软件栈

su huihuang

2021

软件栈

目录

[软件栈 1](#_Toc85530448)

[1. 目录规划 2](#_Toc85530449)

[1.1. 目录说明 2](#_Toc85530450)

[1.2. 软件版本 2](#_Toc85530451)

[2. 一键安装说明 3](#_Toc85530452)

[2.1. 功能说明 3](#_Toc85530453)

[3. 系统源 4](#_Toc85530454)

[3.1. nginx 4](#_Toc85530455)

[3.2. 配置源 4](#_Toc85530456)

[3.3. 客户端的配置文件 4](#_Toc85530457)

[4. 深度学习框架 5](#_Toc85530458)

[4.1. Anaconda 3 5](#_Toc85530459)

[4.2. TensorFlow 1.13 6](#_Toc85530460)

[4.3. PyTorch 1.7.0 6](#_Toc85530461)

[5. 智能加速卡 8](#_Toc85530462)

[5.1. B卡驱动安装 8](#_Toc85530463)

[5.2. caffe 模型转换 10](#_Toc85530464)

[6. 事件分析系统 11](#_Toc85530465)

[6.1. 数据库安装配置 11](#_Toc85530466)

[6.2. 预警模块配置 11](#_Toc85530467)

[6.3. JDK 1.8 11](#_Toc85530468)

[6.4. elasticsearch 5.6.1 12](#_Toc85530469)

[6.5. apache-tomcat 7.0.47 13](#_Toc85530470)

[6.6. 后端配置 14](#_Toc85530471)

[7. 图数据库 15](#_Toc85530472)

[7.1. neo4j-community 3.4.17 15](#_Toc85530473)

[7.2. Janusgraph 0.4 15](#_Toc85530474)

[8. 大数据体系 16](#_Toc85530475)

[8.1. Hadoop 16](#_Toc85530476)

[8.2. Jj 16](#_Toc85530477)

[8.3. L 16](#_Toc85530478)

# 目录规划

## 目录说明

|  |  |  |
| --- | --- | --- |
| 序号 | 目录 | 说明 |
| 1 | /app/soft | 所有软件目录 |
| 2 | /app/conf | 配置文件目录 |
| 3 | /app/public | 公共资料安装目录 |
| 4 | /app/script | 脚本目录 |
| 5 | /app/mirrors | 系统源目录 |
| 6 | /app/biaozhu | 事件系统目录 |

## 软件版本

|  |  |  |
| --- | --- | --- |
| 序号 | 软件名称 | 版本 |
| 1 | Nginx | 1.18 |
| 2 | JDK | 1.8.0\_27 |
| 3 | Mysql | 5.7 |
| 4 | Conda (Anaconda3) | 4.5.12 |
| 5 | Python | 3.7 |
| 6 | Python (test) | 3.6 |
| 7 | TensorFlow (python 3.6 CPU ) | 1.13 |
| 8 | PyTorch (python 3.6 CPU) | 1.7.0 |
| 9 | Elasticsearch | 5.6.1 |
| 10 | Kibana | 5.6.1 |
| 11 | neo4j-community | 3.4.17 |
| 12 | Janusgraph | 0.4 |
| 13 | Caffe-ssd |  |
| 14 | Hadoop |  |
| 15 | Zookeep |  |
| 16 | Kafka |  |

# 一键安装说明

一键安装，为了更安全的操作，所有的操作都以普通用户（kylin）执行，脚本会对系统进行检测当前用户，如果用户不存在，创建用户；如果用户存在，则切换到kylin用户。

另外，为了标准化，所有的安装文件都安装在 /app 目录下面，所有目录严格按照上面的目录规划。

## 功能说明

使用方法：

cd /app/scripts

./install

Usage: install [anaconda|nginx|jdk|mysql|pytorch|tensorflow]

