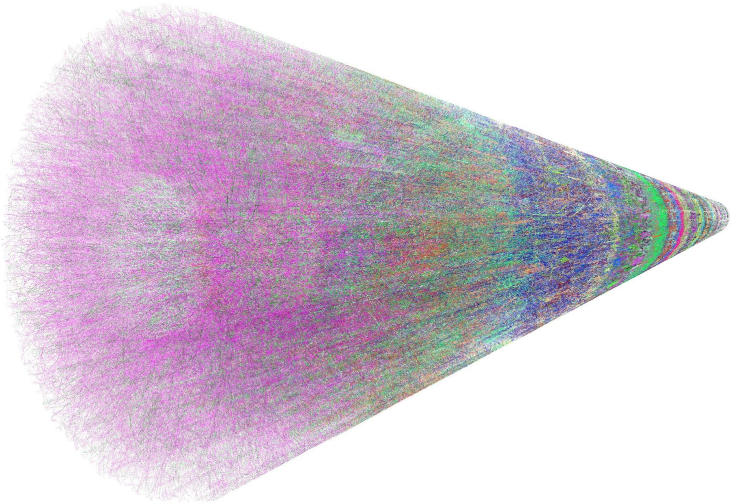
Software R&D Lines 2020 - 2025

Jakob Blomer, Graeme Stewart

On behalf of and with presentation material from the Software Working Group

2018-09-25

HEP Software Challenges for the 2020s



ALICE run 3: TPC 2ms PbPb time frame 50kHz

1. An order of magnitude higher event rates and event complexities at future hh colliders

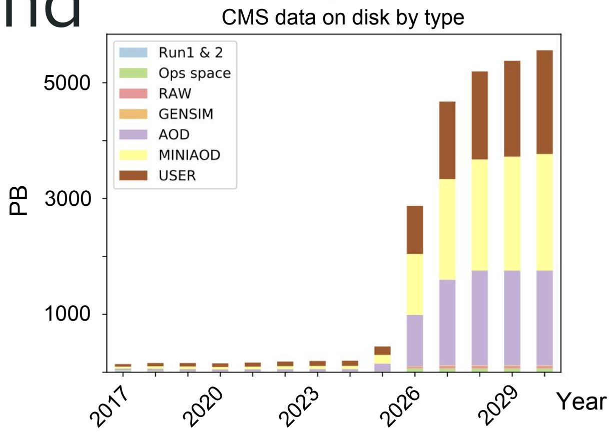
2. Changing hardware landscape
   Specialized and more parallel processors and storage devices

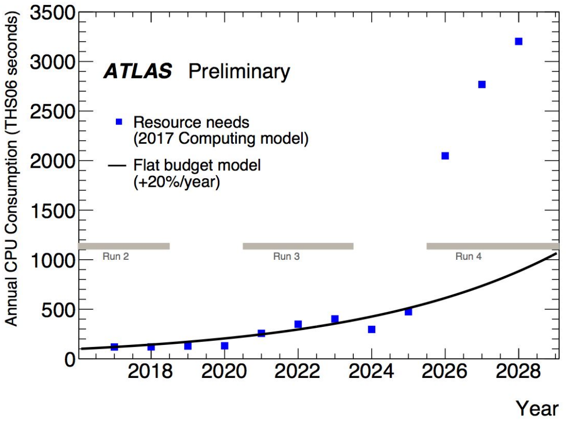
3. Engineering of expressive and robust abstractions and building blocks

Investment in detectors must be matched by software investment

Challenges to the HL-LHC and beyond

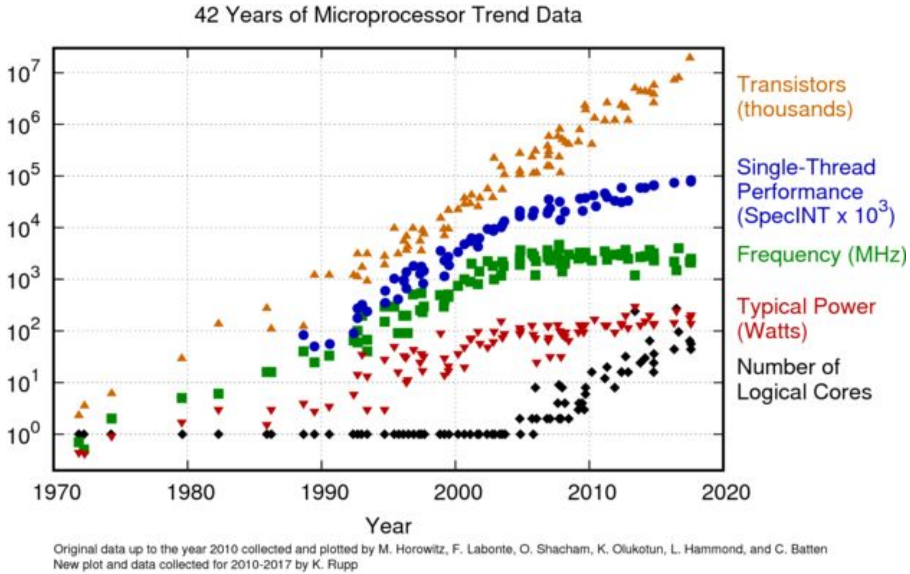
- High-Luminosity LHC is far from being a solved problem for software and computing
  - Extrapolation from today's software is not affordable
- Beyond HL-LHC, there are a number of different options for new machines
  - Lepton colliders (CLIC, FCC-ee wrt. event complexity) have overall less serious computing challenges
    - Require performant, robust, easy to use/deploy software
  - Hadron colliders (HE-LHC, FCC-hh) bring a massive data rate and complexity problem
    - Extreme for everything:
      generators, simulation, reconstruction, analysis
- Whatever the future, we pass through the HL-LHC on the way
  - HEP Software Foundation Community White Paper maps out that path





