Synergies with US ATLAS HL-LHC R&D

Torre Wenaus (BNL)

US ATLAS Facilities Workshop
ANL
December 5 2018

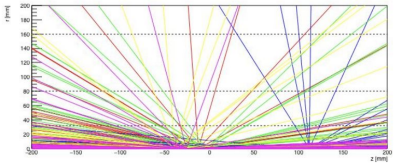
BROOKHAVEN

CERN

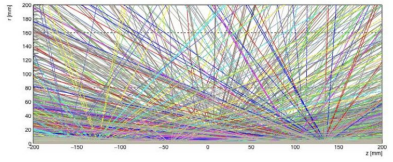
High Luminosity LHC (HL-LHC)

Starting in 2026, the LHC enters a new era

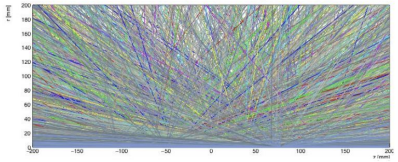
- 5-7x LHC original design luminosity
- Extensively upgraded, more complex detectors
- A 10+ year program of precision and discovery physics with a ten-fold increase in integrated luminosity



2010, <μ>=5



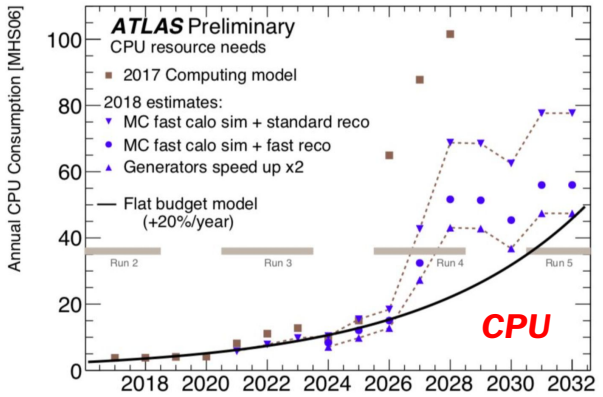
2018, <μ>=40

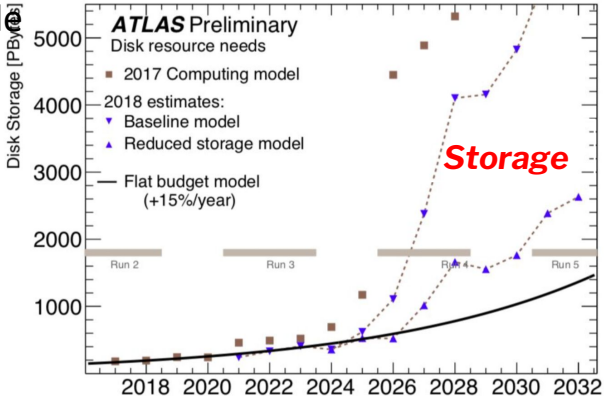


2026, <mu>=200

S&C challenges for the HL-LHC

- Up to **10x the event rate**: much more data, and more complex events
- Pile-up of up to **200 proton-proton interactions** per beam crossing
  - Conventional tracking approaches lead to a combinatorial explosion
- Flat-budget hardware improvements fall **well short of requirements**, extrapolating present computing model
  - Must evolve our processing and data management approaches
- ATLAS is already today **compute-limited** in its science; computing must scale with data volume and complexity or we leave physics on the floor
- Heavily **data-intensive and distributed**, and will become moreso
  - Must feed our often **I/O-bound applications** with data efficiently at scale in the distributed environment
- HEP **software is too serial** for future architectures
  - HEP code is ~1 op/cycle, HPC code ~4, vector instructions up to 8
  - Major re-engineering *beyond multithreading* required to maximize benefit from modern & future CPUs (vectorization, pipelining, accelerators)
- **We cannot afford to buy our way out of the problem with hardware -- we must innovate in software**





Assumptions in the new estimates

Model parameters -- CPU estimates

	2018	2025	2026	2027	2028	2029	2030	2031	2032
Max Luminosity (10^{34})	2.14	0	3	5	5	0	0	7.5	7.5
Integrated luminosity (fb^{-1})	60	0	100	200	300	0	0	450	450
Number of data events	8B	0	40B	50B	60B	0	0	73B	73B
Running time 10^6 s Average trigger rate (kHZ)	7.3 1	0 0	5.3 7.0	7 7.5	7.3 8	0 0	0 0	7.3 10	7.3 10
<mu>	40	0	100	140	140	0	0	200	200
MC events (B)	15	80	80	180	300	300	300	300	300
Fast/Total ratio	1/5	3/4	3/4	3/4	3/4	3/4	3/4	3/4	3/4
Evgen HS06s	1000	1000	1000	1000	1000	1000	1000	1000	1000
Full Sim HS06s	3250	4000	4000	4000	4000	4000	4000	4000	4000
Data Reco HS06s	230	800	800	1000	1000	1000	1000	1200	1200
MC digi+reco HS06s	508	1600	1600	2000	2000	2400	2400	2400	2400
FastChain HS06s	--	700	700	700	700	700	700	700	700

