部署Smartbi-UnionServer

Smartbi UnionServer是一个分布式SQL查询引擎，处在Smartbi的SQL引擎层，为不同的数据源提供统一的SQL解析、跨库查询能力。

1、设置系统环境

① 关闭防火墙。

CentOS 6.X/Redhat 6.X

```
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

CentOS 7.X/Redhat 7.X

```
#
systemctl stop firewalld
#
systemctl disable firewalld
#
systemctl status firewalld
```

②关闭Selinux

```
#selinux
setenforce 0
#Selinux
sed -i 's/=enforcing/=disabled/g'  /etc/selinux/config
```

2、安装Smartbi-UnionServe

上传SmartbiUnionServer.tar.gz到服务器，并解压到/opt目录。

```
tar -zxvf SmartbiUnionServer.tar.gz -C /opt
```

3、 修改配置文件

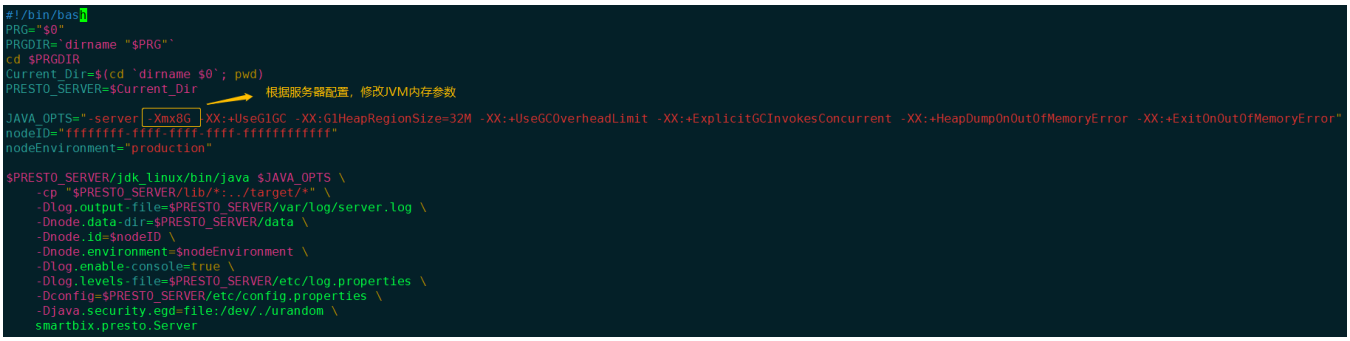
1）JVM配置

通过修改启动文件可以设置JVM的最大内存、GC等参数。

```
cd /opt/SmartbiUnionServer
```

在启动服务之前，需要根据服务器的实际物理内存大小，修改JVM的默认最大内存，如下：

修改run.sh中的-Xmx参数，默认的最大内存值为8G，可根据服务器实际配置进行情况填写：




```
#!/bin/bash
PRG="$0"
PRGDIR=`dirname "$PRG"`
cd $PRGDIR
Current_Dir=$(cd `dirname $0`; pwd)
PRESTO_SERVER=$Current_Dir

JAVA_OPTS="-server -Xmx8G -XX:+UseG1GC -XX:G1HeapRegionSize=32M -XX:+UseGCOverheadLimit -XX:+ExplicitGCInvokesConcurrent -XX:+HeapDumpOnOutOfMemoryError -XX:+ExitOnOutOfMemoryError"
nodeID="ffffffff-ffff-ffff-ffff-ffffffffffff"
nodeEnvironment="production"

$PRESTO_SERVER/jdk_linux/bin/java $JAVA_OPTS \
    -cp "$PRESTO_SERVER/lib/*:../target/*" \
    -Dlog.output-file=$PRESTO_SERVER/var/log/server.log \
    -Dnode.data-dir=$PRESTO_SERVER/data \
    -Dnode.id=$nodeID \
    -Dnode.environment=$nodeEnvironment \
    -Dlog.enable-console=true \
    -Dlog.levels-file=$PRESTO_SERVER/etc/log.properties \
    -Dconfig=$PRESTO_SERVER/etc/config.properties \
    -Djava.security.egd=file:/dev/./urandom \
    smartbix.presto.Server
```

2）参数配置

参数配置文件放在/opt/SmartbiUnionServer/etc目录下，需要修改config.properties。

config.properties的基本配置信息如下：