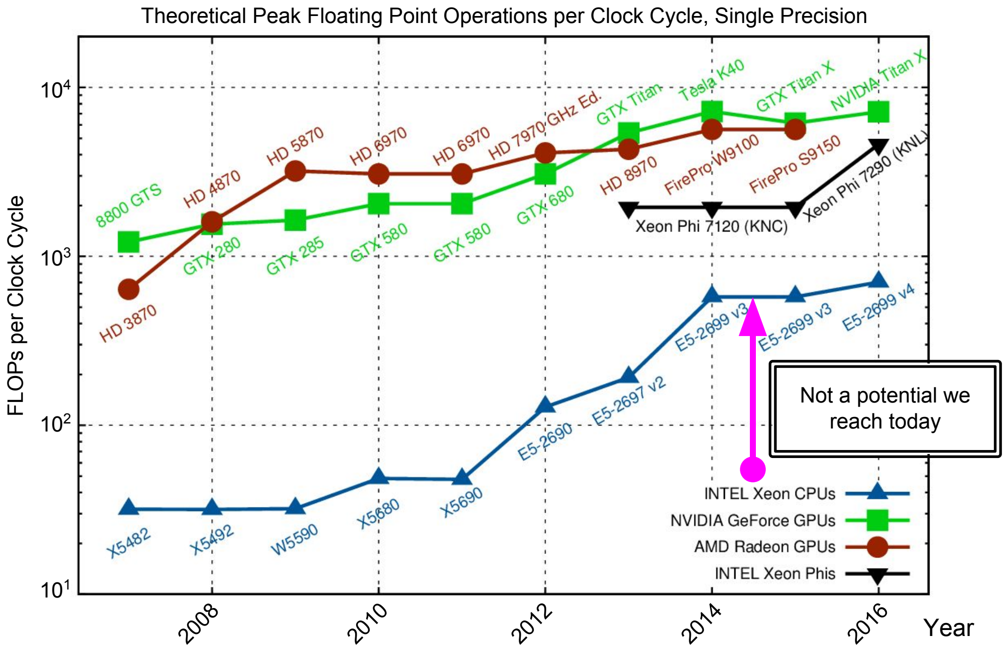
Processor evolution

- Moore's Law continues to deliver increases in transistor density
  - But the rates go down
- Clock speed increase stopped around 2006
  - No longer possible to ramp the clock speed as process size shrinks (Dennard scaling failed)
- We are basically stuck at ~3GHz clocks from the underlying Wm^{-2} limit
  - This is the *Power Wall*
  - Limits the capabilities of serial processing
  - CPU based concurrency still in development for Run 3



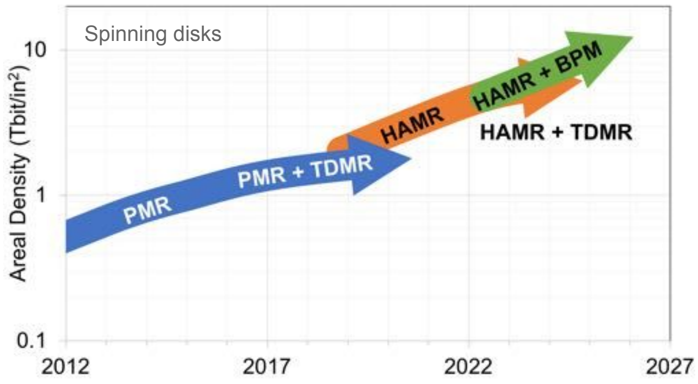
Compute Accelerators (GPU, FPGA, ...)

- It will take continued software effort to exploit the silicon of modern processors
  - Even CPU vector registers are hard for us to exploit
- This gets even more acute with accelerators
  - Different programming model: Many cores, high floating point throughput, but lose a lot of ‘ease of use’
- Accelerators now widely deployed in Supercomputers
- We have to adapt to maintain our ability to use processors effectively

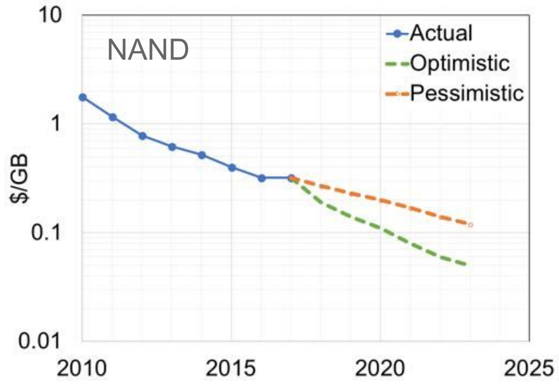


Storage and Network Trends

- Memory
  - DRAM improvements now modest
  - Overall, memory ‘landscape’ becomes more complex
    - Memory/storage boundary blurring
- Storage
  - Spinning disk capacity keeps slowly climbing at ~10%/y
  - SSDs prices dropping but remains likely too high to fully replace HDDs, require highly parallel access for best performance
  - Tape remains cheap to buy, slow to access with few companies left, O(1)
- Networks
  - Capacity increases expected to continue, latency will not change
  - Utilising high-speed networks requires change to software layering



