HEP Software Foundation

HSF Update

Michel Jouvin, CNRS/LAL - Graeme A Stewart, CERN EP-SFT

WLCG Overview Board, 12 June 2019

Community White Paper

- Published in Computing and Software for Big Science
  - https://doi.org/10.1007/s41781-018-0018-8
- Already 19 citations to this CSBS version

Computing and Software for Big Science
December 2019, 3:7 | Cite as

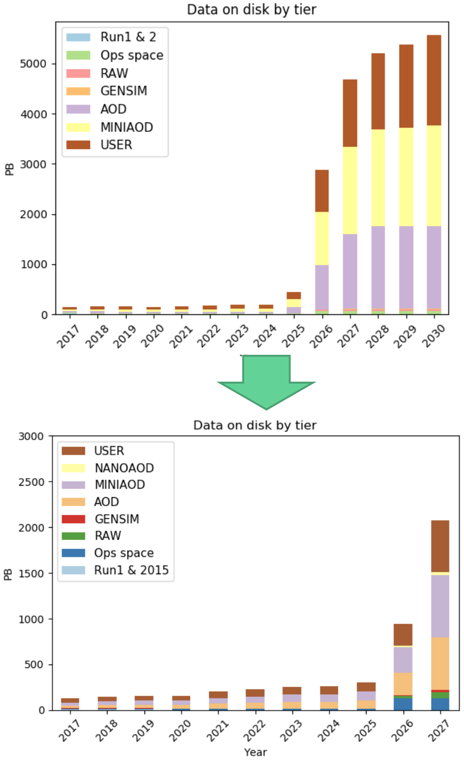
A Roadmap for HEP Software and Computing R&D for the 2020s

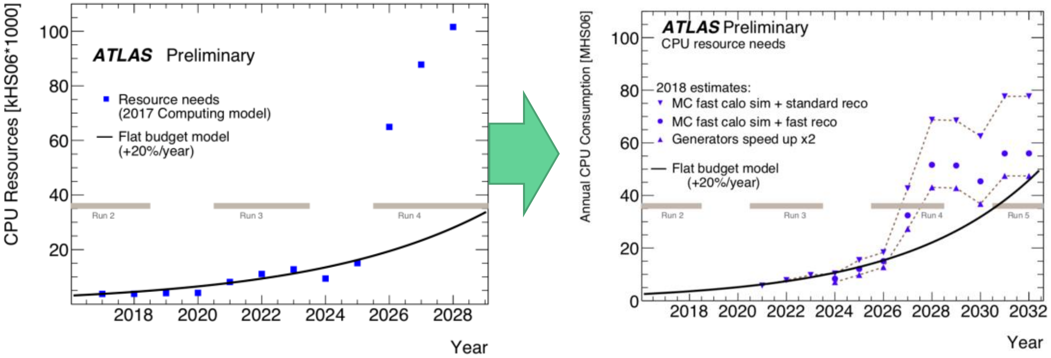
Authors | Authors and affiliations

The HEP Software Foundation, Johannes Albrecht, Antonio Augusto Alves Jr, Guilherme Amadio, Giuseppe Andronico, Nguyen Anh-Ky, Laurent Aphecetche, John Apostolakis, Makoto Asai, Luca Atzori, Marian Babik, Giuseppe Bagliesi, Marilena Bandieramonte, Sunanda Banerjee, Martin Barisits, show 296 more

Meeting the HL-LHC Challenge!

- Already since the Roadmap was written experiments have made great progress in meeting the HL-LHC challenge
  - Bad software, is extremely expensive
  - Good and clever software allows much more physics to fit in the budget





HOW2019 Workshop

- Joint HSF, WLCG, OSG meeting at Jefferson Lab, VA
    - 246 registrations
- Plenary programme covering topics of mutual interest
- Parallel sessions for more focused topics

HOW2019 Highlights

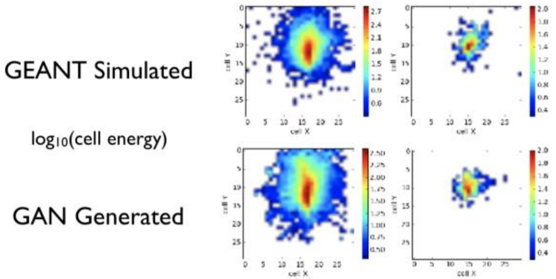
- Overview contributions from non-LHC and non-HEP experiments
  - DUNE, Belle II, Dark Matter, EIC, LSST, LIGO/VIRGO, IceCube
  - Common challenges and problems faced by these communities
- Technology watch and focused session on how we adapt our software for non-CPU and heterogeneous resources
- Sessions from new HSF Working Groups…
  - Analysis, Reconstruction, Simulation
- … and other HSF working groups (+ WLCG + OSG)
  - Training, Software Tools, PyHEP
- Last day discussion of funding initiatives
  - IRIS (UK), IRIS-HEP (US), IDT-UM (DE)

