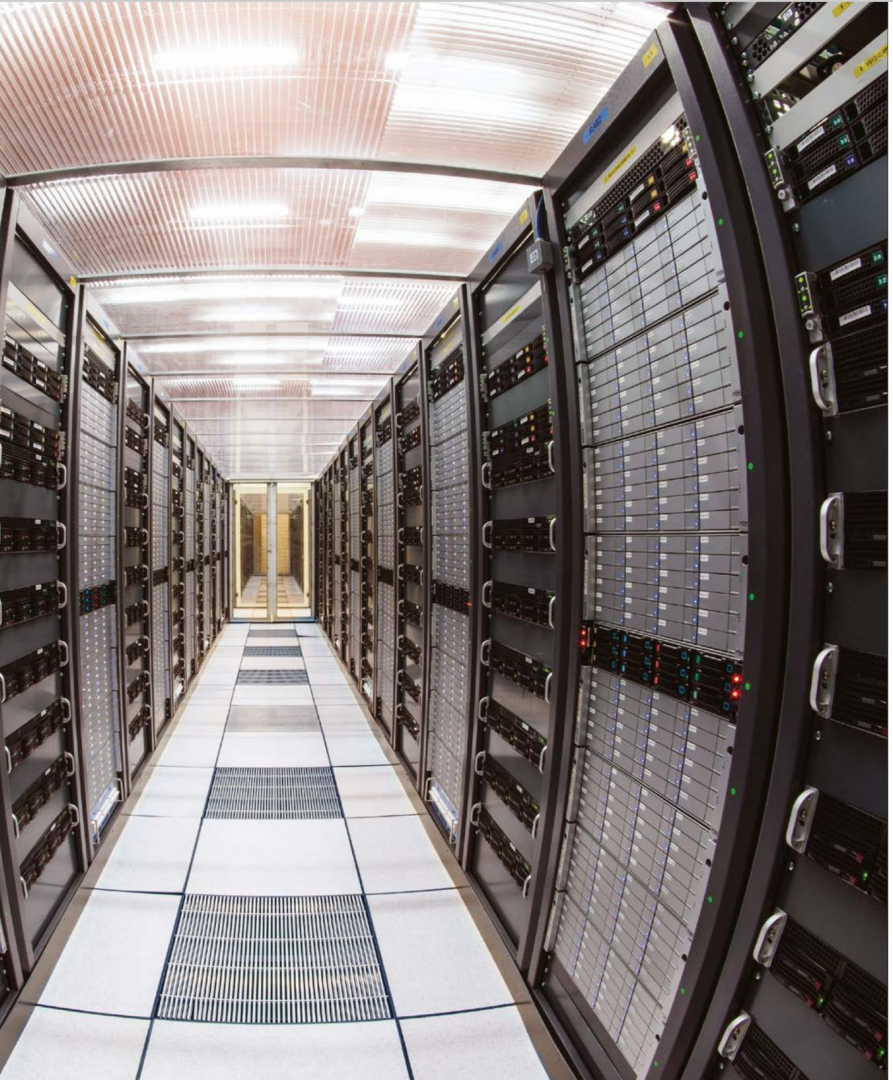
HEP Software Foundation: Update on R&D and Activities

Graeme Stewart, CERN EP-SFT
for HSF Coordination

WLCG Grid Deployment Board

2018-07-18



Software and computing

Time to adapt for big data

Radical changes in computing and software are required to ensure the success of the LHC and other high-energy physics experiments into the 2020s, argues a new report.

It would be impossible for anyone to conceive of carrying out a particle-physics experiment today without the use of computers and software. Since the 1960s, high-energy physicists have pioneered the use of computers for data acquisition, simulation and analysis. This hasn't just accelerated progress in the field, but driven computing technology generally – from the development of the World Wide Web at CERN to the massive distributed resources of the Worldwide LHC Computing Grid (WLCG) that supports the LHC experiments. For many years these developments and the increasing complexity of data analysis rode a wave of hardware improvements that saw computers get faster every year. However, those blissful days of relying on Moore's law are now well behind us (see panel overleaf), and this has major ramifications for our field.

The high-luminosity upgrade of the LHC (HL-LHC), due to enter operation in the mid-2020s, will push the frontiers of accelerator and detector technology, bringing enormous challenges to software and computing (*CERN Courier* October 2017 p5). The scale of the HL-LHC data challenge is staggering: the machine will collect almost 25 times more data than the LHC has produced up to now, and the total LHC dataset (which already stands at almost 1 exabyte) will grow many times larger. If the LHC's ATLAS and CMS experiments project their current computing models to Run 4 of the LHC in 2026, the CPU and disk space required will jump by between a factor of 20 to 40 (figures 1 and 2).

