HEP Software Development in the Next Decade; the Views of the HSF Community

E Sexton-Kennedy[1]
[1]Fermilab, P.O.Box 500, Batavia, IL 60510-5011, USA

E-mail: sexton@fnal.gov

Abstract. We have known for a while now that projections of computing needs for the experiments running in 10 years from now are unaffordable. Over the past year the HSF has convened a series of workshops aiming to find consensus on the needs, and produce proposals for research and development to address this challenge. At this time many of the software related drafts are far enough along to give a clear picture of what will result from this process. This talk will synthesize and report on some of the key elements to come out of this community work.

1. Motivation

When considering the big picture for HEP in the 10 year timescale the community must realize that we are driven by: 1. the physics objectives of our experiments and the timeline for taking that data; 2. the evolution of technologies we use including hardware, operating systems, compilers, and facility provisioning systems for Grid/Cloud and HPC use; 3. Resource limitations or limitations imposed by resource providers. Ultimately we need to guarantee that we can do the desired physics at the scale and for the cost that funding agencies can pay.

2. Cost Projections

Both CMS and ATLAS have made projections of the cost of their HL-LHC program by scaling up their current computing models. These have to be seen as an upper limit, or starting point to bring down, with research and development efforts over the coming decade. For example, CMS used the time per event and event sizes per tier of its current simulation software being used for HL-LHC detector TDR studies. Clearly the software will evolve to take better advantage of detector handles and algorithmic improvements. The point is to highlight the need for this work. The projections are shown in figures 1 and 2 below.

In order to arrive at a cost, it is also necessary to track the evolution of CPU and storage technologies. In this paper, I will use plots shown at the workshop opening the community process at the beginning of 2017[1]. CERN tracks the cost of its CPU server purchases over the year and uses what it knows about market trends and product roadmaps to project costs out to 2026. Comparing the plots in figures 3 and 4 below to the results in 2015 there has been a reduction in uncertainty of almost a factor of 2. Being aggressive and assuming a 20% increase/year for both CPU and disk, these plots project a cost of 2CHF/HS06 and 0.02CHF/GB respectively. Putting it together CMS would have to request 1.6 petaHS06-s over the course of the year (31.5million seconds) which if procured on owned machines would be 50 million HS06 costing about 100 million CHF. It would need 5 exabytes of disk costing another 100 million CHF. ATLAS and CMS projections agree to within the large factor of two margin of error, so combined the experiments would need 400 million CHF. That is comparable to the run 4 detector upgrade costs. This unaffordability is the challenge that drove the need for community building, and a desire for common solutions to mitigate costs as much as possible.

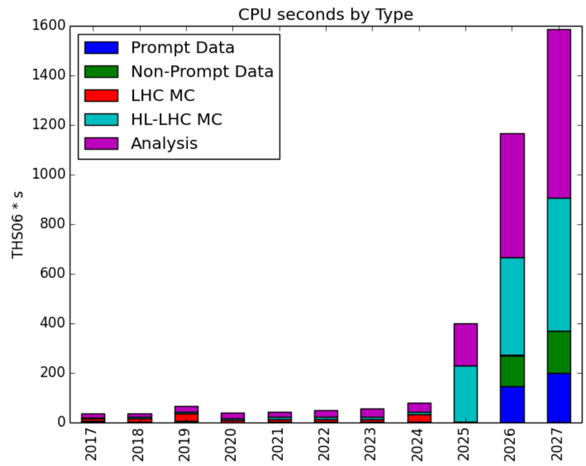

Figure 1. Required CPU second by Type as a function of year.