HSF Working Groups

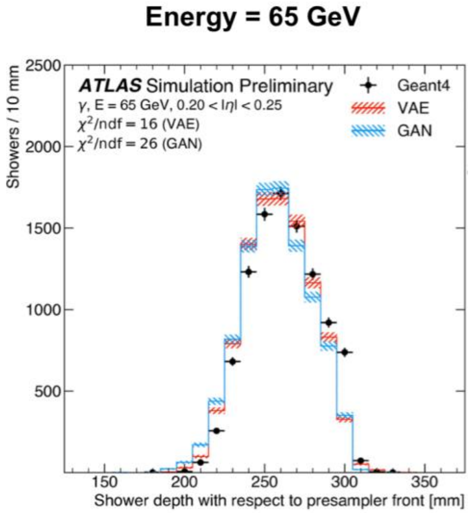
- The Roadmap established what challenges the community faced
    - But it did not spell out *how* to face them in detail
- HSF had adopted a model of working groups from its earliest days
    - These were open groups of people in the community, motivated enough to organise around a common topic, usually at their own initiative
- This model seemed a good one for moving forwards on the key topics
    - We were a little more formal this time around
        - Call for nominations from the whole community, then search committee
        - Significant engagement from LHC experiments and beyond, e.g. Belle II
- The HSF's role is one of an information conduit and meeting point
    - Report on interesting and common work being done
    - Forum for technical comments and discussion
    - Encourage cooperation across experiments and regions

Detector Simulation WG...

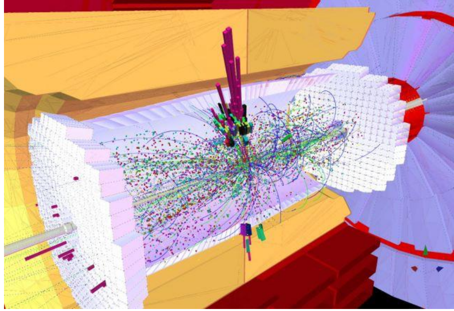
- A major consumer of LHC grid resources today
  - Experiments with higher data rates will need to more simulation
- Faster simulation, with no or minimal loss of accuracy, is the goal
  - Range of techniques have been used for a long time (frozen showers, paramtric response)
  - Key point is deciding when it's good enough for physics
- Machine learning lends itself to problems like this
  - Calorimeter simulations usually targeted
  - Variational Auto Encoders (VAEs) attempt to compress the data down to a 'latent space' - can be randomly sampled to generate new events
  - Generative Adverserial Networks (GANs) train two networks, one to generate events, the other to try to classify as real/fake
  - R&D on lifecycle integration into Geant4 is starting...



LHCb ECal simulated with G4, generated with GAN [F. Ratnikov]



ATLAS VAE and GAN cf. Geant4 simulation [ATL-SOFT-PUB-2018-001.]



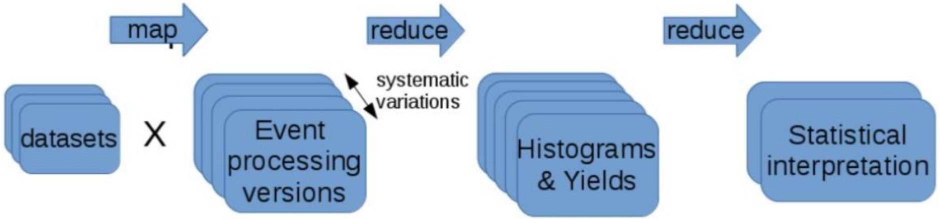
… Detector Simulation WG

- Technical improvement programme helps (and helps *everyone*)
- GeantV R&D modernises code and introduces vectorisation
  - Speed-ups observed
  - Vectorisation introduces small gains
  - Code modernisation seems to help a lot
  - Full report on this R&D after the summer
- Geant4 now have a new R&D working group that will take studies forward
- Some studies of running Geant4 on GPUs have begun
  - US Exascale Computing Project is funding this, motivated by upcoming exaflop supercomputers
    - 90-95% of FLOP capacity in GPUs
  - However, migration of physics code is an incredibly tricky business
    - This would be a long haul, but a huge achievement for all of HEP if we succeed…

Reconstruction and Software Triggers WG

- Software triggers close to the machine required to deal with tremendous rates and to get sufficient discrimination
  - Pressure to break with legacy code is high
  - Lots of experimentation with rewriting code for GPUs
    - In production for ALICE (since Run2)
    - Advanced prototypes for CMS (Patatrack) and LHCb (Allen)
- **Orienting the design around the data** (optimal layouts) is critical
  - This was a key topic identified at JLab HOW Workshop, lots of ongoing discussions since then (including last week meeting, https://indico.cern.ch/event/823263/)
- Real Time Analysis becomes more and more important
  - Produce analysis useful outputs as part of the trigger decision
  - LHCb Turbo strategy here is well known
  - ATLAS and CMS also doing some analysis this way also

