Summary of the cross-experiment HPC workshop

Prepared by the ATLAS, CMS and LHCb computing coordinators: Tommaso Boccali, Concezio Bozzi, James Catmore, Davide Costanzo, Markus Klute. With contributions from Andrea Valassi

We had a workshop on HPCs on 10 May across the experiments, where we had people from ATLAS, CMS, and LHCb, with some participation from ALICE.

1. **Overview.** All experiments report usage of HPC resources, with varying levels of technical difficulties. Integrating HEP experiment workloads on HPC systems poses technical issues and challenges in two distinct areas, namely the management and submission of jobs on HPCs, and the development of the software applications executed within each job:
    a. Distributing computing and data management issues (how to schedule jobs on HPCs, how to connect them with experiments central services, how to access experiment software and calibrations, how to access input data, how to dispatch outputs…): the challenge is that HPC systems often lack external connectivity and do not allow the easy installation of new services that require sysadmin privileges.
    b. Software issues (how to develop applications that utilize efficiently the local hardware architecture): the challenge is that HEP software has so far almost exclusively been developed for traditional x86 CPUs, while HPC systems provide their computing capacity using a variety of technologies including many-core KNL processors, ARM processors, PowerPC processors and, increasingly, GPUs and other accelerators.

2. **Distributed computing issues.** Distributed computing issues may be more or less difficult to overcome depending on the specific technical choices and policies of each HPC center (networking, access, availability of virtualization, possibility to mount specific kernel modules…), but in principle it is possible to solve them with a time-limited integration effort. Some HPCs present themselves as Grid resources (CVMFS, external network connections…) and have been used as such, with no major technical complications.
   The difference in exploiting successfully (or not) some of the "complicated" HPCs is often the expertise provided by the HPC centers themselves, helping and supporting the experiments in integrating their workflows. *FAs should understand this important aspect:* to be able to use effectively these resources, experiments will need concrete manpower help from the HPC center themselves, to help interfacing to the experiments.
   To minimize the duplication of work, it would also make sense to set up a joint team of experts from all experiments, which will share the technical expertise on the various HPCs.

3. **Software issues.** Software issues, conversely, are generally much more difficult to solve and may constitute a blocker for the exploitation in the short-term of some HPC centers (especially those providing a large fraction of their computing power through GPUs), as they require a long-term program of software modernization and reengineering which

may take several years. *FAs should understand this important aspect:* to be able to use effectively these resources, experiments will need a large investment in software in order to bring about a veritable paradigm shift, by retraining the existing personnel as well as hiring/training new experts to reengineer and port/adapt their software applications.
The extent of the problem is, again, different in different HPC centers:

a. Traditional x86 processors at HPC centers can be and have already been readily exploited by all experiments, without the need for specific software work. One example is the Piz Daint supercomputer in Lugano, which is used as a WLCG Tier2 by all of ATLAS, CMS and LHCb.
b. Many-core KNL processors at HPC centers can execute HEP x86 applications without a software port, but they provide a much lower memory per core and they can only be used efficiently by multi-process (MP) or multi-threaded (MT) applications. All experiments have been actively working in the last few years on implementing MP or MT workflows, which have allowed the use of these resources. One example is the CORI Phase 2 supercomputer at NERSC, which CMS has been able to use through its multi-threaded software framework.
c. ARM or PowerPC processors at HPC centers can only execute HEP software applications which have been ported to these architectures. This port has been completed by some experiments for some of their workflows, but it is still ongoing in other cases. Also, physics validation of software stacks on different architectures is a complex procedure, which should not be underestimated. One example is the MIRA supercomputer at Argonne, where ATLAS successfully tested the use of the ALPGEN event generator on Power8 processors. However, no physics-validated productions on ARM or PowerPC have been run yet by ATLAS, CMS or LHCb at any HPC centers.
d. GPUs at HPC centers cannot presently be used efficiently by HEP experiments, except for some workflows (such as ML training) which represent a minimal part of their worldwide-integrated computing needs, as the majority of their software applications have not yet been ported to this architecture. GPUs are presently one of the main challenges for the exploitation of HPC centers by the HEP experiments. The issue is expected to become more important in the future, as the current trend is for new HPC systems to provide an increasingly larger fraction of their computing power through GPUs.

4. **HEP software workflows and HPCs.** Traditional x86 CPU resources on HPCs are used today. Some of these are even already included in our pledges, while others are accounted for as opportunistic (non-pledged) resources. We have monitoring and accounting plots showing their usage and we expect this usage to continue and hopefully expand in the years to come. Full detector simulation (Geant4) represents a significant fraction of worldwide-integrated WLCG computing resources for ATLAS (40%) and even more for LHCb (85%, and expected to grow for Run3): as a consequence, these two experiments have used and plan to continue using HPCs only for this workflow, at least for the next few years. CMS, conversely, feels that HPCs will need to be used as a major part of their infrastructure and should be validated for all workflows, whenever possible, not just simulation: this is because current extrapolations to Phase-II show future lower relative

needs for simulation with respect to reconstruction due to the scaling with PU, absent for simulation and very steep for reconstruction.

