

Software and Computing for Small HEP Experiments¹

Costas Andreopoulos,^{1,2,a,b} Stephen Bailey,^{3,c} Deborah Bard,^{3,4} Wahid Bhimji,^{3,4} Dave Casper (editor),^{5,d,e} Giuseppe Cerati,^{6,f} Jacob Daughhetee,^{7,g} V. Daniel Elvira,^{6,h} Alden Fan,^{8,9,i} Krzysztof Genser,^{6,j,k,l} Paolo Girotti,^{10,m} Scott Kravitz,^{3,i} Robert Kutschke,^{6,j} Maria Elena Monzani (editor),^{8,9,11,i} Benjamin Nachman (editor),^{3,e,n} Vincent R. Pascuzzi,¹² Gabriel N. Perdue,^{6,o} Erica Snider,^{6,f} Elizabeth Sexton-Kennedy,^{6,k} Graeme Andrew Stewart,^{13,e,k} Matthew Szydagis,^{14,i} Eric Torrence,^{15,d,e} Christopher Tunnell^{16,p}

¹ *University of Liverpool*

² *STFC Rutherford Appleton Laboratory*

³ *Lawrence Berkeley National Laboratory*

⁴ *National Energy Research Scientific Computing Center (NERSC)*

⁵ *University of California, Irvine*

⁶ *Fermi National Accelerator Laboratory*

⁷ *Oakridge National Laboratory*

⁸ *SLAC National Accelerator Laboratory*

⁹ *Kavli Institute for Particle Astrophysics and Cosmology*

¹⁰ *INFN Pisa*

¹¹ *Vatican Observatory*

¹² *Brookhaven National Laboratory*

¹³ *CERN*

¹⁴ *SUNY Albany*

¹⁵ *University of Oregon*

¹⁶ *Rice University*

^a *T2K Collaboration*

^b *SBND Collaboration*

^c *DESI Collaboration*

^d *FASER Collaboration*

^e *ATLAS Collaboration*

^f *MicroBooNE Collaboration*

^g *COHERENT Collaboration*

^h *CMS Collaboration*

ⁱ *LZ Collaboration*

¹ Authors have listed their experimental affiliations to illustrate the diversity of experiments represented here, but the opinions, findings, and conclusion or recommendations expressed in the paper are those of the authors and do not necessarily reflect the view of their collaborations.

^j*Mu2e Collaboration*

^k*HEP Software Foundation*

^l*Geant4 Collaboration*

^m*Muon g-2 Collaboration*

ⁿ*H1 Collaboration*

^o*MINERvA Collaboration*

^p*Xenon Collaboration*

E-mail: dcasper@uci.edu, monzani@stanford.edu, bpnachman@lbl.gov

ABSTRACT: This white paper briefly summarized key conclusions of the recent US Community Study on the Future of Particle Physics (Snowmass 2021) workshop on Software and Computing for Small High Energy Physics Experiments.

Submitted to the Proceedings of the US Community Study
on the Future of Particle Physics (Snowmass 2021)

Contents

1	Introduction	1
2	Key Observations and Recommendations	2
2.1	High Performance Computing Centers	2
2.1.1	Key challenge: Porting workflows to new architectures	2
2.1.2	Key challenge: Policy optimization	3
2.1.3	Key challenge: Scaling user support	3
2.2	Requirements beyond FLOPS, bytes, and bandwidth	4
2.3	Software Challenges	4
2.4	Support for Common Tools	5
2.5	Data Storage and Preservation	6
2.6	Computing Personnel	7

1 Introduction

This white paper summarizes the discussion from a dedicated Snowmass workshop on Software and Computing for Small High Energy Physics (HEP) Experiments (<https://indico.physics.lbl.gov/event/1756>). The mandate for the workshop was:

- Identify unique computational challenges of the ‘small’ experiment community.
- Gather input about what is needed in terms of computation for these experiments to be successful.
- Connect members of the ‘small’ experiment community to the computational frontier in Snowmass and encourage participation in topical groups.
- Foster the development and re-use of open-source software, building on the work of the HEP Software Foundation (HSF) and other collaborative efforts within the community.

