H3C 异构服务器

厂商 xx加速卡xx模型推理/训练测试指导

资料版本：5W102-20240731

Copyright © 2025 新华三技术有限公司 版权所有，保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

除新华三技术有限公司的商标外，本手册中出现的其它公司的商标、产品标识及商品名称，由各自权利人拥有。

本文档中的信息可能变动，恕不另行通知。

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 修改描述 | 作者 |
| 2024.6.20 | V0.9 | 初稿 | 刘建/鲍闯/葛强/王新雨/彭超 |
| 2024.6.28 | V1.0 | 根据评审意见修改 | 李于川/刘建/鲍闯/葛强/王新雨/彭超/刘晶 |
| 2024.7.17 | V1.1 | 更新4.4.6、5.4.1和5.5.1章节 | 葛强/鲍闯 |
|  |  |  |  |

目 录

[1 xxx加速卡模型测试 1](#_Toc191997931)

[1.1 常规模型推理测试 1](#_Toc191997932)

[1.1.1 xxx模型推理测试 1](#_Toc191997933)

# xxx加速卡模型测试

## 常规模型（或大模型）推理（或训练）测试

### xxx模型推理（或训练）测试

#### 测试工具版本

|  |  |  |  |
| --- | --- | --- | --- |
| 文件名 | 版本号 | 内容 | 获取方式 |
| xre-ubuntu\_2204\_x86\_64.tar | 4.31.0.1 | R300加速卡驱动 | 用服ftp获取：/New\_Internal\_Versions(新内部版本归档)/11-服务器及商用存储产品/01-Server/GPU卡 |
| modelzoo\_ubuntu\_x86.tar.gz | / | Model zoo压缩包 | \\sqcsidata\TestTools\[GPU]\国产GPU |
| imagenet.tar.gz | / | 数据集压缩包 | \\sqcsidata\TestTools\[GPU]\国产GPU |
| ResNet50.tar.gz | / | ResNet50模型压缩包 | \\sqcsidata\TestTools\[GPU]\国产GPU |

