EPJ Web of Conferences **214**, 03019 (2019) https://doi.org/10.1051/epjconf/201921403019
CHEP 2018

System Performance and Cost Modelling in 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 expected for Run 3 and even more so for Run 4 (HL-LHC) over the next ten years will certainly require radical changes to the computing models and the data processing of the LHC experiments. Translating the requirements of the physics programmes into computing resource needs is a complicated process and subject to significant uncertainties. For this reason, WLCG has established a working group to develop methodologies and tools intended to characterise the LHC workloads, better understand their interaction with the computing infrastructure, calculate their cost in terms of resources and expenditure and assist experiments, sites and the WLCG project in the evaluation of their future choices. This working group started in November 2017 and has about 30 active participants representing experiments and sites. In this contribution we expose the activities, the results achieved and the future directions.

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

1 Introduction

The computing infrastructure for the LHC experiments, managed via the WLCG project [1], has been successfully operating since 2008, with a good match between the resources needed by the experiments and those made available by the funding bodies. In the last few years though it became increasingly clear that Run 3 (for ALICE and LHCb) and Run 4, or HL-LHC (for ATLAS and CMS) will determine very significant increases in the scale of computing, which will not be simply accommodated by technological improvements in the likely scenario of a flat budget evolution. Simply extrapolating the current software performance to the expected trigger rates and average number of overlapping events (pile-up) produces a $O(10)$ discrepancy between the needed and the affordable levels of computing and storage capacity. For this reason, "revolutionary" changes in the software and the computing models of the experiments will be absolutely necessary.

The need of a joint working group between WLCG and the HSF [2] dedicated to the study of performance and cost of computing became apparent and it finally started at the end of 2017, with a long term roadmap that extends to the start of HL-LHC. About thirty members, from experiments, sites and IT and software experts participate in the group activities. The initial focus has been on improving the understanding of current workloads and to establish methodologies and tools to analyse their performance; however, some thought has already been given to the exploration of future scenarios and to try to quantify the gains that could be achieved through paradigm changes. Currently the most important areas of work are:

- the collection of reference workloads from each experiment, to conduct performance studies in controlled and repeateable conditions;
- the definition of the metrics that best characterise the applications and the implementation of tools to measure them;
- the adoption of a common framework for estimating resource needs;
- the adoption of a common process to evaluate the cost of an infrastructure as a function of the experiment needs;
- preliminary studies to explore possible savings in different areas.

2 Workload characterisation and metrics

To model the behaviour of the workloads of the LHC experiments it is essential to understand how the different capabilities that a site provides impact their throughput. To limit the complexity of a performance model, it is desirable to minimise the number of performance characteristics; this requires to find a set of "orthogonal" metrics to which the throughput is sensitive. This knowledge can then be used by software developers to avoid bottlenecks by balancing resource usage, and by site managers to optimise their expenditure. The balance between the amount of memory per core, the disk performance and the memory speed and latency can be optimised on the basis of the characterisation of the relevant workloads. Figure 1 (left) illustrates this relationship.

Finding such a metric set is far from trivial, given the complexity of workloads and their production environments. We started from a representative collection of reference workloads for each experiment. Then, we listed several metrics, grouped in various categories (CPU, I/O, memory, storage, swap and network) and assessed their relevance. Finally, we focused on those that can be measured without significant interference with the running workload on nodes. A tool, PrMon [3], originally developed by ATLAS, has been generalised and allows to measure a large set of parameters, using information from the operating system. Figure 1 (right) shows, as an example, measurements of the memory usage during the execution