Glenn Lockwood, NERSC



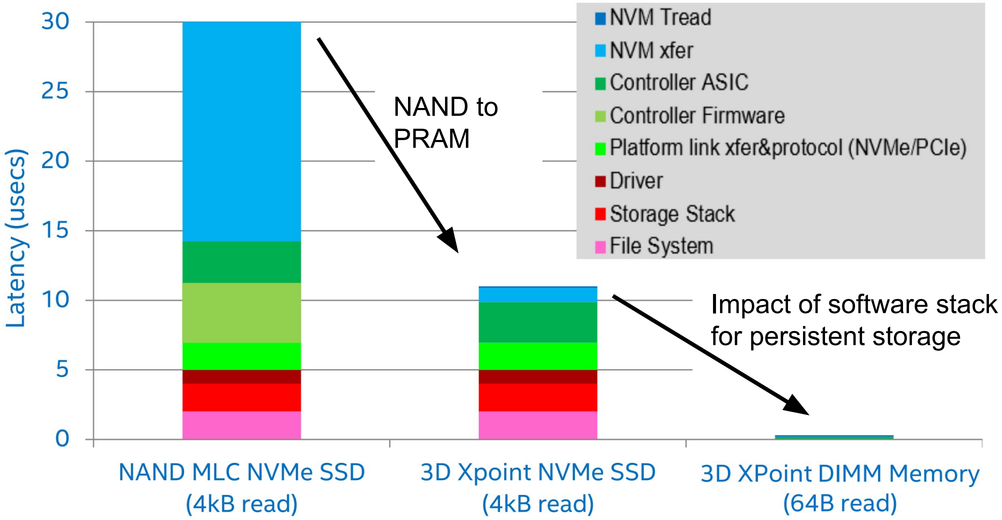
Glenn Lockwood, NERSC

Data Access Trends

- Lower latency of future ultra-fast persistent storage puts pressure on data access software

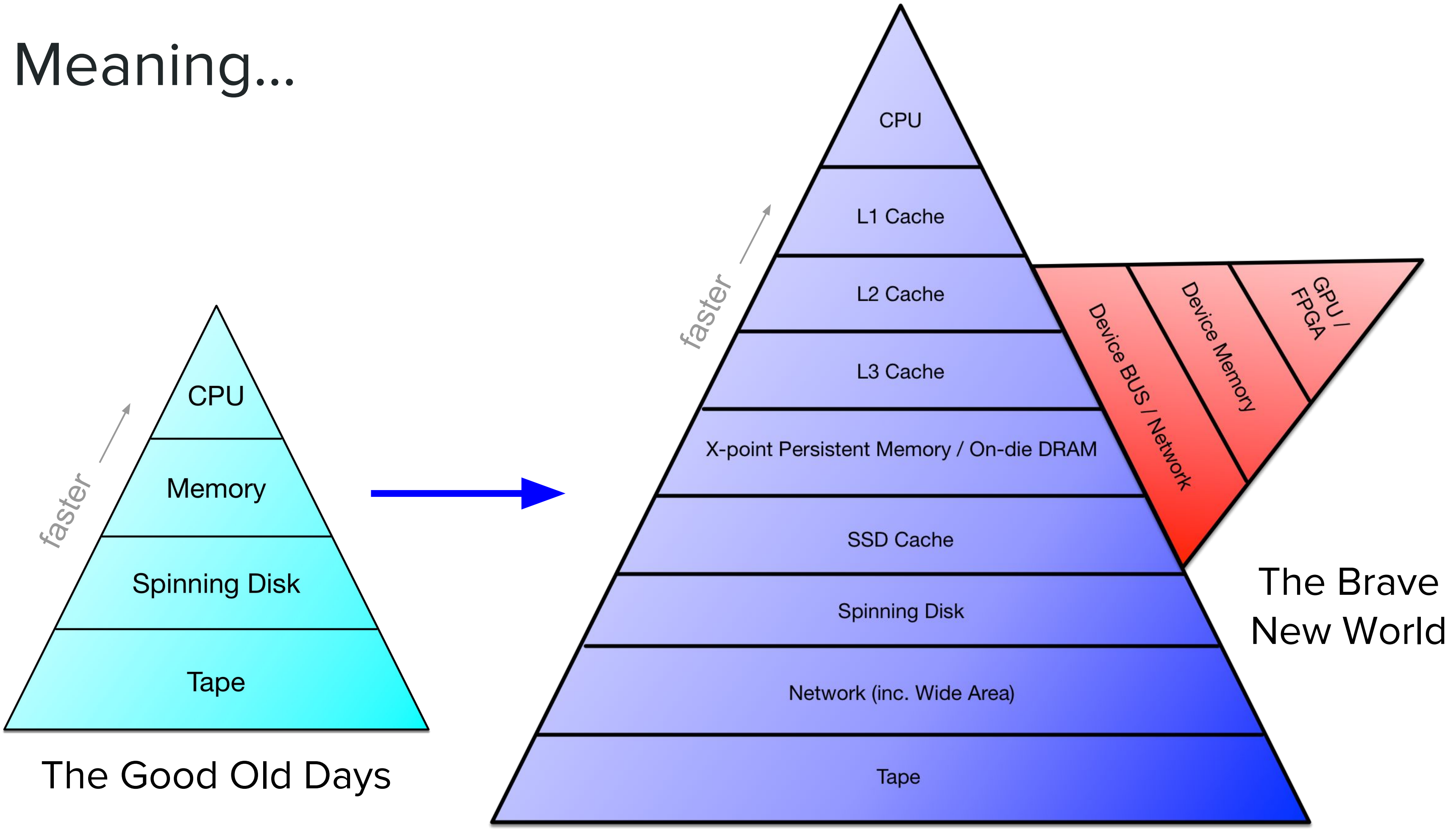
- The file system tree abstraction of storage devices becomes a bottleneck
    - Simple “object stores” show better meta-data scalability

- Shifts task of data organization from off-the-shelf software towards physics applications