```
coordinator=true
node-scheduler.include-coordinator=true
http-server.http.port=48080
query.max-memory=2GB
query.max-memory-per-node=1GB
discovery-server.enabled=true
discovery.uri=http://0.0.0.0:48080
```

参数说明如下：

配置项	说明
http-server.http.port	设置presto的端口，默认为48080，启动时如果端口冲突，需要修改。
query.max-memory	设置单条查询语句最大使用内存，默认为2GB。
query.max-memory-per-node	设置单条查询语句在每个节点上的最大使用内存，默认为1GB。
discovery.uri	设置Smartbi UnionServer的url，默认为http://0.0.0.0:48080，其中0.0.0.0表示本机地址，48080表示端口，必须与http-server.http.port保持一致。

3）日志级别配置

日志级别配置文件放在/opt/SmartbiUnionServer/etc目录下log.propertes，默认日志级别为INFO。

```
com.facebook.presto=INFO
```

日志级别可以选择：DEBUG、INFO、WARN和ERROR，其中DEBUG的日志级别最高，输出的日志最多，ERROR的日志级别最低，输出的日志最少。

4、运维操作

4.1、启动Smartbi-UnionServer

赋予启动脚本可执行权限

```
cd /opt/SmartbiUnionServer
chmod +x run.sh
```

给jdk赋予可执行权限：

```
chmod +x -R jdk_linux/
```

启动SmartbiUnionServer服务有两种方式：

```
#
sh run.sh

# var/log/server.log
nohup ./run.sh > /dev/null 2>&1 &
```

使用前端方式启动presto时，当看到屏幕打印信息：

======== SERVER STARTED ========，说明服务启动成功。

```
2020-05-29T13:53:42.096+0800    INFO    main    Bootstrap       PROPERTY                    DEFAULT  RUNTIME              DESCRIPTION
2020-05-29T13:53:42.096+0800    INFO    main    Bootstrap       resource-groups.config-file  null     etc/queue_config.json
2020-05-29T13:53:43.147+0800    INFO    main    io.airlift.bootstrap.LifeCycleManager   Life cycle starting...
2020-05-29T13:53:43.148+0800    INFO    main    io.airlift.bootstrap.LifeCycleManager   Life cycle startup complete. System ready.
2020-05-29T13:53:43.148+0800    INFO    main    com.facebook.presto.execution.resourceGroups.InternalResourceGroupManager       -- Loaded resource group configuration manag
er file --
2020-05-29T13:53:43.148+0800    INFO    main    com.facebook.presto.security.AccessControlManager       -- Loading system access control --
2020-05-29T13:53:43.149+0800    INFO    main    com.facebook.presto.security.AccessControlManager       -- Loaded system access control allow-all --
2020-05-29T13:53:43.276+0800    INFO    main    com.facebook.presto.server.PrestoServer ======== SERVER STARTED ========
2020-05-29T13:53:46.499+0800    INFO    CatalogMonitorThread    smartbix.presto.CatalogMonitorHandler   Start monitor catalog path /data/SmartbiUnionServer/etc/catalog/smar
tbix
```

如果提示Address already in use，说明端口冲突了，需要修改/opt/SmartbiUnionServer/etc/config.properties里的端口，然后重启SmartbiUnionServer。

