2 related topics : benchmarking and pledge value assignment, physics validation

# Benchmarking

A new CPU benchmarking suite is available and ready to replace HS06 for procurements and pledges

- Based on different categories of workloads from all LHC experiments
- Able to target multiple architectures

This new benchmarking suite is a first and important step to help characterize resources during procurements but is not enough to assign a value to pledges. This value must reflect the “usable capacity”, not only the potential capacity, and take into account more than the CPU. For computing the value, we need to take into account only the workloads appropriate to the non-general purpose resources. The reasoning is that an experiment has a mix of workflow to run and will use the most appropriate resources in its pledges to run each workflow, maximising the usable capacity.

Open questions:

- How to deal between these somewhat different approaches between benchmarking for procurements and pledges based on the usable capacity? It is something new: until now pledges were based on the benchmark execution on a given resource.
- Test the proposed approach for pledging usable capacity on a guinea-pig HPC site using the tools we have currently and learn from it
- How to deal with the possibly evolving “usable capacity” for a given resource? It is particularly true for machines with GPUs. The changes brought in the SW layers by new versions of drivers, CUDA and other portability layers, can increase by an order of magnitude the usable capacity in some cases…
- Future benchmark: work needed to agree on the one number (score) that will reflect the potential capacity of a resource. Do we have one and only one number or do different experiments use a different mix? With what consequences on sites supporting multiple experiments? Which rules for evolving the workloads that are part of the benchmark?
- Is it feasible to have a (composite?) metric that can also account for power efficiency and climate sustainability in addition to computing power?

# Physics Validation

It was the first attempt to have a dedicated session on this aspect with 2 introduction talks: experience with physics validation in ATLAS, a concrete attempt to validate a code on various architectures with CMS Patatrak.

From ATLAS, we learned that physics validation is a well-known process and most of the issues are identified. Only a small part of the work can be and is automatized. A specific source of complication with validation is the non reproducible numbers, in particular those from random seeds in simulation. Maybe some improvements in this area could be discussed (in particular for Geant4, it may be technically feasible to achieve a better reproducibility).

CMS implemented a physics validation step implemented in the development lifecycle of their Patatrack framework (R&D framework for high granularity reconstruction). Patratrack supports both CPU and GPU and the validation is run in both environments. This validation runs as part of the GitLab CI and final plots showing the effects of a PR are attached to it, allowing further tracing. The physics validation is rather expensive (a few hours) and thus is not run on every PR: someone has to decide to run it or not. In addition to physics output, the runtime performance is also checked to avoid any significant regression. An experiment-agnostic infrastructure of this kind doesn't seem feasible.

Open questions:

- How much we can increase automation to reduce the manpower required by physics validation? It should be possible to find the computing resources for it in the grid…
- Geant4: possibility to add a RND state to a track to improve number reproducibility and ease physics comparison by eliminating/reducing a source of differences.