CompF2 Recommendations

Peter Boyle (BNL), Kevin Pedro (FNAL), Ji Qiang (LBL)

Topical Areas

1. Cosmic Calculations
2. Particle Accelerator Modeling
3. Detector Simulation
4. Physics Generators
5. Perturbative Calculations
6. Lattice QCD

CompF2 Report

1. Executive Summary
2. Introduction
3. Cross-cutting computational challenges
4. Topical Area Summaries

   …

10. Conclusions and Recommendations

Report draft URL:
https://www.dropbox.com/s/zy4u7jl9qaxjod1/CompF2_Report_DraftV1.0.pdf?dl=0
(https://snowmass21.org/computational/start#report_drafts)

Computing Hardware Recommendations (1)

- **Faster computer hardware**: There is a need for HPC resources ten times or more faster than planned Exascale computing systems.
- **Support of right-sized CPU clusters**: There should be a right-sized (as large as required, as small as possible) provisioning of general purpose computational cores with high performance memory for important algorithms that do not easily map to acceleration. National Laboratory institutional (non-HPC) clusters contribute to this role. NERSC is a welcome example of providing both CPU and GPU/accelerator resources to handle a wider variety of problems relevant to HEP. These should accommodate a diverse range of requirements for software, large memory, and long running single node jobs.
- **Universal programming interface**: The proliferation of programming interfaces is a barrier to portability and return on investment in software. Simplification and consolidation of the interfaces will improve scientific output.

Computing Hardware Recommendations (2)

- **Software portability**: The difficulty of programming highly accelerated hardware with performance portability is significant. Appropriate support for software development and maintenance is essential to ensure success of much of the science discussed in this report.
- **Automate memory hierarchy**: Hierarchical memory is a significant burden on scientific programmer productivity. Virtual memory paging systems can alleviate this and can be made efficient by using larger page sizes, if necessary. This may need substantial investment by computer vendors with robust encouragement from the Department of Energy.
- **Early access to new computing hardware**: A key element of managing the science program is the early engagement with DOE HPC laboratory sites during the development, and years prior to installation, of major new facilities. The lead time for porting to new architectures lies in the region of multiple years, and early engagement with emerging architectures is required to ensure timely scientific exploitation.

Software Ecosystem Recommendations (1)

- **Accelerator-friendly, portable common tools**: Specific projects to adapt common software packages to run efficiently on GPUs, including those at new HPCs, have shown initial promise with more than order of magnitude speed improvements. These projects should be continued with sufficient effort to deliver complete products and ideally to incorporate portability solutions looking forward to different coprocessor devices and architectures. Adoption of this new software will additionally require expertise devoted to integration and usage in experimental software frameworks, analysis tools, etc., which should be supported as discussed below. The exploration of new machine learning-based methods, as well as technical improvements to existing (CPU-based) software, should also continue.
- **Best practices for common software**: Whenever possible, interoperability and common data structures should be encouraged. Existing standard libraries and underlying software technologies should be used when available and feasible. Interoperability, portability, and common, shared software are fundamental pillars of international scientific collaboration and will become increasingly important in the future.
- **Continuous collaboration with ASCR**: A lot of progress has been made in high energy physics applications through collaborations with programs supported by ASCR under the SciDAC and ECP projects. We believe that such collaborations should be encouraged and continuously supported for future computational modeling in high energy physics.

Software Ecosystem Recommendations (2)

- **Lab-supported software development**: Just as the large experiments require talented permanent staff at the Labs to engineer experiments, theoretical calculation and simulation in HEP experiment, in high energy theory, in cosmology, and in large particle accelerator building have large communities often dependent on sophisticated long-term software systems. Despite the numerous successes and broad adoption of common software tools across all topical areas in this report, the future of these projects is in jeopardy because of an extreme lack of funding. Fixing this requires that career paths be created to retain some of the most talented experts in software and algorithms. One priority should be the development, maintenance, and support for common software tools in the areas of accelerator modeling, detector simulation, and physics generators. A larger permanent laboratory staff of software development experts could help address HPC architectures as they increase in importance within the Computational Frontier.
- **Joint lab-university tenure track appointments**: We believe the DOE should seek to foster the continued development of intellectual leaders in theoretical calculation and simulation. The health of the field requires a similar cohort of individuals at the best universities, reflecting the intellectual vigor and potential of this area to contribute to DOE scientific goals. The creation of such positions can be stimulated by DOE-funded joint, five-year, tenure-track appointments. A good example might be the Jefferson Lab University Relations program of joint and bridged faculty appointments in nuclear physics.
- **Training**: Accessible training in both novel architectures and AI/ML with a low barrier to access for graduate students, postdoctoral researchers, and domain scientists is critical to ensuring the skills base exists for productive science in theoretical calculation and simulation. The use of cross-platform performance portable APIs should be encouraged to maximize return on investment in software. Hackathons and code examples are particularly useful.

Conclusions

- Thanks to all topical area contacts and white paper authors!
    - 18 white papers submitted directly to CompF2
    - ~6 more from other CompF areas referenced in the CompF2 report
    - Other reports from e.g. HSF, as well as numerous scholarly publications also cited
- We open the floor for discussion and feedback