CC-IN2P3 GPU usage in CPPM

28.11.2024
CAF meeting

Minori Fujimoto (CPPM)

CENTRE DE PHYSIQUE DES PARTICULES DE MARSEILLE
CPPM

GPU usage at CPPM

- Using GPUs for machine learning model training for the flavour tagging in ATLAS;

  - Running training for developments for the b-tagging for Run4 with HL-LHC samples
  - Running training for developing the tagger for the Boosted Higgs -> $\tau\tau$

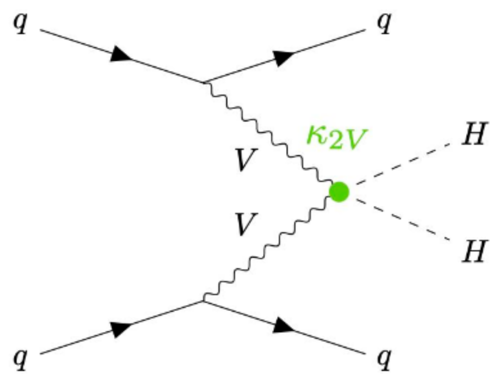
- Multi-GPUs are indispensable for the development of the b-jet tagger for Run4, which uses ~115M jets.

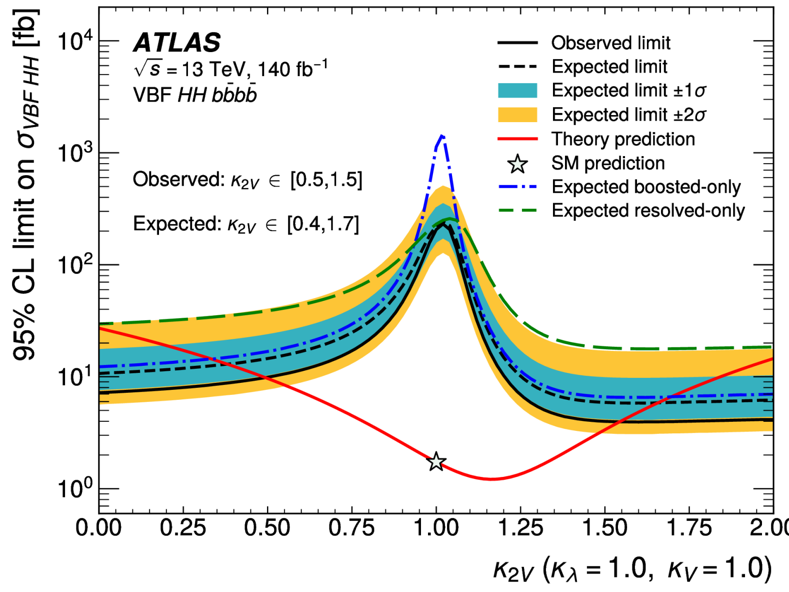
  Today going to report about the jobs on Boosted Higgs -> $\tau\tau$ tagger,
  which uses ~ 20M jets, multi-GPUs are favorable but can also be tested with single-GPU.

Boosted Higgs -> $\tau\tau$ tagger

- Boosted Higgs->$\tau\tau$ tagging can play an important role in

  - High mass resonant searches
  - Exclusion of the BSM coupling for non-resonant analysis
  (ex. κ_{2V} in VBF **HH** analysis)

