
Names (Morning):

- Katy Huff
- Douglas Thain
- Kyle Chard
- Shawn McKee
- Amit Kumar
- Jeff Porter
- Peter Onyisi
- Mark Neubauer
- **Liz Sexton-Kennedy (**Pod A Leader**)**
- Nan Niu
- Danko Adrovic
- Thomas Hacker

S2I2 HEP/CS Workshop Questions

Please write your ideas here for discussion questions for the Thursday sessions. (Including your name is optional.)

What are examples of successful CS-HEP collaborations, and what properties have driven their success? (Small group examples as well as big collaborations.) +++++

- PhD student showed how to multi-thread GEANT4, professionals brought it to production quality.

How to align the CS research mechanisms (3 year grants, student developers, conference pubs) with the longer term needs of big science (30 year projects, production software, journal publications)? ++++

- Can the HEP side help in decomposing problems in “CS sized chunks” ++++
- Could an institute help by mapping FTE/professional effort to short-term projects?
- How could the research deliverables be better tied to sane software production timelines....
- How can the institute bridge the gap between “CS research output” and “working in production”? (Example: GEANT4 multi-threaded approach) +
- Can the needs of the HEP community be concisely be represented as a research agenda for the CS research community?

How to engage a broader slice of the CS community and make scientific computing more respectable within CS circles? (A commonly heard complaint in CS: scientific computing is a “niche" research area.)+

- What are the economic drivers for CS departments? Is it student training? +
- Can we align with industry? In terms of tools and techniques. +
- Can the institute have a training arm that fulfills a software-engineering-related training role (e.g. a massive expansion of Software Carpentry type material, Astro-hack week.)?
    - Would this solve both the skills issue in LHC/HEP as well as help students to be employable if they end up in CS industry instead? ++
- Is the tax (need to address non-interesting problems to get to the interesting ones) too high for CS people to get involved with scientific applications
    - Is there a way to lower the barrier to collaboration?

What CS technologies, techniques, and trends could the HEP community adopt, rather than doing everything internally? (Keeping in mind the long time scales and production needs of HEP.)+

- Ex: Amazon AWS “Batch” was just announced (still in beta), which could supplant existing batch facilities.
- Geoscience Dataspaces: software for data management and reproducibility of scientific research results.
- Could the institute survey artifacts out there, and evaluate them for suitability in HEP?+

As pointed out in Frank’s talk, there are detector differences that demand differences in the software implementation.

- Where can we / should we place that line below which contains the necessarily different implementations? And then above which could benefit from commonality.
- What is scientifically safe to share across experiments that are intended to be independent cross checks on each other?
- If there is a bug in a common component that affects a scientific result, is that too big a risk?
- E.g. there could be common systematics in generators or tracking software, but not likely in software to deliver data and launch workflows onto computer resources. Can the institute help determine the risk/reward proposition for different layers or of the software stack or domain problem?

The S2I2 will not be trying to solve all problems for HL-LHC. Rather, it will be laying out a set of software activities for US Institutions for which the US can play a leading role.

- What are the areas that the S2I2 should play a leading role in, informed by activities within the US HEP and CS communities?

Machine Learning is a very active area of HEP research.

- Is same true for US CS?
- Could ML be a focus area of the S2I2 where the US could play a leading role?

- What areas in LHC physics would benefit from ML?
- What training data and expert classified datasets are open and available for use by the CS community to develop and assess new ML approaches?
- “Simple” interchange formats?
- How can we deal with the issue that our training data may not accurately describe the real world?
- How could the CS community be incentivized to participate in HEP ML activities? What would be the reward for dedicated and sustained efforts from CS students and researchers?
-

The software and computing activities should be driven by the physics that will be done in the HL-LHC era, not the physics being done now. What we that be? Given that we just went through large increases in center-of-mass energy (7 -> 8 -> 13 TeV), we are looking mainly looking for new particle production and also Higgs in previously unobserved channels. If we find such new particles, we’ll be showered with money to follow up and maybe our S&C problems are lessened. But the more likely scenario is that we won’t find such particles and instead we’ll

- How does this change what we do in computing?
- Shouldn’t we just be trying to do these things anyway to be nimble?

Could HEP describe some long-term challenges that don’t need to be solved immediately, but that CS people could go off and think about? (D. Katz)+

What CS research challenges exist within HEP where CS researchers could contribute to HEP but also receive recognition for their work in the CS community? (D. Katz)

How could an HEP software institute facilitate interactions between the CS and HEP communities?

What are the incentives for such collaboration for HEP people? For CS people? For non-CS people? E.g. recognition, funding, publications, students, new problems to solve, new places to apply technologies, new solutions to current problems, pride in working on a global-scale problem.

