January 19, 2017

Workshop 1:
Fostering Collaboration between HEP and Computer Science Communities

December 7-9, 2016
University of Illinois
National Center for Supercomputing Applications (NCSA)

Breakout Sessions Text (Full, Unedited)

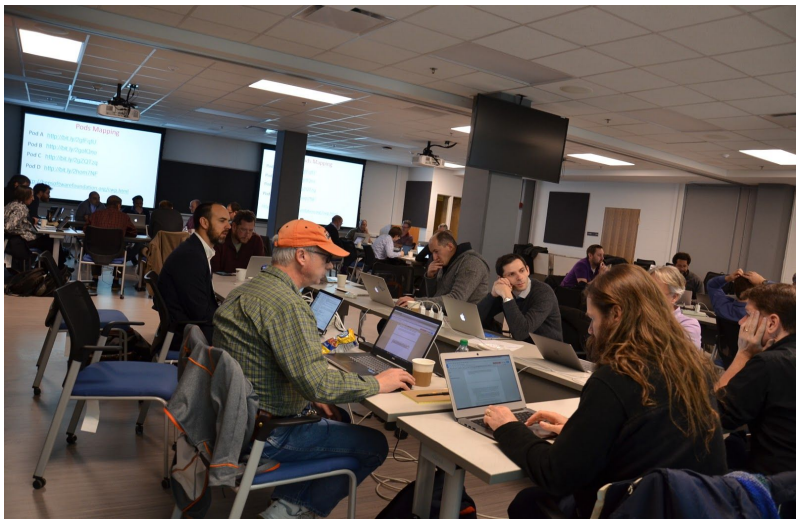

Table of Contents

Pod A 2
Pod B 10
Pod C 19
Pod D 30

Pod A

Names (Afternoon):

- Sergei Gleyzer
- Benedikt Riedel
- Matthew Feickert
- David Lange
- Matt Zhang
- Peter Onyisi
- Mark Neubauer
- Sandra Gesing
- Rob Gardner
- Sally Seidel
- Sumanth Mannam

HEP/CS Collaboration

1) ***How could we proceed to put together a document in the next 6 months summarizing HEP computing challenges in a language that CS people understand and map it to established discipline areas in CS?*** (useful for developing future synergistic and collaborative projects/relationships with CS faculty?)

- As part of the CWP process, each CWP area can put together a page of the challenges (not a part of the CWP document) targeting CS
- Invite CS to read and give feedback/Recruit some CS people that we can partner with during the CWP process
- Ensure CWP is in language understandable to CS
- Talk with CS experts here about mapping CWP areas to CS community
- Possible workshop?

Integrate into existing CWP process: 1) each area summarize challenges targeting CS audience 2) ensure language is understood by CS 3) get CS engaged early and get feedback

2) ***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 could an S2I2-HEP institute create the relevant incentives and promote HEP/CS research collaborations?***

- *CS incentives: large datasets and challenging practical problems, Ph.D. and master theses, internships, research visits, future funding opportunities for CS proposals (domain applications),*
- *HEP incentive: CS expertise and new ideas, access to people with different skillset and possible guidance (should we build what we think we want to build), CS perspectives on industry trends and tools which further HEP research*

- *funding for people/projects at the HEP-CS computing interface*
- *give credit for software work*
- *fellowships*

3) ***What can an S2I2-HEP institute do to create an environment of increased communication and awareness by individual HEP and CS researchers of each other's problems, expertise and research interests?***

- Create broader ties to the communities
- Steering committee with HEP and CS researchers
- Create exchanges: teach each other about what's going on
- Virtual seminar series & meetings?
- Hackathons
- Workshops
- CS Courses (program/curriculum) for HEP graduate students to take
- Send people to external conferences (CS, ML) and (data science, CS, ...) departments
- Host datasets
- Challenge problems

4) ***Will HEP have anything interesting to offer in 5-10 years for CS researchers?***

- Unique datasets
- Practical challenges (e.g., storage, distribution)
- Areas to apply new ideas (eg. ML...)

