GPU computing in ABP

CWG

ABP survey on GPU usage

- Short description of your computations using GPUs (a few lines are sufficient);
- Estimate of your needs for GPUs in the short and medium term (a coarse estimate is sufficient at this stage);
- Required type of access (job submission vs direct login);
- Special requirements in terms of operative system and software;
- Need for other services to be available (storage, access to databases, etc.)
- Interest in GPUs on the HPC cluster (to use GPUs within multi-node MPI simulations)
- Any other constraints for your application.

From Xavier

1GPU - V100
A100 40 or 80 GB, 1.3 faster than V100

PIC simulations of space-charge and wakefields at the SPS.

- ~5E4 h / year (short and medium) [6GPU year full time! (4GPU ABP + Ht condor) done for year and to be continued]
- Job submission is sufficient [perfect!]
- Linux. sixtracklib / xsuite and PyHEADTAIL
- No need for special infrastructure
- No need for multi-GPU

Theoretical investigations of emittance growth in the presence of noise and collective forces

- ~1E4 h/ year (only short term) [1 GPU year full time, half-way]
- Job submission is sufficient
- Linux. Custom code based on cupy
- No need for special infrastructure
- No need for multi-GPU
- Large memory needed (>30Gb GPU RAM, CPU 2 GB)

Self-consistent beam-beam simulations for the FCC-ee.

- >1E5 h / year (medium term) 12 GPU year full time [to be done]
- Job submission is sufficient
- Linux. xsuite
- No need for special infrastructure
- Multi-GPU could be useful, but single nodes with few GPUs are probably enough. (1 bunch for GPU, HT condor enough)

From Frederik (1)

- Short description of your computations using GPUs (a few lines are sufficient)

1) parallelised single-particle tracking with xsuite:
- new methods for DA calculation (Frederik): DA evolution, 4D calculation, longitudinal behaviour
- ML model for the FCC (Davide): loads of loads of long-turn tracking needed
- beam dynamics studies with hollow electron lenses (Pascal)
- simulating diffusion in realistic lattices (Carlo Emilio)

2) gpu-accelerated training of ML models:
- multi-parameter model of the HL-LHC and FCC (Davide)
- high precision DA border detection for large amounts of initial particles (Frederik)

3) generic optimisation of mathematical calculations:
- parallelised tracking of polynomial maps (Carlo Emilio): custom CUDA C++ code wrapped in Python
- parallelisation of specific post-processing steps (Carlo Emilio): FFTs for tune calculation, algebraic operations on large amounts of data

From Frederik (2)

- Estimate of your needs for GPUs in the short and medium term (a coarse estimate is sufficient at this stage)

Pascal: 2-3 GPU-days per week on average for the HEL studies (including his students that work on the topic),
Carlo Emilio: a full V100 for about 0.5-1 month continuous computing time (accumulated since he started using the GPUs). Done, [1 GPU full time, expected to doubling next year]

Davide: 3 sequential DL networks ideally with multi-GPUs. 2 Tesla V100 for 1 month for the first training only.

Tracking needs have constant load, while optimisation results in fluctuating needs: Ideally a large-volume queue for continuous tracking and a dedicated queue for on-the-spot calculations.

Finally, it would be extremely useful to always have a few GPUs available for agile development (and testing and benchmarking).

Summary: ideally 8 V100-type GPUs minimally for the NDC section in perspective

- Required type of access (job submission vs direct login)

Job submission is fine as long as the level of control (driver issues?) is fine enough (obviously to be able to run xtrack, but also to be able to use gpu-accelerated ML training etc).

From Frederik (3)

- Special requirements in terms of operative system and software;?

For tracking nothing special: system compatible with xtrack and docker

For generic optimisation codes, it is important to have a typical software stack that is extendable (CVMFS is always a good starting point).

