



安装手册

SharkBuilder V1.1

北京先进数通信息技术股份公司

编写说明

中文标题: **SharkBuilder-安装手册**

英文标题:

过程分类:

过 程 域:

文 档 类型:

版 本 历 史:

版本	编写	日期	修改说明	审核	批准
v1.0		2019/2/1			
V1.1		2019/5/21	GDS服务配置, Ruby管理		
V1.2					
V1.3					

目录

编写说明	1
1. 安装环境说明	2
1.1. 硬件要求	2
1.2. 软件要求	2
2. MOIA CONTROL 安装	2
3. 作业运行组件安装	2
3.1. 作业运行组件安装环境说明	2
3.2. 作业运行组件安装环境的检查	2
3.2.1. Python运行环境检查	2
3.2.2. python模块安装	3
3.2.3. 数据库环境说明	4
3.3. 作业运行组件安装	4
3.3.1. 作业运行组件安装	4
3.3.2. 配置gds服务（可选）	5
4. WEB 服务管理组件安装	6
4.1. 数据库配置	6
4.2. WEB组件安装步骤	6
5. 安装数据资产服务	8
5.1. 安装RUBY环境	8
5.2. 安装REDIS集群	9
5.3. 数据资产服务安装	10
5.4. 数据资产相关配置	11

1. 安装环境说明

1.1. 硬件要求

- WEB服务器最低要求: 2C+8G+50G

1.2. 软件要求

- 操作系统: RedHat6.6+(CentOs6.6+)
- 调度平台: Moia Control V5.06.001 及以上版本
- 参数库: mysql5.7.2
- 依赖软件: DataStage

2. Moia Control安装



Moia Control安装手册 -V5.04.001.pdf

根据Moia Control 安装手册-V5.04.001.pdf文档安装MoiaControl 服务。

注： 联盟的MoiaWeb端为带有特殊定制接口，且需要jdk1.7 的运行环境。

3. 作业运行组件安装

3.1. 作业运行组件安装环境说明

软件环境:

gcc, gcc-c++, python-devel, openssl-devel, openssl(1.0.1 或以上)
rudy(2.2.2 以上), ruby-devel rubygems rpm-build

3.2. 作业运行组件安装环境的检查

3.2.1. Python 运行环境检查

- 执行 “python --version” 命令确认python的版本，“作业运行组件”需要在python2.7 上运行。

■ 进入python交互行，用import语句检查如下python 模块是否已经安装，如果没有安装请按照如下python 模块列表顺序安装，对于可选部分按照自己的实际需求安装。

- ◆ setuptools (没有则需要安装)
- ◆ hdfs
- ◆ jayDeBeApi
- ◆ requests
- ◆ mysql_connector
- ◆ pyinotify
- ◆ rarfile
- ◆ sqlalchemy
- ◆ traceback2
- ◆ requests_kerberos

3.2.2. python 模块安装

python模块安装有 2 种方式，可自行选择：

离线安装：

- 安装在默认目录下
 - tar -xzvf python模块包名. tar. gz;
 - cd 解压包文件目录;
 - 执行 python setup. py install;
 - 进入python交互，用import命令检查模块是否安装

在线安装：

- pip install 模块名

如：pip install hdfs

模块验证：

```
import hdfs
import jaydebeapi
import traceback
import requests
```

```

import mysql.connector
import pyinotify
import rarfile
import sqlalchemy
import requests_kerberos