Even with optimistic projections of technological improvements there would be a huge shortfall in computing resources. The WLCG hardware budget is already around 100 million Swiss francs per year and, given the changing nature of computing hardware and slowing technological gains, it is out of the question to simply throw

Inside the CERN computer centre in 2017. (Image credit: J Ordan/CERN.)

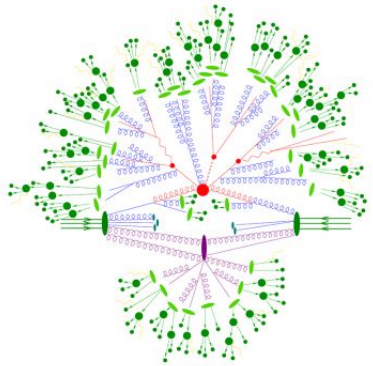
CWP Roadmap

- 13 chapters, 310 authors, 123 institutes, 1712.06982
  - Thank you!
- Article published in April CERN Courier
- Publication in Computing and Software for Big Science in progress
- Notable presentations:
  - CERN Scientific Computing Forum
  - LHCC
  - CHEP Plenary
  - European Committee on Future Accelerators

Individual CWP Papers

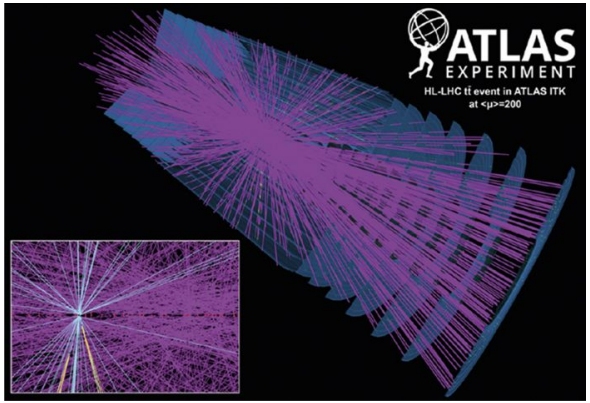
- Progress in getting these finalised and published to arXiv
  - Careers and Training, https://arxiv.org/abs/1807.02875
  - Data Analysis and Interpretation, https://arxiv.org/abs/1804.03983
  - Detector Simulation, https://arxiv.org/abs/1803.04165
  - Machine Learning, https://arxiv.org/abs/1807.02876
  - Software Development, Deployment and Validation, https://arxiv.org/abs/1712.07959
  - Software Trigger and Event Reconstruction, https://arxiv.org/abs/1802.08640 (summary) and https://arxiv.org/abs/1802.08638 (full paper)
- We do still anticipate that there will be arXiv publications for Conditions Data, DOMA, Data and Software Preservation, Frameworks, Facilities and Distributed Computing, Visualisation
  - Check https://hepsoftwarefoundation.org/organization/cwp.html for status

Physics Event Generators



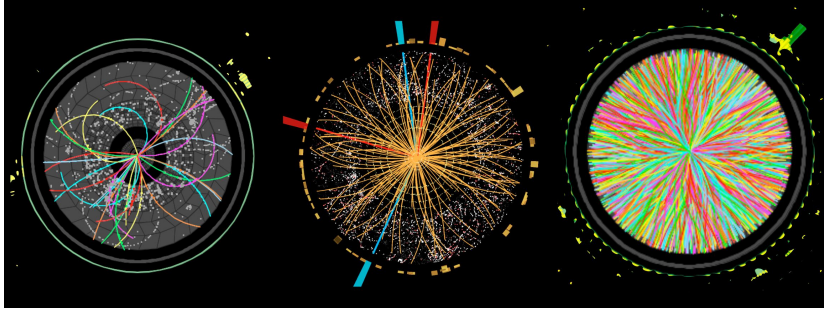
- Physics event generation simulates the physics of underlying Events at the heart of data / monte-carlo comparisons
  - Depending on the precision requested, CPU for event generation ranges from modest to huge
  - At Next-to-Leading Order (NLO) precision used today, CPU consumption can become important
  - Study of rare processes at the HL-LHC will require the more demanding NNLO for more analyses
- Primary contributions from the theory community
  - Need expert help and long term associations to achieve code optimisation
  - Even basic multi-thread safety is problematic for many older, but still heavily used, generators
  - Ongoing maintenance of tools like HepMC, LHAPDF, Rivet is required (with recognition)
- Contact with a number of the key people in the theory community to organise a workshop to address these issues
  - Re-engineering workshop planned - set out main problems and plan for technical improvements