Data Analysis WG



- Improve analysis ergonomics - how the user interacts with the system to express their analysis
  - Streamline common tasks
    - Handle all input datasets; Corrections and systematics
    - Compute per event and accumulate; Statistical interpretations
  - **Declarative models**, building on ROOT's RDataFrame
    - Say *what*, not *how* and let the backend optimise
    - E.g. split and merge, GPU execution
- Notebook like interfaces gain ground, as do containers - lots of high level Python tools
  - Links strongly to PyHEP group
- Interest in data science tools and machine learning is significant for this community - inspiring new approaches (e.g. Scikit-HEP (uproot, awkward array), Coffea, IRIS-HEP)
  - This is an ecosystem into which HEP can contribute
- Links to DOMA and facilities through interest in dedicated analysis clusters

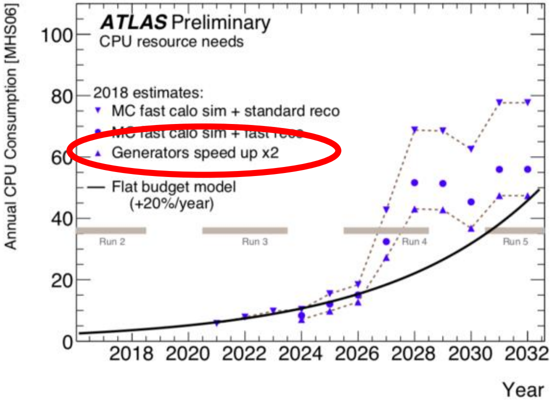
Many analysis frameworks, multiple per experiment, not well generalised

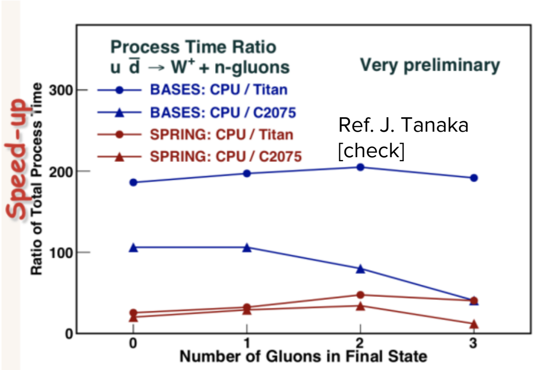
```
# * Jet select/cleaning against loose leptons , jet pt > 25 , jet id
flow.DefaultConfig(jetPtCut=25,jetIdCut=0,jetPUIdCut=0)
flow.SubCollection("CleanJet","Jet",'''
 Jet_pt > jetPtCut &&
 Jet_jetId > jetIdCut &&
 Jet_puId > jetPUIdCut &&
 (Jet_LeptonIdx==-1 || Jet_LeptonDr > 0.3)
''')
```

A. Rizzi, NAIL prototype

Event Generation WG...

- Event generators are the start of the simulation chain
  - At the LHC Run1 only leading order generators were used
  - With Run3/4, higher order generators become much more important and are much more costly to run
- HSF/LPCC workshop in November brought theory and experiment together to look at computing challenges of event generation
  - This was the first workshop of its kind
- Working group tackling technical challenges
  - Setting a baseline for further comparisons
  - Support for technical improvements (e.g. thread safety)