例如：安装 anaconda

cd /app/scripts

./install anaconda

# 系统源

## nginx

nginx 作为系统源的发布服务器，安装二进制安装包版本。

cd /app/soft

tar xf nginx1.18\_linux\_arm.tar.gz -C /app/public

ln -sf /app/public/nginx1.18/sbin/nginx /usr/local/sbin/nginx

cp /app/conf/nginx/nginx.service /lib/systemd/system/nginx.service

systemctl daemon-reload

systemctl enable nginx.service

systemctl restart nginx.service

## 配置源

解压源包到nginx发布目录

cd /app/mirrors/

tar xf ubuntu-ports.tar.gz -C /app/public/nginx1.18/html/ubuntu18.04

## 客户端的配置文件

请修改对应的IP地址

#deb file:///arm\_mirror/ubuntu-ports bionic main universe multiverse restricted

deb http://IP/ubuntu1804 bionic main universe multiverse restricted

# 深度学习框架

深度学习框架，tensoflow 1.13 和 pytorch 1.7.0 都是以python 3.6 环境下编译的CPU版本。

## Anaconda 3

我们采用的是移植 anaconda打包，因为我们把需要的库文件和依赖都安装好。如果你有需要安装全新的anaconda，我们也提供了基于ARM的 anaconda安装包（/app/soft/Archiconda3-0.2.2-Linux-aarch64.sh）。

举粟子：

激活python 3.6 环境：

source /app/public/archiconda3/etc/profile.d/conda.sh

conda activate test

注意：

如果与我的目录不相同，需要修改命令和第一行声明，把你自己的路径写上，如下：

#!/app/public/archiconda3/bin/python

修改成如下：

#!/你的安装路径/archiconda3/bin/python

## TensorFlow 1.13

TensorFlow 使用的是Python 3.6 CPU版本，只需要激活python 3.6 环境，然后引用 tensorflow 即可。具体操作如下：

tensorflow 位置：

/app/public/tensorflow1.13.1-py36

激活 tensorflow 环境：

cd /app/scripts

source start\_tensorflow.sh

激活环境脚本示意：

source /app/public/archiconda3/etc/profile.d/conda.sh

conda activate test

export PYTHONPATH=$PYTHONPATH:/app/public/tensorflow1.13.1-py36/lib/python3.6/site-packages/

## PyTorch 1.7.0

PyTorch 与TensorFlow一样，使用的是 python 3.6 CPU版本，只需要激活 python 3.6环境，然后引用 pytorch 即可。具体操作如下：

pytorch 位置：

/app/public/pytorch1.7.0-py36

激活 pytorch 环境：

Cd /app/scripts

source start\_pytorch.sh

激活环境脚本示意：

cat /app/scripts/start\_pytorch.sh

source /app/public/archiconda3/etc/profile.d/conda.sh

conda activate test

export PYTHONPATH=$PYTHONPATH:/app/public/pytorch1.7.0-py36/lib/python3.6/site-packages

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/app/public/openblas-FT2000+/lib

FAQ：

1. 库文件版本太低

ImportError: /app/public/archiconda3/envs/test/bin/../lib/libstdc++.so.6: version `GLIBCXX\_3.4.26' not found (required by /app/public/pytorch1.7.0-py36/lib/python3.6/site-packages/torch/lib/libtorch\_python.so)

解决办法：

cp /app/public/pytorch1.7.0-py36/lib/libstdc++.so.6.0.28 /app/public/archiconda3/envs/test/lib/libstdc++.so.6