```
presto.server.CoordinatorModule)
Caused by: java.io.UncheckedIOException (same stack trace as error #3)
5) Error injecting constructor, java.io.UncheckedIOException: java.net.BindException: 地址已在使用
  at io.airlift.http.server.HttpServerInfo.<init>(HttpServerInfo.java:46)
  at io.airlift.http.server.HttpServerModule.configure(HttpServerModule.java:66)
  while locating io.airlift.http.server.HttpServerInfo
    for the 1st parameter of io.airlift.http.server.LocalAnnouncementHttpServerInfo.<init>(LocalAnnouncementHttpServerInfo.java:31)
  while locating io.airlift.http.server.LocalAnnouncementHttpServerInfo
  at io.airlift.http.server.HttpServerModule.configure(HttpServerModule.java:78)
  while locating io.airlift.discovery.client.AnnouncementHttpServerInfo
    for the 1st parameter of io.airlift.discovery.client.DiscoveryBinder$HttpAnnouncementProvider.setAnnouncementHttpServerInfo(DiscoveryBinder.java:121)
  at io.airlift.discovery.client.DiscoveryBinder.bindServiceAnnouncement(DiscoveryBinder.java:78) (via modules: com.facebook.presto.server.ServerMainModule -> io.airlift.co
nfiguration.ConditionalModule -> io.airlift.discovery.server.EmbeddedDiscoveryModule)
Caused by: java.io.UncheckedIOException (same stack trace as error #3)
6) Error injecting constructor, java.io.UncheckedIOException: java.net.BindException: 地址已在使用
  at io.airlift.http.server.HttpServerInfo.<init>(HttpServerInfo.java:46)
  at io.airlift.http.server.HttpServerModule.configure(HttpServerModule.java:66)
  while locating io.airlift.http.server.HttpServerInfo
    for the 1st parameter of io.airlift.http.server.LocalAnnouncementHttpServerInfo.<init>(LocalAnnouncementHttpServerInfo.java:31)
  while locating io.airlift.http.server.LocalAnnouncementHttpServerInfo
  at io.airlift.http.server.HttpServerModule.configure(HttpServerModule.java:78)
  while locating io.airlift.discovery.client.AnnouncementHttpServerInfo
    for the 1st parameter of io.airlift.discovery.client.DiscoveryBinder$HttpAnnouncementProvider.setAnnouncementHttpServerInfo(DiscoveryBinder.java:121)
  at io.airlift.discovery.client.DiscoveryBinder.bindServiceAnnouncement(DiscoveryBinder.java:78)
Caused by: java.io.UncheckedIOException (same stack trace as error #3)
6 errors
        at com.google.inject.internal.Errors.throwCreationExceptionIfErrorsExist(Errors.java:543)
        at com.google.inject.internal.InternalInjectorCreator.injectDynamically(InternalInjectorCreator.java:178)
        at com.google.inject.internal.InternalInjectorCreator.build(InternalInjectorCreator.java:109)
        at com.google.inject.Guice.createInjector(Guice.java:87)
        at io.airlift.bootstrap.Bootstrap.initialize(Bootstrap.java:241)
        at com.facebook.presto.server.PrestoServer.run(PrestoServer.java:115)
        at smartbix.presto.Server.main(Server.java:19)
```

使用后台方式启动SmartbiUnionServer时，如果使用后台启动，可以使用ps -ef | grep SmartbiUnionServer查看SmartbiUnionServer进程是否存在，如果存在，则启动成功。如下图所示。

```
[root@smartbi SmartbiUnionServer]# ps -ef | grep SmartbiUnionServer
root      19746  19741 52 13:55 pts/0    00:00:32 /data/SmartbiUnionServer/jdk_linux/bin/java -server -Xmx8G -XX:+UseG1GC -XX:G1HeapRegionSize=32M -XX:+UseGCOverheadLimit -
XX:+ExplicitGCInvokesConcurrent -XX:+HeapDumpOnOutOfMemoryError -XX:+ExitOnOutOfMemoryError -cp /data/SmartbiUnionServer/lib/*:../target/* -Dlog.output-file=/data/SmartbiUn
ionServer/var/log/server.log -Dnode.data-dir=/data/SmartbiUnionServer/data -Dnode.id=ffffffff-ffff-ffff-ffff-ffffffffffff -Dnode.environment=production -Dlog.enable-console
=true -Dlog.levels-file=/data/SmartbiUnionServer/etc/log.properties -Dconfig=/data/SmartbiUnionServer/etc/config.properties -Djava.security.egd=file:/dev/./urandom smartbix
.presto.Server
root      19909  19481  0 13:56 pts/0    00:00:00 grep --color=auto SmartbiUnionServer
```

如果进程不存在，可以查看/opt/SmartbiUnionServer/var/log/server.log，查看报错信息。如果提示Address already in use，说明端口冲突了，需要修改/opt/SmartbiUnionServer/etc/config.properties里的端口，然后重启presto。

