

1、部署数据挖掘

- 1、系统环境准备
 - 1.1 防火墙配置
 - 1.2 安装Java环境
 - 1.3 取消打开文件限制
- 2、部署数据挖掘引擎
 - 2.1 添加主机名映射关系
 - 2.2 安装数据挖掘-执行引擎
 - 2.3 安装数据挖掘-服务引擎
 - 2.4 配置数据挖掘
 - 2.5 测试数据挖掘
- 3、运维操作

数据挖掘包含两大部分：数据挖掘执行引擎、数据挖掘服务引擎

数据挖掘执行引擎：

- 负责接收Smartbi 发送执行请求。
- 通过解析执行定义，生成spark 计算任务或python计算任务，分别发送给spark集群或python集群。
- 本身并不承担计算任务，只负责计算任务的调度跟分发。

数据挖掘服务引擎：

- 提供模型预测服务给第三放系统调用

数据挖掘执行引擎和服务引擎可部署在同一台服务器中，也可以分开部署在不同服务器中。



文档环境

单机部署数据挖掘组件环境如下：

服务器IP	主机名	组件实例	部署目录
10.10.204.248	10-10-204-248	数据挖掘	/data
10.10.204.249	10-10-204-249	Spark, Hadoop	/data
10.10.204.250	10-10-204-250	Python执行节点	/data

PS：可根据实际服务器资源，将数据挖掘及其组件部署在不同服务器，或者部署在同一台服务器中(可能导致性能下降)。

1、系统环境准备



温馨提示

配置防火墙，selinux相关操作，需要管理员权限。

1.1 防火墙配置

为了便于安装，建议在安装前关闭防火墙。使用过程中，为了系统安全可以选择启用防火墙，但必须启用服务相关端口。

1. 关闭防火墙

临时关闭防火墙（立即生效）

```
systemctl stop firewalld
```

永久关闭防火墙（重启后生效）

```
systemctl disable firewalld
```

查看防火墙状态

```
systemctl status firewalld
```

2. 开启防火墙

相关服务及端口对照表：

服务名	需要开放端口
执行引擎	8899, 4040, 7777, [30000-65535]
服务引擎	8900

如果确实需要打开防火墙安装，需要给防火墙放开以下需要使用到的端口
开启端口：8899, 8900, 4040, 7777, [30000-65535]

```
firewall-cmd --zone=public --add-port=8899/tcp --permanent
firewall-cmd --zone=public --add-port=8900/tcp --permanent
firewall-cmd --zone=public --add-port=4040/tcp --permanent
firewall-cmd --zone=public --add-port=7777/tcp --permanent
firewall-cmd --zone=public --add-port=30000-65535/tcp --permanent
```

配置完以后重新加载firewalld，使配置生效

```
firewall-cmd --reload
```

查看防火墙的配置信息

```
firewall-cmd --list-all
```

3. 关闭selinux

临时关闭selinux，立即生效，不需要重启服务器。

```
setenforce 0
```

永久关闭selinux，修改完配置后需要重启服务器才能生效

```
sed -i 's/=enforcing/=disabled/g' /etc/selinux/config
```

1.2 安装Java环境

解压jdk到指定目录：

```
tar -zxvf jdk-8u181-linux-x64.tar.gz -C /data
```

添加环境变量

```
vi /etc/profile
```

在文件末尾添加下面内容：

```
export JAVA_HOME=/data/jdk1.8.0_181
export JAVA_BIN=$JAVA_HOME/bin
export CLASSPATH=: $JAVA_HOME/lib/dt.jar: $JAVA_HOME/lib/tools.jar
export PATH=$PATH: $JAVA_BIN
```

让配置生效

```
source /etc/profile
```

验证安装

```
java -version
```

1.3 取消打开文件限制



温馨提示

取消打开文件限制，配置主机名等操作，需要管理员权限。

修改/etc/security/limits.conf文件在文件的末尾加入以下内容：

```
vi /etc/security/limits.conf
```

在文件的末尾加入以下内容：

```
* soft nfile 65536
* hard nfile 65536
* soft nproc 131072
* hard nproc 131072
```