R. Coulson: The Quest for Low Latency Storage. Senior Fellow, Intel NVM Solutions Group

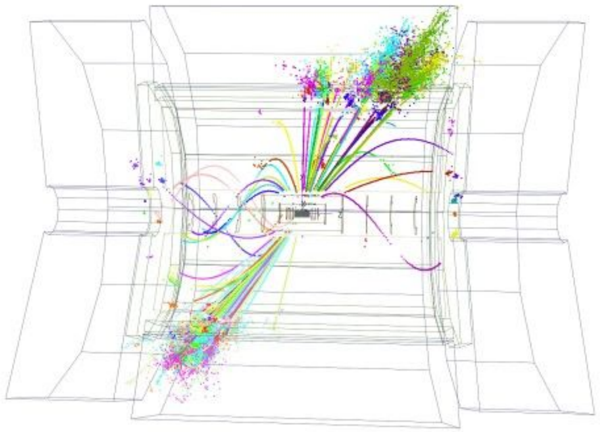
Meaning...


faster
CPU
Memory
Spinning Disk
Tape
The Good Old Days
faster
CPU
L1 Cache
L2 Cache
L3 Cache
X-point Persistent Memory / On-die DRAM
SSD Cache
Spinning Disk
Network (inc. Wide Area)
Tape
Device BUS / Network
Device Memory
GPU / FPGA
The Brave New World

HEP Specific Software Domains

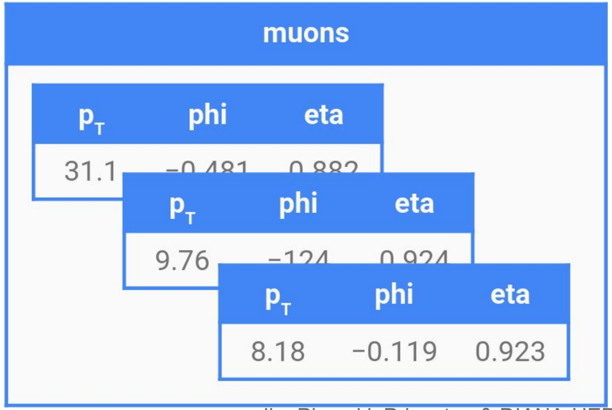
- The Big Data industry tackles computing problems at similar or larger scale than HEP

- Yet there are a number of domains that likely no-one else will solve for us:
  - algorithms describing the physics of detectors
  - data acquisition at the detector
  - the structured representation of HEP event data
  - Uncertainty estimation of approximation methods, including machine learning kernels
  - management of heterogeneous and federated compute and storage resources



CLIC $e^+e^- \rightarrow$ ttbar @ 3TeV

Event data does not map well into flat tables



Jim Pivarski, Princeton & DIANA-HEP

Rationale behind the R&D Line Proposals

- Excellent detectors require excellent software

- Broad software community participation
  - Gather ideas from the whole of the HEP software and computing community
  - Two lightning talk sessions [1] [2]
  - Aligned with HSF community white paper
  - Representative core group to collect input and define R&D lines

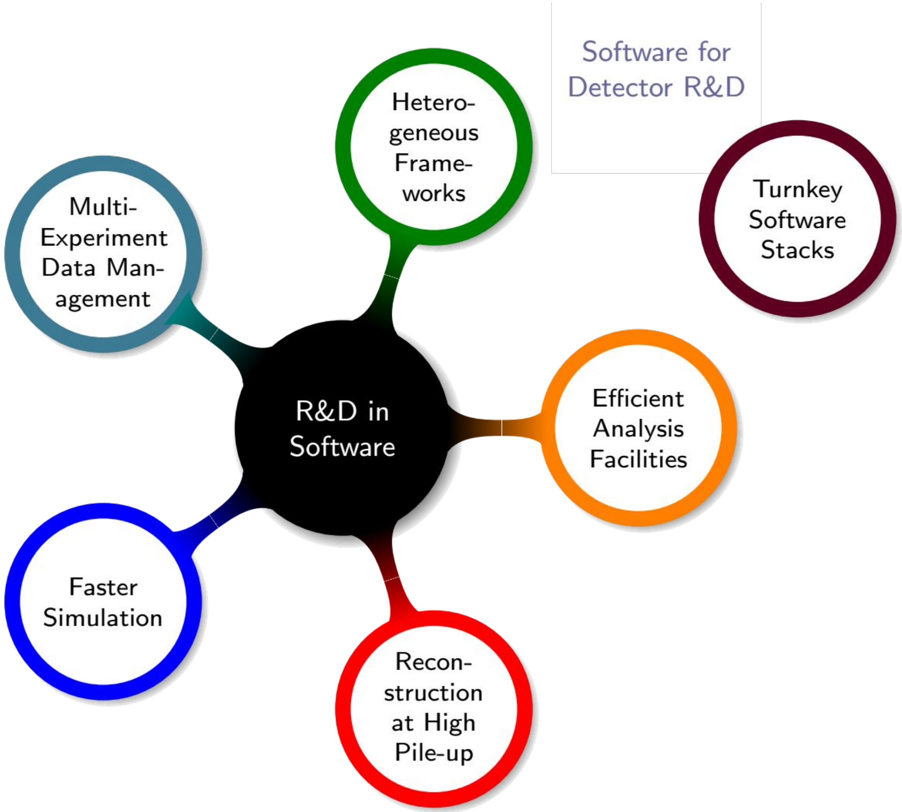
- Work plans adjusted for early user feedback
  - Based on the experience of previous successful software R&D
  - Small and efficient development teams
  - Helps navigate through a quickly changing technology landscape

