==========

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.)