New developments in cost modeling for the LHC computing

Catherine Biscarat[1], *Tommaso* Boccali[2], *Daniele* Bonacorsi[3], *Concezio* Bozzi[4,5], *Davide* Costanzo[6], *Dirk* Duellmann[4], *Johannes* Elmsheuser[7], *Eric* Fede[8], *José* Flix Molina[9], *Domenico* Giordano[4], *Costin* Grigoras[4], *Jan* Iven[4], *Michel* Jouvin[10], *Yves* Kemp[11], *David* Lange[12], *Helge* Meinhard[4], *Michele* Michelotto[13], *Gareth Douglas* Roy[14], *Andrew* Sansum[15], *Andrea* Sartirana[16], *Markus* Schulz[4], *Andrea* Sciabà[4*], *Oxana* Smirnova[17], *Graeme* Stewart[4], *Andrea* Valassi[4], *Renaud* Vernet[8], *Torre* Wenaus[7], and *Frank* Wuerthwein[18]

[1]Univ. Grenoble Alpes, CNRS, Grenoble INP, LPSC-IN2P3, Grenoble, France
[2]INFN Sezione di Pisa, Pisa, Italy
[3]INFN Sezione di Bologna, Università di Bologna, Bologna, Italy
[4]European Organisation for Nuclear Research (CERN), Geneva, Switzerland
[5]Università e INFN, Ferrara, Ferrara, Italy
[6]Department of Physics and Astronomy, University of Sheffield, Sheffield, United Kingdom
[7]Physics Department, Brookhaven National Laboratory, Upton, NY, USA
[8]Centre de Calcul de l'IN2P3 du CNRS, Lyon, France
[9]Centro de Investigaciones Energéticas Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain
[10]LAL, Université Paris-Sud and CNRS/IN2P3, Orsay, France
[11]Deutsches Elektronen-Synchrotron, Hamburg, Germany
[12]Princeton University, Princeton, NJ, USA
[13]INFN Sezione di Padova, Università di Padova, Padova, Italy
[14]SUPA - School of Physics and Astronomy, University of Glasgow, Glasgow, United Kingdom
[15]STFC Rutherford Appleton Laboratory, Didcot, United Kingdom
[16]Laboratoire Leprince-Ringuet, Ecole Polytechnique, CNRS/IN2P3, Université Paris-Saclay, Palaiseau, France
[17]Lunds Universitet, Fysiska Institutionen, Avdelningen för Experimentell Högenergifysik, Box 118, 221 00 Lund, Sweden
[18]University of California, San Diego, La Jolla, CA, USA

Abstract. The increase in the scale of LHC computing during Run 3 and Run 4 (HL-LHC) will certainly require radical changes to the computing models and the data processing of the LHC experiments. The working group established by WLCG and the HEP Software Foundation to investigate all aspects of the cost of computing and how to optimise them has continued producing results and improving our understanding of this process. In particular, experiments have developed more sophisticated ways to calculate their resource needs, we have a much more detailed process to calculate infrastructure costs. This includes studies on the impact of HPC and GPU based resources on meeting the computing demands. We have also developed and perfected tools to quantitatively study the performance of experiments workloads and we are actively collaborating with other activities related to data access, benchmarking and technology cost evolution. In this contribution we expose our recent developments and results and outline the directions of future work.

*e-mail: Andrea.Sciaba@cern.ch

1 Introduction

The preparation for the LHC Run 3, which will see considerable changes in data collection and processing for ALICE and LHCb, and for Run 4, or HL-LHC, has made significant progress in the last year; many factors have contributed to decrease the estimated gap between the estimated amounts of available and needed processing power and storage. In the previous report [1] we quoted a $O(10)$ discrepancy, while now it is closer to a factor 2-3 [2]. The "revolutionary" changes we mentioned as being required to completely close the gap are progressively being introduced or planned.

In addition, thanks to several refinements, the calculation of the cost of computing has improved, both in terms of resources for what is required for the physics program, and in terms of infrastructure costs.

The System Performance and Cost Modeling Working Group, created in 2017 and comprising around thirty members from experiments, sites, IT and software experts, has continued along the roadmap initially defined and even started some new activities, as in the area of data access efficiency, in close collaboration with the DOMA access group [3] and in contributing to the definition of new benchmarks, together with the HEPiX benchmarking working group [4].

In this contribution we will show some recent developments in the areas of work under the domain of this activity.