Software Working Group Core Team

Jakob Blomer (Convener)	Benedikt Hegner (FCC)	Witek Pokorski (Simulation, Generators)
Graeme Stewart (Convener)	Mario Lassnig (Data Management, ATLAS)	Radu Popescu (Programming Languages)
Marilena Bandieramonte (Simulation)	Helge Meinhard (IT R&D)	David Rohr (GPUs, ALICE)
Andrea Bocci (Reconstruction, CMS)	Niko Neufeld (DAQ, LHCb)	Marco Rovere (Reconstruction, CMS)
Marco Cattaneo (LHCb)	Felice Pantaleo (Compute Accelerators, CMS)	André Sailer (CLIC, LCD)
Dirk Duellmann (IT Liaison)	Marko Petric (CLIC)	Andreas Salzburger (Tracking, FCC, ATLAS)
Giulio Eulisse (Systems, ALICE)	Maurizio Pierini (Machine Learning, CMS)	Enric Tejedor (Systems, Analysis)
Andrei Gheata (Simulation)	Danilo Piparo (Analysis)	Sofia Vallecorsa (Simulation, OpenLab)

Six Areas of Investment

- Five research lines addressing HEP software challenges
- Turnkey software stacks supporting of detector studies
- Each activity has a program of work for 2 fellows over 5 years; turnkey software stacks is put at 3 fellows plus 1 fellow per detector study



Proposed Software R&D Lines

Simulation Problem

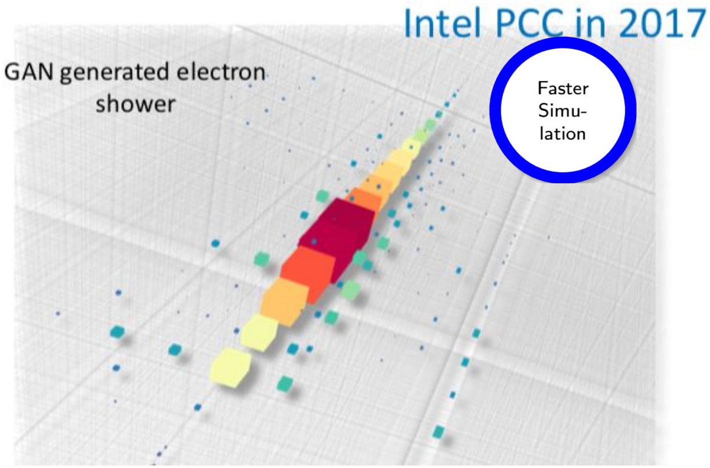
Faster Simu-lation

- Experiments at future colliders need
  an order of magnitude speed-up in detector simulation

- We can realistically expect a speed-up of 2-4 (still to be demonstrated)
  using vectorization of "full" simulation

- "Fast" simulation techniques are becoming an
  essential part of MC simulation applications
  - evolution towards hybrid (full+fast) simulation solution
  - trading speed for precision for specific detectors/regions/particles/physics channels
  - need to be fully embedded within the simulation flow

Technology Exploration



- Machine Learning (ML) can potentially replace "hand-made" parameterization (classical approach for fast simulation) with trained neural networks

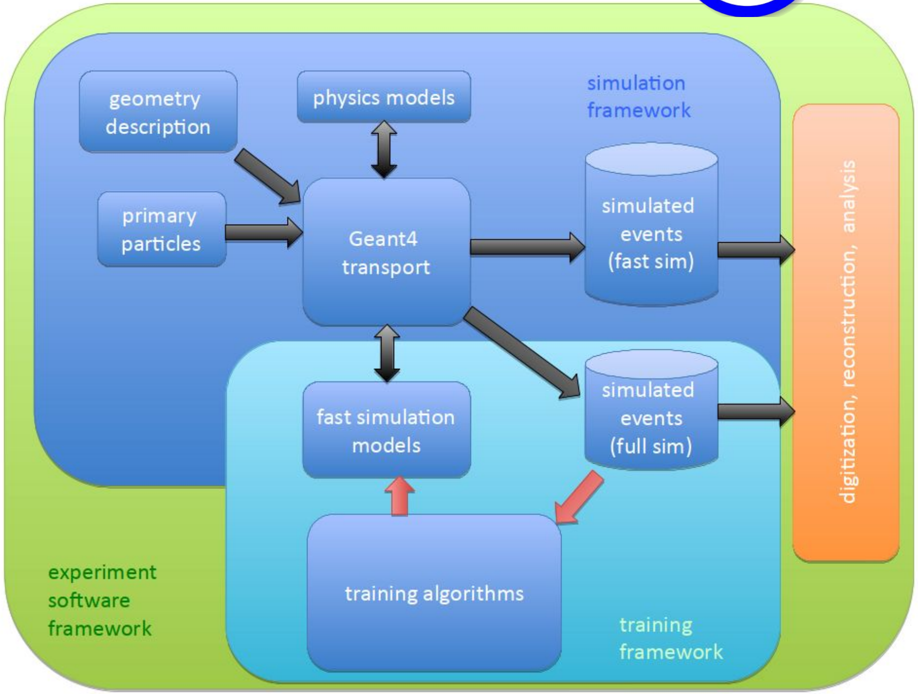
- Numerous possibilities for applying ML, R&D already ongoing in a few directions
    - speeding-up specific CPU consuming calculations (cross-sections)
    - modeling generation of final states
    - generation of full detector response (calorimeter showers, TPC, silicon trackers)
    - pile-up simulation
    - modeling of simulation + reconstruction in one step