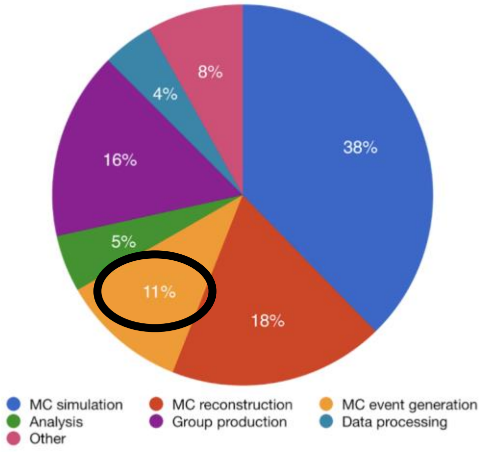




… Event Generation WG

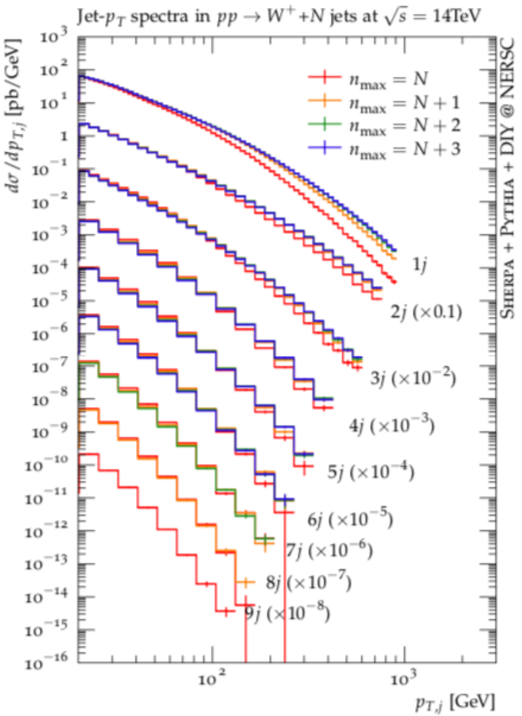
- Better understanding of ATLAS/CMS CPU usage of generators
  - A lot of CMS usage is folded into combined jobs (evgen+sim+reco)
  - Overall usage not as different as previously thought: ~x2-3 rather than x10
  - Insight into different setups and strategies
    - ATLAS filter events more aggressively (increases CPU time, but better populates phase space)
    - Sherpa CPU per event can be improved by x2-3 with a different scale factor choice
- Document summarising these findings is in progress
  - Establish a good baseline of understanding

ATLAS 2018 CPU Report



Event Generation on HPCs and GPUs

- Considerable progress on efficient use of large clusters of machines for event generation
  - Targeting HPC resources in particular
  - Scaling up to 2048 nodes
  - Enables simulation of W/Z+9j with Sherpa/Pythia
- Porting to other architectures
  - Could be very suitable code to do this with (smaller, self contained code bases, numerically intensive)
  - Will also follow up with MadGraph team on their GPU port of some pieces of this generator… but this looks far away from being a working production setup “out of the box”



[arXiv:1905.05120]

Software Nuts and Bolts

HSF Packaging Working Group Report

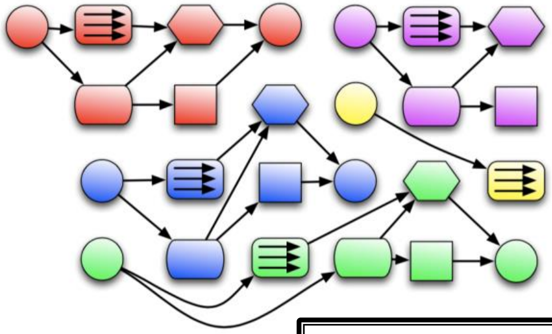
[HSF-TN-2016-03]

L. Sexton-Kennedy[1], B. Hegner[2], B. Viren[3]

[1]FNAL,[2]CERN,[3]BNL

- Software Tools WG
    - Active group promoting best practice for correctness and performance
    - There has been a revolution in adopting best open source practice in recent years
        - git, GitHub, GitLab, CMake, merge requests, code review, ...
    - Topical meeting on a new monitoring tool (Trident, from CERN IT)
    - Best practice in use of static analysers and performance monitoring
- Packaging WG
    - Key component to build an ecosystem and allow to assemble modules as needed
    - Need a software stack, incorporating many components from the open source world and HEP community
        - This touches deeply on license and license combinations
    - Preference for tools that are not home grown and have a wider support base
    - Active protyping activities: Spack (LBNL) in use in Neutrino experiments + FCC, Conda for analysis SW delivery (ROOT for example)

Frameworks and Integration



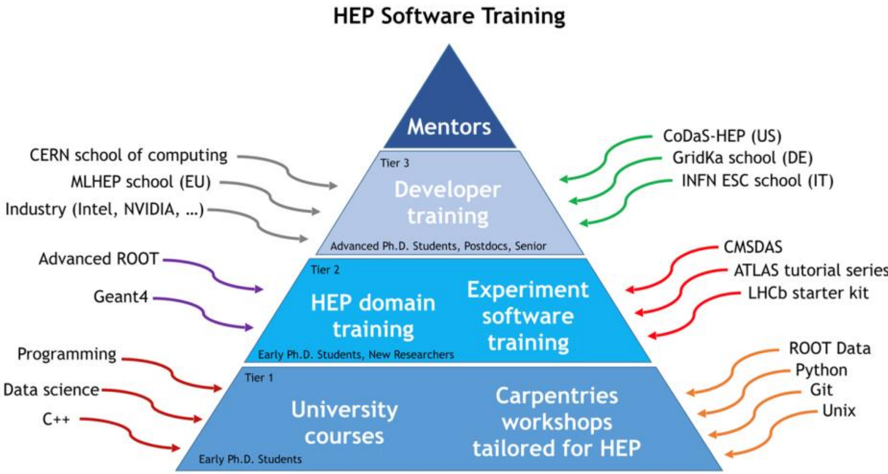
Cartoon of a single job, processing multiple events (colours) through different modules (shapes)