S2I2-HEP Scope

5) The S2I2-HEP will not be trying to solve all problems for HL-LHC or HEP for that matter. 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-HEP should play a leading role in, informed by activities and interests within the US HEP and US CS communities?***

- Machine Learning
- Distributed computing
- Data management and delivery
- Storage
- Languages

- Computer architectures
- Immersive technologies (i.e. VR)
- Outreach and education
- Training
- Fellowships

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?

Pod B

Afternoon session

Names:
Peter Elmer (Princeton University), Shawn McKee, Frank Wuerthwein, Dick Greenwood, Shantenu Jha, Justin Wozniak, Amit Kumar, Kaushik De, Adam Aurisano, Danko Adrovic

HEP/CS Collaboration

1) ***How could we proceed to put together a document in the next 6 months summarizing HEP computing challenges in a language that CS people understand and map it to established discipline areas in CS?*** (useful for developing future synergistic and collaborative projects/relationships with CS faculty?)

It is difficult to document the full set of HEP challenges. Striving for completeness in the next 6 months is too ambitious. Thus, we should focus on a subset of high impact challenges that have the following characteristics:

(i) there is interest in the US community to work on them
(ii) there is expertise and strength in the US community
(iii) there is potential for huge impact to the HL-LHC scientific goals, if solutions can be found that serve more than one experiment
(iv) we probably do not have to worry about matching HEP challenges to CS discipline areas in general. The CS community funded by ACI is probably already diverse enough for this.

Steps: 1) identify HEP challenges, 2) engage CS with "menu" to identify areas of common interest/strength, 3) workshop to create the document.

2) ***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 could an S2I2-HEP institute create the relevant incentives and promote HEP/CS research collaborations?***
The list in e.g. … is an excellent set of incentives. Hard to do better than what is already listed here. An example missing in the e.g. … above is "credit for software and/or data products".

The fellowship model as implemented in MolSSI makes sense for an S2I2-HEP as long as high standards of quality and relevance to the scope of the institute are maintained. The supervision that the students and/or postdocs receive is an important factor for success. It's a big responsibility for the "selection committee" that selects the fellows. In the MolSSI this is planned by attaching 10% of a staff software engineer to each and every fellow.

In addition, the ultimate success (and eventual impact) of a fellowship program will depend on the fellows visibly advancing in their careers (both in HEP and outside of HEP) through the work/research they did as a fellow. Evaluation of this success will only be possible after a latency of a few years, of course.

3) ***What can an S2I2-HEP institute do to create an environment of increased communication and awareness by individual HEP and CS researchers of each other's problems, expertise and research interests?***

The very existence of such an institute will provide a focal point for joint work across experiments in HEP and individuals in CS for the small set of topics chosen to be within the scope of the institute.

4) ***Will HEP have anything interesting to offer in 5-10 years for CS researchers? What?***

Yes. Throughout the HL-LHC period there will be a continued string of challenges in part due to changes in technology and in part due to increases in scale and in part due to operational challenges posed by the reality of the running program.

S2I2-HEP Scope

5) The S2I2-HEP will not be trying to solve all problems for HL-LHC or HEP for that matter. 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-HEP should play a leading role in, informed by activities and interests within the US HEP and US CS communities?***

If we go by "interest, strength, and impact" then we can arrive at some guidance to the scope:

US HEP strengths & interest:
ATLAS: workload management & distributed computing, remote data access, new algorithms (incl. but not limited to machine learning) on new archs, VR
CMS: data management, remote data access, new algorithms (incl. but not limited to machine learning) on new archs,
CS strengths and interests:
High-performance and/or high throughput distributed computing, data science, data transfer and management software, workflow management software

==========

Morning session:

Names:
Peter Elmer (Princeton University)
Sergei Gleyzer (University of Florida)
Fkw (ucsd)
LATBauerdick (Fermilab)
David Lesny (University of Illinois)
Rob Gardner (UChicago)
Sandra Gesing (U of Notre Dame)
Neil Ernst (SEI)
Jim Pivarski (Princeton University)
John towns (UIUC)

S2I2 HEP/CS Workshop Questions

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

High Priority Question

How to put together a document summarizing HEP computing challenges in a language that CS people understand and map it to established discipline areas in CS? (useful for developing future synergistic and collaborative projects/relationships with CS faculty?)