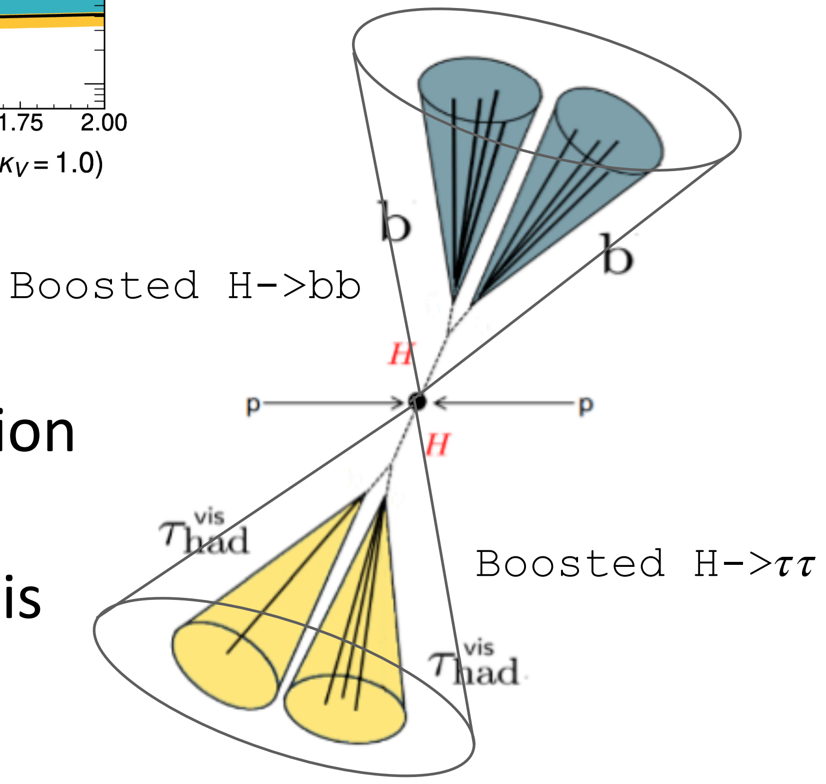




Phys. Lett. B 858 (2024) 139007

- Aiming the use in the boosted HH->bb$\tau\tau$ analysis

- Higgs->$\tau\tau$ tagger is being developed as a natural extension of the existing GNN-based Boosted Higgs->bb tagger (GN2X tagger), expecting easier integration into analysis frameworks



Salt framework

- Using salt framework for the trainings

  - Model-training framework, developed for the flavour tagging algorithm

  - Salt framework is easy to use with prebuilt docker images, via Singularity
    It can be used with a connection to CVMFS

  - Alternative way to use salt framework is use it in a conda environment

How using GPU at CCIN2P3

- GPU batch jobs can be submitted via Singularity container, with sbatch command

```
$sbatch -t 0-10:00 -n 1 --gres=gpu:v100:1 --mem 8G launchSingularity.sh
```

`lauchSingularity.sh`

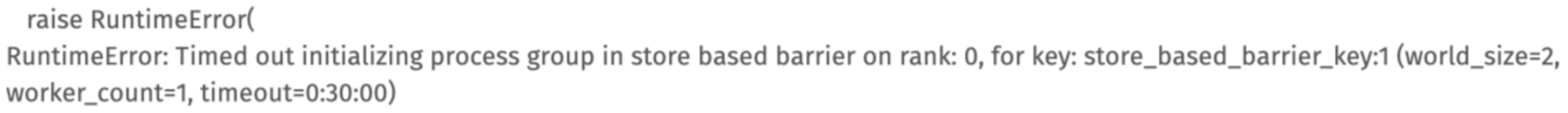
```
#!/bin/bash

# Job name
#SBATCH --job-name=salt

# Job log
#SBATCH --output=serial_test_%j.log

# choose the GPU queue
#SBATCH -p GPU

# requesting one node
#SBATCH --nodes=1

# Only if you really need it!
# #SBATCH --exclusive

# keep environment variables
#SBATCH --export=ALL

# requesting 4 V100 GPU
# (remove the "v100:" if you don't care what GPU)
#SBATCH --gres=gpu:v100:1

# note! this needs to match --trainer.devices!
#SBATCH --ntasks-per-node=1

# number of cpus per task
# don't use this if you have exclusive access to the node
# #SBATCH --cpus-per-task=10

# request enough memory
#SBATCH --mem=8G

# Timelimit
#SBATCH -t 0-10:00

# Change log names; %j gives job id, %x gives job name
#SBATCH --output=/pbs/home/m/mfujimot/HHbbtautau/salt/salt/submit/logs/slurm-%j.%x.out
# optional separate error output file
#SBATCH --error=/pbs/home/m/mfujimot/HHbbtautau/salt/salt/submit/logs/slurm-%j.%x.err

# launch singularity
singularity exec --nv --bind $PWD,/pbs,/cvmfs,/sps \
    /cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/atlas-flavor-tagging-tools/algorithms/salt:0-4 ${PWD}/launch_job.sh
```

`launch_job.sh`

```
#!/bin/bash
#echo ${CUDA_VISIBLE_DEVICES}

# print host info
echo "Hostname: $(hostname)"
#echo "CPU count: $(cat /proc/cpuinfo | awk '/^processor/{print $3}' | tail -1)"

# print gpu info
nvidia-smi

# run the training
cd /pbs/home/m/mfujimot/HHbbtautau/salt/salt
echo "Moved dir, now in: ${PWD}"
echo "Running training script..."
salt fit --help
salt fit --config configs/GN2X.yaml --trainer.devices=1 --trainer.accelerator gpu --force
```

GPU running status at CCIN2P3

- **Works perfectly when requesting single GPU**
- In many cases, need to wait ~1 day before the job starts running due to the batch queue.
- When requesting multi-GPUs (`gres=gpu:v100:2`), it does not run with the error:



```
  raise RuntimeError(
RuntimeError: Timed out initializing process group in store based barrier on rank: 0, for key: store_based_barrier_key:1 (world_size=2,
worker_count=1, timeout=0:30:00)
```

- Lots of helps from the CCIN2P3 HelpDesk : tickets

  This error is still there even after the updates of the Slurm cluster in 06.26.24

  (versions of drivers/CUDA at CCIN2P3, Tesla V100-PCIE-32GB)

```
Nvidia drivers version: 550.54.15
Cuda version: 12.4.1
```

GPUs at CPPM server

- Same salt setup (with the same version of the salt) ran successfully in the CPPM local server with multi-GPUs, with Singularity but without slurm batch

```
$singularity run --nv --bind $PWD,/share/users/fujimoto/
/cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/atlas-flavor-tagging-tools/
algorithms/salt:0-4 salt fit
--config /atlas/fujimoto/BoostedDiTau/salt_latest/salt/salt/configs/GN2X.yaml
--trainer.devices=3
--force
```

(versions of drivers/CUDA at CPPM, 3 GPUs: NVIDIA GeForce RTX 2080)

```
Nvidia drivers version: 545.23.08
Cuda version: 12.3
```

Now later version is used in CCIN2P3.
-> It seems the problem is not because of the versions of the drivers/CUDA.