- Increasingly heterogeneous world requires advanced software support infrastructure
  - Software frameworks support use of different devices as well as insulate developers from many of the details of concurrency and threading models
    - Adapt to the new heterogeneous landscape
    - Latency hiding is critical to maintaining throughout
  - Framework development has traditionally been quite fragmented, but new experiments should offer a chance to increase convergence
    - Better to start off together than try to re-converge later (iLCSoft, LArSoft examples of success, albeit without concurrency; Gaudi for LHCb, ATLAS)
    - E.g. ALFA for ALICE and FAIR experiments
- New HSF working group being established now (draft mandate)
  - Currently in the convenor nomination phase

Google Summer of Code / Season of Docs

- 34 slots granted by Google
  - Up 5 from last year
  - One project was disqualified, so we could use 33 slots
- Google have also launched a programme to improve the quality of documentation
  - Supports a technical writer for open source projects
  - We are exploring this with proposals from ROOT and Rucio
- 3 administrators: 2 from CERN/SFT, 1 external (LAL)
  - Same admins for both programs

Training and Careers



- Many new skills are needed for today's software developers and users
- Base has relatively uniform demands
  - Any common components help us
- LHCb StarterKit initiative taken up by several experiments, sharing training material
  - Links to 'Carpentries' being remade (US training projects) - up the level!
- New areas of challenge
  - Concurrency, accelerators, data science
  - Need to foster new C++ expertise (unlikely to be replaced soon as our core language, but needs to be modernised)
- Careers area for HEP software experts is an area of great concern
  - Need a functioning career path that retains skills and rewards passing them on
  - Recognition that software is a key part of HEP now

Raising Software's Profile and ESPP

- HSF contributed a paper to the European Strategy Update process
  - Considerable HSF discussion and input to talk on Software R&D at Granada
- Mentioned as a critical issues in Granada summary talk on Detectector R&D and computing
  - **Training** - how to equip developers with the correct skills
    - From starting students to refresh for experienced people
  - **Careers**
    - Establish a viable long term career path for HEP software experts: involving them with training activities is helpful, especially through universities
- Discussions started on meaningful ways to develop this activity, involving computer scientists and software engineers
  - Make links with other data intensive sciences with similar challenges
  - E.g. dark matter and astro-particle have expressed interest in this area

Next WLCG/HSF Workshop (Adelaide)

- Date: week-end before CHEP, November 2-3 (noon to noon)
    - Will be officially announced as part of the next CHEP bulletin
- Focused on analysis: *From Future Facilities to Final Plots*
    - Not the usual format reviewing many things as it will be a short meeting
- Program committee (main members): WLCG (Ian B. + C.), HSF (M. Jouvin, G. Stewart), DOMA project (S. Campana), HSF Analysis WG (P. Laycock)
    - Main topics identified
    - Working on session definitions to ensure that they are relevant to both HSF **and** WLCG: don't want 2 workshops in one...
- CHEP will also cover many of the issues tackled by HSF

Conclusions

- We have a wide ranging and ambitious physics programme in HEP and in associated disciplines
    - Our experiments are highly data intensive and require high quality software and computing
- The landscape for software is becoming ever more challenging
    - Working together on common problems is a requirement for efficiency and from our F.A.
- HSF increasing communication between experiments
    - Working groups are active and meeting regularly
    - Forum for exchange of ideas
    - We hope that common development areas will arise from this
- HSF also recognised as playing a role as an advocate for software
    - This raises the profile of software as a critical activity
    - But progress on training and careers really is needed