#### 适用产品

|  |  |  |
| --- | --- | --- |
| 加速卡 | 机型 | 备注 |
|  |  |  |

#### BIOS设置

#### 环境部署

##### 驱动安装

* GPU-R300-32GB-MEZZ

安装linux系统，系统可安装成功，正常登录进系统，查看系统日志无相关GPU error。

将驱动文件xre-ubuntu\_2204\_x86\_64.tar（此处以ubuntu为例）拷贝至系统下。

解压驱动文件

tar -zxvf xre-ubuntu\_2204\_x86\_64.tar

cd xre-ubuntu\_2204\_x86\_64/

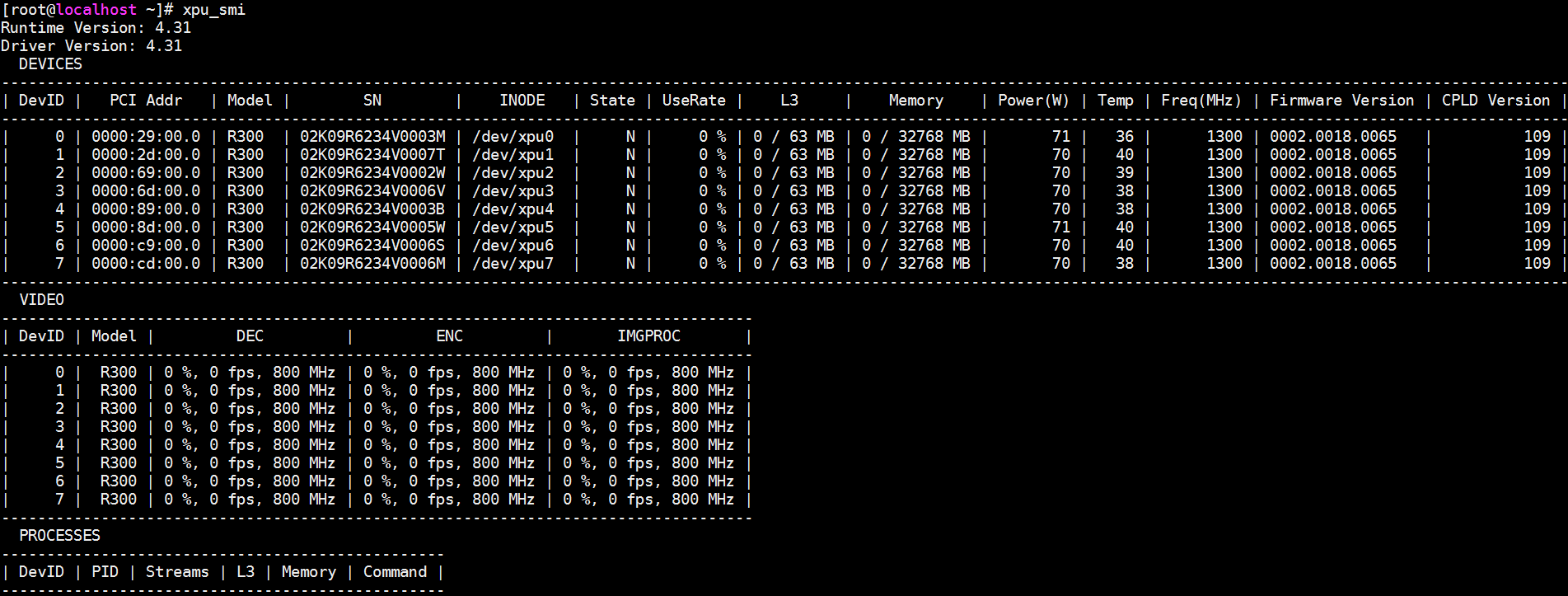
在系统下安装驱动包：

sudo ./install\_rt.sh

确认安装成功：

xpu\_smi

确认回显正常，可以正常显示环境中GPU



##### 安装docker

Docker版本要求：20.10.20及以上的版本

* 离线安装方式

wget https://download.docker.com/linux/static/stable/x86\_64/docker-20.10.20.tgz

wget https://klx-baidu-sp.bj.bcebos.com/musenXie/offline\_deploy\_docker.sh

bash offline\_deploy\_docker.sh

* 在线安装方式

#更新软件包列表：

sudo apt update

#安装所需的软件包，以允许apt使用HTTPS：

sudo apt install apt-transport-https ca-certificates curl software-properties-common

#添加Docker官方的GPG密钥:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

#添加Docker存储库

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

#更新软件包列表（以使新添加的存储库生效）：

sudo apt update

#安装Docker引擎：

sudo apt install docker-ce docker-ce-cli containerd.io

#验证Docker是否已成功安装，运行以下命令：

sudo docker run hello-world

如果一切正常，您将看到一个成功的消息，说明Docker已正确安装并运行。

##### 准备数据集

在目标Linux服务器上下载model zoo压缩包（modelzoo\_ubuntu\_x86）并解压：

|  |
| --- |
| wget <https://baidu-kunlun-public.su.bcebos.com/model_zoo/Latest/modelzoo_ubuntu_x86.tar.gz>  tar -zxf modelzoo\_ubuntu\_x86.tar.gz  cd modelzoo\_ubuntu\_x86/ |

此处以x86平台为例，准备模型测试需要的相关数据文件。在用于存放下载的数据集、产出和模型文件的 assets 目录中，下载模型测试需要的数据集和模型

|  |
| --- |
| cd assets |

* + - * 1. 下载ImageNet验证数据集：

去ImageNet 主页（https://image-net.org/）下载2012版的Validation数据集ILSVRC2012\_img\_val.tar，该数据集大小为6.3G。 或者直接从BOS下载：

wget https://baidu-kunlun-public.su.bcebos.com/model\_zoo/imagenet.tar.gz.00

wget https://baidu-kunlun-public.su.bcebos.com/model\_zoo/imagenet.tar.gz.01

wget https://baidu-kunlun-public.su.bcebos.com/model\_zoo/imagenet.tar.gz.02

cat imagenet.tar.gz.\* > imagenet.tar.gz

tar -zxf imagenet.tar.gz

* + - * 1. 模型可以去ResNet50-v1.5的公共模型地址。（https://zenodo.org/record/2535873/files/ResNet50\_v1.pb）下载，或直接从BOS上下载：