Collaboration Questions:

Summary question:

What are examples of successful CS-HEP collaborations, and what properties have driven their success?

- Local inter-department collaborations with alignment of interests (UC Irvine for ex. In machine learning)
- Productive CS-HEP collaboration on infrastructure software has the following characteristics:
    - Long term engagement. HEP runs infrastructure typically much longer than grant timescales. So the CS teams must be committed and have the means to sustain their products for a decade and more.

    - Strong track record of the CS team to work with HEP on solving operational problems we have with the CS software. Don't just throw software over the fence and walk away.
    - HEP commits and then follows through and uses the software that we agreed we want, and not just make CS folks do work that then gets dumped. Don't just throw requirements over the fence and walk away.
    - HEP needs to be committed to deploy, measure performance, and not reject outright the moment goals aren't quite met, but rather commit to work with the CS team(s) to iteratively improve their artefacts. HEP has a tendency to use initial failure as pretext to reject and build ourselves instead.

Are there examples of CS/domain science collaborations that have worked from which we can learn some lessons to apply to this context? (John Towns)

- Some folks at meeting can speak to general relativity problems pursued in the past
- Domain science provided interesting problems used as CS PhD topics that produced methods and algorithms that also benefitted the advancement of the domain science

How does HEP best present their problems to CS? (S. Gleyzer)
What is the right level of abstraction and how to reach the right audience? (Lothar)
How identify problems that are unique to HEP and those that aren't and can be solved more generally (Neil)

How can HEP contribute to CS (two-way collaboration) ? (S. Gleyzer)

- E.g. Are today's globally distributed systems of HEP big and complex enough to be interesting systems worth studying for CS? (fkw - I think HEP would benefit from CS people analyzing what we do) I.e. can data (accounting, job submission, data transfers, network performance, application performance, …) about our systems be of interest to CS? (non-CS person thoughts: generally these are not very interesting and engage CS community at the wrong point. Suspect CS community would be more interested in involvement in defining these things initially as opposed to observing how they do or do not work. This also misses involvement in e.g. the development of the fundamental algorithms for analysis)
- Can some of the solutions to HEP problems be more broadly useful (S. Gleyzer)
    - The concept of "overlay batch system" as implemented in Panda, Dirac, gWMS has been very widely adopted across all of science. In some cases, both the concept and the product are being used outside the experiment it originated in.
    - Geant, Root, Fluka, (what else?) has been widely used across HEP, Astro, NP, Medical physics, ...
    - WLCG created a globally distributed infrastructure that is starting to be useful to IceCube, LIGO, Nova, Xenon1T, Belle, …. I.e. other international science

collaborations that have the problem of their member institutions wanting to contribute resources to the common good of the collaboration.
    - LHCOne as a global networking infrastructure is being joined now by non-LHC experiments in order to serve their global data distribution needs.
    - Rucio has been adopted by Xenon1T as data management system
    - Cvmfs is being used widely across many sciences. In some cases, it's used for software distribution for large international experiments (e.g. Ligo), in other cases it is used for distributing applications via the modules environment (http://modules.sourceforge.net)
    - HEP people have contributed to a variety of open source projects that have originated outside of HEP, and are predominantly used outside of HEP (e.g. HDFS, … what else …?)
    - HEP people have contributed to commercial product development (Western Digital firmware bug in the early 2000's, … what else … ? Are there serious examples ?)

HEP and emerging fields of Data Science - seems to be a growth area in computer science departments? Are HEP problems of large scale data acquisition, storage, access, quality assurance and analysis of interest to CS? Note, e.g. http://cra.org/data-science/ : "From a computational point of view, **very large data volumes, very high data rates, and very large numbers of users, demand new systems and new algorithms.** New system architectures that can accommodate the heterogeneity and irregular structure in data access and communication are needed. " [R.G.]

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.)

What are the next steps after the workshop we can all contribute to so that we foster the collaboration between HEP and CS? (Sandra Gesing)

Software Institute

Summary question: What is a useful productive structure for S2I2-HEP institute?