2 Software performance

Characterization of software performance is a complex problem, that can be approached from different points of view. While software developers are most concerned with understanding which parts of the code need optimizing, for the computing infrastructure manager it is mainly a question of measuring what is needed to run effectively the experiment workloads. In this case, the application software is to be considered, at a first approximation, as a "black box", and tools like PrMon (which relies on the Linux kernel to extract CPU time, memory, I/O and network metrics for a given process tree) [5] or Trident (which gives access to detailed information on the CPU utilization at the node level using hardware counters) [6] are extremely effective in producing metrics that can be used for infrastructure planning, for benchmarking and for understanding inefficiencies. As an example, Trident was used to quantify how much the experiment workload are similar, or dissimilar, to a given benchmark application, in how they use a CPU [4].

A set of reference workloads from each LHC experiment was analyzed with PrMon, and the resulting values for the metrics are summarized in table 1. The metrics include: number of threads or processes ($N_{thr/proc}$), CPU efficiency (ϵ_{CPU}), time per event (T_{evt}), memory per core (M_c), read rate per core (R_c) and write rate per core (W_c).

As the full PrMon output consists of time series for each metric, we looked into ways to parametrize the time series using a minimal set of parameters. A technique based on CPOP (Continuous piecewise linear Pruned Optimal Partitioning) [7] is able to detect changepoints and therefore reduce a time series to a very small number of points (figure 1). Another work looked into the effect of varying limitations of system resources (memory, network bandwidth and network latency) on the reference workloads, in particular on their wall-clock time. It is planned to combine the two analyses and study the results of the CPOP-based parametrization as functions of the resource restrictions; this would allow to have a very simple input to a

Table 1. Metrics measured by PrMon on a set of reference workloads with an Intel Xeon E5-2630. Values are approximate and may change with different versions of the experiment software

job	$N_{thr/proc}$	ϵ_{CPU}	T_{evt} (s)	M_c (GB)	R_c (MB/s)	W_c (MB/s)
ALICE sim	1	100%	10.9	0.96	0.08	0.17
ATLAS G4	8	100%	270	0.44	0.015	0.009
ATLAS digireco	8	87%	56	1.1	0.3	0.24
ATLAS deriv	8	98%	0.7	1.2	0.7	0.07
CMS gensim	8	99%	21	0.2	0.05	0.04
CMS digi	8	78%	5.9	0.6	0.3	0.3
CMS reco	8	83%	9.8	0.45	0.3	0.2
LHCb gensim	1	100%	180	0.9	0.3	0.01

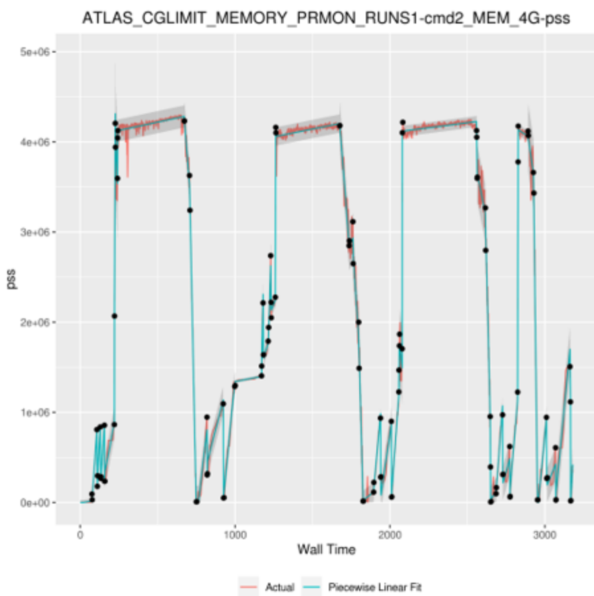

Figure 1. Value of PSS vs. time for an ATLAS digi-reco job when the system memory is restricted to 4 GB

future model of large scale workloads running on a computing infrastructure, like a general purpose batch cluster or an HPC resource.

Studies on the effect of compiler versions and optimizations were done using Geant4 simulation, which showed that statically compiling libraries may achieve a 10% speedup with respect to dynamically compiled libraries, and switching from gcc 4.8.5 to 8.2.0 resulted in a 30% speedup [8]. Consistent results were obtained for CMS simulation [9].

3 Resource estimation

An important step in the process of calculating the cost of computing is to estimate the amount of resources (CPU, disk, tape and network) needed to fulfill the phyics programme of the experiment. Having a sufficiently complex and flexible model is a prerequisite to produce reliable estimates and at some point it was proposed to develop a common framework that could be used by all LHC experiments. Although such framework never came to be, the experiments are now in a much better situation, having modular software-based frameworks instead of the unwieldy spreadsheets that were used for many years. For example, ATLAS and CMS have now frameworks very much comparable in terms of functionality and parameters.

Recent work in CMS [10] focused in adding for the first time estimates for the required tape I/O bandwidth at HL-LHC and for the network capacity; there are still significant uncertainties though, like the future role of GPUs and accelerators, which is impossible to quantify at this point in time.