Additional option in `launch_job.sh` or some parameter tuning in `launchSingularity.sh` might be needed

GPUs at Paris-Saclay

- Salt framework with multi-GPUs also worked at Paris-Saclay mesocenter by Inès Combes (IJCLab)

- Used it in a conda environment (anaconda 3: anaconda3/2022.10/gcc-11.2.0) via slurm batch

```
#!/bin/bash
#SBATCH --job-name=training_GN2X_tau
#SBATCH --output=training_%j.log
#SBATCH --partition=gpua100
#SBATCH --time=24:00:00
#SBATCH --mem=50G
#SBATCH -n 1
#SBATCH --gres=gpu:4
#SBATCH --gpus-per-task=4
#SBATCH --error=errorJob_%j.txt
#SBATCH --export=ALL

# Module load
module purge
module load anaconda3/2022.10/gcc-11.2.0
module load cuda/10.2.89/intel-19.0.3.199


echo "Hostname: $(hostname)"
echo "CPU count: $(cat /proc/cpuinfo | awk '/^processor/{print $3}' | tail -1)"

# Activate anaconda environment code
source activate /gpfs/users/combesi/salt/conda/envs/salt

echo ${CUDA_VISIBLE_DEVICES}

cd /gpfs/users/combesi/salt/salt/
salt fit --config configs/GN2X_tau.yaml --force
```

(versions of drivers/CUDA at Paris-Saclay mesocenter, 4 GPUs: Nvidia A100 GPUs)

```
Nvidia drivers version: 535.104.05
Cuda version: 10.2.89
```

- The conda environment needs to be installed to try this approach on CCIN2P3

Summary

- Doing the flavour tagger training with GPUs with salt framework.
  Multi-GPU usage is necessary.

- At CCIN2P3, it works perfectly when requesting single GPU.
  Many thanks to the CCIN2P3 Helpdesk for lots of helps!
  Multi-GPU functionality appears to have unresolved challenges.

- Multi-GPU usage is supported on some French institute resources,
  and the salt framework operates seamlessly, suggesting the framework itself is not the source of any issues.

Backup

Using multi-GPU with batch

- GPU batch jobs can be submitted via Singularity container, with sbatch command

```
$sbatch -t 0-10:00 -n 1 --gres=gpu:v100:2 --mem 8G launchSingularity.sh
```

lauchSingularity.sh

```
#!/bin/bash

# Job name
#SBATCH --job-name=salt

# Job log
#SBATCH --output=serial_test_%j.log

# choose the GPU queue
#SBATCH -p GPU

# requesting one node
#SBATCH --nodes=1

# Only if you really need it!
# #SBATCH --exclusive

# keep environment variables
#SBATCH --export=ALL

# requesting 4 V100 GPU
# (remove the "v100:" if you don't care what GPU)
#SBATCH --gres=gpu:v100:2

# note! this needs to match --trainer.devices!
#SBATCH --ntasks-per-node= 2

# number of cpus per task
# don't use this if you have exclusive access to the node
# #SBATCH --cpus-per-task=10

# request enough memory
#SBATCH --mem=8G

# Timelimit
#SBATCH -t 0-10:00

# Change log names; %j gives job id, %x gives job name
#SBATCH --output=/pbs/home/m/mfujimot/HHbbtautau/salt/salt/submit/logs/slurm-%j.%x.out
# optional separate error output file
#SBATCH --error=/pbs/home/m/mfujimot/HHbbtautau/salt/salt/submit/logs/slurm-%j.%x.err

# launch singularity
singularity exec --nv --bind $PWD,/pbs,/cvmfs,/sps \
    /cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/atlas-flavor-tagging-tools/algorithms/salt:0-4 ${PWD}/launch_job.sh
```

launch_job.sh

```
#!/bin/bash
#echo ${CUDA_VISIBLE_DEVICES}

# print host info
echo "Hostname: $(hostname)"
#echo "CPU count: $(cat /proc/cpuinfo | awk '/^processor/{print $3}' | tail -1)"

# print gpu info
nvidia-smi

# run the training
cd /pbs/home/m/mfujimot/HHbbtautau/salt/salt
echo "Moved dir, now in: ${PWD}"
echo "Running training script..."
salt fit --help
salt fit --config configs/GN2X.yaml --trainer.devices=2 --trainer.accelerator gpu --force
```