5. **HEP software workflows and GPUs.** All HEP experiments have been actively working in the last few years on the reengineering of their software workflows and also on their porting to heterogeneous hardware environments including at least GPUs. This is an important activity which the experiments will continue in the future to be able to more efficiently exploit the available computing resources, whether at WLCG sites, at HPC centers or elsewhere. The status of software development on GPUs and the outlook for the future exploitation of HPC systems including GPUs is different for different software components and workflows:

    a. Experiment software frameworks are evolving to allow offloading some algorithms to GPUs and other accelerators. LHCb has considered adapting its core software framework to a heterogeneous environment including GPUs, but there are currently no plans in this direction as this is a complex task with uncertain results in terms of software efficiency. CMS has deployed in the last software release a heterogeneous framework where different libraries (targeting different architectures) can be chosen for a given part of the workflow; the choice can be at submission time via configuration, or as late as at the start of the job, which can auto-discover the local available accelerators. The software framework orchestrating workflows, e.g. through scheduling of calculations, runs on the CPU, while the implementation of selected number crunching algorithms depends on some specific software technologies such as NVidia CUDA. In order to cope with such couplings, solutions that allow to abstract from the underlying runtime libraries, such as Kokkos or Alpaka, are also being considered. In any case, while prototypes are available, production usage of this new code is only expected to start around Run3, i.e. in a few years from now, with increased usage towards Run4.
    b. Full detector simulation (Geant4), as mentioned, is one of the largest consumers of WLCG computing resources for LHCb (85%), ATLAS (40%) and CMS (25%). In spite of past and ongoing R&D efforts on its vectorization and on its port to GPUs, there are no clear indications at the moment that this workflow can be executed on GPU-based HPCs in the short term. Fast detector simulation using parametrized calorimeter responses is an area where R&D efforts on GPUs are also ongoing, for instance using ML algorithms, but similarly there are no indications that this can reach production quality to be executed on GPU-based HPCs in the short term.
    c. Event reconstruction is an area where the R&D for the exploitation of GPUs has been active in all HEP experiments. This is important as event reconstruction is also a large consumer of WLCG computing resources, at least for ATLAS and CMS. In particular, CMS is planning to port to GPUs a sizeable fraction of its HLT algorithms for Run3, in order to run commissioning tests for a Run4 transition, but no algorithms for event reconstruction are in production yet. There

are thus no plans to deploy event reconstruction workflows in production on GPU-based HPCs in the short term.

d. Event generation is currently a significant consumer of WLCG computing resources for ATLAS (10%), more than in CMS, and it is expected that the fraction of time spent in event generation will grow in both experiments with the increased need for theoretical precision at HL-LHC. Porting these workflows to GPUs would therefore be very useful, and we welcome the efforts from the theory community in this direction. Some R&D on GPUs was already done in the past for MadGraph, but this activity has now only just been revived (for this and other generators) and it is too early to predict if and when it will reach production quality to be executed on GPU-based HPCs. The fraction of time spent in generators by LHCb is instead small with respect to simulation and it is expected that it will stay so, as most of the generation is performed with PYTHIA and higher order corrections are needed only in a minor part of the LHCb physics program.
e. GPUs can be used already today for Machine Learning (ML) training and hyperparameter optimisation, and also to run software applications that execute maximum likelihood fits (such as GooFit, and possibly zfit in the future). ATLAS has demonstrated the possibility of submitting jobs with a GPU payload to sites where GPUs are available. However, the fraction of worldwide-integrated computing resources that ML training or fits may represent (mainly at Tier3s) is minimal, with respect to that required by the other workflows mentioned above.

6. **Benchmarking and accounting on HPCs and GPUs.** A monitoring mechanism to track HPCs and GPU usage by the experiments should also be put in place to better understand usage patterns. We welcome the WLCG decision to task the Benchmarking group for the definition of criteria on which evaluate the usability and contribution to experiment workflows for non-traditional resources such as GPUs and HPCs.

7. **HPCs, GPUs and pledges.** It is clear that, in an ideal world with no limitations on computing resources, the experiments would be happy to continue using the Run 1-2 models, without reverting to new and potentially disrupting technologies. Integrating HEP experiment workloads on HPCs poses technical challenges both in the distributed computing and software areas, because these systems use architectures and policies that are sometimes very different from those which have been traditionally used in HEP so far. The effort that the HEP experiments need to invest to efficiently exploit HPCs is to some extent an effort that was needed and ongoing in any case, especially in the modernization and optimization of their software stacks, but there is otherwise no real push from the experiments to adopt these new technologies. Still, we have to be prepared to a situation in which our FAs will ask us to use HPC resources (including GPUs) as a sizeable part of our pledges -- somehow regardless of the probability it will happen (which we anyhow consider medium/high). This would be a significant difference in our usage of HPC centers with respect to today, when HPCs are mostly seen as opportunistic (non-pledged) resources: in particular, it would imply that any computing time offered on

an HPC center (even if in the form of a GPU) would decrease the computing time available on a more traditional x86 CPU resource.

We experiments think that, while on our side we are committed to do our best, we should be facilitated by the FAs in these aspects. The best approach would be to have HEP experts to be part of the definition process of both architectures and policies, having hence the HEP use case as a first-class citizen for HPCs. Sadly, this is not happening today, with HEP entering the game only when HPC machines have already been built and put into production.

HPC centers are not all equivalent to one another, because they exhibit different policies and architectures. It should be made clear that HEP experiments cannot possibly efficiently exploit HPC centers unless these policies and architectures meet some minimal criteria. In particular, HEP experiments are not ready today to receive a sizeable part of their pledges in the form of GPU-based HPCs, because the HEP software workflows which represent the (current, and projected) main consumers of worldwide-integrated computing resources have not yet been ported to GPUs.

As mentioned above, it should also be made clear to the FAs that, in order to efficiently exploit HPC resources, help in two areas is needed: concrete manpower help from the HPC centers themselves, to help interfacing to the experiments and integrating their computing workflows and their job submission and data management practices; a large investment in the HEP experiment software, to retrain the existing personnel and hire new experts who can contribute to the reengineering and porting of their software applications to new architectures.