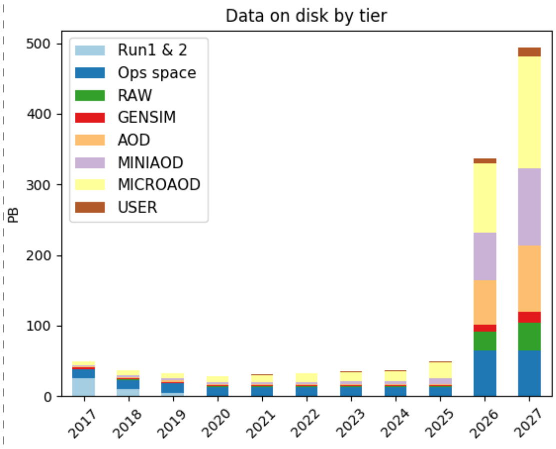

Figure 2. Required data on disk as a function of year.

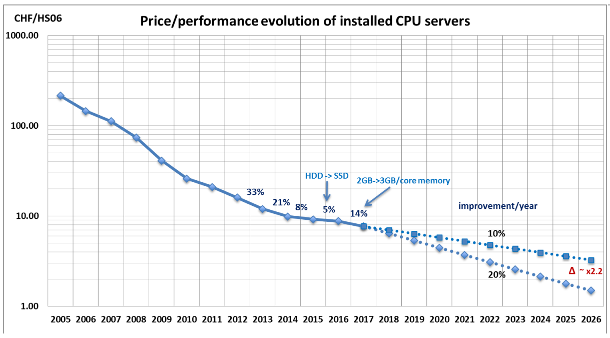

Figure 3. Price per performance evolution of installed CPU as a function of year

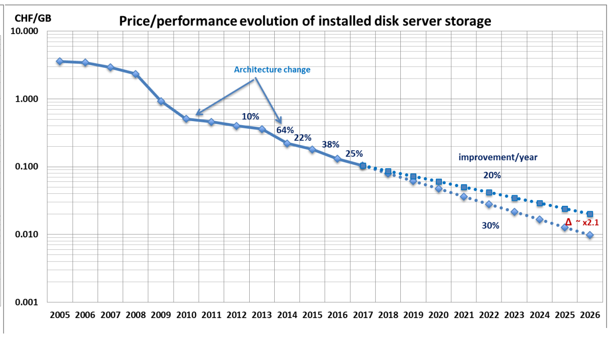

Figure 4. Price per performance evolution of disk server storage as a function of year

3.The Community White Paper, CWP Process

A coordinated response to the above technical and budgetary challenges is clearly needed. In the fall of 2016 the WLCG leadership charged the HEP Software Foundation, HSF to write a community white paper presenting an overall strategy and roadmap for carrying out a "software upgrade" analogous to the detector upgrades for HL-LHC [2]. The process consisted of a series of workshops held throughout the first half of 2017 starting with a kickoff meeting in January, and finishing up with a workshop at the end of June. Participation was open, and all were invited, but special care was taken to ensure representation across international funding agencies and HEP sub-disciplines.

4. Domains covered by the CWP

During the first workshop about a dozen working groups were formed with an initial list of participants who worked on creating charges that defined their area's scope. There was not enough time in the conference to cover all of the work, by all of these groups so I chose to summarize the work of the groups that drive CPU costs such as simulation, generators, reconstruction, frameworks and those that drive the data movement and storage costs such as analysis.

