

AIPowerMeter implementation in a Docker container

Green AI

Green AI UPPA & Prof En Poche

11 février 2022



What is Docker ?

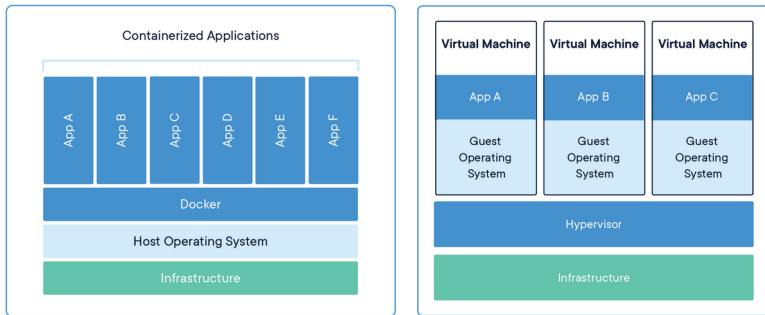


Figure: Difference between docker containers and VM

Docker is a tool used throughout the development lifecycle for fast, easy and portable application development. Unlike a VM (for Virtual Machines), multiple containers can share a same OS kernel each running as isolated processes.

Why using Docker with AI ?

We can use Docker containers for (at least) two main ideas for ML/AI project :

- To test a solution in a isolated environment and make sure we don't override an existing library or tool installed directly on the machine (the same advantage as a python virtual environment, but we push the idea further by adding an initial script (the Dockerfile) and encapsulate the whole experience)
- To make sure the solution is deployable and is working outside our own environment (here we can stick our solution only to the key dependencies and then share it to anyone more easily)

Example with the Speech Recognition system DeepSpeech

A Dockerfile is available directly from the source of DeepSpeech and help anyone to make his own training without much knowing anything or getting trouble for installing dependencies on his machine

Problem with naive AIPowerMeter implementation

Our goal is to implement a solution of AIPowerMeter¹ to track the consumption of electricity of the CPU/GPU but if we add naively the code for tracking the consumption inside a docker we get this error when running a container :

```
(tmp-docker-env) profenpoch@ADAIA:/media/profenpoch/data/testNicolas/AIPM_docker$ sudo docker run --gpus all -it --rm --mount type=bind,src=/media/profenpoch/data/testNicolas/docker_repo,dst=/mnt_aipm test 0:latest
Downloading: "https://download.pytorch.org/models/alexnet-owt-7be5be79.pth" to /root/.cache/torch/hub/checkpoints/alexnet-owt-7be5be79.pth
100.0%
Using cuda device
Experience 1 / 10 is running
RAPL not available: cannot find rapl directory in /sys/class/powercap/intel-rapl/
GPU power will be measured with nvidia
we'll take the measure of the following pids [1]
Process Process-1:
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/multiprocessing/process.py", line 297, in _bootstrap
    self.run()
  File "/usr/local/lib/python3.7/multiprocessing/process.py", line 99, in run
    self._target(*self._args, **self._kwargs)
  File "/AIPM/deep_learning_power_measure/power_measure/experiment.py", line 108, in process_func
    raise e
  File "/AIPM/deep_learning_power_measure/power_measure/experiment.py", line 102, in process_func
    ret = func(self, queue, *args, **kwargs)
  File "/AIPM/deep_learning_power_measure/power_measure/experiment.py", line 276, in measure_yourself
    self.measure(queue, pid_list, period-period)
  File "/AIPM/deep_learning_power_measure/power_measure/experiment.py", line 295, in measure
    metrics.gpu = gpu_power.get_nvidia_gpu_power(pid_list)
  File "/AIPM/deep_learning_power_measure/power_measure/gpu_power.py", line 174, in get_nvidia_gpu_power
    per_gpu_per_pid_utilization_absolute = get_gpu_use_pmon(nsample-nsample)
  File "/AIPM/deep_learning_power_measure/power_measure/gpu_power.py", line 72, in get_gpu_use_pmon
    raise ValueError("Problem with output in nvidia-smi pmon -c 10")
ValueError: Problem with output in nvidia-smi pmon -c 10
```

Figure: Difference between docker containers and VM

¹<https://github.com/GreenAI-Uppa/AIPowerMeter>

Fixing the problem

This error is due to a bad output of the command

```
nvidia-smi pmon -c 10
```

which is unable to output the consumption per pid. We decide then to artificial add a line in a branch of AIPowerMeter called `docker_implementation`, but because of that, we can only measure the whole consumption of the GPU instead of the consumption of each training. It means that you have to make sure you're the only one using the GPU if you want to get more relevant measures.

Results

Our solution can be found in the repository `AIPM_docker`². It contains an implementation of alexnet in torch and the Dockerfile to launch it in a container. You can also find all the documentation for a personal use, a notebook with the consumption of a model training and lastly the python environment with all libraries necessary for the notebook.

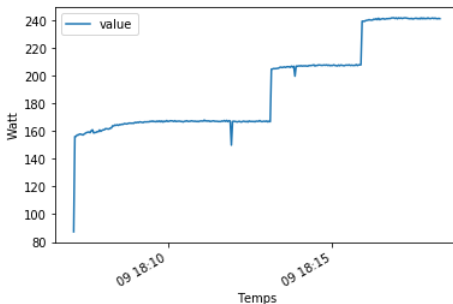


Figure: Consumption over time for three different image sizes

²https://github.com/NicoR2T/AIPM_docker

Conclusion and going further

With this work, you can give it a try or take some part for your personal model trainings and you'll be able to track the power consumption afterwards. An improvement will be to be able of multiple training tracking at the same time.

Feel free to contact me for more informations or suggestions :
ntirel@univ-pau.fr