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

What is a useful structure for the S2I2-HEP institute? (S. Gleyzer)

What kind of task would a HEP Software Institute take on, on what kind of time scales (short-term initial, mid-term etc)

Data and Knowledge Exchange

Summary question: What is a useful data and knowledge exchange model between HEP and CS?

How can HEP be more exposed to up-to-date CS ideas, technologies and tools (S. Gleyzer)

How can HEP become a data-repository to be shared with CS?
How hard/how much work will it take to create a set of standard HEP datasets for replication with ML, systems, etc.? (similar to R dataframes like sepal width or Netflix movie rating)

How to best educate young (HEP) analysts in CS (S. Gleyzer)

How to provide career path for people working interdisciplinary in CS and HEP (Sandra Gesing)

How can HEP and CS support the Open-Source community (S. Gleyzer)

Original Questions

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)

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.)

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).

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)? 1-year conference paper cycles, large number of HEP authors on papers

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)

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.

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 - fkw: yes, this is what I meant).
    - This is wrong - public here means really public, released externally sometimes with the software needed to run over it (S. Gleyzer) opendata.cern.ch
        - Fkw: in my slides public meant public to the collaboration. I did not address the question of public to the rest of the world. That's a much more complicated problem that I tried to avoid.
- 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.
    - Fkw: there is and has always been this "black hole". We deal with it by having two independent teams do the same and confirm each other for any high profile analysis. To be very clear, it is inconceivable that CMS (or ATLAS) would ever claim a discovery of anything without multiple independent teams taking the data and arriving at the same results.
- In the future we want "published" results to come with published data, and software, allowing for reproducibility by future analysts.
    - Fkw: what does this mean? What about the HLT? The L1 trigger? What does "data" mean here? What does "software" mean here? What does "reproducibility" mean here? By whom, and for what purpose? Does the public benefit from

Petabytes of RAW data from the LHC? Is there any agency on the planet prepared to fund the curation of those Petabytes, incl. all the necessary calibrations and software (reconstruction and simulation) and documentation?
        - Corrollary: even Astronomy that has a long tradition of making data public do not in general make all the RAW data public. There is a conscious choice being made what level data products are useful in the public domain. And that fundamentally limits the meaning of reproducibility. The data that is made public can generally not be reproduced from the RAW data, which is generally not public. See e.g. plans for LSST, or practice by Fermi LAT, or ...
- 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)

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.)

Pod C

Mike Sokoloff
Marc Paterno
Myron Livny
Jim Pivarski
Kyle Chard
Thomas Hacker
Aaron Elliott (aaron@aegisresearchlabs.com)
Robert Kalescky
Jim Kowalkowski

HEP/CS Collaboration

1) ***How could we proceed put together a document in the next 6 months summarizing HEP computing challenges in a language that non-HEP (CS, and more) people understand and map it to established discipline areas in CS?*** (useful for developing future synergistic and collaborative projects/relationships with CS faculty?)

- Articulate grand challenge science problems (domain science)
- Identify difficult computing problems that need answers from CS (for example)
  - more flops, more computation → distributed → must rethink domain-specific code
  - lower bandwidth: "thin" the data, lossy but determine what parts of the data are more important than others (again, domain-specific)
  - hardware budget and power budget
- Form a group of computer scientists/computational scientists to distill a computing research agenda from the HEP computing problems. [integrate computer scientists into the process explicitly.]
- Make sure it's an iterative process.

2) ***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 could an S2I2-HEP institute create the relevant incentives and promote HEP/CS research collaborations?***

Incentives can be constructed and used to encourage the types of behavior deemed to be useful or productive for HEP efforts.

Who are the CS communities? Faculty, researchers, students, software engineers, operations staff, etc. Each of these groups would be motivated by different incentives. The incentives for software engineering and faculty are likely to be very different.

Who are the HEP people?

The sociological environment and incentives for multidisciplinary efforts need to be well understood and carefully addressed to increase the chances of success.

3) ***What can an S2I2-HEP institute do to create an environment of increased communication and awareness by individual HEP and CS researchers of each other's problems, expertise and research interests?***

A. serve as a matchmaker; respected and trusted expertise + source of seed funding, not welfare
B. Innovative multidisciplinary efforts require depth and sustained commitment from all of the disciplines engaged in the collaborative effort.
C. The sociological elements of trust, respect, affirmative acknowledgement of mutual and differing disciplinary needs, and frequent communication should be deliberately and specifically addressed and managed as sociological "investments" that help to accelerate productivity and reduce potential misunderstandings and friction.
D. Generate a glossary of terms. For example, "framework" has had at least three different meanings in today's discussions.

