Running containers in Metacentrum
Sitola – Laboratory seminar 6. 10. 2021

Jan Hoidekr hoidekr@cesnet.cz

Running containers in Metacentrum

- Containers with Singularity
- Machine/Deep Learning frameworks, Alphafold2 in Metacentrum
- Tips for running containers

Singularity in Metacentrum

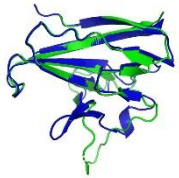
- Singularity
  - Containers in HPC world
    - “Integration is more important then isolation”
  - Runs in user-space
  - Singularity Image Format
  - All frontend/compute nodes in Metacentrum
    - Default binded /storage/* + support GPU, Infiniband
  - builder.metacentrum.cz – create/modify SIF images
    - group ***builders,*** subuid/subgid feature
    - definition files
  - Performance note: loop ***inside*** 1 container vs. loop running containers

First use of singularity

- Run docker image `$ singularity run docker://busybox`
  - In cache – download + build image + run ; 2nd run from cached image
- Build and run SIF image
  - ```
    $ singularity build BB.SIF docker://busybox
    $ singularity shell BB.SIF
    Singularity>  shell inside container, Ctrl-d to exit
    ```
- Modify image in 3 steps – use argument `-f` *OR run as root*
  - ```
    $ singularity build -f -s BB.sbox docker://busybox
    $ singularity shell -f –w BB.sbox   in shell `touch /MY_FILE` and exit
    $ singularity build –f BBmod.SIF BB.sbox
    + check if /MY_FILE file exists $ singularity exec BBmod.SIF ls /
    ```
- Definition files

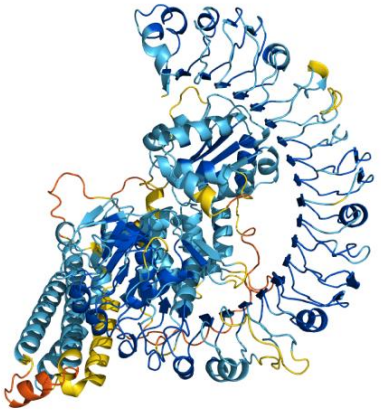
GPU jobs in Metacentrum

- PBS scheduling system
  - Queue – gpu – limits walltime to 24h, longer jobs possible in gpu_long
  - Resources gpu_cap – cuda35 to cuda80
- Hardware
  - more GPU clusters
  - 250 GPU cards – from Tesla K40 (2013) to A100 (2020)
  - 2021 – new cluster 90x GPU



Alphafold2 in Metacentrum

- **Highly accurate protein structure prediction with Alphafold**, see [1]
- Published in July 2021
  - Docker image and 2TB of data
  - *“DNN inference on GPU”*
- Wiki https://wiki.metacentrum.cz/wiki/AlphaFold
  - Prepared scripts to run in Metacentrum on GPU
    - Parameters – input fasta file(s) + output dir
- cca 1h+ jobs
  - CPU+GPU computation
  - Memory 200G+



[1] Jumper, J., Evans, R., Pritzel, A. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596,** 583–589 (2021)

NVIDIA GPU CLOUD

- NVIDIA GPU CLOUD – NGC https://ngc.nvidia.com/

- AI, Machine/Deep Learning containers with GPU support
  - CUDA, libs, … inside images, needs only drivers in Metacentrum
  - TensorFlow, PyTorch + many other tools
  - “Tuned” docker images with documentation
    - **Singularity support**

- SIF images in Metacentrum
  /cvmfs/singularity.metacentrum.cz/NGC/

https://wiki.metacentrum.cz/wiki/NVidia_deep_learning_frameworks

NVIDIA GPU CLOUD in Meta

- Pytorch example - MNIST Word Language Model
  - qsub MNIST-WLM.job

```
#!/bin/bash
#PBS -q gpu
#PBS -l select=1:ncpus=2:ngpus=1:mem=64gb:scratch_local=8gb:gpu_cap=cuda61
#PBS -l walltime=1:00:00
cd $SCRATCHDIR  && wget https://github.com/pytorch/examples/archive/refs/heads/master.zip
unzip -q master.zip &&  cd examples-master/word_language_model/
singularity exec --nv –B $SCRATCHDIR --pwd  $PWD \
    /cvmfs/singularity.metacentrum.cz/NGC/PyTorch\:21.09-py3.SIF  python ./main.py --cuda --epochs 6
clean_scratch
```