HL-LHC is a challenge and also a great opportunity to improve HEP software

Useful Links

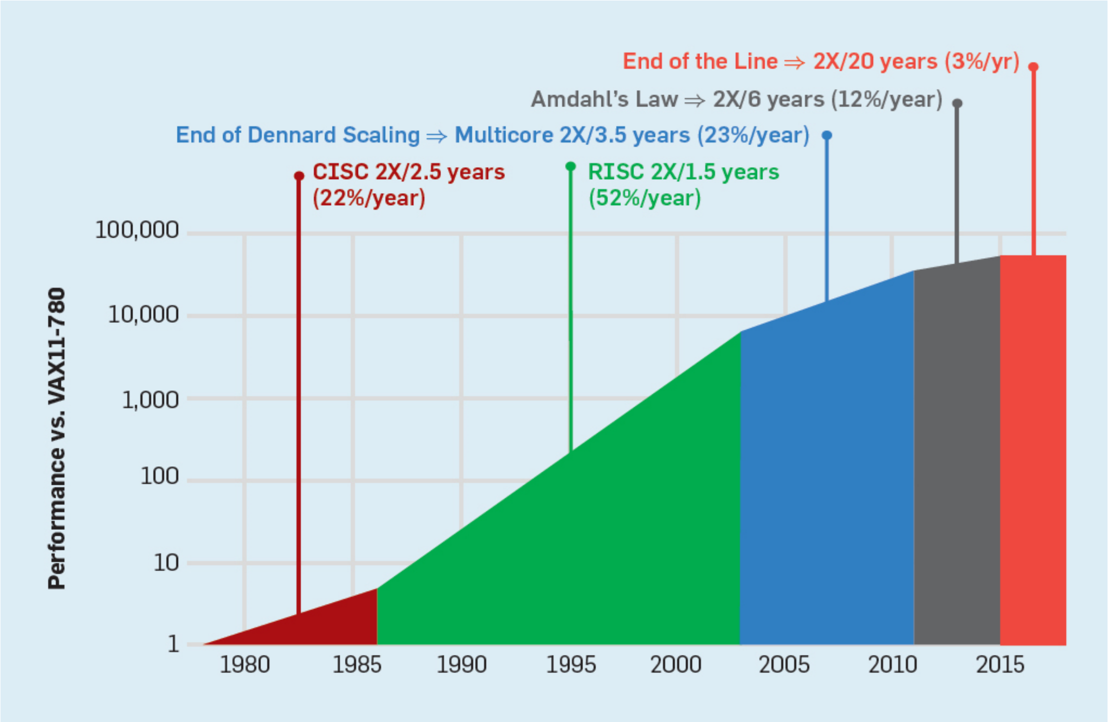
- HSF web site: https://hepsoftwarefoundation.org
- ESPP Open Workshop, Granada, May 2019
  - HEP Computing Software R&D, G. Stewart: https://indico.cern.ch/event/808335/contributions/3367988/attachments/1843865/3025660/eppsu-software-rd.pdf
  - Summary on Instrumentation and Computing: https://indico.cern.ch/event/808335/contributions/3365081/attachments/1845683/3028368/summary-instrumentation-computing.pdf
- Software update report @LHCC, G. Stewart, June 2019
  - https://indico.cern.ch/event/754732/contributions/3127504/attachments/1855646/3047775/Software_Update_2019-06.pdf

Backup Slides

Decreasing Returns over Time

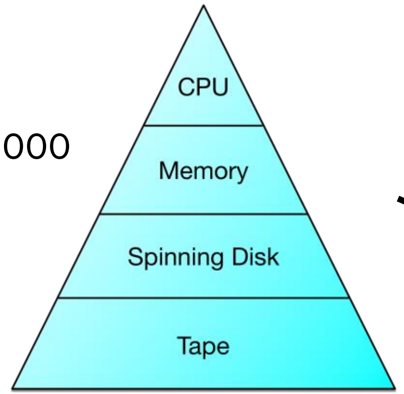
- Conclusion is that diversity of new architectures will only grow
- Best known example is of GPUs

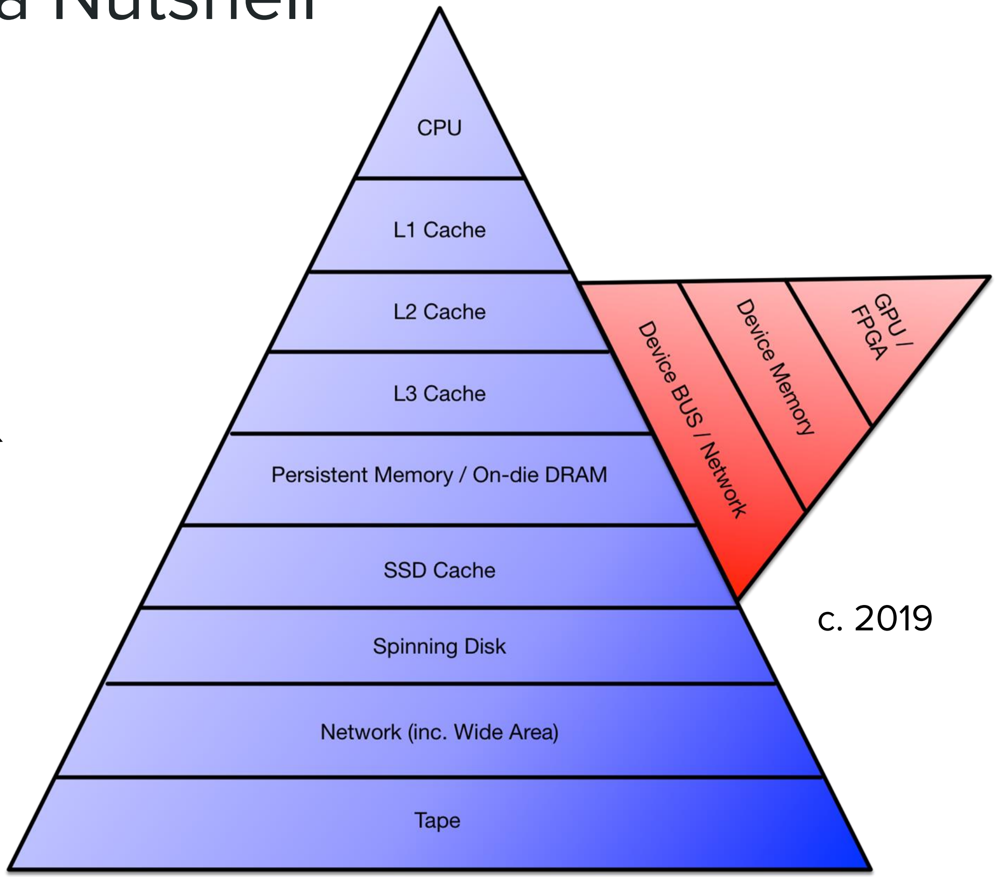
[link]