4) ***Will HEP have anything interesting to offer in 5-10 years for CS researchers?***

- Unique environment in scale for distributed computing; publicly releasing data on how data is accessed, distributed, etc. could be of interest to the CS community.
- FPGAs for "in-flight" analysis
- Provide access to software development process for "anthropological" studies.
- "Industry scale, academic openness™."

S2I2-HEP Scope

5) The S2I2-HEP will not be trying to solve all problems for HL-LHC or HEP for that matter. 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-HEP should play a leading role in, informed by activities and interests within the US HEP and US CS communities?***

==========

Morning session:

Names:
Matthew Feickert
Amir Farbin
David Lange
Daniel S. Katz
A. Aurisano
Jim Kowalkowski
Justin M. Wozniak
P. Calafiura
Sally Seidel
Sumanth Mannam
Ilija Vukotic
Carlos Maltzahn

S2I2 HEP/CS Workshop Questions

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

Is novelty necessary for computer scientists to work with HEP or can it be just an engineering collaboration?

What are the distinct software domains/communities in HEP? (e.g. Distributed Computing, Core Computing, Tracking, Calorimetry, Machine Learning, Analysis) What are their problems? Can we describe this problems in a way that can be comprehended by and is appealing to CSist?

What software components will have to be retained? What components can be rewritten from scratch? How do we decide? When do decisions need to be made? Are there different types of software for which decisions need to be made at different times?

Should we formally train HEP students in software engineering practices (Agile, Scrum)?

- (see https://www.youtube.com/watch?v=oyLBGkS5ICk for an interesting very recent talk and thoughts about how large collaborations and software dependencies can work better than they do today)

Should we run HEP software development like a software company?

How much should we rely on industry offerings?

Should we make a distinction between CS/HEP collaboration aimed at developing something novel, versus collaboration aimed at engineering a solution to a problem?

How does HEP approach CS professionals vs CS researchers? HEP probably needs to work with both, but both are looking for different things from the collaboration (e.g. and simplistically, salary vs papers)

Are there fundamentally different ideas that could completely change how HEP software is developed, that would involved bringing together relatively new or non-well-known CS techniques into HEP? (for example, automated code generation from physics algorithms that would be provable correct and future-proof)

How do the Scientists and Engineers work on gathering requirements?
Does the innovations in CS influence the HEP innovations? (can it generate new HEP problems)?
How to encourage CS people to work for HEP problems?
Is there any agile methodology to work across different contents and timezones?

There is a lot of domain-specific vocabulary and language used within the HEP and CS communities. How will they come together to understand each other and what the needs actually are?

What are examples of successful CS-HEP collaborations, and what properties have driven their success?

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)?

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 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.)

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) (C. Maltzahn)

- 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. (C. Maltzahn)
- (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. (C. Maltzahn)
- Red Hat also provides great resources for open source in education. (C. Maltzahn)

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)

What are software domains/communities in HEP? E.g. Distributed Computing,

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)

Do we have a software architecture for the HL-LHC software? If not, how will we get to one? (D. Katz)

What would such an HL-LHC software architecture cover? Everything? Facilities, Operations, physics (sim, production, analysis)?

What is trustworthy software?
How to balance trust and security? (C. Maltzahn)

How do you maintain trust in software in a large organization without being bogged down? Do you rely on communities (“eyeballs”), formal methods (e.g. automatic provers), reuse mechanisms such as (certified) software containers? Who is trusted for what? What does “adult supervision” look like?

What role can software containers play in HEP software? Can they help us interface with external software - particularly new machine learning systems?

What are the venues for communication between the CS and HEP communities? (Workshops, tutorials, etc.)

Are there different levels of interactions depending on what part of the overall computing problem is being tackled? When is direct interaction with physicist-researchers needed? When is interactions with laboratory operations most important? When is interactions with software development staff at laboratories most interesting?