For gpu-accelerated ML training, it is important to have up-to-date CUDA drivers and the ability to install dedicated software that can fully talk to the hardware (docker should be fine as long as it can access the installed drivers) like RAPIDS (cuML) and Horovod-UBER (or tools like h2o4gpu if a license would be available).

- Need for other services to be available (storage, access to databases, etc.)?

For ML studies a local file storage with high-performance I/O (fast SSD) is crucial, furthermore we need 6-10 CPU cores per GPU to feed the latter, and a way to transfer data easily is needed (AFS/EOS). Access to CVMFS is very useful.

- Interest in GPUs on the HPC cluster (to use GPUs within multi-node MPI simulations)?

Not needed in our current computation environment.

From Frederik (4)

- Any other constraints for your application.

There are several issues with the current GPU environment at CERN:

- lack of straightforward documentation of the file systems (AFS/EOS) and the bottlenecks they might (and will) impose proper usage of AFS/EOS requires (non-trivial) extra implementations
  - Carlio Emilio: detailed list of known issues for AFS/EOS
- no alternatives to AFS/EOS:
  - a local file storage could solve a lot of the bottlenecks (to be verified why /scratch or /tmp is no needed

- different HTCondor nodes sometimes have different CUDA runtimes installed (Carlo Emilio)
- no solutions (so far) for developing code within docker containers (Carlo Emilio)
- the queue system for gpu is not clear: not clear when GPUs are available, and also the fair-share system is not clear (and maybe not so fair-share atm) (Pascal, Carlo Emilio, Kostas suggested condor_status)

Guido

4 GPU year /user 20 GPU year

Kostas

GPU exploitation in beam dynamics simulations

- Tracking single particle, incoherent multiparticle, coherent multiparticle; Machine learning for post processing and machine optimization.
- Usage: beam-beam, lifetime, space charge, collimation, e-cloud

Software stack: Custom kernel code written in C/C++ written by 1 to 10 developers, leveraging Python packages such as cupy, pyopencl. Also high level libraries such as RAPIDS, cuML and Horovod-UBER will likely to be used.

Present hardware resources

Development and small studies for double-precision:

- Workstation: 4x Nvidia TitanV (V100, 12 GB)
- Workstation: 1X Nvidia TitanV, 1x AMD Radeon VII, 2x GTX1050
- Bologna cluster: 4x V100 (16 GB)

Large scale studies:

- HTCondor: ~20 GPUs on average (V100 because of double precision)

We are in contact with IT to get the amount of GPU hours used in the last year

Outlook for the future development service

Workstations not suitable when number of developer increases as developers do not develop full time, moving to central services seems more efficient.

- Developers needs:
  - machines immediately available for extensive number of hours
  - well behaving file system ~300GB surviving between sessions
  - software stack configurable and operating system (e.g. nvidia toolkit version)
  - OS: linux. CentOS is fine, but for some applications, ubuntu/Debian could be beneficial.
  - Access to NXCals
- Comments on options:
  - HTCondor not suitable due to queuing time and session timeout
  - Private virtual machines ideal solution if not too costly.
  - Lxplus-like could also work provide
    - session duration for several days (but not weeks) if active
    - session could be killed if inactivity for >4 hours
    - file system should allow git, makefile, compilers to work quickly and flawlessly similar to local storage

Outlook for the future HTCondor

- Studies expected to increase load as more software is being written for GPUs with double precision most of time.
- Still expecting time varying demands.
- Peaks of ~100 GPU (V100 equivalent GPU ~7.5TFlops double precision not tensor core) can be reached. Single precision and tensor cores can be needed for ML studies.
- Most of the single particle jobs can run using 1/3 V100 equivalent.
- Few exception require large memory (80GB) and many GPU cores (5000k cores).
- Combination with MPI (HPC) not expected, at the least in the near future.
- Requirements on the host are typically small (1 core/2GB ram) but few exception could require (16 core/16GB ram), besides ML studies for which we need CPU and fast IO.
- Fast file system for I/O and persistent shared file system for software stack neded