How can we create “crystallization points”, shared artifacts that allow the encoding of tools and practices of the two communities and that can be improved over time? (Successful examples are wikipedia, linux kernel, docker registry)

- Along these lines DIANA HEP (in particular Kyle Cranmer and Lukas Heinrich) is working on some of this in the form of preserving analyses with use of Docker (c.f. RECAST). Though Docker has some problems with HPC envs(?) (M. Feickert) This

article has useful pointers to efforts making software containers viable for HPC (C. Maltzahn).
-

Re: data "privatization" in Frank's presentation (commentary here from R. Gardner):

- "Public" and "private" have different meanings in collaborations. In Frank's talk "public" meant datasets available collaboration-wide, e.g. public to the collaboration, and private meaning the end-stage datasets specific to an analysis and not necessarily registered in the experiment's official catalogs, even after publication (Frank correct me if this is wrong).
- The implication of this if true is that it prevents full reproducibility of a published result; there's a little "black hole" of data (and potentially software) between the collaboration datasets and the final plots and figures data.
- In the future we want "published" results to come with published data, and software, allowing for reproducibility by future analysts.
- How can CS help here? (many projects out there - are they addressing problems relevant to the scale and timeframe of HL-LHC?)
    - Check out the Popper convention -- this effort views reproducibility as a software engineering problem (the dev/ops community has already sophisticated tools to reproduce behaviors in a continually evolving software artifacts) and is partly funded by the Big Weather Web NSF SI2-SSI project. It's a convention, not a particular tool set (although tools need to be "scriptable"). So it should be applicable to a wide variety of domains. It's also scalable because it uses git for provenance and the git repositories include large resources by reference.
- (D. Katz: see https://mpsopendata.crc.nd.edu for some work in this area)

What role common data formats play in fostering collaboration with computer scientists. I.e. moving away from ROOT formats to open formats, those used e.g. in other data driven sciences? (R. Gardner)

How can CS help build frameworks/organization/processes that incubate software from the S2I2 into open source projects? Do organizations like AMPLab – UC Berkeley which build tools that have strong industry-coupling and support apply? How do we avoid building HEP unicorns, but technologies that are potentially of broad interest and with large, open development communities? (R. Gardner)

- Check out Center for Research in Open Source Software (CROSS) at UC Santa Cruz. The research project portfolio is currently skewed towards storage systems but the goal is to create a career path for Ph.D. students to become open-source software leaders. The membership agreement and the bylaws are strongly inspired by NSF's U/ICRC concept.

- Red Hat also provides great resources for open source in education.

What open source tools supported by industry can the HEP community use to solve its problems? Some good examples are OpenStack and LLVM (Spark, Tensorflow, also various commercial "AI as service" offerings, see IBM Watson for example), are there more out there? (L. Sexton-Kennedy)

Which of the many specialities in CS is most useful for HEP? (consider: machine learning, software engineering, computer vision, programming languages, networks, databases, complexity theory, robotics, human computer interaction, systems, architecture, ...) (N Ernst)

- This isn't an answer in full, but some examples of where applications of the above are currently being used in HEP (M. Feickert)
    - Machine Learning (DOI's and e-Prints): 10.1038/ncomms5308, arXiv:1609.00607, arXiv:1612.01551
    - Machine Learning and (some) Computer Vision: 10.1088/1748-0221/11/09/P09001, arXiv:1611.05531
    - Programming languages (incomplete, others please add): C++ (ATHENA, ROOT), Python (PyROOT, applications of scikit-learn)
- It may be helpful to look toward branches of physics, science, and engineering that have similar challenges to HEP, such as complex and enormous data, shared algorithms for data processing, large scale simulation validation needs, a tension between public and private data etc. (K. Huff)
    - These disciplines may be using CS-developed tools, workflows, etc. which might have worth in HEP.
    - Alternatively, perhaps these fields have learned lessons through software engineering / algorithmic / data management pitfalls that HEP might like to avoid.
    - I personally think of Astronomy (see: SDSS), Nuclear Engineering (see: evaluated nuclear reaction cross sections).

Is it fair to ask what level of involvement an institution what's to have in HL-HEP and CS-HEP (Amit K.)

What are the challenges of today's HEP software, and its adoption and scalability on emerging hardware or OS virtualization software that one has to think beyond those? What pieces of this software CS-HEP collaboration can be sliced for CS community to work on with a clear definition of expectations? (Amit K.)

Given the life cycle (for lack of a better term) of a researcher in HEP (transient employment… 5 years grad -> 3 years postdoc -> 3 or 5 year grant timelines etc.), what software workflows used in CS/SoftwareEngineering/Industry could reduce (even better, completely automate) the enormous burden of legacy software maintenance in HEP applications? Is greater adoption of

continuous integration sufficient, or is there a broader integration strategy that is necessary? (K.Huff)

Can an institute help incubate grassroots cross-collaboration projects? There are quite a number of small, 1-5 developer projects in HEP that are interesting but suffer from sustainability problems and likely overlap with other projects. What can we do to give interesting projects more visibility and avoid fragmentation?

Can we enumerate the ways in which HEP poses different CS problems than other scientific fields, and the ways in which we should join in broader development efforts across sciences?