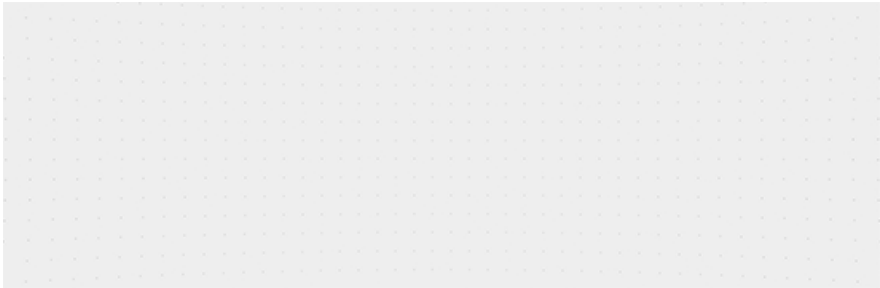


实操教程 | 使用Docker为无网络环境搭建深度学习环境

原创 有为少年 极市平台 2021-08-21 22:00:00 手机阅读 𐄞

↑ 点击蓝字 关注极市平台



作者 | 有为少年

编辑 | 极市平台



月发文数目: **:

月平均阅读: **:

文章工具

已发文

采集图文 合成多

采集样式 查看

极市导读

本文尝试利用独立配置好的容器环境来作为服务器端运行代码的环境，通过借助于Docker，成功的在服务器端训练起了模型。 >>加入极市CV技术交流群，走在计算机视觉的最前沿



前言

最近需要在一个特殊的环境下训练模型，虽然硬件设备都不错，也可以向服务器拷贝东西，但是重要的一个问题是，服务器端没有连接外网，这导致搭建深度学习模型训练环境的时候，不能自动通过网络来安装特定的软件了。

虽然说，可以将代码用到的所有的包都下载好，将其安装传输到服务器中安装，但是由于python包的迁移太过麻烦琐碎，实操起来并不方便。于是就尝试利用独立配置好的容器环境来作为服务器端运行代码的环境。

最终，通过借助于Docker，成功的在服务器端训练起了模型。

下面记录下大致的步骤和细节。

关于Docker的基本使用可参考我的另一篇总结docker指令的文档：<https://www.yuque.com/lart/linux/fp6cla>

安装过程

Docker本体

首先准备好Docker的安装包。

查阅官方文档可以看到有提供关于安装方法的指引：<https://docs.docker.com/engine/install/ubuntu/#installation-methods>

这里我们选择手动安装：<https://docs.docker.com/engine/install/ubuntu/#install-from-a-package>

64位Ubuntu18.04的系统对应的安装包下载仓库：<https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd64/>

要注意，这里存放了三个包，`containerd.io` `docker-ce-cli` `docker-ce` 这些都是需要安装的。都需要下载，如果没有特别需求的话，直接都选最新版本下载即可。

GPU相关插件

这里主要为Docker提供GPU的基础支持。从Docker的文档https://docs.docker.com/config/containers/resource_constraints/#gpu中可以看到相关内容。

首选需要确保你的服务器已经装好了显卡驱动。并留意驱动版本（最好版本新一些，这样可以用更新的NVIDIA提供的具有针对性优化的框架的Docker镜像）。

之后准备下载相关的插件，由上面的页面中可以看到提供了一个github.io的页面：<https://nvidia.github.io/nvidia-container-runtime/>

这个页面中可以看到对应的linux发行版的支持情况，如果确保无误，可以直接略过该页面的其他内容，直接进入对应的github仓库<https://github.com/NVIDIA/nvidia-container-runtime/tree/gh-pages/stable/ubuntu18.04/amd64>下载包文件。

这里要注意到，这里同样提供了多个包，我们需要下载的是：

- `nvidia-container-toolkit`
- `nvidia-container-runtime`

另外，我们还需要去<https://github.com/NVIDIA/libnvidia-container/tree/gh-pages/stable/ubuntu18.04/amd64>下载依赖包：

- `libnvidia-container-tools`
- `libnvidia-container1`

这下基础的环境就准备好了，接下来开始准备打包镜像。

构建并导出镜像

由于使用的是pytorch，并且想要使用由NVIDIA提供的镜像，所以我们需要从<https://ngc.nvidia.com/catalog/containers/nvidia:pytorch>这里查找对应版本的镜像。

这里<https://docs.nvidia.com/deeplearning/frameworks/pytorch-release-notes/index.html>提供了不同镜像发布版本具体的内部的包含的库的版本信息，选择对应版本下载。

注意，这里需要保证使用的镜像中的CUDA和服务器的驱动互相兼容。可见各个页面下的Driver Requirements一节中的提示。

这里我们选择使用https://docs.nvidia.com/deeplearning/frameworks/pytorch-release-notes/rel_20-09.html#rel_20-09这一版本的镜像。

本地需先执行：`sudo docker pull nvcr.io/nvidia/pytorch:20.09-py3`。

之后通过 `sudo docker images` 查看对应镜像的名字和标签（tag）。

将镜像pull到本地后，基于它首先启动一个容器：

```
sudo docker run --gpus\
    -it \
    --ipc host \
    -v /Code:/workspace/Code \ # NVIDIA提供的Docker镜像的工作目录是/workspace，这里为了方便直接将代码指向其中的文件夹，这里为了不增加未知
    -e ENV_VARS="Something" \
    IMAGE_REPOSITORY:IMAGE_TAG \ # 或者直接使用对应的ID也可以
    bash
```