```
[root@smartbi log]# pwd
/data/SmartbiUnionServer/var/log
[root@smartbi log]# ll -h
总用量 580K
-rw-r--r--. 1 root root    0 5月  29 13:53 http-request.log
-rw-r--r--. 1 root root   20 5月  29 13:53 http-request.log-2019-03-15.0.log.gz
-rw-r--r--. 1 root root 494K 5月  29 14:02 server.log
[root@smartbi log]#
```

4.2、停止SmartbiUnionServer

通过命令：ps -ef | grep SmartbiUnionServer

查到Smartbiunionserver的进程号：

然后使用kill -9 <进程号>命令杀掉Smartbiunionserver进程。

4.3、设置开机启动

Linux部署SmartbiUnionServer开机启动设置方式：

4.3.1方法一：

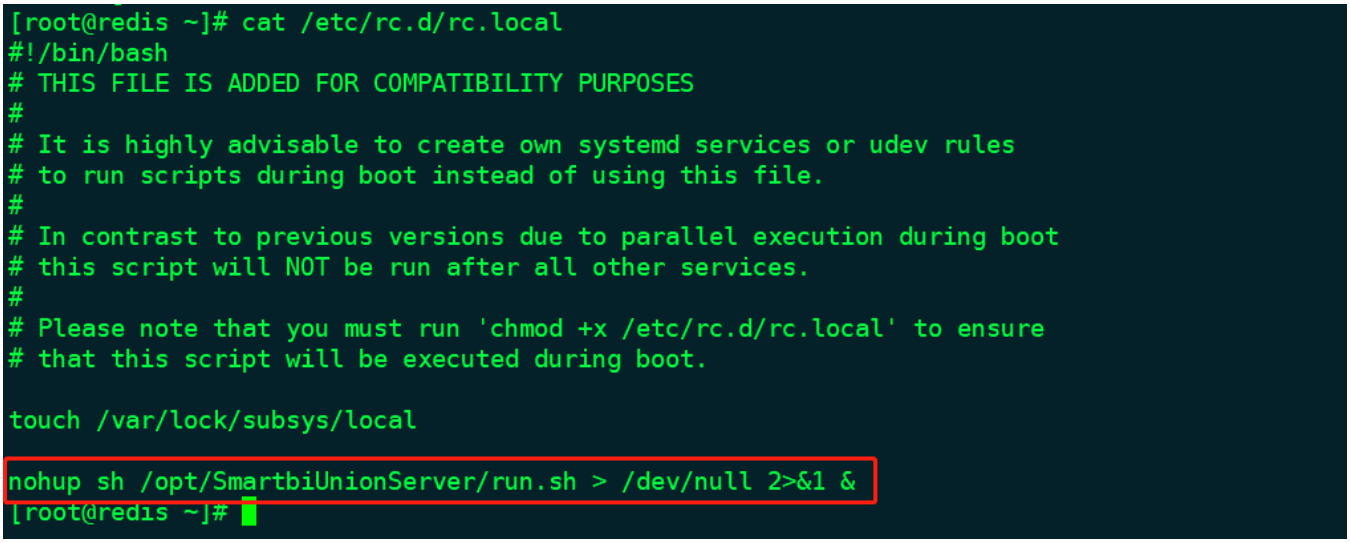
Centos6.x

①赋予脚本可执行权限（/opt/SmartbiUnionServer/run.sh是SmartbiUnionServer的脚本路径）

```
chmod +x /opt/SmartbiUnionServer/run.sh
```

②在/etc/rc.d/rc.local文件末尾增加添加SmartbiUnionServer的脚本启动命令，保存退出

```
vi /etc/rc.d/rc.local
#
nohup sh /opt/SmartbiUnionServer/run.sh > /dev/null 2>&1 &
```



Centos7.x

① 赋予脚本可执行权限（/opt/SmartbiUnionServer/run.sh是SmartbiUnionServer的脚本路径）

```
chmod +x /opt/SmartbiUnionServer/run.sh
```

② 在/etc/rc.d/rc.local文件末尾增加添加SmartbiUnionServer的脚本启动命令，保存退出

```
vi /etc/rc.d/rc.local
#
nohup sh /opt/SmartbiUnionServer/run.sh > /dev/null 2>&1 &
```