```

3.2.3. 数据库环境说明

当前数据集成开发平台参数库支持mysql。

数据库用户需要具有连接和开发权限。

3.3. 作业运行组件安装

3.3.1. 作业运行组件安装

■ 作业运行组件安装

将作业运行组件程序包【Didp.tar.gz】上传至作业运行组件安装环境，按下面的操作步骤执行。

注：X_0 表示实际的版本号

操作说明	命令
步骤一：目录切换	<i>cd \$HOME (安装目录)</i>
步骤二：解压程序包	<i>tar -xzvf didp_linux_*.*.*.tar.gz</i> <i>ln -fs didp_linux_*.*.* Didp</i>
步骤三：赋权限	<i>chmod -R 755 \$HOME/Didp</i>
步骤四：修改配置文件	<p>进入Didp目录</p> <p><i>vi setenv (参照如下配置进行设置，粗体为变更区域)</i></p> <pre> ## 基础环境变量 export DIDP_HOME=/home/moiaagent/Didp #安装位置 export DIDP_LOG_LEVEL=INFO export DIDP_JDBC_DRIVER_PATH=\$DIDP_HOME/drivers # 把所有驱动包加入CLASSPATH中 export CLASSPATH= find \$DIDP_JDBC_DRIVER_PATH -name *.jar/tr '\n' ':' :\$CLASSPATH export JAVA_TOOL_OPTIONS="-Ddb2.jcc.charsetDecoderEncoder=3" export LD_LIBRARY_PATH=\$DIDP_HOME/lib:\$LD_LIBRARY_PATH export DIDP_CFG_DB_DATA_SOURCE=didp_db_source ## 配置库 </pre>

	<pre> export DIDP_CFG_DB_USER=didp export DIDP_CFG_DB_PWD=didp export DIDP_CFG_DB_DATA_SOURCE=didp_source # MYSQL #export DIDP_CFG_DB_JDBC_CLASS=com.mysql.jdbc.Driver #export DIDP_CFG_DB_JDBC_URL="jdbc:mysql://10.10.100.224:3306/didp_db?useSSL=true&characterEncoding=utf-8" # ORACLE #export DIDP_CFG_DB_JDBC_CLASS=oracle.jdbc.driver.OracleDriver #export DIDP_CFG_DB_JDBC_URL="jdbc:oracle:thin:@IP地址:端口:SID" # DB2 #export DIDP_CFG_DB_JDBC_CLASS=com.ibm.db2.jcc.DB2Driver #export DIDP_CFG_DB_JDBC_URL="jdbc:db2://IP地址:端口/库名" # SQLSERVER #export DIDP_CFG_DB_JDBC_CLASS=com.microsoft.sqlserver.jdbc.SQLServerDriver #export DIDP_CFG_DB_JDBC_URL="jdbc:sqlserver://IP地址:端口;databaseName=库名" # DATASTAGE #export APT_CONFIG_FILE=/opt/IBM/InformationServer/Server/Configurations/default.apt ## Moia服务连接配置 export DIDP_MOIA_IP=10.10.100.154 export DIDP_MOIA_PORT=8090 export DIDP_MOIA_USER=admin export DIDP_MOIA_PWD=admin # DATASTAGE export APT_DSHOME=/opt/IBM/InformationServer/Server/DSEngine . \$APT_DSHOME/.denv export DSHOME=\${APT_DSHOME} export APT_ORCHHOME=/opt/IBM/InformationServer/Server/PXEngine export APT_STRING_PADCHAR=0x20 export APT_MAX_DELIMITED_READ_SIZE=2097152 export APT_DEFAULT_TRANSPORT_BLOCK_SIZE=2097152 export LD_LIBRARY_PATH=/usr/lib64:\$APT_ORCHHOME/lib:\$LD_LIBRARY_PATH export PATH=\$APT_ORCHHOME/bin:\$DSHOME/bin:\$PATH export APT_OPERATOR_REGISTRY_PATH=\$DIDP_HOME/lib:\$APT_ORCHHOME/etc export APT_CONFIG_FILE=/opt/IBM/InformationServer/Server/Configurations/default.apt ## 激活virtualenv #source \$DIDP_HOME/venv/bin/activate </pre>
步骤 5 添加环境变量	<pre> vi ~/.bash_profile source \$HOME/set_env </pre>

3.3.2. 配置 gds 服务（可选）

◆ 若加载目标为GaussDb，则需要启动gds服务。

解压gds.tar至moia执行节点。

- 启动gds服务：

```

gds -d /home/bdp/file/middle -p 10.10.0.62:5000 -H 10.10.1.0/24 -
1 /home/moiaagt/gds/gds_log.txt -D -t 2

```

参数含义：

- d : 保存有待导入数据的数据文件所在目录，此处填写Middle区路径
 - p : gds 监听IP和监听端口, 填写当前节点IP即可
 - H : 允许哪些主机连接和使用GDS服务
 - l: gds log文件存放路径。
 - D: 后台运行GDS。
 - t : 设置GDS并发线程数
- 修改\$DIDP_HOME/etc/didp_run.cfg文件, 配置正确的
GdsIp, GdsPort, GdsDataPath

```
[didp.run]
UnCheckConfigFile=/home/moiaagent/Didp/ext/DECIMAL_UNSTRONG_CHECK_TABLES_CONFIG
LoggerBaseDir=/tmp/didp_log
Compression = yes
FileMonitorMaxRunTime=7200
GdsIp=10.10.0.62
GdsPort=5000
GdsDataPath=/home/bdp/file/middle
[didp.dw]
SDSCodeMappingErrorCode=##
FDSpTyErrorPrefix=##
~
```

