper是用于分布式应用程序的协调服务。它公开了一组简单的API，分布式应用程序可以基于这些API用于同步，节点状态、配置等信息、服务注册等信息。其由JAVA编写，支持JAVA 和C两种语言的客户端。



znode 节点

zookeeper 中数据基本单元叫节点，节点之下可包含子节点，最后以树级方式程现。每个节点拥有唯一的路径path。客户端基于PATH上传节点数据，zookeeper 收到后会实时通知对该路径进行监听的客户端。



二、部署与常规配置

zookeeper 基于JAVA开发，下载后只要有对应JVM环境即可运行。其默认的端口号是2181运行前得保证其不冲突。

版本说明：

2019年5月20日发行的3.5.5是3.5分支的第一个稳定版本。此版本被认为是3.4稳定分支的后续版本，可以用于生产。基于3.4它包含以下新功能

* 动态重新配置
* 本地会议
* 新节点类型：容器，TTL
* 原子广播协议的SSL支持
* 删除观察者的能力
* 多线程提交处理器
* 升级到Netty 4.1
* Maven构建

另请注意：建议的最低JDK版本为1.8

文件说明：

* apache-zookeeper-xxx-tar.gz 代表源代码
* apache-zookeeper-xxx-bin.tar.gz 运行版本

下载地址：<https://zookeeper.apache.org/releases.html#download>

具体部署流程：

#下载

wget https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/current/apache-zookeeper-3.5.5-bin.tar.gz

#解压

tar -zxvf apache-zookeeper-3.5.5-bin.tar.gz

#拷贝默认配置

cd {zookeeper\_home}/conf

cp zoo\_sample.cfg zoo.cfg

#启动

 {zookeeper\_home}/bin/zkServer.sh

常规配置文件说明：

# zookeeper时间配置中的基本单位 (毫秒)

tickTime=2000

# 允许follower初始化连接到leader最大时长，它表示tickTime时间倍数 即:initLimit\*tickTime

initLimit=10

# 允许follower与leader数据同步最大时长,它表示tickTime时间倍数

syncLimit=5

#zookeper 数据存储目录

dataDir=/tmp/zookeeper

#对客户端提供的端口号

clientPort=2181

#单个客户端与zookeeper最大并发连接数

maxClientCnxns=60

# 保存的数据快照数量，之外的将会被清除

autopurge.snapRetainCount=3

#自动触发清除任务时间间隔，小时为单位。默认为0，表示不自动清除。

autopurge.purgeInterval=1

客户端命令：

**基本命令列表**

**close**

关闭当前会话

**connect host:port**

重新连接指定Zookeeper服务

**create [-s] [-e] [-c] [-t ttl] path [data] [acl]**

创建节点

**delete [-v version] path**

删除节点，(不能存在子节点）

**deleteall path**

删除路径及所有子节点

**setquota -n|-b val path**

设置节点限额 -n 子节点数 -b 字节数

**listquota path**

查看节点限额

**delquota [-n|-b] path**

删除节点限额

**get [-s] [-w] path**

查看节点数据 -s 包含节点状态 -w 添加监听

getAcl [-s] path

**ls [-s] [-w] [-R] path**

列出子节点 -s状态 -R 递归查看所有子节点 -w 添加监听

**printwatches on|off**

是否打印监听事件

**quit**

退出客户端

**history**

查看执行的历史记录

**redo cmdno**

重复 执行命令，history 中命令编号确定

removewatches path [-c|-d|-a] [-l]

**删除指定监听**

set [-s] [-v version] path data

**设置值**

**setAcl [-s] [-v version] [-R] path acl**

为节点设置ACL权限

**stat [-w] path**

查看节点状态 -w 添加监听

**sync path**

强制同步节点

**node数据的增删改查**

# 列出子节点

ls /

#创建节点

create /luban "luban is good man"

# 查看节点

get /luban

# 创建子节点

create /luban/sex "man"

# 删除节点

delete /luban/sex

# 删除所有节点 包括子节点

deleteall /luban

三、Zookeeper节点介绍

知识点：

1. 节点类型
2. 节点的监听(watch)
3. 节点属性说明(stat)
4. 权限设置(acl)

zookeeper 中节点叫znode存储结构上跟文件系统类似，以树级结构进行存储。不同之外在于znode没有目录的概念，不能执行类似cd之类的命令。znode结构包含如下：

* **path**:唯一路径
* **childNode**：子节点
* **stat**:状态属性
* **type**:节点类型

节点类型

|  |  |
| --- | --- |
| 类型 | 描述 |
| PERSISTENT | 持久节点 |
| PERSISTENT\_SEQUENTIAL | 持久序号节点 |
| EPHEMERAL | 临时节点(不可在拥有子节点) |
| EPHEMERAL\_SEQUENTIAL | 临时序号节点(不可在拥有子节点) |