wget https://baidu-kunlun-public.su.bcebos.com/model\_zoo/ResNet50.tar.gz

tar -zxf ResNet50.tar.gz

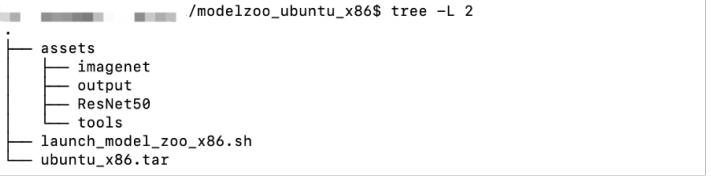
* + - * 1. 3.解压x86平台Docker镜像文件。

cd ..

tar -zxf ubuntu\_x86.tar.gz

最终所有文件均放置在modelzoo\_ubuntu\_x86文件夹下，如图所示。

目录结构



##### 环境准备

确保镜像文件和容器加载成功，此处以x86平台为例，具体操作如下：

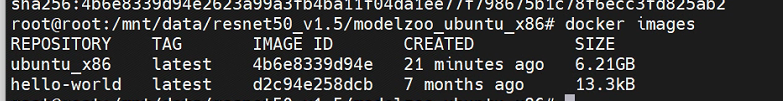
通过如下命令，加载镜像文件。

docker import ubuntu\_x86.tar ubuntu\_x86:latest



执行如下命令，如果显示类似如下信息，则代表镜像文件已经加载成功。

docker images



加载容器

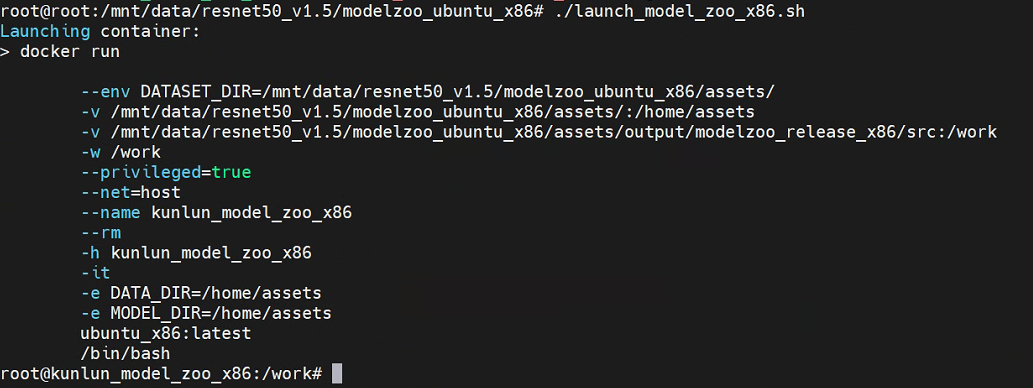
* + - * 1. 执行chmod +x xxx.sh命令，为启动脚本增加执行权限。