```
[root@test ~]# cat /etc/rc.d/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
nohup sh /opt/SmartbiUnionServer/run.sh > /dev/null 2>&1 &
[root@test ~]#
```

③ 在centos7中，/etc/rc.d/rc.local的权限被降低了，所以需要执行如下命令赋予其可执行权限

```
chmod +x /etc/rc.d/rc.local
```

Suse12

① 赋予脚本可执行权限（/opt/SmartbiUnionServer/run.sh是SmartbiUnionServer的脚本路径）

```
chmod +x /opt/SmartbiUnionServer/run.sh
```

② 在/etc/rc.d/after.local文件末尾增加添加SmartbiUnionServer的脚本启动命令，保存退出

```
vi /etc/rc.d/after.local
#
nohup sh /opt/SmartbiUnionServer/run.sh > /dev/null 2>&1 &
```

```
smartbi:~ # cat /etc/rc.d/after.local
#! /bin/sh
#
# Copyright (c) 2010 SuSE LINUX Products GmbH, Germany.  All rights reserved.
#
# Author: Werner Fink, 2010
#
# /etc/init.d/after.local
#
# script with local commands to be executed from init after all scripts
# of a runlevel have been executed.
#
# Here you should add things, that should happen directly after
# runlevel has been reached.
#
nohup sh /opt/SmartbiUnionServer/run.sh > /dev/null 2>&1 &
smartbi:~ #
```

③ 给/etc/rc.d/after.local添加执行权限

```
chmod +x /etc/rc.d/after.local
```

4.3.2 方法二

进入/etc/init.d目录，创建unionserver启动配置文件

```
vi /etc/init.d/unionserver
```

配置参考如下：

```
#!/bin/bash
# chkconfig: 345 80 20
# description: start the unionserver deamon
#
# Source function library
. /etc/rc.d/init.d/functions

prog=unionserver
UNIONSERVER_HOME=/home/smartbi/SmartbiUnionServer/  #smartbi unionserver
export UNIONSERVER_HOME

case "$1" in
start)
    echo "Starting unionserver..."
    $UNIONSERVER_HOME/run.sh &
    ;;

stop)
    echo "Stopping unionserver..."
    kill -9 $(ps -ef | grep SmartbiUnionServer | grep jdk_linux | awk '{print $2}')
    ;;

restart)
    echo "Stopping unionserver..."
    kill -9 $(ps -ef | grep SmartbiUnionServer | grep jdk_linux | awk '{print $2}')
    sleep 2
    echo
    echo "Starting unionserver..."
    $UNIONSERVER_HOME/run.sh &
    ;;

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

设置开机启动

```
chmod +x /etc/init.d/unionserver    #
chkconfig unionserver on            #
chkconfig --list                    #
```

5、日志文件

SmartbiUnionServer 的日志路径：<SmartbiUnionServer >/var/log/server.log。

如果出现启动失败时，可以通过分析日志来判断问题点。

6、版本更新

1）停止现有的SmartbiUnionServer服务：

```
# ps -ef| grep SmartbiUnionServer
# kill -9 id
```

2）升级

① 备份原来的SmartbiUnionServer/plugin目录

```
# mv plugin pluin_back
```

复制解压出来的新版SmartbiUnionServer/plugin到原来的目录文件

```
# cp -r <SmartbiUnionServer>/plugin <SmartbiUnionServer>/plugin
```

② 备份原来的SmartbiUnionServer/lib目录

```
# mv lib lib_back
```

复制解压出来的新版SmartbiUnionServer/lib到原来的目录文件

```
# cp -r <SmartbiUnionServer>/lib <SmartbiUnionServer>/lib
```

③ 复制SmartbiUnionServer/etc/queue_config.json 到etc目录

```
# cp -r <SmartbiUnionServer>/etc/queue_config.json <SmartbiUnionServer>/etc/
```

④ 复制SmartbiUnionServer/etc/resource-groups.properties 到etc目录

```
# cp -r <SmartbiUnionServer>/etc/resource-groups.properties <SmartbiUnionServer>/etc/
```

3）启动

```
# nohup ./run.sh &
```

4）测试验证

使用 Smartbi 连接跨库联合数据源验证，详情请参考 跨库联合数据源。