Storage models used

For MC: Assume the size is 1.5x data
Run two scenarios:
a) Parameterization as in the fit - 30% (AMSG-R3 goal)
b) "Reduced storage model":

Reduced Storage model

- DAOD / AOD ratio for MC = 0.31
  (eg 30% for derivations and 1% for a common analysis format) **vs. 0.8**

- Only one version (**vs 2 versions**) of MC AOD and DAOD is on disk.
  1 copy for AOD and 2 copies for DAOD. Some work will have to be done from tape

- No copies of the previous year simulation kept on disk **vs 1 AOD version** from the previous year

- Only one version of current year data AOD **vs 2 versions**
  (eg remove prompt-AOD from disk as soon as derivations available)

ATLAS computing upgrades towards HL-LHC

2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4
Run 1: 7-8 TeV, 0.7×10^{34} (μ≈20), 25 fb^{-1}			**LS1**			**Run 2:** 13 TeV, 1.9×10^{34} (μ≈55), 150 fb^{-1}			**LS2**

2020	2021	2022	2023	2024	2025	2026	2027	2028	2029
Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4
LS2	**Run 3:** 14 TeV, 3×10^{34} (μ≈80), 300 fb^{-1}			**LS3**			**Run 4:** 14 TeV, 7.5×10^{34} (μ≈200), 3000 fb^{-1}		

Upgrade	Shutdown	LHC Luminosity Target	Main ATLAS S&C Changes
Phase-I	2019-21 ***LS2***	2-3 × design*	• Multithreaded framework AthenaMT in production • Updated analysis model: study group active • Software upgrades for new detectors • **Expanded role for fast simulation** • **First exascale HPC experience**
HL-LHC (Phase-II)	2024-26 ***LS3***	5-7.5 × design*	• New tracking software capable of pileup 200 • Fast HPC-optimized generators • **Further expanded role for fast simulation** • **Wholesale integration of ML, accelerators, ...** • **"Data lake" based data + workflow management**

* Original Luminosity Target = 1×10^{34} $cm^{-2}s^{-1}$

Leaping over familiar ground

- Skipping (mostly) the challenges for CPU, for storage, the complexities of HPCs, …
  - Moved to supplementary slides
- Going directly to:
  - Walking through the WBS 2.4 “HL-LHC Computing” structure
  - How it is presently populated with R&D topics
  - **Where there is synergy with facilities**
- Based on the draft WBS breakdown in our working document, to which many have contributed, including of course the L3 managers
- I take responsibility for the specifics of what I’m presenting and how I’m presenting it: a view for discussion!
- It is very early, all is open to change

Organizing HL-LHC R&D in US ATLAS

How US ATLAS is organizing itself for HL-LHC R&D: the “HL-LHC Computing” WBS activity, WBS 2.4

Level 5 items appearing in the WBS are areas in which work is underway or soon will be

2.4.1 Software reengineering and algorithm development

- **2.4.1.1 Event generation**
  - 2.4.1.1.1 Event generators on next-gen HPCs
- **2.4.1.2 Simulation**
  - 2.4.1.2.1 Fast calo sim on accelerators
  - 2.4.1.2.2 Fast chain on accelerators
  - 2.4.1.2.2 Machine learning for simulation
- **2.4.1.3 Reconstruction**
  - 2.4.1.3.1 Algorithm development for HL-LHC
  - 2.4.1.3.2 Algorithm development for accelerators
- **2.4.1.4 Analysis**
  - 2.4.1.4.1 Clusters designed for analysis
- **2.4.1.5 Framework and services**
  - 2.4.1.5.1 Adaptation of framework and services to handle accelerators
  - 2.4.1.5.2 Development of EDM and data layout for accelerators and next-gen architectures
- **2.4.1.6 Event I/O and persistency**
  - I/O planning will be elaborated soon

2.4.2 Workflow porting to new platforms

- **2.4.2.1 Common platform support infrastructure**
  - 2.4.2.1.1 Performance measurement and monitoring
  - 2.4.2.1.2 I/O infrastructure
  - 2.4.2.1.3 Information systems
  - 2.4.2.1.4 Scalable Systems Laboratory
  - 2.4.2.1.5 Release installation tools
  - 2.4.2.1.6 Resource provisioning
- **2.4.2.2 HPC and exascale platforms**
  - 2.4.2.2.1 Implementation of reconstruction workflows
  - 2.4.2.2.2 Implementation of derivation workflows
  - 2.4.2.2.3 Implementation of simulation workflows
  - 2.4.2.2.4 Implementation of distributed training workflows
  - 2.4.2.2.5 Adaptation of workflows to exascale
- **2.4.2.3 Cloud, commercial and other platforms**
  - 2.4.2.3.1 Commercial cloud R&D