R&D Deliverables

Faster Simu-lation

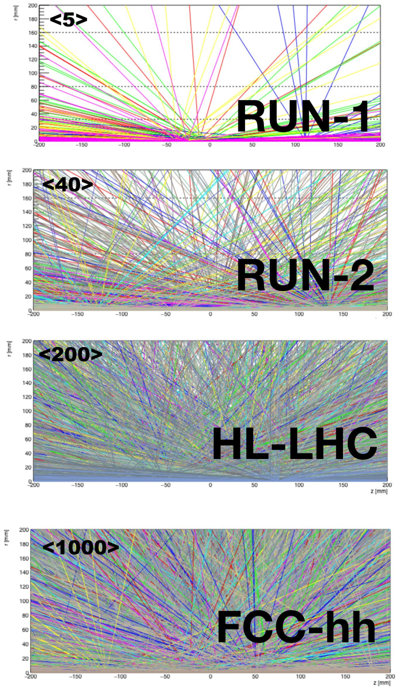
- Work packages
  - fast parameterisation of CPU-intensive physics processes
  - detector response simulation
  - combined fast simulation and reconstruction

- Need to explore how best to do the ML training in the context of the overall workflow
  - crucial to provide complete solutions where both the training and inference are possible to plug into existing simulation frameworks/toolkits
  - systematic uncertainties need to be understood



Reconstruction Problem

Recon-
struction
at High
Pile-up



- High pile-up affects reconstruction disproportionally due to combinatorial explosion
- Timing detectors and highly granular calorimeters require new algorithmic foundations for their reconstruction
- Vectorised and hardware-accelerated algorithms are necessary to sustain higher event rates

Reconstruction R&D

Recon-
struction
at High
Pile-up

- Development of the mathematical framework for 4D detectors (including timing) and 6D track model, transport and fitting
- Extension and application of tracking concepts to High Granularity Calorimeters reconstruction; validation and optimization of prototypes with software for HL-LHC, CLIC or FCC-hh
- Reference library of parallel algorithms for clustering
- Development of dynamic occupancy driven domain decomposition for parallel processing with special treatment of high multiplicity regions

Code optimization	algorithmic R&D	(low n) Concurrency	Hardware acceleration	Machine Learning

Data Analysis Challenge

Efficient Analysis Facilities

- We expect a one to two orders of magnitude increase in the number of events at future colliders, as well as an increased event complexity
- In order to maintain short window between data arrival and analysis result, smarter analysis data flows need to be developed
- We need to reduce the overhead of transferring unneeded data, as well reduce the number of times each dataset is sent from storage to compute

Change towards declarative programming model: no event loop

```
ROOT::EnableImplicitMT();
ROOT::RDataFrame df(dataset);
auto df2 = df.Filter("x > 0")
             .Define("r2", "x*x + y*y");
auto rHist = df2.Histo1D("r2");
df2.Snapshot("newtree", "out.root");
```

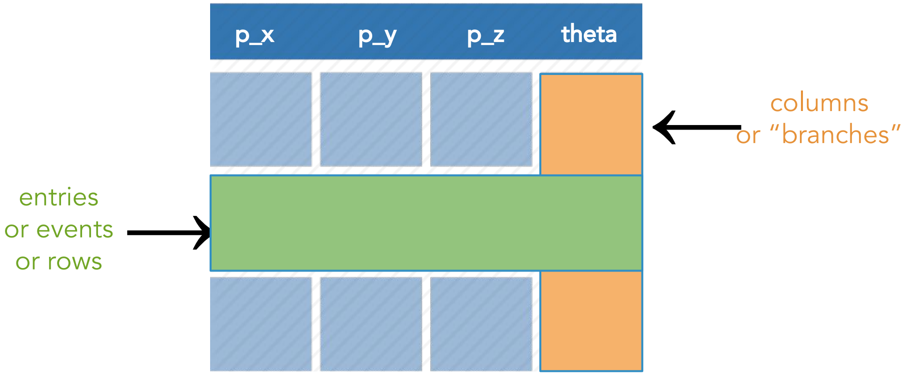
Data Analysis Technology Exploration

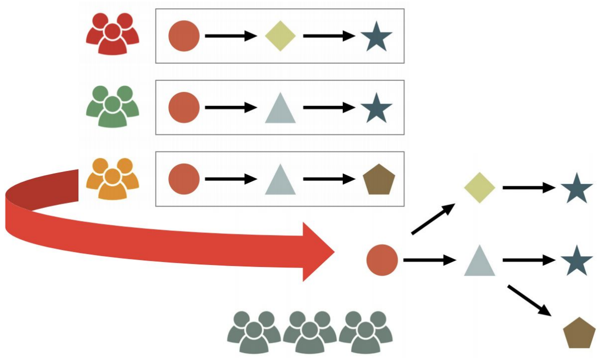
- Foundational R&D investigates data layout improvements in order to optimally exploit parallelism in future storage devices

- In order to scale the data bookkeeping, application software needs to interface directly with object stores (“file-less” I/O, cf. data management R&D)

- Switching from imperative analysis languages (the “How”) to declarative analysis languages (the “What”) allows for automatic query optimization

- Integration with distributed analysis farms enables *sharing* of transient data sets from several users

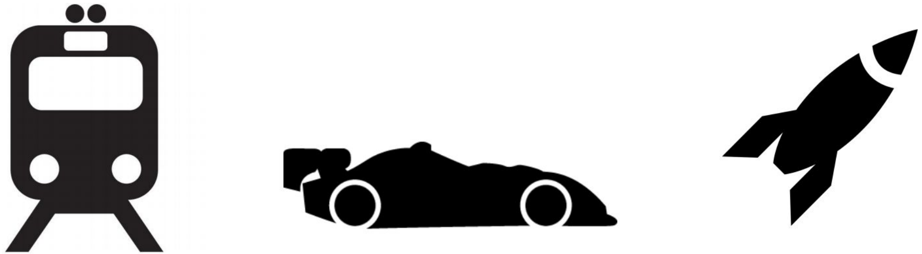




Heterogeneous Computing

Hetero-
geneous
Frame-
works

- Limiting HEP software to CPUs will have a detrimental effect on our physics reach
- Use of hardware accelerators (GPU, FPGA, TPU, ASIC, etc.) follows market driven development in order to keep the computing affordable
- There is a need for next generation frameworks capable of exploring and integrating resources of different types across a large number of nodes