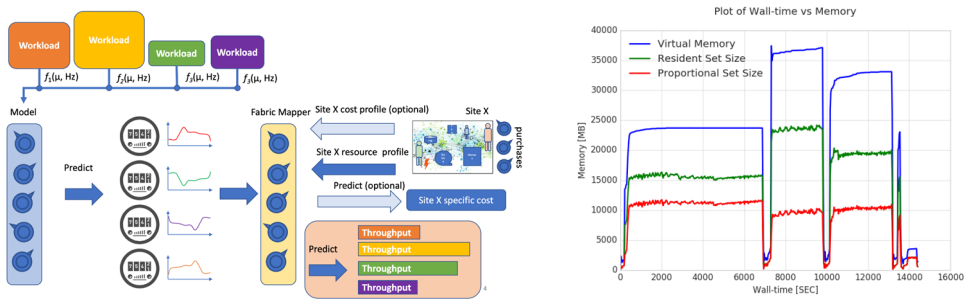

Figure 1. *(left)* Workload resource requirements depend on the pile-up and the trigger rate. The model predicts the demands on the different resources. The Fabric Mapper abstracts the characteristics of a fabric and produces estimates for throughputs. Optional sites can add their local cost structure to the model. *(right)* Memory consumption over the execution of an ATLAS Digi-Reco job from PrMon; the different processing steps are clearly visible. The periods with very low memory consumption are due to the merging of intermediate files

of a workload that performs digitisation with pile-up events and reconstruction for Monte Carlo events. PrMon allows to estimate which capabilities of a system limit the throughput and how efficiently multi-process and multi-threaded applications use their allocated cores. To study the dependencies on latency, bandwidth and memory restrictions, another tool has been developed that runs workloads repeatedly with increasingly restricted bandwidth, memory and increased latency, measuring the throughput for each configuration.

Yet another tool, Trident [4], was developed to provide a convenient access to the information contained in CPU hardware counters. These counters provide information on the level of parallelism exploited by a workload, memory access, cache utilisation and vector capabilities, etc. in an abstract and CPU-model dependent way. This gives the developer an estimate on how much improvement is at best possible in exploiting the available resources and quantitatively understand the limiting factors. The site managers can use the insights gained from Trident in decisions concerning trade-offs between different capabilities, like memory speed vs. size or system disk speed and network.

2.1 Next steps

While the time series measured by PrMon give valuable insights, they cannot be directly used as input for modelling the behaviour of the workloads. Work on this process has started to parametrise metric time series, as processing can be described as sequence of steps, each one looping over events. During each of these steps the resource usage can be described by a small number of parameters and the number of processed events. This, in combination with the dependencies on latency, bandwidth and memory, will allow to predict the throughput of the current workloads under different conditions.

For the resource modelling under future running conditions the dependency on the average pile-up has to be integrated into the model. For data size and simulation time, the dependency is at most linear. For reconstruction, an extrapolation to HL-LHC levels is currently very uncertain and it would be exponential with the current algorithms. ATLAS plans to address this issue by introducing closely spaced tracking layers, to provide precise starting vectors, while CMS will take advantage of a very high time resolution to distinguish between hits from different collisions.

3 Resource estimation

In order to sensibly plan their activity and estimate their costs, experiments need to translate the requirements of their physics programmes into computing resources needs. They thus need a modelling tool that takes as input the details of the physics activities and the features of the computing model and outputs the amount and characteristics of computing resources required.

Currently, all LHC experiments have their own tools for this task, often some complex spreadsheets. Our purpose is to provide a common framework that could standardise, generalise and possibly improve what each experiment does, in its own specific way, today. This framework should be as generic and customisable as possible, it should define a schema for the parameters that describe the expected activity and the characteristics of the computing model, and provide some standard calculations on these parameters. Besides providing the experiment with a standardised way to compute their needs, such framework will allow us to easily play with different computing model scenarios and explore potential gains.

3.1 Current status

A first version of the framework [5] was obtained by forking, refactoring and slightly generalising existing code used by CMS to estimate HL-LHC requirements [6]. It is a set of modules and scripts that read some input files and produces plots and tables with the required storage and CPU resources. Inputs are loaded hierarchically so that we can easily overwrite a generic set of definitions with different special scenarios. This is handled by the classes and functions defined in the *ResourceModel* module. Parameters include:

- LHC parameters: trigger rates, live fractions, shutdown years, etc.
- Computing model: event sizes and processing times, software improvement factors, etc.
- Storage model: numbers of versions, replicas, etc.
- Infrastructure: capacity model, Tier-1 disk and tape, etc.

The framework computes the number of data and Monte Carlo events per year using the LHC parameters and some physics information like the required Monte Carlo/data ratio. The functions defined in the *CPUModel* and *StorageModel* modules translate the number of events into CPU and Storage requirements for the different activities (reconstruction, Monte Carlo, analysis). The module *ModelOut* defines functions to create plots and tables.