2、部署数据挖掘引擎



对于自定义驱动的业务库数据源，在部署数据挖掘引擎时需要将驱动包放到部署挖掘引擎主机的<smartbi-engine-bin>/conf/lib对应数据库的目录里面。然后重启数据挖掘服务。

```
[root@smartbi lib]# pwd
/smartbi/smartbi-mining-engine-bin/conf/lib
[root@smartbi lib]# ll
总用量 0
drwxr-xr-x 2 root root 24  9月  3 15:17 clickhouse
drwxr-xr-x 2 root root 24  9月  3 15:17 db2
drwxr-xr-x 2 root root 24  9月  3 15:17 dm
drwxr-xr-x 2 root root 24  9月  3 15:17 gbase
drwxr-xr-x 2 root root 24  9月  3 15:17 guassdb
drwxr-xr-x 2 root root 24  9月  3 15:17 hana
drwxr-xr-x 2 root root 24  9月  3 15:17 hive
drwxr-xr-x 2 root root 24  9月  3 15:17 kafka
drwxr-xr-x 2 root root 24  9月  3 15:17 maria
drwxr-xr-x 2 root root 24  9月  3 15:17 mysql
drwxr-xr-x 2 root root 24  9月  3 15:17 odps
drwxr-xr-x 2 root root 24  9月  3 15:17 oracle
drwxr-xr-x 2 root root 24  9月  3 15:17 postgresql
drwxr-xr-x 2 root root 24  9月  3 15:17 presto
drwxr-xr-x 2 root root 24  9月  3 15:17 sqlserver
drwxr-xr-x 2 root root 24  9月  3 15:17 sybase
drwxr-xr-x 2 root root 24  9月  3 15:17 vertica
```

举例：

- 1、若星环是自定义驱动的，则将驱动包放进hive目录中，重启数据挖掘服务。
- 2、若oracle是自定义驱动的，则将驱动包放进oracle目录中，重启数据挖掘服务。

2.1 添加主机名映射关系

将数据挖掘组件中的服务器主机名映射到hosts文件中

```
vi /etc/hosts
```

文件末尾添(根据实际环境信息设置):

```
10.10.204.248 10-10-204-248
10.10.204.249 10-10-204-249
10.10.204.250 10-10-204-250
```



注意!

部署smartbi服务器的/etc/hosts, 需要添加所有数据挖掘组件的主机和IP地址映射

2.2 安装数据挖掘-执行引擎



温馨提示

- 1、安装部署、启动数据挖掘执行引擎, 服务引擎等, 可以使用普通用户权限进行操作。
- 2、部署过程中用普通用户操作, 则后续的所有运维操作等, 都需要用普通用户来执行。如果切换其他用户操作, 可能会因为权限问题导致服务启动失败。
- 3、数据挖掘的执行引擎或服务引擎的端口配置如果小于1024, 也需要管理员权限才能启动服务。

- 1、解压Smartbi-engine安装包到指定的安装目录

```
tar -zxvf SmartbiMiningEngine-V10.0.64186.21183.tar.gz -C /data
```

- 2、启动数据挖掘执行引擎

```
cd /data/smartbi-mining-engine-bin/engine/sbin/
./experiment-daemon.sh start
```

2.3 安装数据挖掘-服务引擎



注意事项

由于文档中服务引擎与执行引擎部署在同一台服务器, 所以无需重复设置系统环境。
如果服务引擎与执行引擎部署在不同服务器时, 服务引擎系统环境设置可参考执行引擎的配置。

- 1、启动数据挖掘服务引擎

```
cd /data/smartbi-mining-engine-bin/engine/sbin/
./service-daemon.sh start
```

2.4 配置数据挖掘



此处仅测试数据挖掘本地模式, 未连接Spark等计算节点

- 1、数据挖掘连接测试:

①浏览器访问Smartbi, 打开**系统运维 - 数据挖掘配置 - 引擎设置**, 引擎地址和服务地址设置成正确的数据挖掘引擎地址, 并**点击保存**

③打开系统运维 - 数据挖掘配置 - 执行引擎—计算节点配置，参考下图设置，修改完成后点击保存

由于未部署spark计算节点，需要设置成单机local模式：

④打开系统运维 - 数据挖掘配置 - 服务引擎，参考以下格式修改“系统api地址”（根据实际环境修改），修改完成后点击保存：

[http:// Smartbi_IP_地址 : port /smartbi/smartbix/api/monitor](http://Smartbi_IP_地址:port/smartbi/smartbix/api/monitor)