In order to be inclusive, we did not impose a definition of ‘small’ and have asked experiments to self-select. The workshop was organized into two days: a first day for perspectives from representative experiments and a second day about computational tools. In particular, on the first day, we heard from CERN-based experiment FASER [1], Fermilab-based experiments MicroBooNE [2] and g-2 [3], and non-CERN/FNAL experiments LUX-ZEPLIN (LZ) [4], COHERENT [5], and DESI [6]. These experiments covered all of the HEP frontiers. On the second day, there were speakers covering the HSF, High Performance Computing (HPC), event processing frameworks, reconstruction frameworks, event generators,

detector simulations, and machine learning. The slides contain a wealth of information about each of these topics and are preserved at the [indico link](#) above. The goal of this white paper is to describe some of the key takeaway points from the presentations and discussions. If the entire workshop had to be summarized in one phrase, it would be¹

small experiment $\neq$ small data volume or small computing problem!

2 Key Observations and Recommendations

Below is a summary of important points discussed at the workshop; key recommendations are **offset in bold**. The four subsections below are in no particular order.

2.1 High Performance Computing Centers

Increasingly, small experiments need to rely on HPC resources for a significant fraction of their computing budget. At the workshop, we heard about challenges by experiments in porting to HPC resources and about challenges from HPC center staff about accommodating small experiment users.

From the perspective of an HPC center, ‘small’ experiments look very similar to ‘large’ experiments. They have similar scales in terms of compute hours, storage used and number of users, and they all need peripheral services such as databases and workflow controls, as well as allocations on the big systems. For an HPC center, this means both ‘small’ and ‘large’ experiments present the same demands on resources. What distinguishes ‘small’ experiments is that many of the common issues faced by all experiment teams working at HPC centers are exacerbated by the lack of human resources available to work on these issues. It is vital that science teams and HPC centers partner to address these challenges. These issues fall into roughly three areas:

- Getting codes running on new architectures
- Adapting to HPC center policy
- Tools that are easy to stand up and maintain

2.1.1 Key challenge: Porting workflows to new architectures

The DOE Office of Advanced Scientific Computing Research (ASCR) operates three computing and one networking user facilities. All have a joint mission to advance both science and the state of the art in computing. ASCR facilities engage with vendors years in advance of a system’s delivery, to ensure that production systems push the limits of existing technology. There are three key drivers in this advancement that pose a challenge to scientific workflow design:

¹Quoted from E. Snider’s presentation.

- Energy efficiency. Scientific computing needs are ever-increasing, but there is a limit to the amount of power and cooling we can safely bring into a building. Energy efficient architectures are here to stay, including specialized devices that perform specific calculations at ultra-high efficiency (e.g., for AI/ML).
- New storage technologies. Object stores are increasingly prevalent, and massively parallel file systems do not always respond well to data-intensive workloads.
- Peripheral services. We are seeing an increasing use and reliance on containers, scalable databases, and interfaces to other computing sites (e.g., local clusters, the cloud).

A pressing question for both experiment teams and HPC center staff is how to optimize the entire end-to-end workflow, not just individual applications. The NESAP program at NERSC (the National Energy Science Research Computing center) partners with application development teams and vendors to port and optimize codes to new architectures and platforms. Lessons learned from this process are shared with the wider NERSC community via documentation and training. NESAP is evolving with the types of systems we deploy, and will need to support workflow optimisation in the future.

Programs like NESAP are extremely valuable to small experiments and we advocate for a continuation and/or expansion of this program. However, we also heard about difficulties in recruiting NESAP postdocs. This is discussed in more detail in Sec. 2.6.

2.1.2 Key challenge: Policy optimization

Some of the biggest challenges faced by users of HPC centers are due to security and policy concerns, rather than technical barriers. Supporting the needs of experiment workflows has pushed NERSC to change policies around (for example) near-realtime access to systems, and support for federated ID.

In addition, NERSC has recognised that some workflows highly value continued access to NERSC services - for example, if an experiment is operating 24/7. Through careful planning and investment, we are now able to keep most infrastructure up during outages using generators and new system software capabilities for rolling upgrades. We are also supporting research in several projects to support more portable cross-facility workflows [7].

