UBT 后台部署

第 1 话　微服务 update-service 部署

1.1　在本地 Repository 中安装 jar

```
# 若该服务引用到的项目有所改动，到相应项目的根目录中执行命令：
mvn clean && mvn install

# 引用到的项目
dubbo-base、task-runner、ubt-micro-parent、ubt-update-api
```

1.2　利用 maven 命令打包项目

```
# 进入到 update-service 根目录（以我的目录为例）
D:\workspace\ubtechinc-update-service

# 执行打包命令
mvn clean && mvn package -Dmaven.test.skip=true

# 打包成功后生成的压缩文件位置
D:\workspace\ubtechinc-update-service\target\ubtechinc-update-
service-1.0.0-assembly.tar.gz

# 解压压缩文件并找到以下 jar 包
jar 包 1：ubtechinc-update-api-1.0.0.jar
jar 包 2：ubtechinc-update-service-1.0.0.jar
```

1.3　上传部署文件并重启服务

```
# 连接服务器（Xshell + Xftp）
主机：10.10.20.30
密码：ubt83474428
```

```
# 将文件 1 和文件 2 上传到以下目录，覆盖前请先备份
cd /usr/local/ubtechinc-update-service-1.0.0/lib

# 重启服务
cd /usr/local/ubtechinc-update-service-1.0.0/bin
./ubtechinc-update-service restart

# 打印实时日志，查看是否启动成功
tail -f ../logs/wrapper.log
```

第 2 话　webapp 部署

2.1　在本地 Repository 中安装 jar

```
# 若该服务引用到的项目有所改动，到相应项目的根目录中执行命令：
mvn clean && mvn install
```

2.2　打包项目

```
# 进入 ubt_webapp 根目录
 D:\workspace\ubt_webapp

# 执行打包命令
mvn clean && mvn package -Dmaven.test.skip=true

# 打包成功后生成的压缩文件的位置
D:\workspace\ubt_webapp\target\ubt_webapp.war
```

2.3　上传部署文件并重启服务

```
# 将 2.2 中得到的压缩文件解压并上传至目录
/usr/local/apache-tomcat-7.0.52-web/webapps/ROOT
```

上传说明：

1、千万不能直接替换 ROOT 文件下所有文件，特别注意不能替换：
/WEB-INF 文件下的 classes 文件中存在配置文件，web.xml 也为配置文件。

2、仅替换修改过的文件，未修改的文件尽量不要替换，替换前一定要先备份。
html 相关：jsp 目录；
js 相关：static 目录；
依赖的 jar 文件相关：lib 目录；
相关配置相关：classes 目录。

```
# 关闭服务器
cd /usr/local/tomcat7-ubt_webapp/bin
./shutdown.sh

# 找出相关 java 进程 pid，并 kill 掉
ps -ef|grep java
kill -9 [pid]

# 开启服务器
./start.sh

# 打印实时日志，查看是否启动成功
tail -f ../logs/catalina.out
```

第 3 话　Zookeeper 的安装与部署

3.1　基本介绍

3.1.1 安装包下载地址

http://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/

3.1.2　ZooKeeper 软件的文件和目录

bin 目录：

zk 的可执行脚本目录，包括 zk 服务进程，zk 客户端，等脚本。其中.sh 是 Linux 环境下的脚本，.cmd 是 Windows 环境下的脚本。

conf 目录：

配置文件目录。zoo_sample.cfg 为样例配置文件，需要修改为自己的名称，一般为 zoo.cfg。log4j.properties 为日志配置文件。

lib 目录：

zk 依赖的包。

contrib 目录：

一些用于操作 zk 的工具包。

recipes 目录：

zk 某些用法的代码示例

ZooKeeper 的安装包括单机模式安装，以及集群模式安装。在开发测试环境下，我们常使用单机模式。当然在单台物理机上也可以部署集群模式，但这会增加单台物理机的资源消耗。故在开发环境中，我们一般使用单机模式。但是要注意，生产环境下不可用单机模式，这是由于无论从系统可靠性还是读写性能，单机模式都不能满足生产的需求。

3.1.3 修改服务器环境变量

修改 JAVA 和 ZOOKEEPER 环境变量：

```
# 编辑文件/etc/profile，在文件末尾加上：（均为冒号）
80  export JAVA_HOME=/usr/java/jdk1.7.0_67
81  export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
82  export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
83
84  export ZOOKEEPER_HOME=/usr/local/zookeeper-3.4.6
85  export PATH=$ZOOKEEPER_HOME/bin:$PATH


# 编辑成功后使配置文件立即生效
source profile
```