4.1. Simulation

In the domain of simulation there is already a dominant and common simulation engine and related tools like geometry modelers. That is the Geant [3] family of projects. Geant modernization projects have delivered a multi-threaded application (Geant4) capable of running in experiment frameworks on the GRID well before Run 2 giving the LHC experiments enough time to physics validate it. Continuing on that success there is also an R&D project focused on development of a fully vectorized version (GeantV). Adoption of this next step in modernization will be even more difficult then the last multi-threaded step. This CWP working group makes the case that the best strategy for moving forward is to continue this multi-pronged approach. Firstly Geant4 will continue to deliver new or refined functionalities both in physics coverage and accuracy, whilst introducing software performance improvements whenever possible. At the same time, continue the R&D program; investigating new approaches that aim to benefit from modern computing architectures. Its main feature is track-level parallelization, bundling particles with similar properties from different events to process them in a single thread. This approach combined with SIMD vectorization coding techniques and use of data locality is expected to yield large speed-ups, which are being measured in a realistic prototype under development. In addition, the work on Fast Simulation is accelerating with a view to producing a flexible framework that permits Full and Fast simulation to be combined for different particles in the same event. The overriding requirement is to ensure the support of experiments through continuous maintenance, support and improvement of the Geant simulation toolkit with minimal API changes visible to these experiments at least until production versions of potentially alternative engines, such as those resulting from ongoing R&D work, become available, integrated, and validated by experiments. The agreed ongoing strategy to meet this goal is to ensure that new developments resulting from the GeantV R&D program can be deployed immediately in Geant4. GeantV will need to make intrusive data marshaling API changes required to get the dramatic speed improvement that can take advantage of the wide vectors available on modern chips. This in turn will have to wait for the experiment frameworks to catch up and support toolkits like GeantV before any wide adoption in LHC experiments can proceed.

Of course there is more that goes into making simulation samples for experiments: 1. The outputs of Geant, hits in the detectors, have to be made into digitized detector signals. A further complication of this step is that the hard scatter signals (dependent on the generator process being modeled) has to be combined with a large number of minimum bias (low pT QCD events) interactions according to a model of the luminosity profile expected or observed. The issues with simulation overlays is not unique to the LHC. Unless the detector is deep underground it will have model cosmic overlay events. 2. Monte Carlo truth information hast bee handled and passed to analysis. 3. Digitized signals must then be compressed into the RAW data format of the experiment, so that the same reconstruction is run as you would run on the collected detector data.

The following program of work was agreed upon by the simulation community:

1. Continued work on VecGeom, facilitating experiment adoptions and back porting to Geant4 in order to make the modeling of complex detectors computationally more efficient, modular, and transparent.

2. Promote common solutions for detector geometry descriptions that service the needs of simulation and experiment reconstruction, DD4hep.

3. Review the physics models assumptions, approximations and limitations of the simulation engine in order to achieve higher precision, and to extend the validity of models up to FCC energies.

4. Improve the technical performance on emerging computing architecture, e.g., Single Instruction Multiple Data (SIMD) vectorization, Non-uniform Memory Access (NUMA) hierarchies, offloading to Many Integrated Core (MIC), Graphic Processing Unit (GPU), and Tensor Processing Unit (TPU) architectures;

5. Explore fast simulation options, including common frameworks for fast tuning and validation using Machine Learning (ML);

6. Develop techniques for background modeling, including contributions of multiple hard interactions overlapping the event of interest (data overlay, ML)
7. Revisit digitization algorithms to improve performance by means of vectorization and sub-system parallelization techniques. Exploring opportunities for code sharing among experiments when sharing readout electronics.

4.2. Physics Generators
Generator software is another domain in which the community shares tools. Historically generator packages come from small groups of theorist authors that have not needed to worry about computational performance. During the Tevatron and first run of the LHC, the event generation phase of the simulation took a very small fraction of the total time needed to simulate and reconstruct monte carlo samples. This has changed in Run 2 as the have there is a need to invest in higher precision calculations. The fact that theoretical approaches as well as implementations can vary offers an opportunity to validate and cross-check the predictions in a similar way that ATLAS and CMS provide systematic checks of experimental measurements. How much precision for which analysis samples is worth the cost, is still an open question. There is a lots of diversity in generator options and a wide difference in cost among the options. ATLAS reports that in 2016 they spent 20% of their GRID capacity on the generation phase of their application chain. Their average CPU time per event went from 1.6sec./event to 91.4sec/event. In CMS there was an increase as well but not quite as dramatic. It should be noted that the computational challenges involved with modeling QCD at higher orders lend themselves to super computer exploitation. Two areas of future work, with the goal of increasing the precision of generators needed for analyses of the HL-LHC while aiming to reduce our computing requirements, have been identified; 1. the development of a new scalable adaptive Monte-Carlo integrator that can take advantage of highly parallel computing architectures and can be used by existing frameworks; 2. work on re-weighting methods which attach to a given partonic event a new weight corresponding to a new theory. Those new weights allow to predict accurately all the differential distributions at parton-level and to perform a single detector simulation for all the models under consideration. The application of such re-weighting will allow a large reduction in the time spent on the generation of beyond standard model samples.