2.1.3 Key challenge: Scaling user support

NERSC is increasingly supporting more users and projects from experiment facilities (including other DOE user facilities). HPC centers are facing a sea change in the number and type of users they support - without any change in the number of staff available to support them. This challenge requires a new approach to user support, to make it easier for users to develop and adapt their workflows without direct support from NERSC staff. If done successfully, this will also make it easier for ‘small’ experiments to adapt to HPC centers.

The LBNL Superfacility Project [8] was designed to leverage and integrate work being done across NERSC, ESnet and research divisions at LBNL to provide a coordinated and

coherent approach to supporting experimental science at DOE facilities. The principles behind the project were to provide an integrated, scalable and sustainable framework for experiment science, working closely with a range of science teams to get the design right and using existing, industry standard and open source tools wherever possible. This 3-year Project was completed at the end of 2021, having achieved its aim, although work continues very actively on Superfacility topics. NERSC is now able to support automated pipelines that analyze data from remote facilities at large scale, without routine human intervention, using capabilities such as near-realtime computing, dynamic networking, API-driven automation, HPC-scale notebooks with Jupyter [9], state of the art data management tools, federated ID and container-based peripheral services.

2.2 Requirements beyond FLOPS, bytes, and bandwidth

Classic high performance computing (HPC) centers have traditionally focused on maximizing the number of floating point operations per second (FLOPS) delivered to massively parallel programs, coupled with large bandwidth to disk and tape storage. Small experiments often have requirements that go beyond just FLOPS, bytes, and bandwidth, e.g. databases, data access portals, I/O robustness, complex workflows with many interdependent smaller jobs, collaboration accounts accessible by multiple users to coordinate data processing, queue waittime, and jupyter notebook servers. High uptime, low mean time to failure (MTTF), and access to realtime and interactive computing resources are particularly important for the effective use of computing centers by smaller experiments. Supporting these requirements takes tangible specific effort from the computing centers, and tangible specific effort from the experiments to adapt. As such, HPC centers need dedicated support for non-HPC workloads with appropriate metrics for evaluating success, e.g. minimizing MTTF and providing interactive access, and not just FLOPS delivered. Similarly, experiments need dedicated support to adapt and maintain their workflows to take full advantage of the opportunities at HPC centers.

2.3 Software Challenges

High Energy Physics has a vast investment in software estimated to be greater than 50M lines of C++ code. It is a critical part of our physics production pipeline, from triggering and simulation all the way to analysis and final plots. This legacy represents a huge investment over the years. It requires ongoing support in human effort to adapt to changing computing technologies. As shown in figure 1 1, improved single threaded performance will not come from the hardware. Computing gains come from increasing concurrency in the code, especially at HPC centers. Concurrent programming skills must be learned and not all experimenters want to invest in becoming proficient in concurrent programming. There is no longer a large supply of graduate students and postdocs, the historical sources of experiment software, that can write or even maintain codes for these complicated architectures. This hits smaller experiments especially hard, because there may not be a critical mass of people with the critical skills needed. See Sec. 2.6 for suggestions on how to address personnel needs. Efforts to share code developed by the large experiments should be funded to have broader impact for the community, and thereby maximize return on