3.2 单机模式

3.2.1 修改配置文件

复制样例配置文件 zoo_sample.cfg，重命名为 zoo.cfg，配置参数介绍：

tickTime：

时长单位为毫秒，为 zk 使用的基本时间度量单位。例如，1 * tickTime 是客户端与 zk 服务端的心跳时间，2 * tickTime 是客户端会话的超时时间。tickTime 的默认值为 2000 毫秒，更低的 tickTime 值可以更快地发现超时问题，但也会导致更高的网络流量（心跳消息）和更高的 CPU 使用率（会话的跟踪处理）。

clientPort：

zk 服务进程监听的 TCP 端口，默认情况下，服务端会监听 2181 端口。

dataDir：
无默认配置，必须配置，用于配置存储快照文件的目录。如果没有配置 dataLogDir，那么事务日志也会存储在此目录。

3.2.2 启动 ZooKeeper

```
# 在 Linux 环境下，进入 bin 目录，先获取权限：
chomod -R u+x *

再执行命令：
./zkServer.sh start
# 如果想在前台中运行以便查看服务器进程的输出日志，可换为命令：
./zkServer.sh start-foreground
```

使用文本编辑器打开 zkServer.sh 文件，可以看到其会调用 zkEnv.sh 脚本。zkEnv 的作用是设置 zk 运行的一些环境变量，例如配置文件的位置和名称、java 环境变量等。

3.2.3 连接 ZooKeeper

```
# 如果是连接同一台主机上的 zk 进程，那么直接运行 bin/目录下的 zkCli.cmd（Windows 环境下）或者 zkCli.sh（Linux 环境下），即可连接上 zk。
./zkCli.sh

# 直接执行 zkCli.sh 命令默认以主机号 127.0.0.1，端口号 2181 来连接 zk，如果要连接不同机器上的 zk，可以使用 -server 参数，例：
./zkCli.sh -server 192.168.0.1:2181
```

3.3 集群模式

单机模式的 zk 进程虽然便于开发与测试，但并不适合在生产环境使用。在生产环境下，我们需要使用集群模式来对 zk 进行部署。

注意：在集群模式下，建议至少部署 3 个 zk 进程，或者部署奇数个 zk 进程。如果只部署 2 个 zk 进程，当其中一个 zk 进程挂掉后，剩下的一个进程并不能

构成一个 quorum（仲裁）的大多数。因此，部署 2 个进程甚至比单机模式更不可靠，因为 2 个进程其中一个不可用的概率比一个进程不可用的概率还大。

3.3.1 修改配置文件

在集群模式下，所有的 zk 进程可以使用相同的配置文件（是指各个 zk 进程部署在不同的机器上面），例如如下配置：

```
1 tickTime=2000
2 dataDir=/home/myname/zookeeper
3 clientPort=2181
4 initLimit=5
5 syncLimit=2
6 server.1=192.168.229.160:2888:3888
7 server.2=192.168.229.161:2888:3888
8 server.3=192.168.229.162:2888:3888
```

tickTime：

tickTime 则是上述两个超时配置的基本单位，例如对于 initLimit，其配置值为 5，说明其超时时间为 2000ms * 5 = 10 秒。

dataDir：

其配置的含义跟单机模式下的含义类似，不同的是集群模式下还有一个 myid 文件。myid 文件的内容只有一行，且内容只能为 1 - 255 之间的数字，这个数字亦即上面介绍 server.id 中的 id，表示 zk 进程的 id。

initLimit：

ZooKeeper 集群模式下包含多个 zk 进程，其中一个进程为 leader，余下的进程为 follower。 当 follower 最初与 leader 建立连接时，它们之间会传输相当多的数据，尤其是 follower 的数据落后 leader 很多。initLimit 配置 follower 与 leader 之间建立连接后进行同步的最长时间。

syncLimit：

配置 follower 和 leader 之间发送消息，请求和应答的最大时间长度。

server.id=host:port1:port2

其中 id 为一个数字，表示 zk 进程的 id，这个 id 也是 dataDir 目录下 myid

文件的内容。host 是该 zk 进程所在的 IP 地址，port1 表示 follower 和 leader 交换消息所使用的端口，port2 表示选举 leader 所使用的端口。

注意：如果仅为了测试部署集群模式而在同一台机器上部署 zk 进程，server.id=host:port1:port2 配置中的 port 参数必须不同。但是，为了减少机器宕机的风险，强烈建议在部署集群模式时，将 zk 进程部署不同的物理机器上面。