1. PERSISTENT（持久节点）

持久化保存的节点，也是默认创建的

#默认创建的就是持久节点

create /test

1. PERSISTENT\_SEQUENTIAL(持久序号节点)

创建时zookeeper 会在路径上加上序号作为后缀，。非常适合用于分布式锁、分布式选举等场景。创建时添加 -s 参数即可。

#创建序号节点

create -s /test

#返回创建的实际路径

Created /test0000000001

create -s /test

#返回创建的实际路径2

Created /test0000000002

1. EPHEMERAL（临时节点）

临时节点会在客户端会话断开后自动删除。适用于心跳，服务发现等场景。创建时添加参数-e 即可。

#创建临时节点， 断开会话 在连接将会自动删除

create -e /temp

1. EPHEMERAL\_SEQUENTIAL（临时序号节点）

与持久序号节点类似，不同之处在于EPHEMERAL\_SEQUENTIAL是临时的会在会话断开后删除。创建时添加 -e -s

create -e -s /temp/seq

节点属性

# 查看节点属性

stat /luban

其属性说明如下表：

#创建节点的事物ID

cZxid = 0x385

#创建时间

ctime = Tue Sep 24 17:26:28 CST 2019

#修改节点的事物ID

mZxid = 0x385

#最后修改时间

mtime = Tue Sep 24 17:26:28 CST 2019

# 子节点变更的事物ID

pZxid = 0x385

#这表示对此znode的子节点进行的更改次数（不包括子节点）

cversion = 0

# 数据版本，变更次数

dataVersion = 0

#权限版本，变更次数

aclVersion = 0

#临时节点所属会话ID

ephemeralOwner = 0x0

#数据长度

dataLength = 17

#子节点数(不包括子子节点)

numChildren = 0

节点的监听：

客户添加 -w 参数可实时监听节点与子节点的变化，并且实时收到通知。非常适用保障分布式情况下的数据一至性。其使用方式如下：

|  |  |
| --- | --- |
| 命令 | 描述 |
| ls -w path | 监听子节点的变化（增，删） |
| get -w path | 监听节点数据的变化 |
| stat -w path | 监听节点属性的变化 |
| printwatches on|off | 触发监听后，是否打印监听事件(默认on) |

acl权限设置

ACL全称为Access Control List（访问控制列表），用于控制资源的访问权限。ZooKeeper使用ACL来控制对其znode的防问。基于scheme:id:permission的方式进行权限控制。scheme表示授权模式、id模式对应值、permission即具体的增删改权限位。

**scheme:认证模型**

|  |  |
| --- | --- |
| 方案 | 描述 |
| world | 开放模式，world表示全世界都可以访问（这是默认设置） |
| ip | ip模式，限定客户端IP防问 |
| auth | 用户密码认证模式，只有在会话中添加了认证才可以防问 |
| digest | 与auth类似，区别在于auth用明文密码，而digest 用sha-1+base64加密后的密码。在实际使用中digest 更常见。 |

**permission权限位**

|  |  |  |
| --- | --- | --- |
| 权限位 | 权限 | 描述 |
| c | CREATE | 可以创建子节点 |
| d | DELETE | 可以删除子节点（仅下一级节点） |
| r | READ | 可以读取节点数据及显示子节点列表 |
| w | WRITE | 可以设置节点数据 |
| a | ADMIN | 可以设置节点访问控制列表权限 |

**acl 相关命令：**

|  |  |  |
| --- | --- | --- |
| 命令 | 使用方式 | 描述 |
| getAcl | getAcl <path> | 读取ACL权限 |
| setAcl | setAcl <path> <acl> | 设置ACL权限 |
| addauth | addauth <scheme> <auth> | 添加认证用户 |

**world权限示例**

语法： setAcl <path> world:anyone:<权限位>

注：world模式中anyone是唯一的值,表示所有人

1. 查看默认节点权限：

#创建一个节点

create -e /testAcl

#查看节点权限

getAcl /testAcl

#返回的默认权限表示 ，所有人拥有所有权限。

'world,'anyone: cdrwa

1. 修改默认权限为 读写

#设置为rw权限

setAcl /testAcl world:anyone:rw

# 可以正常读

get /testAcl

# 无法正常创建子节点

create -e /testAcl/t "hi"

# 返回没有权限的异常

Authentication is not valid : /testAcl/t

**IP权限示例：**

语法： setAcl <path> ip:<ip地址|地址段>:<权限位>

**auth模式示例:**

语法：

1. setAcl <path> auth:<用户名>:<密码>:<权限位>
2. addauth digest <用户名>:<密码>

**digest 权限示例：**

语法：

1. setAcl <path> digest :<用户名>:<密钥>:<权限位>
2. addauth digest <用户名>:<密码>

注1：密钥 通过sha1与base64组合加密码生成，可通过以下命令生成

echo -n <用户名>:<密码> | openssl dgst -binary -sha1 | openssl base64

注2：为节点设置digest 权限后，访问前必须执行addauth，当前会话才可以防问。

1. 设置digest 权限

#先 sha1 加密，然后base64加密

echo -n luban:123456 | openssl dgst -binary -sha1 | openssl base64

#返回密钥

2Rz3ZtRZEs5RILjmwuXW/wT13Tk=

#设置digest权限

setAcl /luban digest:luban:2Rz3ZtRZEs5RILjmwuXW/wT13Tk=:cdrw

1. 查看节点将显示没有权限

#查看节点

get /luban

#显示没有权限访问

org.apache.zookeeper.KeeperException$NoAuthException: KeeperErrorCode = NoAuth for /luban

1. 给当前会话添加认证后在次查看

#给当前会话添加权限帐户

addauth digest luban:123456

#在次查看

get /luban

#获得返回结果

luban is good man

ACL的特殊说明：

权限仅对当前节点有效，不会让子节点继承。如限制了IP防问A节点，但不妨碍该IP防问A的子节点 /A/B。