在容器里，我们首先将代码需要安装的依赖包全部安装好，之后输入 `exit` 退出后，使用 `sudo docker ps -a` 查看这个退出的容器的ID。

然后使用 `sudo docker commit <CONTAINER ID> <NEW_IMAGE_NAME : NEW_IMAGE_TAG>` 将之前更新后的容器打包成一个新的名字为 `<NEW_TAG>` 的镜像。

最后使用 `sudo docker save -o <NEW_IMAGE_TAR_NAME>.tar <NEW_IMAGE_NAME: NEW_IMAGE_TAG>` 这里我们将其打包为TAR文件。

拷贝大文件

一般这个导出的镜像文件会>10G，这对于某些特定的文件系统而言，可能无法拷贝（假设我们这里使用U盘拷贝数据）。所以我们首先切割成小文件再拷贝。

本地 -> U盘：`cat <NEW_IMAGE_TAR_NAME>.tar | split -b 3G - <NEW_IMAGE_TAR_NAME>.tar.gz`。这里将文件切割成了数个最大为3G的文件。（注意这里的 `-` 不可忽略）

U盘 -> 服务器：拷贝到服务器后，使用 `cat <NEW_IMAGE_TAR_NAME>.tar.gz.* > <NEW_IMAGE_TAR_NAME>.tar` 合并文件。

服务器端安装

服务器端就很简单了，直接按照下面几部操作即可：

```
# 安装docker
$ sudo dpkg -i docker-ce_20.10.7_3-0_ubuntu-bionic_amd64.deb \
    containerd.io_1.4.6-1_amd64.deb \
```

```
docker-ce-cli_20.10.7-3-0_ubuntu-bionic_amd64.deb
# 安装显卡支持
$ sudo dpkg -i nvidia-container-runtime_3.5.0-1_amd64.deb \
  nvidia-container-toolkit_1.5.1-1_amd64.deb \
  libnvidia-container-tools_1.4.0-1_amd64.deb \
  libnvidia-container1_1.4.0-1_amd64.deb
# 导入镜像
$ sudo docker load --input <NEW_IMAGE_TAR_NAME>.tar
# 查看镜像
$ sudo docker images
# 启动容器
$ sudo docker run --gpus '"device=0,2"' \ # 指定GPU
  -it --ipc host \
  -v /Datasets:/Datasets \
  -v /Code:/workspace/Code \
  -e ENV_VARS="YOUR_ENV_VARS" \
  <YOUR_IMAGE_NAME>:<YOUR_IMAGE_TAG> \
  bash
```

进入容器后便可以开始进行正常的代码训练了。

后台运行容器

为了后台运行容器，这里可以首先将容器中的程序后台运行，或者是不用管，直接运行后，在容器里按快捷键 `ctrl+p+q` 来将容器放入后台。这时查看容器状态可以看到它是 `UP` 。可参考：<https://www.cnblogs.com/davis12/p/14456227.html>

如果再次进入，即通过 `sudo docker exec -it <YOUR_CONTAINED_ID> bash` 进入后可以看到，没有了实时的输出。所以请确保训练脚本会自动将训练过程中的日志写到文件中，以便于后续查看。

那么，这里是否可以将训练过程中的内容再次打印到终端中呢？这时可以考虑终端复用工具 `screen` 或者是 `tmux` 。

结束

一切OK，就让程序慢慢跑着吧！

相关链接

原文档：<https://www.yuque.com/lart/blog/nxomkh>

博客链接：https://blog.csdn.net/p_lart/article/details/119170889

如果觉得有用，就请分享到朋友圈吧！



极市平台

专注计算机视觉前沿资讯和技术干货，官网：www.cvmart.net

624篇原创内容

公众号

▲点击卡片关注极市平台，获取最新CV干货

公众号后台回复“CVPR21分割”获取CVPR2021全景分割论文下载~

极市干货

深度学习环境搭建：如何配置一台深度学习工作站？

实操教程：OpenVINO2021.4+YOLOX目标检测模型测试部署 | 为什么你的显卡利用率总是0%？

算法技巧（trick）：图像分类算法优化技巧 | 21个深度学习调参的实用技巧

极市算法打榜来啦！

 早鸟奖励+分分超越奖励

 免费算力+真实场景数据集

 奖金激励+持续订单分成，实时提现

挑战不断升级、奖金翻倍增加

小试牛刀榜

入门

30+场景数据集

炉火纯青榜

进阶

100+场景数据集

孤独求败榜

高级

200+场景数据集



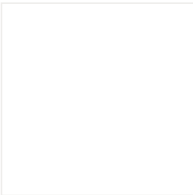


长按扫码了解详情



CV技术社群邀请函

.....



△长按添加极市小助手

添加极市小助手微信 (ID : cvmart4)

备注：姓名-学校/公司-研究方向-城市（如：小极-北大-目标检测-深圳）

即可申请加入极市目标检测/图像分割/工业检测/人脸/医学影像/3D/SLAM/自动驾驶/超分辨率/姿态估计/ReID/GAN/图像增强/OCR/视频理解等技术交流群

每月大咖直播分享、真实项目需求对接、求职内推、算法竞赛、干货资讯汇总、与 10000+来自港科大、北大、清华、中科院、CMU、腾讯、百度等名校名企视觉开发者互动交流~

觉得有用麻烦给个在看啦~

阅读原文

喜欢此内容的人还喜欢

15个目标检测开源数据集汇总
极市平台