**Environment configuration**

**一 Tensorflow**

The project needs tensorflow environment, tensorflow is a machine learning tool, you can install tensorflow by yourself or use docker. The image name is hsuppr/tensorflow:1.15.2-py3-jupyter-jdk8 which also includes jdk8 and jupyter.

**二 psycopg2**

The psychopg2 library is a third-party library used by Python to operate the PostgreSQL database. Installation is required before use.

pip install psycopg2

**三 Database environment configuration**

**1. stand-alone version**

**① PostgreSQL database with FDW**

As the stand-alone version, this project uses PostgreSQL database and adds FDW extension to it.

You can download and install it directly or use docker image named vidardb/postgresql:rocksdb. Use the following docker statement to install it:

*docker run --name db -itd --net=host -p 5432:5432 --log-opt max-size=10m --log-opt max-file=3 -v pgdata:/var/lib/postgresql/data \vidardb/postgresql:rocksdb-6.2.4*

**② Create table structure**

Then we set up a database called test to build kV\_ FDW extension, and kV\_ Server service:

*CREATE DATABASE test;*

*\c test*

*CREATE EXTENSION kv\_fdw;*

*CREATE SERVER kv\_server FOREIGN DATA WRAPPER kv\_fdw;*

**2. distributed version**

As the distributed version we use Greenplum database.Here is how to configure the Greenplum database：

**① Interconnection between docker containers across physical machines.**

Install docker under each centos7 physical machine.

Download Image for docker of each machine: *docker pull pivotaldata/gpdb-dev:centos7*

Configure docker 0 on the host, and configure the IP address of docker container for each physical machine.

On the host’s File(/etc/docker/daemon.json) configuration: {"bip": "172.17. X.252/24"}

Restart the docker service to make the modified docker 0 network segment take effect.

Configure the routing table for all hosts and add routing rules.

Touch /etc /sysconfig /network scripts/route XXX (XXX is EM3) Write each rule 172.17.x.0/24 via host IP dev network card name in the file.

Create containers on each host:

*docker run --name mdw -it --privileged - d pivotaldata/gpdb-dev:centos7 /usr/sbin/init*

**②Docker container environment configuration**

Modify the host name of each server:  *vi /etc/hostname*

In each docker container, modify the /etc/hosts file to establish the mapping between container ID, host name and IP.

Modify the /etc/sysconfig/network file in all nodes to keep it consistent with the host name:

*vi /etc/sysconfig/network NETWORKING=yes HOSTNAME=mdw*

Modify the file open limit on each node:

*vi /etc/security/limits.conf*

*# End of file*

*\* soft nofile 65536*

*\* hard nofile 65536*

*\* soft nproc 131072*

*\* hard nproc 131072*

Close the firewall on each node, and close SELinux.

In each docker container, configure the kernel parameter:

*cat /proc/sys/kernel/sem*

*sysctl -w kernel.sem="250 32000 32 350"*

**③Install Greenplum database**

Installing greenplus in gpadmin: Su gpadmin

To install the greenplus installation package under the master node:

git clone <https://github.com/greenplum-db/gpdb>

Install packages: ./readme.Centos.bash

Compile:

./configure --with-libxml --disable-orca --prefix=/home/gpadmin/lib/gpdb

make

make install

Create hostlist and seg\_hosts files under the master node to determine the IP address of segments and master:

mkdir -p /home/gpadmin/conf

vi /home/gpadmin/conf/hostlist

mdw

sdw1

sdw2

vi /home/gpadmin/conf/seg\_hosts

sdw1

sdw2

Configure the gpssh secret free connection under the master node:

source /home/gpadmin/lib/gpdb/greenplum\_path.sh

gpssh-exkeys -f /home/gpadmin/conf/hostlist

Packing greenplug under master node:

tar -cf gp.tar /home/gpadmin/gpdb

tar -cf lib.tar /home/gpadmin/lib

Send from master to all child nodes:

gpscp -f /home/gpadmin/conf/seg\_hosts gp.tar=:/home/gpdb/

gpscp -f /home/gpadmin/conf/seg\_hosts lib.tar=:/home/gpdb/

Batch operation child node, decompression file:

gpssh -f /home/gpadmin/conf/hostlist

cd /home/gpadmin

tar -xf gp.tar

tar -xf lib.tar

**④Greenplum database initialization**

Establish resource directory:

source /home/gpadmin/lib/gpdb/greenplum\_path.sh

gpssh -f /home/gpadmin/conf/hostlist

mkdir gpdata

cd gpdata

mkdir gpdatap1 gpdatap2 gpdatam1 gpdatam2 gpmaste

Configure environment variables in master, send them to child nodes and make them effective:

vi /home/gpadmin/.bash\_profile

source /home/gpadmin/lib/gpdb/greenplum\_path.sh

export MASTER\_DATA\_DIRECTORY=/home/gpadmin/gpdata/gpmaster/gpseg-1

export GPPORT=5432

export PGDATABASE=Greenplum

Send to child node:

gpscp -f /home/gpadmin/conf/seg\_hosts /home/gpadmin/.bash\_profile=:/home/gpdb/

gpssh -f /home/gpadmin/conf/hostlist

source .bash\_profile

Configuration initialization file:

cp -r gpdb/gpMgmt/doc/gpconfigs/gpinitsystem\_config /home/gpadmin/conf/

Modification content:

declare -a DATA\_DIRECTORY=(/home/gpadmin/gpdata/gpdatap1

/home/gpadmin/gpdata/gpdatap1 /home/gpadmin/gpdata/gpdatap1

/home/gpadmin/gpdata/gpdatap1 /home/gpadmin/gpdata/gpdatap1

/home/gpadmin/gpdata/gpdatap1 /home/gpadmin/gpdata/gpdatap2

/home/gpadmin/gpdata/gpdatap2 /home/gpadmin/gpdata/gpdatap2

/home/gpadmin/gpdata/gpdatap2 /home/gpadmin/gpdata/gpdatap2

/home/gpadmin/gpdata/gpdatap2)

MASTER\_HOSTNAME=mdw

MASTER\_DIRECTORY=/home/gpadmin/gpdata/gpmaster

MASTER\_PORT=5432

DATABASE\_NAME=greenplum

MACHINE\_LIST\_FILE=/home/gpadmin/conf/seg\_hosts

Initialize database:

export GPPORT=5432

export MASTER\_DATA\_DIRECTORY=/home/gpadmin/gpdata/gpmaster/gpseg-1

. ~/.bash\_profile

gpstart -a

createdb test

select \* from gp\_segment\_configuration ;