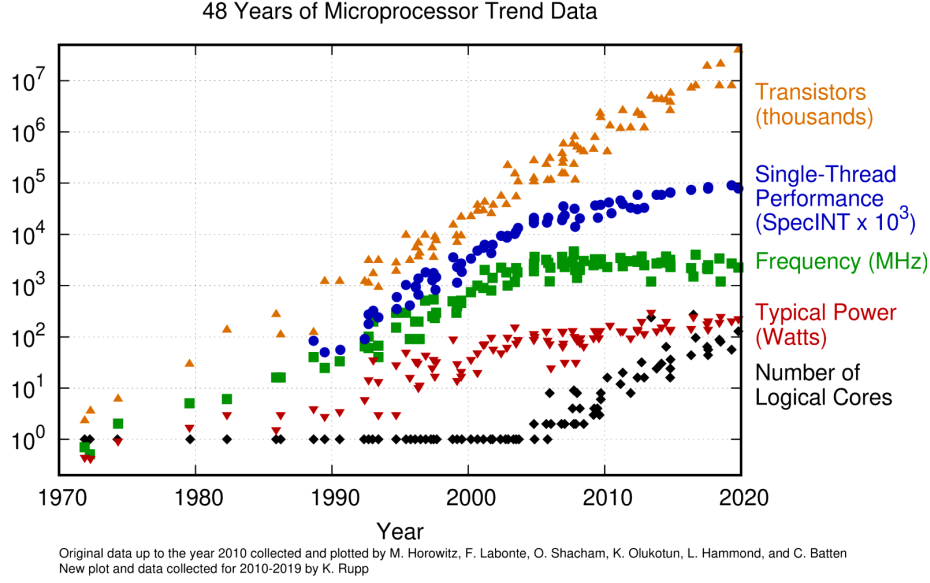


Figure 1. Microprocessors reach the power wall limit circa 2010

those investments. This means adding time for documentation, training, and outreach. Whenever possible the code should be open sourced.

2.4 Support for Common Tools

A reoccurring theme in nearly all of the experiment vignettes was the need for common tools. A distinguishing feature of small experiments compared to large ones is that they often do not have many/any dedicated software and computing experts. This means that there is limited/no resources to develop new data acquisition/simulation/reconstruction tools from scratch as large experiments are often able to do. Some small experiments benefit from shared expertise with large experiments, especially those located at CERN and FNAL. Yet many small experiments do not have such natural overlap, and they are left with few avenues to obtain the resources necessary to develop their required suite of tools.

In some cases, larger experiments have (at least partially) abstracted and open-sourced their software in a way that it can be re-used by small experiments. Two great examples of this are the software packages ACTS [10] and LArSoft [11] (<https://larsoft.org>) as well as the event processing frameworks Gaudi [12], Art [13] and Stitched (<https://github.com/cms-sw/Stitched>) an abstraction of the CMSSW framework[14]. ACTS (A Common Tracking Software) is a package for tracking at collider experiments, initially based on ATLAS code. It provides experiment-independent implementations of track and vertex reconstruction algorithms which are available for usage by any experiment once their geometry and material description is implemented in ACTS. Successful examples include ATLAS, FASER, sPHENIX, Belle-II. LArSoft is a shared base of simulation, reconstruction, and analysis software across liquid argon time projection chamber (LArTPC) neutrino experiments. LArSoft is currently used by all Fermilab-based LArTPC experiments, in-

cluding DUNE and smaller-scale experiments (SBN, MicroBooNE, LArIAT, ArgoNeuT). Both ACTS and LArSoft provide ready-to-use software solutions as well as a platform for R&D work spanning usage of AI methods, algorithm parallelization, and deployment at HPC, which can significantly reduce the required workload for smaller experiments.

However, some open source tools still require significant adaptation or development and thus require experiment-independent support. The best example of this is the widely-used detector simulation software Geant4 [15–17]. Even for large experiments, significant resources are required to integrate Geant4 into the detector-specific setup. Many small experiments have physics needs beyond those covered by large experiments and additional development is necessary. Some of this is happening organically (e.g. NEST [18] for dark matter direct detection), but this is not always the case. In the US the HEP support for Geant4 comes from the computing operations budgets of large experiments and intensity frontier operations at Fermilab. This leads to a stove-piping of Geant support such that issues specific to those experiments are addressed because they are needed for the operation of the experiment. Other experiments that use Geant are left out in the cold. In addition Geant’s common software, such as physics models, no longer receives any US maintenance funding. This is not sustainable for the long term viability of small experiments. **Long term, experiment-agnostic Geant4 support is critical for the success of small experiments.** At the workshop, we also heard that many small experiments do not update Geant4 versions frequently (or ever!) and so they are unable to take advantage of new developments, while maintaining many old versions by the Geant4 collaboration is unsustainable. This means that small experiments often use no longer supported Geant4 versions and miss out on recent physics developments and computational innovations, such as multi-threading or the ongoing integration with Graphical Processing Units (GPUs). To strengthen the collaboration and communication between the Geant4 Collaboration and the experiments, the Geant4 Collaboration had recently introduced a status of a Geant4 contributor which should facilitate a faster inclusion of codes developed by the experiments, benefiting the entire community using Geant4. The physics Generators used in the field (eg. Pythia, GENIE, Madgraph, Sherpa) also suffer from lack of stable funding in the same way as Geant does.