Detector Simulation



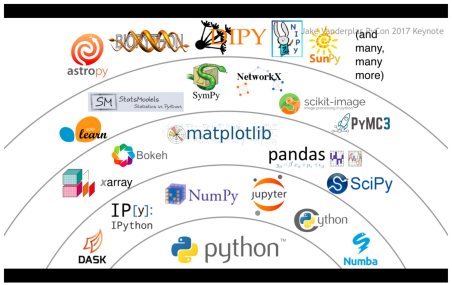
- **Simulating our detectors consumes huge resources**
  - However, this is a recognised critical activity for LHC success
  - Further gains needed for HL-LHC and intensity frontier experiments in particular
- **Main R&D topics**
  - Improved physics models for higher precision at higher energies (HL-LHC and then FCC)
  - Adapting to new computing architectures
    - Can a vectorised transport engine actually work in a realistic prototype (GeantV early releases)? How painful would evolution be (re-integration into Geant4)?
    - CMS testing the GeantV alpha release now
  - Faster simulation - develop a common toolkit for tuning and validation of fast simulation
    - How can we best use Machine Learning profitably here? Multi-level approach, from *processes* to *entire events*
  - Geometry modelling
    - Easier modelling of complex detectors, targeting new computing architectures

Software Trigger and Event Reconstruction



- **Move to software triggers is already a key part of the program for LHCb and ALICE already in Run 3**
  - ‘Real time analysis’ increases signal rates and can make computing more efficient storage and CPU
- **Main R&D topics**
  - Controlling charged particle tracking resource consumption and maintaining performance
    - Do current algorithms’ physics output hold up at pile-up of 200 (or 1000) and maintain low p_T sensitivity?
    - High granularity calorimeters bring a new type of detector into the tracking domain
  - Detector design itself has a big impact (e.g., timing detectors, track triggers)
  - Improved use of new computing architectures
    - Multi-threaded and vectorised CPU code, use of GPGPUs and possibly FPGAs
  - Robust validation techniques when information will be discarded
    - Using modern continuous integration, multiple architectures with reasonable turnaround times
  - Reconstruction toolkits can help adapt to experiment specificities: ACTS, TrickTrack, Matriplex
    - Ideas and concepts can certainly be shared, code is a greater challenge

Data Analysis and Interpretation

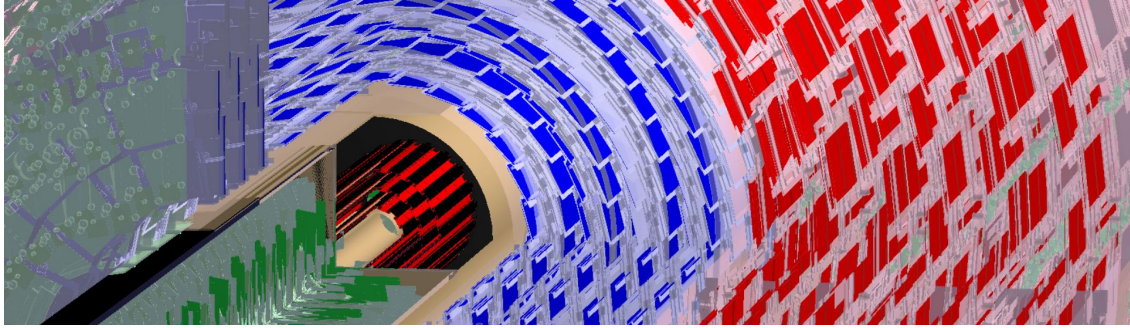


- **Today we are dominated by many cycles of data reduction**
  - Aim is to reduce the input to an analysis down to a manageable quantity that can be cycled over quickly on ~laptop scale resources
  - Key metric is ‘time to insight’
- **Main R&D topics**
  - How to use the latest techniques in data analysis that come from outside HEP?
    - Particularly from the Machine Learning and Data Science domains
    - Need ways to seamlessly interoperate between their data formats and ROOT
      - Python is the *lingua franca* here, thus guaranteeing our python/C++ bindings is critical
    - Functional / declarative expressions of analysis are more robust than imperative
  - New Analysis Facilities
    - Skimming/slimming cycles consume large resources and can be inefficient
      - Grid resources not usually optimised for high I/O load analysis jobs
      - Dedicated analysis clusters may do better
    - Can interactive data analysis clusters be set up? SWAN, Spark, Dask interesting
      - Characterised by rapid column-wise access reads, with writes of new columns