- `--nv for GPU`, -B binds $SCRATCHDIR
- SIF image from Metacentrum storage

NVIDIA GPU CLOUD in Meta

- TensorFlow example – modify NGC image
  - `$ singularity run TensorFlow:21.09-tf2-py3.SIF pip list | grep addons`
    `tensorflow-addons 0.13.1`
  - Definition file for building singularity image, latest version of tf-addons

```
Bootstrap: localimage
From: /cvmfs/singularity.metacentrum.cz/NGC/TensorFlow:21.09-tf2-py3.SIF
%post
pip install tensorflow-addons==0.14.0
```

  - `$ singularity build -f TF-addons0.14.SIF TFaddons.def`
  - `$ singularity run TF-addons0.14.SIF pip list | grep addons`
    `tensorflow-addons 0.14.0`

Tips 1: Jupyter and Papermill

- Jupyter notebooks – interactive jobs
  - jupyter.cloud.metacentrum.cz
  - PBS job + singularity + image with Jupyter (e.g. NGC TF, PyTorch)
    - https://wiki.metacentrum.cz/wiki/NVidia_deep_learning_frameworks
    - web access to compute node
- Papermill
  - *tool for parameterizing, executing, and analyzing Jupyter Notebooks*
  - saved input.ipynb -> set parameters ->
    -> run PBS job -> output.ipynb

```
import papermill as pm
pm.execute_notebook(
  'path/to/input.ipynb',
  'path/to/output.ipynb',
  parameters = dict(alpha=0.6, ratio=0.1)
)
```

Tips 2: repeatability, reproducibility

- Singularity image – saved READ-ONLY workspace
  - File vs. directory with virtual environment
    - Easy to transfer, and share
  - stable environment

- Use PYTHONUSERBASE
  - add python modules for development, then modify image
  - best way: for each image different directory

Tips 3: repeatability, reproducibility

- Depreacted or changed functions in frameworks
  - live development -> version skip -> lost of warning on deprecated/changed functions
- New colleagues onboarding
  - same environment
  - time saving
- ML Reproducibility challenge https://paperswithcode.com/rc2021
  - Task: „replicate the main claim described in papers“

Transformers benchmark – PyTorch, TensorFlow

PyTorch

```
pip install transformers py3nvml


from transformers import PyTorchBenchmark,
PyTorchBenchmarkArguments

args = PyTorchBenchmarkArguments(
    models=["bert-base-uncased"],
    batch_sizes=[8, 16, 32],
    sequence_lengths=[8, 32, 128, 512]
    training=True,
    verbose=True,
    env_print=True,
)

benchmark = PyTorchBenchmark(args)
results = benchmark.run()

print(results)
```

TensorFlow

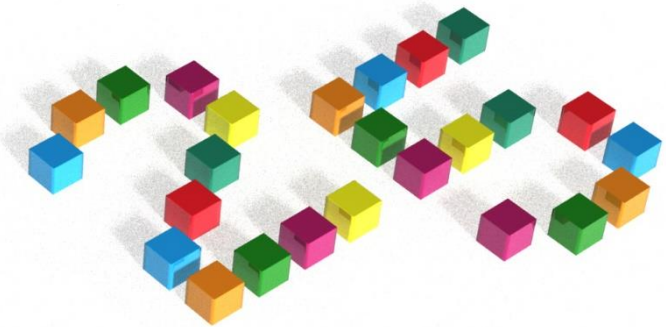
```
pip install transformers py3nvml


from transformers import TensorFlowBenchmark,
TensorFlowBenchmarkArguments

args = TensorFlowBenchmarkArguments(
    models=["bert-base-uncased"],
    batch_sizes=[8, 16, 32],
    sequence_lengths=[8, 32, 128, 512]
    training=True,
    verbose=True,
    env_print=True,
)

benchmark = TensorFlowBenchmark(args)
results = benchmark.run()

print(results)
```



Thanks for your attention!

Jan Hoidekr, hoidekr@cesnet.cz