Benchmark Docker 申请流程和使用指南

尊敬的客户您好，欢迎试用寒武纪Benchmark Docker。为了更好展示寒武纪AI芯片的强大性能，我们为您准备了这个试用环境。通过这个试用环境中的一系列demo您可以亲身体验寒武纪产品的使用运行。

如果您想了解更多的销售信息请联系我们的销售人员：sales@cambricon.com

1. 申请流程

要试用Benchmark Docker请将以下信息填写后，以邮件形式发送至eco@cambricon.com。请填写真实有效的信息，以确保审核成功后我们能联系到您。

邮件标题：寒武纪云平台企业体验版申请。

|  |  |  |  |
| --- | --- | --- | --- |
| 基本信息 | | | |
| 姓名 |  | 所在地 | *请填写您的所在地* |
| 单位 | *请填写公司名称* | 职位 | *请填写您的职位* |
| 行业 | *请填写所属的行业* | 微信 | *请填写个人微信号* |
| 邮箱 | *请务必填写企业邮箱，环境的相关信息我们将发送至此。* | 论坛ID | *方便与您联系发送邀请邮件等事宜* |
| 用途及应用场景描述 | | | |
| *请填写具体的应用场景如： 电力巡检* | | | |
| 您主要关注的领域 | | | |
| *如机器学习、语音、自然语言处理、推荐系统等* | | | |
| 您关注哪些资讯平台 | | | |
| *如新智元、机器之心、量子位等* | | | |