# 智能加速卡

## B卡驱动安装

B 卡驱动安装之前，请确认B卡是否已经安装。如未安装，驱动将安装不上。

查看B 卡设备：

lspci | grep BM1684

查看驱动是否安装：

ls /dev/bm-sophon\*

lsmod | grep bms

安装驱动：

tar xf bmnnsdk2-bm1684\_v2.0.3.tar.gz -C /app/public

cd bmnnsdk2-bm1684\_v2.0.3/scripts

# 安装库文件

./install\_lib.sh nntc

# 安装 ufw

source envsetup\_pcie.sh ufw

# 安装 bmnetu

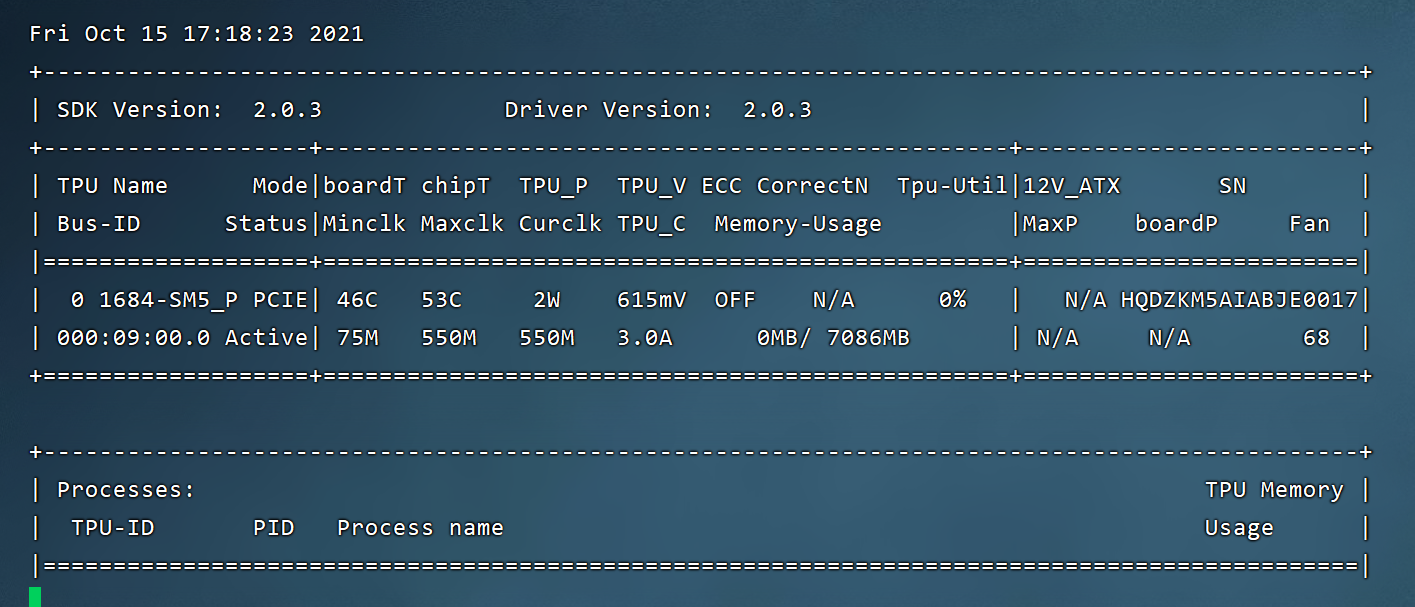
source envsetup\_pcie.sh bmnetu

检查B 卡状态：

cd /app/public/bmnnsdk2-bm1684\_v2.0.3/scripts

source envsetup\_arm\_pcie.sh

bm-smi



返回成功，即安装成功！

## caffe 模型转换

为什么要将caffe模型移植到咱们的定制卷积加速器上呢？

因为咱们平台上选用了这款加速器，并且它是一款自主可控、能效比高的加速器。这款加速器主要是在推理阶段对模型进行优化并实现高效运行。

而模型量化可以在精度损失微乎其微的的前提下，减少模型参数与计算量，我们的加速器提供了一套完整的量化工具链来实现模型的量化，并且加速器拥有定制计算单元，可以实现量化模型推理阶段的高效运行。

为了更好的适配我们的加速器的硬件设计，我们需要将训练好的caffemodel转换成面向加速器的深度神经网络模型文件。

caffe-ssd 位置：

/app/soft/caffe-ssd.tar.gz

tar xf caffe-ssd.tar.gz -C /app/public/

使用方法：

引入环境变量(source activate\_ft.sh)

cat activate\_ft.sh

export LD\_LIBRARY\_PATH=/app/public/caffe-ssd/build/lib:/usr/local/lib/3libopencv:$LD\_LIBRARY\_PATH