2.5 Data Storage and Preservation

Data preservation is necessary for scientific repeatability, while helping maximize the “return on investment” of our experiments. Re-analysis of data is of particular interest, as advances in analysis methods and physics models become available such as more detailed simulations and new applications of machine learning. Well-understood datasets are often the best source of training for young physicists, and open-access datasets support outreach efforts by targeting both data literacy and science skills. In addition, standardized, trusted data repositories are critical for developing and testing new machine-learning algorithms [19]. Encouraging the experiments to make their data publicly available will lead to a richer ecosystem of available datasets, and help with long-term preservation of data and experimental knowledge. Most important is the publication of data following the end of an experiment. One way of ensuring data are usable for post-experiment use is to periodi-

cally release data throughout the lifetime of the experiment. Another model is to internally future-proof data formats and analysis software to always be able to analyze older datasets. In both cases, person-power is required for an effective implementation.

Computing clusters do not typically guarantee hosting for significant lengths of time, nor do they have mechanisms to easily share these data with the public. However, services for long-term data and software preservation (<https://heasarc.gsfc.nasa.gov> and <https://lambda.gsfc.nasa.gov>) have been used for decades in astronomy, and the HEP community has become quite familiar with them. Ensuring data usability long after the end of an experiment is not quite as simple as “dumping everything” in a publicly available repository. Detailed detector information is needed, as well as access to metadata such as slow-control and calibration databases. Moreover, community standards specifying data format and interfaces must be agreed upon. Finally, prototype analysis codes must be published with the datasets, further lowering the barrier to data access and ensuring that the data is analyzed correctly. It has become more commonplace in our community to take advantage of general purpose data and code services like Zenodo (<https://www.zenodo.org>), which provide most of the infrastructure required for effective data preservation.

2.6 Computing Personnel

The general challenge of maintaining software and computing talent is exacerbated in small experiments by the lack of long term, permanent positions within the experiments. Many of the software tasks are carried about by young collaborators and there is significant turnover from year to year. A special case is an experiment that is both small and far from data, for which it is common that the community of junior people working on software is not large enough to be self-sustaining. This problem can be partly mitigated by increased use of common tools but that does not address the task of developing experiment specific configurations, customizations and glueware.

It is therefore essential that there is **funding for permanent software and computing experts**. The careers of these researchers should be evaluated appropriately with rewards to efficiency, stability, and robustness. A more transparent approach to software development and data sharing would go a long way towards improving the career prospects of software and computing experts, as it would allow individuals to claim credit for their work and be evaluated appropriately. There is little funding for software maintenance, even though this is critical. One possibility is the creation of research software engineer (RSE) positions that have long term funding independent of experiments, but for the support of existing and planned experiments. Another area of opportunity is the increase of joint particle physics and data science appointments at universities, which have become marginally more common over the last decade. We also need to increase the community investment in general software literacy of physicists, through (continuing) education initiatives and collaborations with industry, national labs and academia.

The diversity issues pervasive in HEP are exacerbated in the computing domain due to the (perceived) technical nature of the work, and we must ensure that faculty, staff, and trainee positions are viable career options for a diverse group of people. Our workforce development efforts should explicitly include equity, and recognize that diversity is foun-

dational to our success, as it demonstrably increases the creativity of solutions and variety of approaches. Diversifying the computational workforce will require efforts to diversify physics in general, and to leverage our partnership with the astronomy and industry communities, who have made substantial efforts in this direction over the last decade. This remains a challenging problem, but not insurmountable, as STEM identity formation for underrepresented minorities has become a priority in a variety of educational settings, from early childhood to higher education [20].