New Working Groups

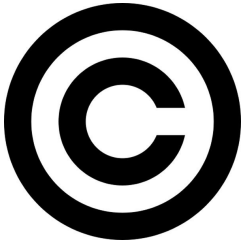
- Intention to form working groups in these three key areas of HEP software:
  - Simulation
  - Reconstruction
  - Analysis
- Building on work in the CWP
- Raise awareness of work being done in these areas in HEP
  - Not all projects are as known as they should be
  - New projects can begin with a broad scope and common goals
- These will be areas reviewed by the LHCC next year
  - These groups will be able to answer the charge of whether we really have learned to work together or not
- *We have been in touch with experiments, but in the ‘do-ocracy’ spirit, please let us know if you’re interested!*

Software Forum



- HSF has relaunched the Software Forum
  - Meetings that can
    - Showcase common software projects
    - Introduce tools that help us face challenges like concurrency or vectorisation
    - Open dialogue with other like-minded communities
- First meeting looked at the DD4hep geometry modeling package
  - Adopted by CLIC, FCC and now CMS; LHCb very interested
- Today's meeting (18 July) will look at VecCore and SOAContainer
  - Common libraries addressing vectorisation and data layout
- Presentations on HEP analysis in the Python ecosystem and Tracking software scheduled for after summer
- We would be very happy to hear suggestions for other topics
  - Please use this as a way to connect to the wider HEP software community and different experiments

Copyright and Licensing



- We continue to work in this much neglected area in HEP software
  - Much code exists with no clear copyright or licence
  - The issues of large and deep stacks of experiments' software and license combinations were often neglected up to now
    - *Does impact on our ability to collaborate*
- Experiments moving to be more open with their software
  - Goal is to maximise our useful user base and interactions with others
  - Significant progress in opening software: CMS, LHCb, ALICE; ATLAS in progress; Belle II discussing
- GPL licenses have become disfavoured as they place obligations on users and can inhibit collaboration (e.g., industrial)
  - ATLAS and CMS want non-GPL licenses
    - Matches shifts at CERN, e.g., Indico moving from GPL to MIT
  - We made significant progress in moving packages like HepMC and DD4hep to *LGPL*
  - Widespread use of GPL by theory community still affects us greatly (Fastjet in particular)

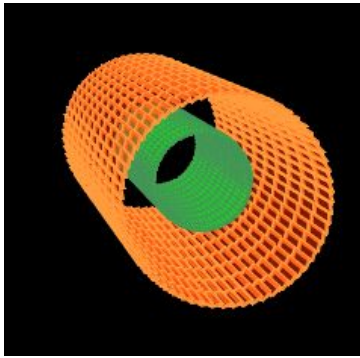
Packaging

- Packaging is one of the de facto areas of common interest between exp
  - Building and deploying our software is a significant task and there is much duplicated effort
- HSF Packaging Group decided to formalise the problem we are trying to solve
  - Write down the actual use cases we have
  - Recognise that CVMFS and Containers simplified the problem a lot for us
    - Use cases can be enabled or become redundant as technology develops
    - We should be independent of site installed base OS
- Key drivers for the future:
  - Containers for deploying work at sites
  - CVMFS in *most* places (with work arounds where it's not)
- R&D Projects looking at some of the directions for the future
  - Nix - pure functional package manager, build everything (really, even `libc`)
  - Portage - from the Gentoo Linux distribution, prefix distribution isolated from system, consistent harmony with `RPATH`
  - Spack - from LLNL, widely used scientific build orchestrator, very multi-version friendly

Other Working Groups

Software Tools Working Group

- Meeting on performance analysis software and how to share data
  - Common work on warehousing and visualisation possible
- Will also look at static analysers and grid tools



Visualisation

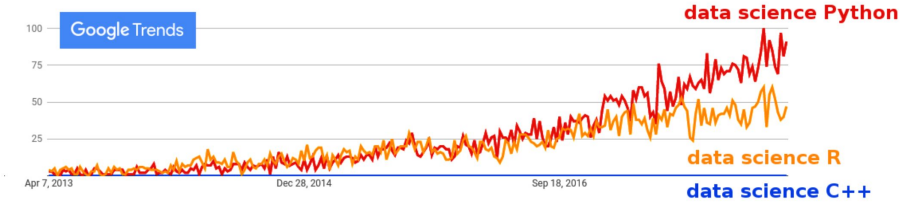
- Ed Moyse's WebGL event display now an HSF project (Phoenix)