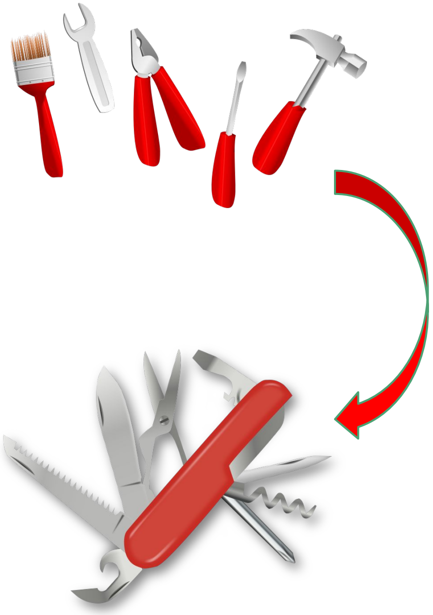
Heterogeneous Computing

Hetero-
geneous
Frame-
works

- Besides the algorithm side, tasks need to be packed and scheduled to CPUs and accelerators such that the utilization of the available hardware is balanced

- Message passing is a promising paradigm to forward compute tasks to heterogeneous compute devices
  - Little or no global synchronization
  - Message passing network mediates differences of latency and throughput

- Numerous challenges to overcome
  - Health and performance monitoring and swift recovery from failures
  - Integration with existing task-based software frameworks
  - Resource pool might not have a fixed size (HLT vs. Cloud)

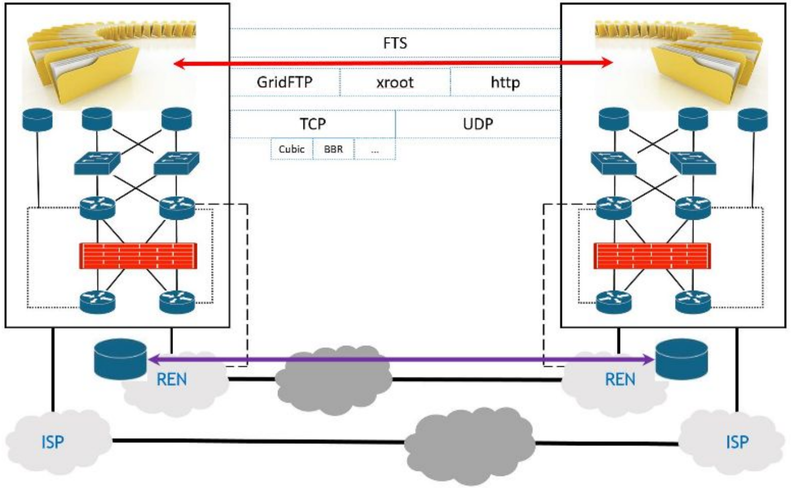


Data management across experiments

Multi-Experiment Data Management

- Data management provides an abstraction of the physical data infrastructure
  - Provide a global federated view on the experiment data
  - Schedules all dataflows on the storage and network
  - Shields users from the details of the resources and provides access

- Scientific infrastructure will be shared between HEP and other big sciences
  - Scale of DUNE and SKA will be HL-LHC equivalent
  - There is not enough storage and network for all experiments to operate *"as-is"*



Currently: resource federation of static allocations

- We move from *static resource allocation to dynamic provisioning*, which requires automated cooperation between experiments

Data management across experiments

Multi-Experiment Data Management

- Dataflow planning across experiments will allow dynamic resource sharing
  - Storage performance classes, e.g., *low-latency*, *throughput*, *archival*, …
  - Data lifetime classes, e.g., *permanent RAW*, *intermediate compute results*, …
  - Data types classes, e.g., *files, objects, events, …* *(cf. Analysis R&D)*
  - Network routes and flow configuration, e.g., *SDN, NFV, …*
  - Subject to computational requirements

- As an example, idle data sets of experiment *A* might temporarily migrate to archival storage to provide a short term resource lease for an analysis workflow of experiment *B*

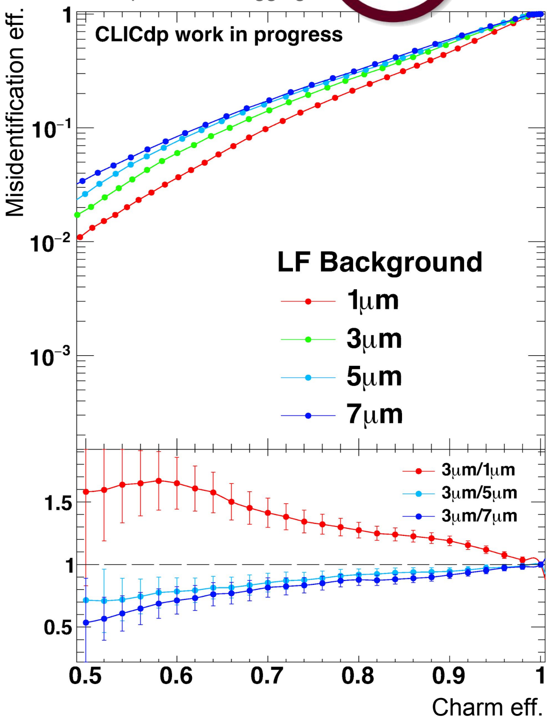
- Experiment software groups in EP have spearheaded the development of data management systems, e.g., *Rucio*, *DIRAC*, *AliEn*