Hardware Evolution in a Nutshell

c. 2000

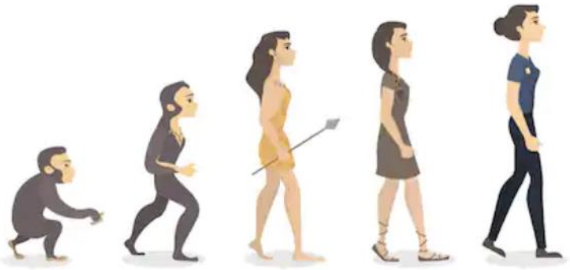




c. 2019

Oh brave new world!
That has such people in it...

Drivers of Technology Evolution



- Low power devices
  - Driven by mobile technology and Internet of Things
- Data centre processing
  - Extremely large clusters running fairly specialist applications
- Machine learning
  - New silicon devices specialised for training machine learning algorithms, particularly low precision calculations
- Exascale computing
  - Not in itself general purpose, but poses many technical problems whose solutions can be general - HEP pushed to use HPC centres, especially in US
- Energy efficiency is a driver for all of these developments
  - Specialist processors would be designed for very specific tasks
  - Chips would be unable to power all transistors at once: dark silicon is unlit when not used

Software Challenges and Opportunities

Concurrency

- The one overriding characteristic of modern processor hardware is concurrency
    - SIMD - Single Instruction Multiple Data (a.k.a. vectorisation)
        - Doing exactly the same operation on multiple data objects
    - MIMD - Multiple Instruction Multiple Data (a.k.a. multi-theading or multi-processing)
        - Performing different operations on different data objects, but at the same time
- Because of the inherently parallel nature of HEP processing a lot of concurrency can be exploited at rough granularity
    - Run many jobs from the same task in parallel
    - Run different events from the same job in parallel
- However, the push to highly parallel processing (1000s of GPU cores) requires **parallel algorithms**
    - This often requires completely rethinking problems that had sequential solutions previously, e.g. finding track seeds via cellular automata (TrickTrack library, CMS and FCC)

Heterogeneity

- There are a lot of possible parallel architectures on the market
  - CPUs with multiple cores and wide registers
    - SSE4.2, AVX, AVX2, AVX512, Neon, SVE, Altivec/VMX, VSX
  - GPUs with many cores; FPGAs
    - Nvidia (many generations - often significantly different), AMD, Intel, …
- In addition there are ‘far out’ architectures proposed, like Intel’s Configurable Spatial Architecture
- Many options for coding, both generic and specific:
  - Cuda, TBB, OpenACC, OpenMP, OpenCL (→ Vulcan), alpaka, Kokkos, …
- Frustratingly no clear winner, mutually exclusive solutions and many niches
  - One option for now is to isolate the algorithmic code from a ‘wrapper’ that targets a particular device or architecture - approach of ALICE for their GPU/CPU code
  - Hiding details in a lower level library (e.g. VecCore) also helps insulate developers