4.3. Reconstruction
Whether it is reconstructing the simulated data or detector data, reconstruction is the most expensive processing step for many experiments operating 10 years from now. The reconstruction working group has outlined the needs and opportunities to achieve affordability by working on the software tools and methods. They have identified the following broad areas of R&D for algorithms and approaches:

1. Enhanced vectorization programming techniques
2. Algorithms and data structures to efficiently exploit many-core architectures
3. Algorithms and data structures for non-x86 computing architectures (e.g., GPUs, FGAs)
4. Enhanced QA/QC approaches for reconstruction techniques
5. Real-time analysis
6. Precision physics-object reconstruction, identification and measurement techniques
7. Fast software trigger and reconstruction algorithms for high-density environments

Another avenue of attack is to design detectors with reconstruction costs in mind. ATLAS made a factor of 2 difference in cost between their LOI detector and their TDR. CMS collaborators have studied how variations on the tracker layout can significantly change the cost of seeding and track finding [4]. In fact detector handles, which add information to the event that enable new algorithms such as 4D pattern recognition, offer the most hope for reducing costs.

4.4. Analysis

In the analysis domain it is very difficult to separate thoughts about the software from the facility. Analysis is IO and compression limited so more capable facilities, the so called big data facilities, like Spark clusters, can enable new methods of doing analysis that were not available previously. As outlined in section 2, data movement and storage is a significant cost driver and since analysis drives the use of storage any innovations in analysis can make analysis and time to incite more affordable. We learned at the Analysis Ecosystem workshop[5] that while the different experiments draw the line between central and distributed analysis differently, we all do the same things. Namely filter, slice and flatten the experiment's centrally produced analysis samples, until they can fit on a user's personal storage space. Partially this is the nature of the problem; users need rapid turn around when creating plots for publication. Partially this is the result of using the same tool set, ROOT[6]. Historically analysis users are our least savvy software and computing colleagues. As the scale of data needing access increases by an order of magnitude these users will need more help. The Analysis working group outlines many promising avenues of investigation for doing this:

1. Establish PyROOT support and identify other needed measures to promote python as a first class language.
2. Modularization and plugin support plan and roadmap for ROOT
3. Inventory the needed bridges, connectors and converters, identify which are missing or need further work, develop workplans for them
4. Interfacing external ML tools: inventory the needs and develop workplans
5. ROOT I/O performance analysis and tuning
6. Assess the relevance of AaaS and notebook for the analysis community
7. R&D on how new hardware technologies could affect the analysis tool chain, Spark clusters and similar…

4.5. Frameworks

Frameworks are not cost drivers themselves, however they enable novel approaches, as already mentioned in the section discussing vectorized Geant. Without support in the experiment frameworks for innovative techniques, those approaches can not be integrated and deployed for the wides spread use in the production workflows of the experiments. It is recognized that frameworks have a central role in supporting the porting of applications onto advanced architectures and HPCs.

The group of people who have architected and designed previous experiment frameworks is not large, and the CWP effort has brought together most of the people involved in this over the last 2 decades. The biggest challenge in this area is the social challenge of dealing with large sets of legacy software, even though it is recognized that much of that software will have to be modernized. As a starting point this group has attempted to define common terms that help us understand that the different experiments do not have widely different needs. There is disagreement about the feasibility of a common solution even though it is recognized that this is needed for modernization and sustainability. The group proposes to investigate the possibilities of increasing the sharing of existing frameworks and framework components. In an attempt to answer the question, "Is it possible to reduce the number of existing frameworks?" Event without the answer to that question, there are some obvious areas in which common work can be done. Those include:

1. Investigate alternatives to TBB which include support for heterogeneous resources and HPC facilities.
2. Improve the understanding of the interplay between workload management and framework, and the tradeoffs between multi-threaded application, and heavy weight cooperative process frameworks.
3. Devise ways of accommodating multi-event tools like GeantV in the framework.

