**GGC环境搭建**

1. Gemfire与Greenplum 版本适配

| **GemFire-Greenplum Connector version** | **GemFire version** | **Greenplum version** |
| --- | --- | --- |
| 3.1.1 | 9.3 | 4.3.x, 5.x |
| 3.1.0 | 9.1 | 4.3.x, 5.0.x, and 5.1.x |

1. 安装jdk，配置环境变量
2. 下载gemfire-greenplum-3.1.0.jar
   1. 下载地址：<https://network.pivotal.io/products/pivotal-gemfire/>
3. 安装Gemfire，本例以Gemfire9.1.1 为例
   1. 下载地址：<https://network.pivotal.io/products/pivotal-gemfire/>
   2. 解压：unzip pivotal-gemfire-9.1.1.zip
   3. 在home/gemfire/创建目录

mkdir bin lib config gfsh log data locator server backup

* 1. 将步骤3下载包放置在home/gemfire/lib 目录
  2. 配置环境变量

vim /etc/profile

GEMFIRE=/home/gemfire/pivotal-gemfire-9.1.1

PATH=$PATH:$GEMFIRE/bin:/home/gemfire/bin

CLASSPATH=$CLASSPATH:/home/gemfire/lib/\*

CLASSPATH=$CLASSPATH

* 1. 验证
     1. 出现下面界面安装成功



* 1. 创建对象Parent对象，

int id

String name

BigDecimal income

对象打包，放置在home/gemfire/lib目录

* 1. 在home/gemfire/config添加如下配置
     1. locator.properties

bind-address=192.168.119.134[12345]

mcast-port=0

log-level=config

log-file=/home/gemfire/locator/locator.log

locators=192.168.119.134[12345]

enable-network-partition-detection=true

conserve-sockets=false

* + 1. gemfire.properties

bind-address=192.168.119.134

mcast-port=0

locators=192.168.119.134[12345]

log-level=config

cache-xml-file=/home/gemfire/config/cache-greenplum.xml

statistic-sampling-enabled=true

conserve-sockets=false

statistic-sample-rate=1000

statistic-archive-file=test2.gfs

off-heap-memory-size=256M

enforce-unique-host=true

* + 1. cache-greenplum.xml

<?xml version="1.0" encoding="UTF-8"?>

<cache

xmlns="http://geode.apache.org/schema/cache"

xmlns:gpdb="http://schema.pivotal.io/gemfire/gpdb"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://geode.apache.org/schema/cache http://geode.apache.org/schema/cache/cache-1.0.xsd http://schema.pivotal.io/gemfire/gpdb http://schema.pivotal.io/gemfire/gpdb/gpdb-2.4.xsd"

version="1.0">

<!--对象序列化-->

<pdx read-serialized="true" persistent="false">

<pdx-serializer> <class-name>org.apache.geode.pdx.ReflectionBasedAutoSerializer</class-name>

<!--映射对象-->

<parameter name="classes"> <string>io.pivotal.gemfire.demo.entity.\*</string>

</parameter>

</pdx-serializer>

</pdx>

<!--数据源-->

<jndi-bindings>

<!--数据库、登录信息-->

<jndi-binding jndi-name="DemoDatasource" type="SimpleDataSource"

jdbc-driver-class="org.postgresql.Driver" user-name="gpadmin"

password="pivotal" connection-url="jdbc:postgresql://localhost:5432/DemoDatasource">

</jndi-binding>

</jndi-bindings>

<!--gemfire中region与对象映射-->

<region name="Parent">

<region-attributes refid="PARTITION" off-heap="true" >

<partition-attributes redundant-copies="1" recovery-delay="300000" startup-recovery-delay="600000"

total-num-buckets="29" />

</region-attributes>

<!-- 指定的数据库 -->

<gpdb:store datasource="DemoDatasource">

<gpdb:types>

<!-- 对象 -->

<gpdb:pdx name="io.pivotal.gemfire.demo.entity.Parent"

schema="public"

table="parent"> <!-- 数据表 -->

<gpdb:id field="id" />

<!-- 对象与数据表映射 -->

<gpdb:fields>

<gpdb:field name="id" column="id" />

<gpdb:field name="name" />

<gpdb:field name="income" class="java.math.BigDecimal" />

</gpdb:fields>

</gpdb:pdx>

</gpdb:types>

</gpdb:store>

</region>

</cache>

* 1. home/bin 创建脚本
     1. locator启动脚本

startlc.sh

#!/bin/sh

NAME=`hostname`

gfsh start locator --port=12345 --dir=/home/gemfire/locator --name=locator\_${NAME} --J=-Dgemfire.jmx-manager-port=20001 --J=-Xms256m --J=-Xmx256m --J=-Dgemfire.enable-cluster-configuration=false --J=-Dgemfire.use-cluster-configuration=false --include-system-classpath --properties-file=/home/gemfire/config/locator.properties &

* 1. cacheServer启动脚本

startServer.sh

#!/bin/sh

HOSTNAME=`hostname`

DATE=`date +%Y-%m-%d-%H-%M`

gfsh start server --name=${HOSTNAME}\_gem\_1 --locators=192.168.119.134[12345] --server-port=60020 --include-system-classpath --properties-file=/home/gemfire/config/gemfire.properties --J=-Xms256M --J=-Xmx256M --J=-Xss256k --J=-XX:+UseParNewGC --J=-XX:+UseConcMarkSweepGC --J=-XX:CMSInitiatingOccupancyFraction=70 --J=-XX:+UseCMSInitiatingOccupancyOnly --J=-XX:+PrintGCDetails --J=-Xloggc:/home/gemfire/log/gc\_${DATE}.log --J=-Dgemfire.enable-cluster-configuration=false --J=-Dgemfire.use-cluster-configuration=false &

1. 验证连接

connect

help import gpdb

输出一下内容，连接成功

IS AVAILABLE

true

1. 从Greenplum导出数据到Gemfire

import gpdb --region=Parent

1. 从Gemfire 导出数据到Greemplum

export gpdb --region=Parent --type=UPSERT

Type说明：

UPSERT：更新已存在内容，插入新的数据

INSERT\_ALL： region中数全部插入greenplum

INSERT\_MISSING：插入新的数据，不更新已存在数据

UPDATE：更新已存在的数据

https://help.github.com/articles/about-pull-requests/