Given the maturity reached by this estimation process, it is reasonable to assume that the cost model working group will not need to play a role any longer, but for facilitating exchange of information among experiments and identifying possible gaps that would need to be addressed.

4 Site cost estimation

blah

5 Other improvements

A very promising work has started, to study the impact of site caches on the cost of computing for smaller sites. The idea behind it is the consolidation of managed storage at a few large sites forming "data lakes", and its replacement with storage caches elsewhere. This would bring several advantages: lessen the need to create long-lived replicas of datasets (simpler data management), reduce the amount of "cold data" (less storage used), replace complex storage services with simpler ones (lighter site operations) and allow for less redundant and expensive storage configurations (less cost for TB). This work is using actual file access data from ATLAS and CMS and is described in detail elsewhere in these proceedings [11].

6 Cost effectiveness of HPC resources

An aspect of the computing evolution that has not yet been completely understood is the scenario, particularly likely in certain regions, where computing resources at a national level are increasingly provided by supercomputing (HPC) centers at the detriment of more "traditional" computing centers, like those operating as Tier-1s and Tier-2s in WLCG. Using HPC centers for the LHC experiments workloads presents considerable challenges, among which:

- hardware heterogeneity: non-x86 CPUs, GPUs, accelerators
- different authorization/authentication systems, network restrictions, time-limited resource allocations, etc.
- quantification of the usable resources with respect to each relevant workload

The last point in particular was the subject of a preliminary proposal by the cost model and the HEPiX benchmarking working groups, to define a practical procedure to map the capacity of an HPC resource to a WLCG "pledge", briefly described here.

The first step consists in running at a small scale an eligible workload A_1 for a certain time on a certain number of cores and accelerators (not shared with other workloads) and measure the amount of T_{CPU} cores·s and T_{acc} accelerators·s needed to process one event. Subsequently, the same workload is run on a "traditional" system and the amount of HS06·s to process one event is measured [?]. By equating the computing resource usages in the two cases, one can translate a given WLCG pledge to a certain amount of processing time on a given HPC resource. In reality, one will have to take into account also bottleneck effects, for example when the accelerator is underutilized and the CPU is the limiting factor. It is important to know that this equivalence is in principle different for each workload, as each one may use differently the resources, and different for each HPC system, potentially leading to a very large number of combinations; in practice, we can expect only a handful of systems to be used on the few workloads that can use them more efficiently.

Another important metric to be defined would be a measure of how "well" one is exploiting an HPC resource, that we could call *realised potential*. A possibility is to use as reference the R_{max} FLOPS measured by LINPACK [**?**] to express the maximum achievable performance of the HPC system; integrating it over the length of the time allocation, it produces a certain amount of FLOP for the CPUs and the GPUs. While the workload is run, the CPU and GPU utilization levels are measured (as a percent), multiplied by the amounts of FLOP calculated above, and the total FLOP utilized is divided by the total FLOP achievable. This ratio would represent how much of the computing power of the HPC resource is actually used, and it could be used to determine which workloads are best run on the resource, and which resources are most cost-effective for the pledge provider.

In reality, experience shows that running on an HPC resource usually needs a considerable amount of preparation, which is not taken into account in the metrics defined above. It is also clear that, at this time, very limited use can be done (if any) of non-CPU resources for production workloads; however, this is going to change...

7 Conclusions

blah

References

[1] C. Biscarat *et al*, EPJ Web of Conferences, **214**, 03019 (2019)

[2] ATLAS preliminary resource extrapolations, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults

[3] X. Espinal *et al*, *The Quest to solve the HL-LHC data access puzzle. The first year of the DOMA ACCESS Working Group*, these proceedings

[4] A. Valassi *et al*, *Using HEP experiment workflows for the benchmarking and accounting of computing resources*, these proceedings

[5] G. Stewart and A.S. Mete, PrMon, https://doi.org/10.5281/zenodo.2554202

[6] S. Muralidharan and D. Smith, EPJ Web of Conferences, **214**, 08024 (2019)

[7] P Fearnhead, R Maidstone and A Letchford, Journal of Computational and Graphical Statistics **28(2)**, 265-275 (2019)

[8] C. Marcon, S. Muralidharan and O. Smirnova, *Impact of different compilers and build types on Geant4 simulation execution time*, these proceedings

[9] José Flix Molina, private communication

[10] D. Lange, *Modeling of the CMS HL-LHC computing system*, these proceedings

[11] A. Sciabà, M. Schulz and V. Matskovskaya, *Analysis and modeling of data access patterns in ATLAS and CMS*, these proceedings