4.6. Crosscutting Topics and Solved Problems

There are many other topics addressed by CWP working groups that I did not have time to cover even in this summarized form but which do deserve mentioning. The area of Software Development, Deployment and Validation/Verification is a cross cutting area that is absolutely vital to the advancement of all of the domains covered above. These tools allow us to build on our long legacy through detailed regression testing. The software captures our domain knowledge, from matrix element calculations, to the detector understanding built into the reconstruction, and without the tools to prove that new techniques do not destroy that understanding, we can not adopt and deploy them.

The area of machine learning is another cross cutting technique that is applicable to analysis, reconstruction and simulation. It is a promising tool that offers exciting possibilities to solve these challenges.

Attention to Careers, Staffing and Training is a necessary investment to successful carry out this ambitious program of work.

While the areas of conditions databases and visualization are solved problems, the target of opportunity in these areas is to increase commonality among experiments and thereby decrease manpower. The Visualization working group has also concluded that there is a need to modernize and adopt industry tools, thereby getting much more bang for the same about of effort.

5. Conclusions

There is a field wide, large challenge ahead of us on the timescale of 10+ years. Limitations of power, cooling, and funding will not allow us to keep our current software paradigm, nor our current mode of computing resource provisioning. In the face of so much uncertainty it is vital that we continue to talk to each other. We need to attack this challenge from many angles; by maximizing the scientific computing infrastructures available to do science 10 years from now; by developing innovative and efficient software solutions for the domains we know today; by designing detectors that give us handles to minimize computational costs, throughout the data flow, from detector to insight. I have attempted to summarize the work done in 2017 to define a software roadmap for R&D targeting their challenge. As of this writing, the first draft of the CWP itself, it is under review and should be delivered by the end of Oct. For the most up to date information about the status of the paper check the HSF CWP web page [7].

6. References

[1] Helge Meinhard, and Bernd-Panzer Steindel, "Computing Evolution: Technology and Markets" HSF workshop San Diego, 2017 https://indico.cern.ch/event/570249/contributions/2404412/

[2] Charge for Producing a HSF Community White Paper, July 2016, http://hepsoftwarefoundation.org/assets/CWP-Charge-HSF.pdf

[3] Geant4 project, "Geant" [software], version 10.3.2, 2017. Available from https://github.com/Geant4/geant4/releases/tag/v10.3.2 [accessed 2017-08-17]

[4] V. Krutelyov et. al. "Impact of Tracker Layout and Algorithmic Choices of Cost on Computing at High Pileup", 38th International Conference on High Energy Physics, August 2016 https://pos.sissa.it/282/194/pdf

[5] The HSF Analysis Workshop, May 2017 https://indico.cern.ch/event/613842/

[6] Rene Brun and Fons Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also "ROOT" [software], https://doi.org/10.5281/zenodo.848819

[7] The HEP Software Foundation, "A Roadmap for HEP Software and Computing R&D for the 2020s" http://hepsoftwarefoundation.org/activities/cwp.html

Acknowledgments

This work was supported by Fermilab; Operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.
The HSF Community White Paper workshops have been supported and organized by many funding agencies and institutions including the University of California at San Diego, the San Diego Supercomputer Center (SDSC), CERN, Nikhef, the Laboratoire d'Annecy de Physique des Particules (LAPP) and the S2I2-HEP project supported by National Science Foundation grants ACI-1558216, ACI-1558219, and ACI-1558233.