3.2 Next steps

This first version of the framework elicited strong interest from other LHC experiments and has been tentatively agreed as a common basis for future development. Generalising it to all LHC experiments will require rewriting some of the most CMS-specific parts and making the parameters schema more flexible. Even if a common framework was not finally adopted by all LHC experiments, having them to adopt a similar approach will greatly benefit the ability to produce consistent and flexible resource estimates. Other foreseen improvements are a generic time granularity and an estimation of network resources.

4 Site cost estimation

The purpose of site cost estimation is to understand and measure what the data centres typical expenses are, and predict what they may be in the future. Our approach is to model those

expenses taking into account the diversity of national contexts across sites in terms of funding, procurement procedures, and local market conditions. These results will help experiments plan new computing strategies that will improve the cost-effectiveness of their resource usage.

4.1 First results

Some of the first elements to address are the diversity of expenses across sites and their definition. A simple and quick exercise was made by four sites, aiming to get a first estimate of the financial cost to run a given workflow and to store a given amount of data, in terms of IT resources and power consumption. The answers to this exercise differed up to a factor of two from site to site. The reasons were found to originate from the intrinsic variety of costs, the measurement method, and the understanding of what a given metric means.

Those results clearly showed the need of a consolidated and common model and method to measure costs across WLCG data centres. For example, one may consider the cost of a server providing a given capacity, including (or not) the equipment that is shipped with it (rack, switch, adaptor etc.). One may also include (or not) in a unit of capacity of tape storage the investments made in library, drives, disk cache etc. that are needed for such system to work. Finally, the measurement of the electrical consumption may include (or not) the Power Usage Effectiveness of the data centre in order to take into account UPS or HVAC system contributions to the final power bill. It is therefore fundamental for site cost estimation to establish a precise definition of the cost-related metrics in order to build a reliable model.

4.2 Next steps

An attempt to address the total cost of ownership (TCO) through cost modeling was shown in [7]. This model assumes that a data centre invests each year a constant budget in the following assets: batch system, disk storage and tape system capacities. If this hypothesis is satisfied (even roughly), one can show that budget (B) and available capacity (K) over time are bound together by a quantity (c^*) that depends on hardware cost evolution and lifetime:

$$B(t) = K(t) \times c^*(t) \tag{1}$$

In the case where hardware unitary costs decrease exponentially over time with a rate r (e.g. 0.2 for a 20% yearly decrease) and hardware is replaced after τ years, one can show that

$$c^*(t) = c(t)\frac{r}{1-(1-r)^\tau} \tag{2}$$

and the site capacity for a flat budget turns out to be exponentially increasing.

This model however does not address all the components of a TCO, like manpower. Site cost studies may leverage this model to estimate the financial impact of a variation in the usage that experiments make of data centre resources. But the quantity c^* may differ significantly from site to site, so an extension of this model at global scale will have to take into account as many as site-dependent parameters as possible to establish c^*.

In order to get a better idea of the variations of the expenditure at different sites, a survey is being conducted at all Tier-1 sites, although the results are not yet available.

5 Areas of improvement

Since the start of the LHC the community constantly improved the throughput of the main workflows; however, studies conducted by the Understanding Performance team and this working group found several areas in which potential improvements could be achieved.

5.1 Compiler and software improvements

WLCG sites provide a variety of CPU models and HEP code is usually built to run on the oldest available architectures. This, and advancements in compilers motivated further studies.

Studies on GEANT simulation, reconstruction and NLO generator code show gains by compiler and link-based optimisations to be around 20-25%, including compilation for individual target CPUs, the use of Intel's commercial compiler and the use of feedback-directed optimisation (AutoFDO from Google and Intel). Compiler-based vectorisation of current production code showed little or no improvement. Reducing the overhead of shared libraries by building large libraries resulted in large gains on older architectures (e.g., a 10% on Ivy Bridge for ATLAS simulation). It has to be noted that the use of feedback-directed optimisation requires that the objects are built statically, which is not always trivial.