export PYTHONPATH=/app/public/caffe-ssd/python:$PYTHONPATH

# 事件分析系统

## 数据库安装配置

使用系统源来安装数据库：

apt-get install mysql-server mysql-client

在使用系统源安装之前，请先同步时间。

## 预警模块配置

模型所需的模块已安装到 conda 环境中，解压后，直接启动即可。

tar xf action\_prediction.tar.gz -C /app/biaozhu/

cp /app/conf/python/predic.service /lib/systemd/system/

systemctl daemon-reload

systemctl enable predic.service

systemctl start predic.service

## JDK 1.8

只要在arm服务器上需要安装arm版本的JDK。

cd /app/soft

tar xf jdk-8u271-linux-aarch64.tar.gz -C /app/public/

mv /app/public/jdk1.8.0\_271/ /app/public/jdk1.8

sed -i.ori '$a export JAVA\_HOME=/app/public/jdk1.8\nexport PATH=$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$PATH\nexport CLASSPATH=.$CLASSPATH:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib:$JAVA\_HOME/jre/lib:$JAVA\_HOME/lib/tools.jar' /etc/profile

source /etc/profile

## elasticsearch 5.6.1

Elasticsearch 必须要普通用户才能运行，请先确定你使用哪个用户运行elasticsearch。

useradd pdl

cd /app/soft

tar xf elasticsearch-5.6.1.tar.gz -C /app/biaozhu/

chown -R pdl.pdl /app/biaozhu

cp /app/conf/elasticsearch/elasticsearch.service /lib/systemd/system/

systemctl daemon-reload

systemctl enable elasticsearch.service

systemctl start elasticsearch.service

FAQ：

报错日志：

[1]: max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]

[2]: max virtual memory areas vm.max\_map\_count [26214] is too low, increase to at least [262144]

问题 1 解决办法：

echo “\* soft nofile 65536” >>/etc/security/limits.conf

echo “\* hard nofile 65536” >>/etc/security/limits.conf

问题2 解决办法：

echo “vm.max\_map\_count = 262144” >>/etc/sysctl.conf

sysctl -p

## apache-tomcat 7.0.47

tomcat 用于发布前端应用，只需要修改配置文件IP地址(这里在脚本中替换IP地址)，然后启动即可。

cd /app/soft

tar xf apache-tomcat-7.0.47.tar.gz -C /app/biaozhu/

IP=`ip a|grep enp|grep inet|awk -F '[ /]' '{print $6}'`

sed -i s#10.107.17.70#$IP#g /app/biaozhu/apache-tomcat-7.0.47/webapps/event-anno/resource/httpRequest.js

sed -i s#10.107.17.70#$IP#g /app/biaozhu/apache-tomcat-7.0.47/webapps/event-anno/views/event-wall/app.a9ed35ad.js

chown -R pdl.pdl /app/biaozhu

cp /app/conf/tomcat/tomcat.service /lib/systemd/system/

systemctl daemon-reload

systemctl enable tomcat.service

systemctl start tomcat.service

## 后端配置

后端应用，静态资源在配置文件中写绝对路径，需要注意，所用于依赖于前面的服务，所以必需最后启动。

cp /app/soft/zlxsfs-0.0.1-SNAPSHOT.jar /app/biaozhu/

tar xf annotation.tar.gz -C /app/biaozhu/

cp /app/conf/event/event.service /lib/systemd/system/

cp /app/conf/event/event\_start.sh /app/biaozhu/

systemctl enable event.service

systemctl start event.service

注意： Jar 包的静态资源和服务的启动顺序。

# 图数据库

## neo4j-community 3.4.17

neo4j-community 为社区版本。需要先安装JDK。

neo4j 安装：

cd /app/soft

tar xf neo4j-community-3.4.17-unix.tar.gz -C /app/public/

cd /app/public/neo4j-community-3.4.17/bin

./neo4j start

## Janusgraph 0.4

Janusgraph 目前版本迭代比较快，这里只提供一个比较新的安装包。Janusgraph 是基于java 开发，所有只需要安装JDK即可。对于一些周边工具请查看对应的版本所对应工具集版本号：

<https://github.com/JanusGraph/janusgraph/releases>

Janusgraph 的安装包位置：

/app/soft/janusgraph/

<https://github.com/JanusGraph/janusgraph/tags>

# 大数据体系

## Hadoop

l

## Jj

L;kj

l

## L

kk