Acknowledgments

We thank the participants of the Software and Computing for Small HEP Experiments workshop <https://indico.physics.lbl.gov/event/1756> for many useful discussions.

This work was supported by the US Department of Energy, Office of Science, under contract numbers DE-AC02-76SF00515 (SLAC) and DE-AC02-05CH11231 (LBNL), and by the Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract DE-AC02-07CH11359 with the US Department of Energy.

References

- [1] J. L. Feng, I. Galon, F. Kling and S. Trojanowski, *ForwArd Search ExpeRiment at the LHC*, *Phys. Rev. D* **97** (2018) 035001, [[1708.09389](#)].
- [2] MICROBOONE collaboration, R. Acciarri et al., *Design and Construction of the MicroBooNE Detector*, *JINST* **12** (2017) P02017, [[1612.05824](#)].
- [3] MUON G-2 collaboration, J. Grange et al., *Muon (g-2) Technical Design Report*, [1501.06858](#).
- [4] LZ collaboration, D. S. Akerib et al., *The LUX-ZEPLIN (LZ) Experiment*, *Nucl. Instrum. Meth. A* **953** (2020) 163047, [[1910.09124](#)].
- [5] COHERENT collaboration, D. Akimov et al., *COHERENT 2018 at the Spallation Neutron Source*, [1803.09183](#).
- [6] DESI collaboration, M. E. Levi et al., *The Dark Energy Spectroscopic Instrument (DESI)*, [1907.10688](#).
- [7] K. B. Antypas, D. J. Bard, J. P. Blaschke, R. Shane Canon, B. Enders, M. A. Shankar et al., *Enabling discovery data science through cross-facility workflows*, in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 3671–3680, 2021, [DOI](#).
- [8] B. Enders, D. Bard, C. Snively, L. Gerhardt, J. Lee, B. Totzke et al., *Cross-facility science with the superfacility project at lbl*, in *2020 IEEE/ACM 2nd Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP)*, pp. 1–7, 2020, [DOI](#).
- [9] R. Thomas and S. Cholia, *Interactive supercomputing with jupyter*, *Computing in Science Engineering* **23** (2021) 93–98.
- [10] X. Ai et al., *A Common Tracking Software Project*, [2106.13593](#).
- [11] E. L. Snider and G. Petrillo, *LArSoft: Toolkit for Simulation, Reconstruction and Analysis of Liquid Argon TPC Neutrino Detectors*, *J. Phys. Conf. Ser.* **898** (2017) 042057.
- [12] G. Barrand et al., *GAUDI - A software architecture and framework for building HEP data processing applications*, *Comput. Phys. Commun.* **140** (2001) 45–55.

- [13] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren and Q. Lu, *The Art Framework*, *J. Phys. Conf. Ser.* **396** (2012) 022020.
- [14] C. Jones, M. Paterno, J. Kowalkowski, L. Sexton-Kennedy and W. Tanenbaum, *The new cms event data model and framework*, *Proceedings for Computing in High-Energy Physics (CHEP'06), Mumbai, India* **13** (2006) .
- [15] GEANT4 collaboration, S. Agostinelli et al., *GEANT4—a simulation toolkit*, *Nucl. Instrum. Meth. A* **506** (2003) 250–303.
- [16] GEANT4 collaboration, J. Allison et al., *Geant4 developments and applications*, *IEEE Trans. Nucl. Sci.* **53** (2006) 270.
- [17] GEANT4 collaboration, J. Allison et al., *Recent developments in Geant4*, *Nucl. Instrum. Meth. A* **835** (2016) 186–225.
- [18] M. Szydagis, N. Barry, K. Kazkaz, J. Mock, D. Stolp, M. Sweany et al., *Nest: a comprehensive model for scintillation yield in liquid xenon*, *Journal of Instrumentation* **6** (Oct, 2011) P10002–P10002.
- [19] D. Dua and C. Graff, *UCI machine learning repository*, 2017.
- [20] A. Singer, G. Montgomery and S. Schmoll, *How to foster the formation of stem identity: studying diversity in an authentic learning environment*, *International Journal of STEM Education* **7** (12, 2020) .