3.3.2 启动 zookeeper

假如我们打算在三台不同的机器：192.168.229.160、192.168.229.161、192.168.229.162 上各部署一个 zk 进程，以构成一个 zk 集群。 三个 zk 进程均使用相同的 zoo.cfg 配置。

在三台机器 dataDir 目录（ /home/myname/zookeeper 目录）下，分别生成一个 myid 文件，其内容分别为 1，2，3。然后分别在这三台机器上启动 zk 进程，这样我们便将 zk 集群启动了起来。

3.3.3 连接 ZooKeeper

使用以下命令来连接一个 zk 集群，该命令执行后会随即打印日志。客户端连接上哪台机器的 zk 进程是随机的，通过日志可以看到连接的具体信息。

```
./zkCli.sh -server 192.168.229.160:2181,192.168.229.161:2181,
192.168.229.162:2181
```

第 4 话　Dubbo 部署

4.1　Dubbo Admin（管理控制平台）部署

4.1.1 安装 tomcat

下载 tomcat 安装包

下载地址：http://tomcat.apache.org/

上传到目录 /usr/local

修改 conf 目录下 server.xml 中访问端口（如改为 8086），避免端口冲突

打开防火墙，修改文件：/etc/sysconfig/iptables，添加端口

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8082 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8010 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8001 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8040 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8086 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

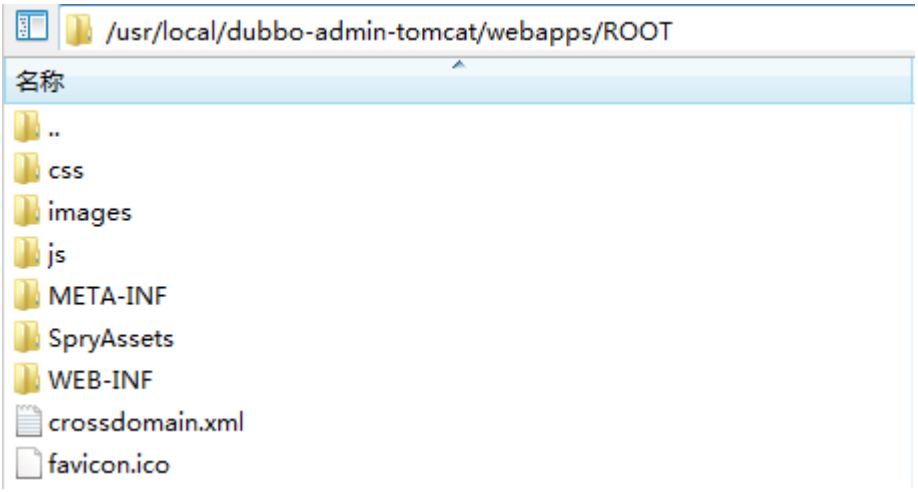
4.1.2 安装 Dubbo Admin

直接百度搜索：dubbo admin 下载，下载其 war 包

修改 WEB-INF/dubbo.properties 中 zk 地址

```
dubbo.registry.address=zookeeper://10.10.1.43:3181
dubbo.admin.root.password=root
dubbo.admin.guest.password=guest
```

上传 war 包内容至：/usr/local/tomcat/webapps/ROOT



4.1.3 启动并访问

```
# 进入 /usr/local/tomcat/bin 启动 tomcat
./start.sh

# 启动成功后访问对应地址，如：http://120.10.10.10:8086
  默认用户名：root，密码：root
```

4.2 Dubbo Monitor（监控平台）部署

4.2.1 下载 Dubbo Monitor

```
# 直接百度搜索：dubbo monitor 下载，下载其 war 包

# 修改 conf/dubbo.properties 配置：
```

```
dubbo.container=log4j,spring,registry,jetty
dubbo.application.name=simple-monitor
dubbo.application.owner=
#dubbo.registry.address=multicast://224.5.6.7:1234
dubbo.registry.address=zookeeper://10.10.1.43:2181
#dubbo.registry.address=redis://127.0.0.1:6379
#dubbo.registry.address=dubbo://127.0.0.1:9090
dubbo.protocol.port=7070
dubbo.jetty.port=8081
dubbo.jetty.directory=${user.home}/monitor
dubbo.charts.directory=${dubbo.jetty.directory}/charts
dubbo.statistics.directory=${user.home}/monitor/statistics
dubbo.log4j.file=logs/dubbo-monitor-simple.log
dubbo.log4j.level=WARN
```

dubbo.registry.address：zookeeper 地址

dubbo.jetty.port：访问地址的端口号

4.2.2 启动并访问

```
# 进入 /bin 目录，执行：
./start.sh

# 启动成功后访问地址：http://120.10.10.10:8081
```