Turnkey Software Stacks

Turnkey Software Stacks

- Future detector studies critically rely on well-maintained software stacks to model detector concepts and to understand a detector's limitations and physics reach
- We have a scattered landscape of specific software tools on the one hand and integrated frameworks tailored for a specific experiment on the other hand
- Aim at a low-maintenance common core stack for FCC and CLIC that can "plug-in" a detector concept under study
    - Relying on common detector description toolkit, (partially) common EDM and reconstruction algorithms, common framework

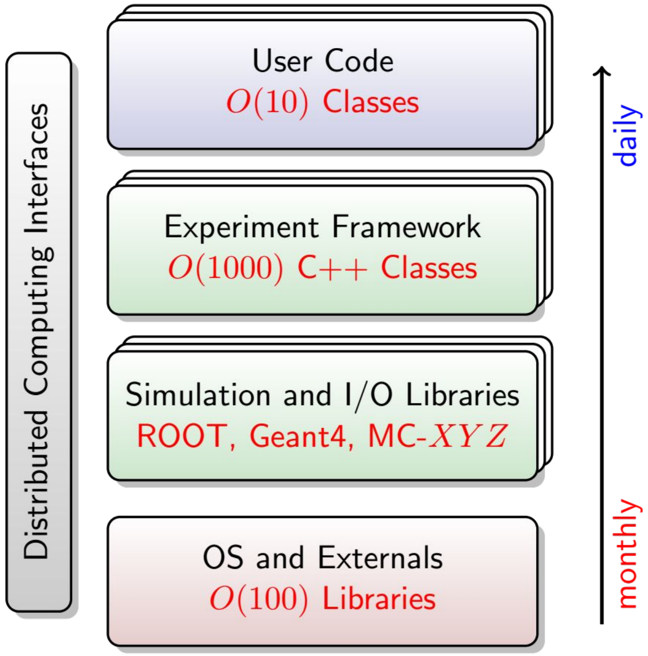
Example: charm tagging



This stack provides a natural test bed for the other R&D activities

Software Integration

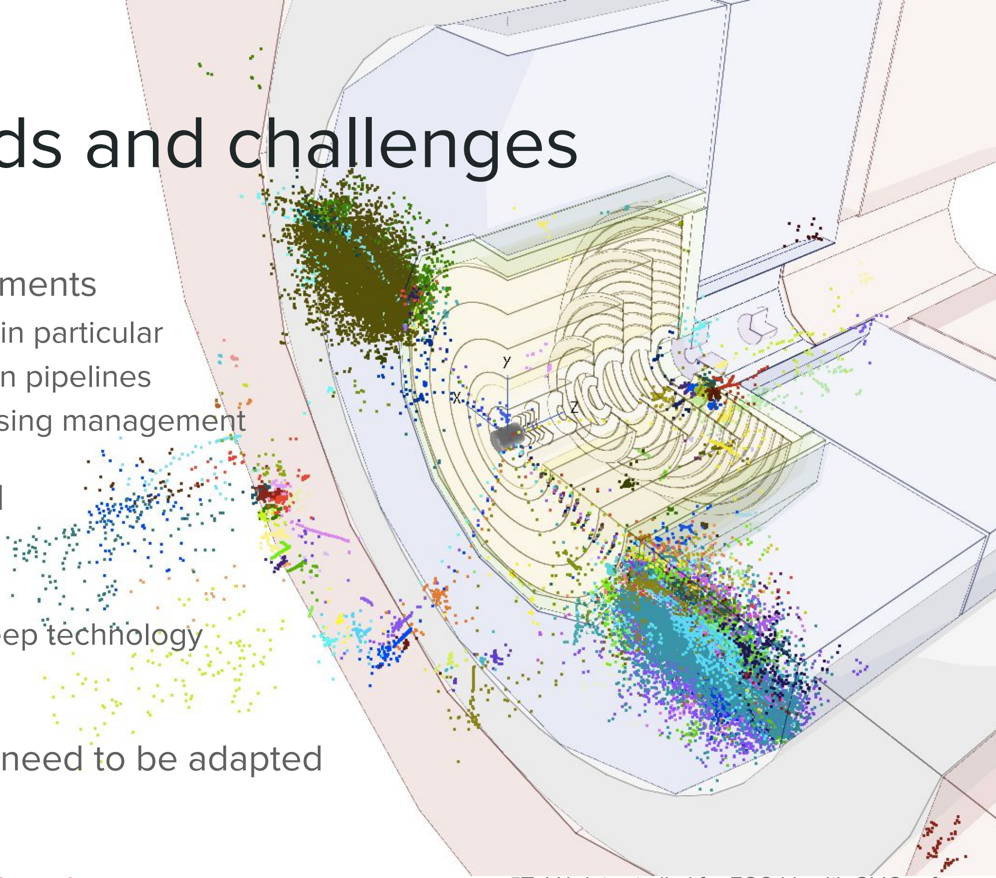
Turnkey Software Stacks



- Evolution and management of massive code bases created over many years
    - Current software is the base from which we design future detectors
- A non-trivial integration problem
    - Millions of lines of code
    - Thousands of classes
    - Hundreds of packages
    - Half a dozen languages
- Benefits from continuous improvements by the industry standard software engineering toolbox

Conclusion: software needs and challenges

- Meet the software challenges of future experiments
  - Very complex events - hard for reconstruction in particular
  - High rates - efficient, high speed data reduction pipelines
  - Huge volume - massive scale data and processing management
- Landscape for software becomes more varied
  - No more ‘free lunch’ from Moore’s Law
  - Harder to exploit hardware -
    need to adapt to compute accelerators and deep technology
    stack for data flow
- Advances from other fields are promising but need to be adapted
- These are not problems that can be solved without investment
  - Software R&D program, running alongside detector R&D itself



5TeV b-jets studied for FCC-hh with CLIC software