chmod +x launch\_model\_zoo\_x86.sh

* + - * 1. 执行以下命令，启动Docker。

./launch\_model\_zoo\_x86.sh

如果显示类似如下信息，则代表加载成功已进入到Docker环境。



#### 测试执行

* + - * 1. 在Docker环境中添加环境变量：

source /home/assets/tools/kl2\_set\_env.sh



* + - * 1. 通过make run命令执行模型推理：

ResNet50大算子INT8量化的推理命令：

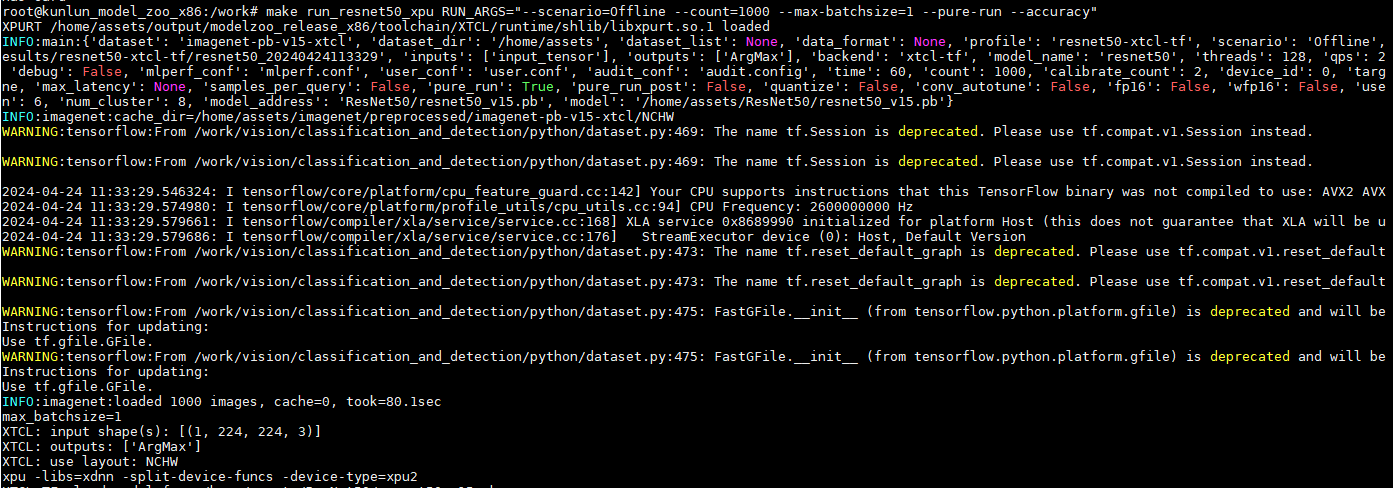
|  |
| --- |
| make run\_ResNet50\_custom\_xpu RUN\_ARGS="--scenario=Offline --count=1000 --max-batchsize=1,4,8,16,24,32,48,64,192 --pure-run --accuracy" |

ResNet50大算子FP16量化的推理命令：

|  |
| --- |
| make run\_ResNet50\_custom\_xpu RUN\_ARGS="--scenario=Offline --count=1000 --fp16 --max-batchsize=1,4,8,16,24,32,48,64,192 --pure-run --accuracy" |

INT8自动量化的推理命令：

|  |
| --- |
| make run\_ResNet50\_xpu RUN\_ARGS="--scenario=Offline --count=1000 --max-batchsize=1,4,8 --quantize --pure-run --accuracy" |



参数说明

* --count：测试样本数。
* --scenario：测试的情景，可选：[SingleStream, MultiStream, Server, Offline]。
* --max-batchsize：batchsize的数量，同时也支持批量输出性能数据，例如：1,8批量输出。
* batchsize=1 和batchsize=8的性能数据。
* --pure-run：pure run性能统计的开关，只统计推理部分的时间。
* --accuracy：输出性能数据的同时输出精度数据。
* --fp16: fp16推理。
* --quantize: INT8量化推理。

结果输出

现在支持两种输出格式，分别是界面的打印输出和文件保存的方式

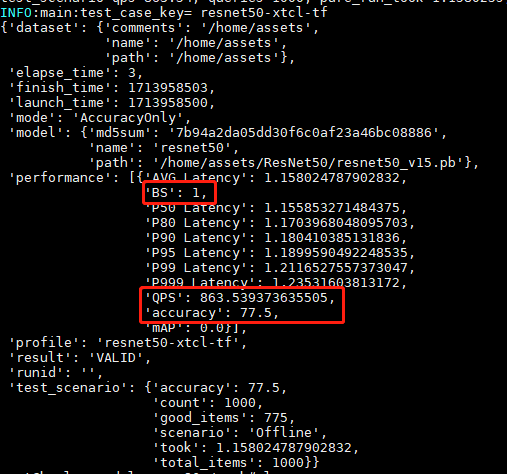
打印结果中主要关注BS、QPS、accuracy这三个字段

BS：表示batchsize的数量

QPS:表示（Query Per Second）每秒可处理的请求，越大性能越好

Accuracy：表示计算准确率

界面输出



文件自动保存，输出目录的位置和格式：对应任务/results/配置/模型\_时间/results.json。 示例： /work/vision/classification\_and\_detection/results/ResNet50-xtcl-tf /ResNet50\_2022032512xxxx/results.json