How does the HEP community adapt new concepts and changes, given the goals to produce physics, especially when experiments are in operations? Is it always a long-term process? How does that relate to mechanisms of reuse and leveraging the work of communities outside the HEP community?

There is big gap between downloading software and successful deployment of software -- what mechanisms should the HEP community use to bridge this gap, using automation and reuse? (C. Maltzahn)

How do we achieve consensus on software system (or product) changes across organizations for things that are shared, especially when requirements (including here organizational constraints) diverge or differ? Is consensus necessary?

Should the HEP community connect with the DevOps community? (C. Maltzahn)

The gap between practical software development / engineering (needed by the physicist) and the computer science research that required means to be a problem. How is this gap narrowed? In others words, how is the CS research coupled to applied software needs of an experiment in HEP?

What topics are useful research topics in both HEP and CS?

How is HEP software development investment measured (what is the metric)?

How do we get datasets to open communities so they can participate in challenges on regular basis in order to rapidly grow knowledge bases of new technologies or techniques?

How do you assess the value of software used for HEP research? How do computer scientists, software engineers, IT experts, and professional consultants help increase returns on that value or decrease costs?

What are the relevant computer science domains and what are their respective incentives to collaborate to bring value?

How do you leverage industry for the application challenges in which CS theorists do not find research value, but the HEP community could benefit from deeper knowledge bases?

Many computer scientists are not interested in application, what scientific value do highly generalized (and successful) machine learning algorithms such as Deep Learning or Random Forest have to computer scientific researchers?

Pod D

Afternoon Session

Names
Neil Ernst (SEI)
Paolo Calafiura
Nan Niu (Univ. of Cincinnati)
Douglas Thain (ND)
Henry Schreiner
David Lesny (UIUC)
Carlos Maltzahn (UC Santa Cruz)
Liz Sexton-Kennedy (FNAL)

1) *How could we proceed put together a document in the next 6 months summarizing HEP computing challenges in a language that CS people understand and map it to established discipline areas in CS?* (useful for developing future synergistic and collaborative projects/relationships with CS faculty?)

There is a CS language. There has to be a person from each domain working on the plan / document together. Paolo says the CS person has to write it. But then the CS person might not understand the domain problem. Can the HEP reader of the CS written white paper understand it? There is a problem if you have one computer scientist you need different perspectives of different types.

Would like to start from a list of problems together with their priorities. This would help create a mapping between the problem and the domain of CS that is most relevant.

The DevOps community may be more relevant for some problems which are not research topics of interest for CS.

Adding a link to a software engineering workshop for sciences
http://se4science.org/workshops/se4science17/

2) *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 could an S2I2-HEP institute create the relevant incentives and promote HEP/CS research collaborations?***

HEP can offer an interesting test-bed - for example, in storage systems of 150PB and 2 orders of magnitude increase planned. Could take a fraction of this and do something experimental for CS. Instead of having to be in Google.

A huge benefit is to have connections to institutions that recruit members of one community to work with the other. Small cross disciplinary collaborations can train students that become more of an expert in solving a unique problem that they become more expert in then the original advisors. That student has to do well in both disciplines.

There are some CS people that like contributing to big "cool" science. CS people were not aware that you can be a highly functional experimental HEP practitioner that writes code all day long.

3) ***What can an S2I2-HEP institute do to create an environment of increased communication and awareness by individual HEP and CS researchers of each other's problems, expertise and research interests?***

Short term fellowships are an obvious suggestion. One year long sabbaticals for CS profs.
Curating an overlay journal.
Foster careers for people who do computational (software) science
Edit/continue the Big Data Science Springer journal (HSF)
Increase visibility of "CHEP" - Computing in High Energy Physics conference (counter-productive?)
Workshops
Match making institutions like openlab do really work.
In exploring working with open source software project you join a group that the students can grow into and even become leaders of. Then they are employable by the company that sponsors that open source software.

Would the institute fund the project? No it would fund the student.

Is the embedded engineer the right model? This has been successful in the past.