引擎配置

引擎服务端口：

8900

引擎高可用时连接zookeeper地址：

10.10.204.239:2181

引擎高可用设置,默认为不可用：

false

系统api地址：

http://10.10.204.248:8080/smartbi/smartbix/api/monitor

初始值（空白）

节点数据是否存储：

true

初始值（true）

节点数据是否计数：

true

初始值（true）

节点数据目录：

/data/smartbi-mining-engine-bin/data

节点日志目录：

/data/smartbi-mining-engine-bin/logs

节点数据存储行数：

100

初始值（100）

python插件存储目录：

/data/smartbi-mining-engine-bin/conf/plugins/python

java插件jar包存储目录：

/data/smartbi-mining-engine-bin/conf/plugins/java

节点数据hdfs存储目录：

webhdfs://enginecluster/mining/

初始值（webhdfs://enginecluster/mining/）

节点数据hdfs访问控制列表：

mining

导航 | 数据挖掘配置 x

引擎设置 执行引擎 服务引擎 作业流

替换成实际的访问地址

⑤执行引擎和服务引擎完成设置、保存后，返回 **系统运维-数据挖掘配置-引擎设置**，重新点击测试，如下提示表示配置成功：

2.5 测试数据挖掘

3、运维操作

```
cd /data/smartbi-mining-engine-bin/engine/sbin
./experiment-daemon.sh restart #
./experiment-daemon.sh stop #
./experiment-daemon.sh status #
./service-daemon.sh restart #
./service-daemon.sh stop #
./service-daemon.sh status #
```

```

# Common options for all engine
#JAVA_HOME, to point java home (e.g: /opt/smartbi/jdk1.8.0_181) (Default: system env JAVA_HOME)
#JAVA_HOME=/data/Smartbi_CentOS_x86_CN_V10.5.15/Jdk
#
#
#
# Options for experiment engine
#EXPERIMENT_ENGINE_XMS, to point experiment engine jvm xms (e.g: 4096M, 4G) (Default: 5G)
#EXPERIMENT_ENGINE_XMS=5G
#EXPERIMENT_ENGINE_XMX, to point experiment engine jvm xmx (e.g: 4096M, 4G) (Default: 5G)
#EXPERIMENT_ENGINE_XMX=5G
#
#
# Options for agent engine
#AGENT_ENGINE_XMX, to point agent engine jvm xmx (e.g: 4096M, 4G) (Default: 3G)
#AGENT_ENGINE_XMX=3G
#AGENT_ENGINE_XMS, to point agent engine jvm xms (e.g: 4096M, 4G) (Default: 3G)
#AGENT_ENGINE_XMS=3G
#
#
# Options for service engine
#SERVICE_ENGINE_XMS, to point service engine jvm xms (e.g: 4096M, 4G) (Default: 1G)
#SERVICE_ENGINE_XMS=1G
#SERVICE_ENGINE_XMX, to point service engine jvm xmx (e.g: 4096M, 4G) (Default: 3G)
#SERVICE_ENGINE_XMX=3G
#
~
~
~

```

配置jdk路径

执行引擎最小内存

执行引擎最大内存

代理服务最大内存

代理服务最小内存

服务引擎最小内存

服务引擎最大内存



温馨提示

如需修改配置信息，需要删除对应配置行开头的#号

4、设置数据挖掘开机启动



温馨提示

配置开机启动，需要管理员权限。

进入/etc/init.d目录，创建数据挖掘-执行引擎启动配置文件

```
vi /etc/init.d/mining-engine
```

配置参考如下：


```
#!/bin/bash
#
# tomcat startup script for the mining-engine server
# chkconfig: 345 80 20
# description: start the mining-engine daemon
#
# Source function library
. /etc/rc.d/init.d/functions

prog=mining-engine
JAVA_HOME=/data/jdk1.8.0_181/      #JAVA
export JAVA_HOME
MINING_HOME=/data/smartbi-mining-engine-bin/engine/      #
export MINING_HOME

case "$1" in
start)
    echo "Starting mining-engine..."
    $MINING_HOME/sbin/experiment-daemon.sh start
    ;;

stop)
    echo "Stopping mining-engine..."
    $MINING_HOME/sbin/experiment-daemon.sh stop
    ;;

restart)
    echo "Stopping mining-engine..."
    $MINING_HOME/sbin/experiment-daemon.sh stop
    sleep 2
    echo
    echo "Starting mining-service..."
    $MINING_HOME/sbin/experiment-daemon.sh start
    ;;

*)
    echo "Usage: $prog {start|stop|restart}"
    ;;
esac
exit 0
```

进入/etc/init.d目录，创建数据挖掘-服务引擎启动配置文件

```
vi /etc/init.d/mining-service
```

配置参考如下：

```
#!/bin/bash
#
# tomcat startup script for the mining-service server
# chkconfig: 345 80 20
# description: start the mining-service daemon
#
# Source function library
. /etc/rc.d/init.d/functions

prog=mining-service
JAVA_HOME=/data/jdk1.8.0_181/      #JAVA
export JAVA_HOME
MINING_HOME=/data/smartbi-mining-engine-bin/engine/      #
export MINING_HOME

case "$1" in
start)
    echo "Starting mining-service..."
    $MINING_HOME/sbin/service-daemon.sh start
    ;;

stop)
    echo "Stopping mining-service..."
    $MINING_HOME/sbin/service-daemon.sh stop
    ;;

restart)
    echo "Stopping mining-service..."
    $MINING_HOME/sbin/service-daemon.sh stop
    sleep 2
    echo
    echo "Starting mining-service..."
    $MINING_HOME/sbin/service-daemon.sh start
    ;;

*)
    echo "Usage: $prog {start|stop|restart}"
    ;;
esac
exit 0
```

设置开机启动

```
chmod +x /etc/init.d/mining-engine      #
chmod +x /etc/init.d/mining-service      #
chkconfig mining-engine on              #
chkconfig mining-service on             #
chkconfig --list                        #
```