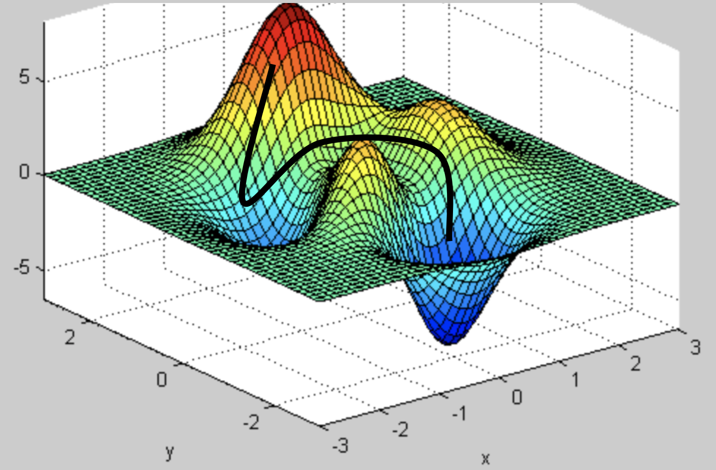
Data Layout and Throughput

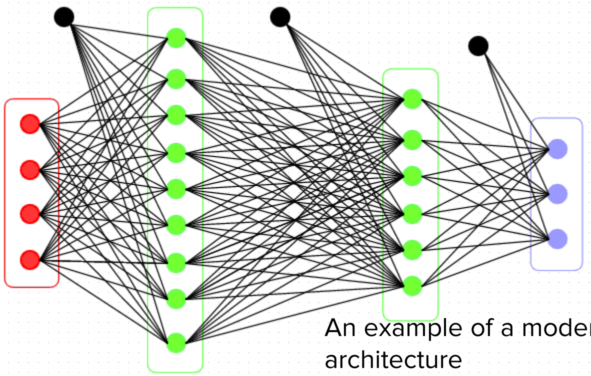
- Original HEP C++ Event Data Models were heavily inspired by the Object Oriented paradigm
  - Deep levels of inheritance
  - Access to data through various indirections
  - Scattered objects in memory
- Lacklustre performance was ~hidden by the CPU and we survived LHC start
- In-memory data layout has been improved since then (e.g. ATLAS xAOD)
  - But still hard for the compiler to really figure out what's going on
  - Function calls non-optimal
  - Extensive use of 'internal' EDMs in particular areas, e.g. tracking
- iLCSoft / LCIO also proved that common data models help a lot with common software development
- Want to be flexible re. device transfers and offer different persistency options
  - e.g. ALICE Run3 EDM optimised for message passing and the code generation approaches in FCC-hh PODIO EDM generator

Machine Learning

- Machine learning, or artificial intelligence, used for many years in HEP
  - Algorithms learn by example (training) how to perform tasks instead of being programmed
- Significant advances in the last years in ‘deep learning’
  - Deep means many neural network layers
  - Fast differentiability and use of GPUs
- Rapid development driven by industry
  - Vibrant ecosystem of tools and techniques
  - *Highly optimised for modern, specialised hardware*

ML minimisation problem - do this minimisation with 10^6 variables...





An example of a modern ML architecture

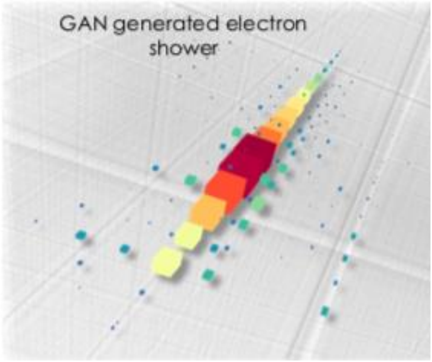
Machine Learning in HEP

- Better discrimination
  - Important input for analysis (see improvements with Higgs)
  - Also used at HLT as inference can be fast (N.B. training can be slow!)
  - HEP analogies to image recognition or text processing
- Replace expensive calculations with trained output
  - E.g. calorimeter simulations and other complex physical processes
- There are significant opportunities here
  - Need to combine physics and data science knowledge
  - Field evolves rapidly and we need to deepen our expertise
- Integration into our workflows is not at all settled
  - Resource provision, efficient use, heterogeneity and programming models pose problems
  - Training deep models may require *significant* resources

Table 1 | Effect of machine learning on the discovery and study of the Higgs boson

Analysis	Years of data collection	Sensitivity without machine learning	Sensitivity with machine learning	Ratio of P values	Additional data required
CMS[24] $H \to \gamma\gamma$	2011–2012	2.2σ, $P = 0.014$	2.7σ, $P = 0.0035$	4.0	51%
ATLAS[43] $H \to \tau^+\tau^-$	2011–2012	2.5σ, $P = 0.0062$	3.4σ, $P = 0.00034$	18	85%
ATLAS[99] $VH \to bb$	2011–2012	1.9σ, $P = 0.029$	2.5σ, $P = 0.0062$	4.7	73%
ATLAS[41] $VH \to bb$	2015–2016	2.8σ, $P = 0.0026$	3.0σ, $P = 0.00135$	1.9	15%
CMS[100] $VH \to bb$	2011–2012	1.4σ, $P = 0.081$	2.1σ, $P = 0.018$	4.5	125%

Machine learning at the energy and intensity frontiers of particle physics, https://doi.org/10.1038/s41586-018-0361-2



Use of Generative Adversarial Networks to simulate calorimeter showers, trained on G4 events (S. Vallacorsa)