4) *Will HEP have anything interesting to offer in 5-10 years for CS researchers?*

The needs of HEP are different than the needs of industry, and that will be interesting.
Namespaces are an example of something HEP knew that it needed a while ago and now industry is realizing it needs them too.
The cloud computing will start to move to addressing HPC problems.
The answer is yes, just the scale is interesting. HEP will probably be in the forefront of using new devices
Will HEP students 5-10 years for now still be coding all of the time?
Some believe we will not, some only in reconstruction, not analysis. Some think this is crazy.
The multi-exabyte data stores will require innovation in storage systems, and potentially in the way we organize our hierarchical data store.

How do we find the interesting work of others that might be interesting to us?

S2I2-HEP Scope

5) The S2I2-HEP will not be trying to solve all problems for HL-LHC or HEP for that matter. 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-HEP should play a leading role in, informed by activities and interests within the US HEP and US CS communities?*

Doesn't the US have leadership in all of HEP computing?
DevOps community has good experience that can give us flexibility.

==========

Morning Session

Names:
Robert Kalescky
Marc Paterno
Henry Schreiner
Jeff Carver
Mike Sokoloff
Miron Livny(ML)
Matt Zhang (MZ)
Benedikt Riedel (BR)
Dick Greenwood
Kaushik De (KD)
Don Petravick.

S2I2 HEP/CS Workshop Questions

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

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)? How do align such collaboration with the life-cycle of Phd thesis in CS? (*ML*)
(is it not substantially about networks of people and incentives -- DLP)

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)

- What makes a format “open”?
  - Should be independent of a framework
  - ROOT is very HEP-centric (BR)
  - HDF5 (already used in some experiments for higher level data), netCDF (popular in meteorology), zipped BSON, boost archive formats, etc. are used across disciplines and in industry (BR)

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)

How do we reward grad students/post-docs/more senior physicists for writing good (sustainable) code and for cleaning up the existing code base (M. Sokoloff)?

Some tools are common, others are specific. How do we identify the common set of tools for the HL-LHC for us to focus on while keeping requirements late (just-in-time/agile)? (KD)

- Discussion - do we need incentives, to promote better/common code development?
- Discussion - maybe we should let good ideas rise to the top? The best ideas and practises should be encouraged, and they will win.

How are the Software Engineering needs in HEP distinct from the Computer Science needs? (JC)

What are examples of successful CS-HEP collaborations, and what properties have driven their success? How do we measure success of such a collaboration? (*ML*)

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 can a software institute provide "consulting services" to help experiments review their architectural decisions, the state of their legacy code, their documentation and code review processes, etc. in the context of "best practices" identified by the computer science/software engineering community (M. Sokoloff)?

How can we make sure code is modular and well-documented enough, where a new developer (grad student), when given a specific task to work on, can immediately make useful additions without having to understand the entire code framework? (MZ)

- What if we add "And without further complicating the software" to this question

How do we formulate the HEP challenges in terms of CS principles? (*ML*)

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)? How do align such collaboration with the life-cycle of Phd thesis in CS? (*ML*)

How specifically is the current software and its development strategy deficient? There has been discussion of new data and compute time requirements and broad estimates, however a better understanding of these requirements and the compute capabilities that will be available may help to guide the direction that solutions can take in fulfilling those requirements. (R. Kalescky)

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)

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)

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 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.)

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. (n.b HEP typically runs two experiments that probe the same phenomena, is this not a kind of substitute for the topic)
- 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)

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)

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.)

There are many different types of HEP software. How do our issues differ, depending on whether we are talking about "infrastructure code" (e.g. event-processing frameworks) or

“physics code” (e.g. the implementation of a tracking algorithm) versus “analysis code” (often ntuple analysis)? How does “offline” differ from “online”? How do we deal with different timescales for “life” of such software? (M. Paterno)

Can documentation and training be included as a component that can be enforced by a design, review, and/or reward system, to lower the barrier for those that are not as familiar with the specific software? (H. Schreiner)

Can the software stack be modularized (libraries instead of monolithic tools, e.g. LLVM’s strategy) so that multiple groups can pick and choose what to use? The institute could promote and reward efforts in this direction. (R. Kalescky)

Can the institute promote standards (data, libraries, etc.)? Either specific to HEP or otherwise. (R. Kalescky)