申请通过之后我们会将服务器的登录信息发送到您的申请邮箱中。例如：

登录地址： 120.236.247.203:31843

用户名：root

密码：flzx!@#

服务器创建时间：2020-05-22 09:59

服务器到期时间：2020-05-25 09:59

请注意！**账号有效期为3天**，服务器到期时会自动收回，容器只用于展示Benchmark，**请勿保存文件**。

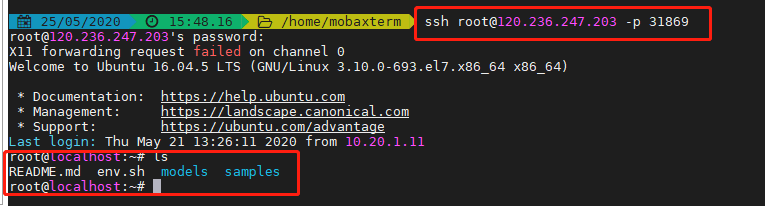
1. 使用方法

**登录服务器：**

首先请使用ssh 客户端登录服务器。

例如：ssh root@120.236.247.203 -p 31843

登录成功之后您将看到如下内容：



**编译demo：**

请依次执行以下命令。

source env.sh *#设置环境变量*

cd samples/

./build.sh *#执行编译脚本*

**执行demo：**

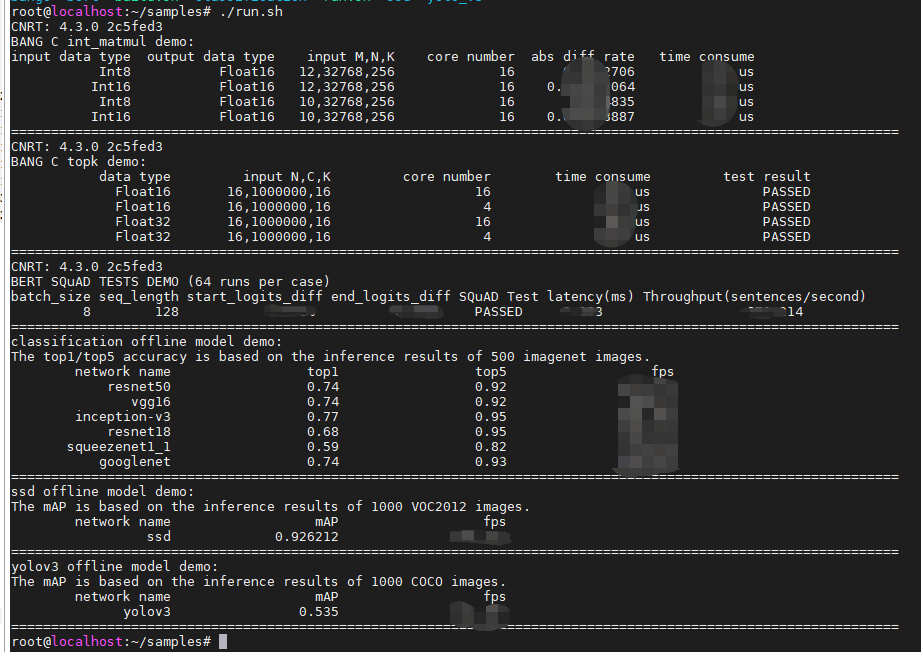
source env.sh *#设置环境变量*

cd samples/

./run.sh *#执行运行脚本*

除了使用以上编译脚本和运行脚本之外，您也可以独立编译和运行每个demo。

**运行效果示例：**



1. Demo内容介绍

目录结构：

models/ 包括了以上网络的离线模型

samples/

├─ bangc

├── mlu\_bangc\_gemm 矩阵乘的BANG C实现

├── mlu\_bangc\_topk topk 的BANG C实现

├── bert BERT网络的BANG C实现

├── classification 常用分类网络的离线模型和程序

├── ssd SSD检测网络的离线模型和程序

└── yolo\_v3 YOLOv3 检测网络的离线模型和程序

|  |  |
| --- | --- |
| demo | 包含内容 |
| mlu\_bangc\_gemm | BANG C device端代码，Host端代码 |
| mlu\_bangc\_topk |
| bert | BANG C device端代码，Host端代码，推理依赖的数据 |
| classification | 离线模型运行代码（基于CNRT），离线模型，推理依赖的数据 |
| ssd |
| yolo\_v3 |

运行输出内容：

**mlu\_bangc\_topk 输出内容：**

data type: 输入数据类型, 包括了Float16 和Float32。

input N,C,K： N 是 batchsize, C 是输入长度, K是Topk所要的找的K个最大的元素。

core number： 运行时MLU调用到的核数。

time consume： 总的MLU硬件计算时间。

test result： 和CPU的比较结果。

**mlu\_bangc\_gemm输出内容：**

input data type: 输入数据类型。

output data type: 矩阵乘结果的输出数据类型。

input M,N,K: 输入数据规模, 左矩阵规模M \* K, 右矩阵规模K \* N。

core number: 运行时MLU调用到的核数。

abs diff rate: 和CPU计算结果的比较。具体的误差计算公式：sum(abs(mlu[i] - cpu[i]))/input\_length。

time consume: 总的MLU硬件计算时间。

**bert 输出内容：**

batch size: 输入的batch size。

seq length: 输入句子的长度。

start logits diff: start logits和CPU相比的计算误差。

end logits diff: end logits 和CPU相比的计算误差 对于问答类任务，需要2个logit：一个用于开始位置，一个用于结束位置。 根据这2个logit，您有一个答案范围（以开始/结束位置表示）。

SQuAD Test: 基于SQuAD (The Stanford Question Answering Dataset.) 的测试结果

latency(ms): 总的MLU硬件计算时间。

Throughput(sentences/second): SQuAD测试的吞吐量。

**classification输出内容：**

network name: 离线模型名字。

top1：基于500张magenet图片的top1推理精度。

top5：基于500张magenet图片的top5推理精度。

fps：每秒推理帧数。

**ssd 输出内容：**

network name: 离线模型名字。

mAP: 基于1000张VOC2012图片的SSD推理mAP。

fps: 每秒推理帧数。

**yolo\_v3 输出内容：**

network name: 离线模型名字。

mAP: 基于1000张COCO图片的YOLOv3推理mAP。

fps: 每秒推理帧数。