From code profiling and a detailed analysis of the dynamic use of memory allocation, it is known that the current code spends up to 25% of the time on memory/object management, due to the frequent creation and destruction of small objects [8]. A 60-90% of allocations exist for less than 100μs and are smaller than 64 bytes. By using better strategies for object management, such as object pools and static vectors, this could be reduced to less than 10% with relatively minor code refactorisation. At the same time the data structure layout can be improved for more efficient utilisation of caches and improved memory access.

The current HEP code executes on modern cores only $0.8 - 1.5$ instructions per cycle (IPC). With the large vector registers provided by current CPUs the theoretical limit is above 20 IPC for fully vectorised code, and tuned complex code for HPC systems can reach about 4 IPC, which can be seen as an upper limit for our code base. Approaching this limit requires at least a change of the used data structures and a re-implementation or factorisation of our algorithms, which should be taken into consideration when designing new algorithms.

Gains from new algorithms and the impact of new detector components cannot be treated here. As the development of the cellular automaton based, HLT track reconstruction code for the ALICE HLT has shown, large factors $O(100)$ can in some cases be achieved [9].

5.2 Storage

The LHC community has recently agreed to consider a scenario where managed storage is consolidated at a few, very large data centres (the "data lake") as the most promising to achieve significant cost savings [10].

For what concerns operational effort, the 2015 WLCG site survey [11] showed that on average Tier-1 sites require 2.5 full time equivalent (FTE) for operating storage and Tier-2 sites 0.75 FTE, with a weak dependence on the amount of storage. By concentrating managed storage at a few sites and using only disk caches at most sites, one can estimate a decrease of the overall number of FTE for storage operations from around 100 to around 60.

Nearly 80% of used disk in WLCG consists of data formats used for analysis. Data popularity studies show that on average datasets exist at ~ 2 sites and are accessed less than 10 times over a period of six months, with most accesses in the first month. We can argue that data needs to be stored only once in the data lake, and temporarily cached as needed depending on the client load. Less popular data may be purged from disk and kept on tape. Again, analysis of popularity data and storage system monitoring indicates that only a small fraction of the produced data is active at any time and a significant fraction (15-20%) of the data could be moved to a different storage layer.

Potential savings depend on the retention strategy and the use of hierarchical storage: what is gained from replacing some amount of disk with tape is partially offset by the need of more tape drives and some added complexity in the data migration.

Table 1. Summary of data access studies for different types of workloads, data location with respect to the processing node, corresponding network latency, access method and relative running time with respect to processing local data; the results show that effective latency hiding can be achieved at the application level (e.g. CMS) or at the caching layer (e.g. ATLAS with a cache)

Workload	Data location	Latency (ms)	Access method	Rel. time
ATLAS Digi-Reco	remote	25	direct	1.90
ATLAS Digi-Reco	rem., empty cache	25	cached	1.08
ATLAS Derivation	remote	25	direct	8.30
ATLAS Derivation	rem., empty cache	25	cached	1.05
CMS Digi-Reco	remote, added latency	10	direct	1.04
CMS Digi-Reco	remote, added latency	20	direct	1.11
CMS Digi-Reco	remote, added latency	50	direct	1.24

The concentration of data at a few sites requires limiting the impact of bandwidth limitations and latency on the workload throughput. We measured the impact of latency on throughput of various workloads, showing that for latencies up to 25ms the reduction of throughput can be limited to 5% when using Xcache [12] as an additional layer (table 1).

Data redundancy within storage systems is achieved either by replication (more performant but expensive) or some form of error encoding (cheaper but less performant). Even larger savings could be achieved by not using data redundancy at all and re-staging from tape, or even regenerating, any lost data. Based on the observed disk failure rate of about 1% per year in the CERN disk storage, the relative total cost of storage and computing at CERN (around 4 HS06/TB)[1], the amount of CPU time to generate AOD events ($\sim$ 850 HS06$\cdot s$) and their size ($\sim$ 400 kB), one can estimate that the computing cost to re-generate the AOD data lost to disk failures is $\sim$ 20% of the cost to make the storage redundant by full replication.

In reality, most of the times the lost data would be replicated elsewhere; one can conservatively estimate that 30-50% of the disk costs can be saved in this way. For this approach to be efficient, the process of recreating individual files has to be automated, which is highly desirable since other failure modes (software bugs, human errors, etc.) lead to data loss on a comparable scale.