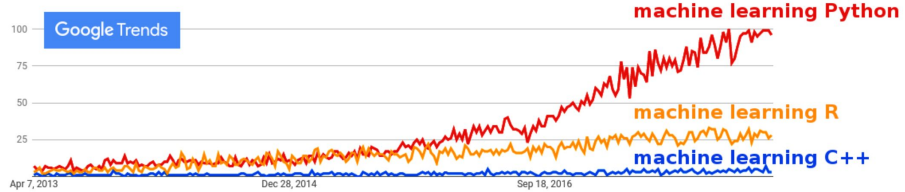
Frameworks

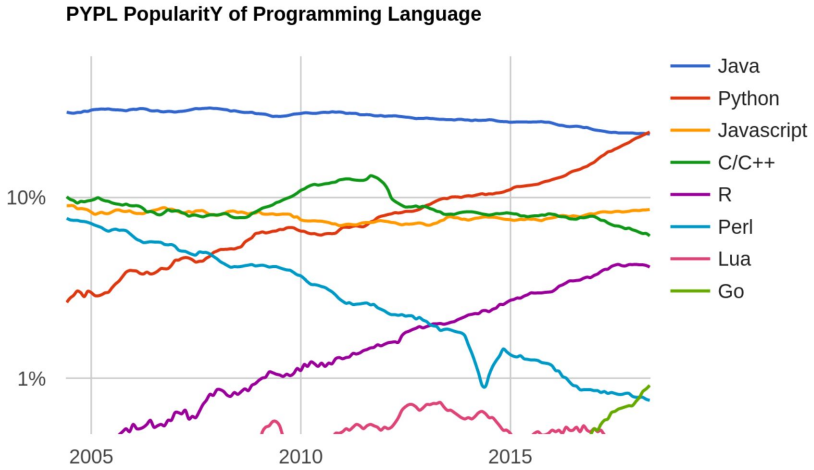
- Pre-CHEP meeting focusing on data models
- Use of accelerators is a common concern; tools are of common interest
- Will continue to meet in context of Software Forum meetings

PyHEP

- First conference looking at the role of Python in HEP organised





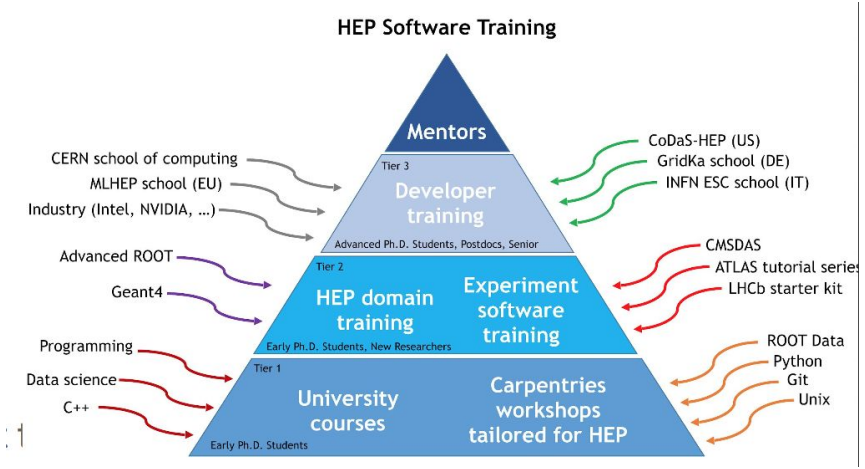


From Jim Pivarski's opening talk

PyHEP

- 70 people attended
  - Good representation from most LHC experiments, Belle II, many smaller experiments; lots of IT/lab representation as well
    - Informative pre-workshop questionnaire
- Key topics
  - Python used in HEP data analysis
  - Binding Python to other languages
  - Robust distribution of Python
  - Core software
  - Education and training
  - Migration to Python 3
- Outcomes
  - Continue with this as a workshop series
  - Build a community of interested users and developers
    - Setting up a gitter channel now for interested people

Training and Careers



Training

- Recognition of training ‘pyramid’, from core skills to expert
- Organising some federation of training schools
- Working on a curated set of training materials
  - StarterKit organisers decided that the HSF was a good place to host common HEP training material, https://github.com/hsf-training
- We would like to establish closer links with the Software Carpentry community
- Key point is to foster a community of trainers
- NSF has recently funded a 3 year training project with these objectives
  - Will support HSF efforts in this area

Conclusions

- We advanced a long way in understanding the problems of the next decade
  - And the areas where we can work together profitably
- HSF continues to act as a focal point for common software efforts
  - Continued work on important technical matters: licensing, packaging, software tools, frameworks
  - Inventory of software projects and tools; advice on publication and dissemination; training
  - Communication channels (hsf-forum, hsf-tech-forum lists) are vital
- New working groups will form nuclei of solving the grand challenges for HL-LHC
- New software projects will come
  - The should be agile and cooperative from the outset

There are many opportunities to be involved and shape our common work in the field