4. Web服务管理组件安装

Web服务管理组件包括web程序包(didp-*.jar), 和web启动脚本(start.sh)。

注意:Web服务组件需要jdk1.8版本。

4.1. 数据库配置

目前数据集成开发平台的参数库仅支持mysql

安装步骤:

1. 数据库执行建表语句脚本 didp_create_table.sql
2. 数据库执行初始化数据脚本(初始化数据sql下的所有sql文件)

4.2. web组件安装步骤

1. 修改didp_xx.jar中的\BOOT-INF\classes\application-dev.yml
文件, 将其中的的数据库地址改为实际的数据库地址

```
application-dev - 记事本
文件(F) 帮助(H) 格式(O) 查看(V) 帮助(I)
ipring:
datasource:
type: com.alibaba.druid.pool.DruidDataSource
driverClassName: com.mysql.jdbc.Driver
# url: jdbc:mysql://36.33.97.138:3306/zzbank?allowMultiQueries=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false
# username: sharkbuilder
# password: 123456
url: jdbc:mysql://10.10.100.224:3306/didp_db?allowMultiQueries=true&useUnicode=true&characterEncoding=UTF-8&useSSL=false
username: didp
password: didp
initial-size: 10
max-active: 100
min-idle: 10
max-wait: 60000
pool-prepared-statements: true
max-pool-prepared-statement-per-connection-size: 20
time-between-eviction-runs-millis: 60000
min-evictable-idle-time-millis: 300000
#validation-query: SELECT 1 FROM DUAL
test-while-idle: true
test-on-borrow: false
test-on-return: false
stat-view-servlet:
enabled: true
url-pattern: /druid/*
```

2. 将web包和启动脚本上传至服务器
3. 启动start.sh
4. 登录浏览器, 检查Web是否正常启动。

注: 若Tomcat默认端口 8080 被占用, 修改didp_xx.jar中的\BOOT-INF\classes\application.yml文件

```
server:
tomcat:
uri-encoding: UTF-8
max-threads: 1000
min-spare-threads: 30
max-http-header-size: 10485760
max http post size: -1
port: 8080
servlet:
context-path: /didp
session:
timeout: 30m

spring:
profiles:
active: dev
```

5. 安装数据资产服务

数据资产服务是微服务组件, 用于事件依赖的发送, 安装数据资产服务需要先安装Redis集群。

注意: 需求 1.8 版本的JDK

5.1. 安装ruby环境

Redis集群依赖ruby环境, 需要事先安装ruby

- yum install ruby
- yum install rubygems
- gem install redis

若ruby版本低于 2.2.2, 需要对其进行升级

解决办法是先安装rvm, 再把ruby版本提升至 2.3.3:

1. 安装curl

```
sudo yum install curl
```

2. 安装RVM

```
curl -L get.rvm.io | bash -s stable
```

```
3. source /usr/local/rvm/scripts/rvm
```

4. 查看rvm库中已知的ruby版本

```
rvm list known
```

5. 安装一个ruby版本

```
rvm install 2.3.3
```

6. 使用一个ruby版本

```
rvm use 2.3.3
```

7. 设置默认版本

```
rvm remove 2.0.0
```

8. 卸载一个已知版本

```
ruby --version
```

9. 再安装redis就可以了

```
gem install redis
```

5.2. 安装Redis集群

Redis集群根据实际需求安装，但是至少需要 6 个节点。现在以 2 台机器 (ip1, ip2)，6 台节点为例，搭建Redis集群。

1. 下载redis。并上传到 2 台机器上。

```
wget http://download.redis.io/releases/redis-3.2.4.tar.gz
```

2. 解压redis安装包，并安装

- tar -zxvf redis-3.2.4.tar.gz
- cd redis-3.2.4
- make && make install

3. 添加快捷方式

```
cp src/redis-trib.rb /usr/local/bin/
```

4. 创建 Redis 节点

首先在 IP1 机器上 /root/software/redis-3.2.4 目录下创建 redis_cluster 目录；

- mkdir redis_cluster
- cd redis_cluster
- 在 redis_cluster 目录下，创建名为 7000、7001、7002 的目录，并将 redis.conf 拷贝到这三个目录中
- mkdir 7000 7001 7002
- cp redis.conf redis_cluster/7000
- cp redis.conf redis_cluster/7001
- cp redis.conf redis_cluster/7002