5.3 Gains from improvements in operations

Scheduling inefficiencies in WLCG can arise from a mismatch between cores in a system and memory requirements, a mismatch between requested cores and cores grouped on nodes (the tessellation problem), batch system or pilot service inefficiencies, delays due to data staging, I/O waits etc. Site managers have identified some of these problems and $20-30\%$ of resources may be lost due to them. With advanced backfilling and more complex job placement strategies, efficiencies above 90% can be reached, at the expense of added complexity in workload management systems. Another source of scheduling inefficiencies stems from breaking the processing chains from raw data to data analysis objects into several individual steps that exchange data via files: especially when parallel processing threads/processes write individual files, the merging steps create inefficiencies, as they are single threaded. By using asynchronous I/O these inefficiencies can be reduced. The overall impact is difficult to measure, but from ATLAS step chain activity logs it can be estimated to be about 5%.

Losses due to occasional job failures are unavoidable. Currently most of the steps can recover from failure with the help of automated retry mechanisms, but complete jobs are

[1]HEPSpec06 is the benchmark commonly used to measure CPU power for HEP applications [13].

Table 2. Estimated potential gains and associate efforts

Change	Effort for sites	Effort for users	Potential gain
Managed storage only at few sites	some on large sites	little	−40% effort
Reduced data redundancy	some on large sites	some	−30-50% cost
Scheduling and site inefficiencies	some	some	+10-20% CPU
Reduced job failure rates	little	some - massive	+5-10% CPU
Compiler and build improvements	none	little - some	+15-20% CPU
Improved memory usage	none	some	+10-15% CPU
Exploiting modern CPU arch.	none	massive	+100% CPU

sometimes run up to 20 times before successful completion. The overall loss in walltime due to job failures is between 10 and 15%. Improved procedures to make jobs more resilient or fail very early can reduce these losses. Table 2 summarises the identified potential gains.

6 Conclusions

The WLCG/HSF joint working group on systems performance and cost modelling has started only recently, but it is already active on several of the issues that need to be addressed in order to fulfill our computing challenges in the coming years. This involves pursuing performance and efficiency gains in all aspects of software and computing, optimising the usage and the purchase of resources and exploring new ideas in a quantitative way. Collaborations are in place with other activities in the HEP community, including HEPiX for benchmarking [14] and technology aspects, the WLCG DOMA working group and the rest of the HSF.

The work we described clearly indicates that substantial gains are achievable, provided that adequate effort is invested. The activities of the working group will continue, not only to further develop our understanding of the cost of LHC computing and its evolution, but also to increase the knowledge and the awareness of cost and performance issues in our community.

References

[1] Worldwide LHC Computing Grid, http://wlcg.web.cern.ch
[2] HEP Software Foundation, https://hepsoftwarefoundation.org/
[3] G. Stewart, A.S. Mete, PrMon, https://doi.org/10.5281/zenodo.2554202
[4] S. Muralidharan, D. Smith, *Trident: A three pronged approach to analysing node utilisation*, these proceedings
[5] A. Sartirana, Resource modeling, https://github.com/sartiran/resource-modeling/tree/wlcg-wg-new
[6] D. Lange *et al*, *CMS Computing Resources: Meeting the demands of the high-luminosity LHC physics program*, these proceedings
[7] R. Vernet, J. Phys.: Conf. Ser. **664**, 052040 (2015)
[8] S. Kama and N. Rauschmayr, J. Phys.: Conf. Ser. **898** 072031 (2017)
[9] D. Rohr *et al*, J. Phys.: Conf. Ser. **396** 012044 (2012)
[10] HEP Software Foundation, *A Roadmap for HEP Software and Computing R&D for the 2020s*, arXiv:1712.06982 (2018)
[11] M. Alandes Pradillo *et al*, J. Phys.: Conf. Ser. **664** 032025 (2015)
[12] W. Yang *et al*, *Xcache in ATLAS Distributed Computing*, these proceedings
[13] M. Michelotto *et al*, J. Phys.: Conf. Ser. **219** 052009 (2010)
[14] D. Giordano *et al*, *Next Generation of HEP CPU Benchmarks*, these proceedings