分别修改这三个配置文件，修改如下内容

```
port 7000 //端口7000,7002,7003
bind 本机ip //默认ip为127.0.0.1 需要改为其他节点机器可访问的ip 否则创建集群时无法访问对应的端口，无法创建集群
daemonize yes //redis后台运行
pidfile /var/run/redis_7000.pid //pidfile文件对应7000,7001,7002
cluster-enabled yes //开启集群 把注释#去掉
cluster-config-file nodes_7000.conf //集群的配置 配置文件首次启动自动生成 7000,7001,7002
cluster-node-timeout 15000 //请求超时 默认15秒，可自行设置
appendonly yes //aof日志开启 有需要就开启，它会每次写操作都记录一条日志
```

5. 接着在另外一台机器上（IP2），的操作重复以上三步，只是把目录改为 7003、7004、7005，对应的配置文件也按照这个规则修改即可
6. 启动各个节点

第一台机器上执行

```
redis-server redis_cluster/7000/redis.conf
```

```
redis-server redis_cluster/7001/redis.conf
```

```
redis-server redis_cluster/7002/redis.conf
```

第二台机器上执行

```
redis-server redis_cluster/7003/redis.conf
```

```
redis-server redis_cluster/7004/redis.conf
```

```
redis-server redis_cluster/7005/redis.conf
```

7. 安装redis

```
gem install redis
```

如遇到源错误问题，可以将源替换为<http://gems.ruby-china.com>

8. 启动集群

```
redis-trib.rb create --replicas 1 IP1:7000 IP1:7001
```

```
IP1:7002 IP2:7003 IP2:7004 IP3:7005
```

注：redis_trib.rb方式依赖ruby<2.0 版本以上。

5.3. 数据资产服务安装

1. 将assets.tar 上传到服务器
2. 解压assets.tar
3. 修改application.properties, 设置moia地址和redis地址

```
cd assets/producer-asset-manager-1.0.0/config;
vi application.properties
```

```
###主配置
spring.application.name=asset-manager
#使用环境
spring.profiles.active=dev

# 连接数据库
spring.datasource.type=com.alibaba.druid.pool.DruidDataSource
spring.datasource.url=jdbc:mysql://10.10.100.224:3306/idp_db
spring.datasource.username=idp
spring.datasource.password=idp
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
## 初始化大小,最小,最大
spring.datasource.druid.initial-size=5
spring.datasource.druid.min-size=2
spring.datasource.druid.max-active=20
spring.datasource.druid.min-idle=5
## 隔离级别
spring.datasource.druid.max-wait=60000
## 配置间隔多久才进行一次检测,检测需要关闭的空间连接;单位是毫秒
spring.datasource.druid.validation-time-interval=10000
## 配置一个连接在池中最小生存的时间,单位是毫秒
spring.datasource.druid.minEvictableidleTime-millis=300000
spring.datasource.druid.validationQuery=select 1 from dual
spring.datasource.druid.test-on-borrow=false
spring.datasource.druid.test-on-return=false
spring.datasource.druid.pool-prepared-statements=true
spring.datasource.druid.max-pool-prepared-statement-per-connection-size=20
#logging.level.com.adtec.sharkata.mapper.mybatis=warn

#redis 集群节点
spring.redis.cluster.nodes=10.10.100.224:7001,10.10.100.224:7002,10.10.100.224:7003,10.10.100.225:7004,10.10.100.225:7005,10.10.100.225:7006
spring.redis.timeout=5000
#最大空闲连接数
spring.redis.maxIdle=10
#最大连接数
spring.redis.maxTotal=50
#连接超时时间
spring.redis.maxWait=5000
spring.redis.maxWaitMillis=3000
##redis尝试链接的次数
spring.redis.retryAttempts=3
##redis 队列
spring.redis.queue.jobstatus=asset:jobstatus
spring.redis.queue.jobevent=asset:jobevent
##消息处理超时时间 (s)
spring.redis.process.timeout=5
```

4. 启动数据资产服务

assets/producer-asset-manager-1.0.0/bin/start.sh

5.4. 数据资产相关配置



数据资产管理使用
指南.V1.0.docx