2.4.3 Distributed computing development

- **2.4.3.1 DOMA (data organization, access and management**
  - 2.4.3.1.1 Fine grained dataflows
  - 2.4.3.1.2 DDM integration for HPC dataflows
  - 2.4.3.1.3 Data movement, caching and access
  - 2.4.3.1.4 Services for storage tiering (SSD, disk, tape)
  - 2.4.3.1.5 Data lake and CDN services and interfaces
- **2.4.3.2 Workload and workflow management**
  - 2.4.3.2.1 Fine grained workflows
  - 2.4.3.2.2 Workload/workflow management integration with DOMA
  - 2.4.3.2.3 Distributed training tools
- **2.4.3.3 Analysis services**
  - 2.4.3.3.1 Analysis support as a service
- **2.4.3.4 Common infrastructure**
  - 2.4.3.4.1 Site standardization
  - 2.4.3.4.2 Information services

As Kaushik said, distinctions with other parts of WBS are topical, not people; same people share effort across this and other WBSes. ie: built for synergy

R&D activities in our current WBS: 2.4.1

Bold: prospects for R&D synergy with Facilities

- 2.4.1 Software reengineering and algorithm development
  - Event generators on HPCs. Requested effort to work on matrix element calculators usable by multiple generators (starting with Sherpa). Couple to Taylor's SciDAC
  - FastCaloSim and FastChain on HPCs and accelerators
    - Working group with BNL CSI underway to start addressing this
    - Effort underway in ATLAS simu group on using ML (GANs) in fast sim
      - **Reference for later: GANs are very CPU expensive. Benefit from distributed learning.**
  - GPU-enabled reco algorithm development in AthenaMT
    - Early stages of exploring how to develop GPU-enabled code in C++. Scott S, Attila have reported in recent core software meetings. Early impression: tool stack is immature.
    - Reco team is consumed between now and Run-3 with AthenaMT migration and ACTS (new tracking) integration. Very little prospect of substantial new innovations reaching production.
    - And even if GPU-enabled algorithms were integrated, how much of an improvement to the overall walltime? How much GPU utilization? Also topics of study.
  - **Analysis: clusters designed for analysis.**
    - **Enormous synergy here in an exciting new area, as just discussed.**

R&D activities in our current WBS: 2.4.1

- 2.4.1 Software reengineering and algorithm development - continued
  - **Event data model and data layout for accelerators, next-gen HPCs.**
    - **Large synergy here with optimizing layouts/formats for transport/streaming on the wire, data transformations in moving from disk->tape, how layout relates to caching hierarchy, integrating (de)compression in workflows to hide latency, ...**
  - **Event I/O and persistency**
    - **Loads of synergy**
    - **Scalable intra-machine data flows and I/O on HPCs and exascale**
    - **scatter-gather operations to supply processors with data and to collect data from them**
    - **Transient, persistent and streaming data organization to optimize for HPC/exascale dataflows, streaming dataflows, GPU data formats and flows. (c.f. layout above)**
    - **Persistent data organization to leverage emerging storage platforms and paradigms**
    - **Storage footprint minimization consistent with efficient I/O**

Parenthesis: Thinking outside the box

- In the HL-LHC S&C R&D meeting at UTA last month, had an interesting brainstorming session
- Discussed alternative to conventional AthenaMT
  - Extract a GPU-capable algorithm (e.g. CaloCell creation) to a lightweight kernel that processes all events in a task
  - Amortizes GPU overheads over huge number of events, gives GPU lots of work to do
  - Leverages AthenaMT as the overall orchestrator and to define algorithm behavior precisely
  - Straightforward (in principle) to validate: outputs identical to the original
  - If algorithm is standalone, why require that it's C++ with a weak GPU toolkit? Make it possible to use python + SciPy, with excellent GPU support and performance?
- Would have big implications for workflow and dataflow
  - Builds directly on our plans for event streaming service, but lots to build
  - Deliver (only) the data needed for CaloCell building
  - Orchestrating the disjoint workflow of AthenaMT => standalone GPU stack => AthenaMT
    - Do it dynamically as one workflow or in distinct stages?
- Felt by those in attendance to be worth some R&D
- Would certainly be fun and interesting, and would have synergy with Facilities
  - Streaming data delivery of sub-events: basically let's get going with event streaming
  - Role of caching
  - Workflow orchestration

R&D activities in our current WBS: 2.4.2

- **2.4.2 Workflow porting to new platforms**
  - **Fully synergistic**
  - **This is really all about HPCs, at least at the moment.**
  - **New workflows on existing machines**
    - **Porting reco, derivations, fast simulation/fast chain, distributed training services, … to existing machines**
  - **New machines**
    - **Porting all our HPC workflows to new HPCs**
  - **Transition from 2.4 to Facilities is when a new development is production-capable and begins integration, on its way to production and stable ops**
  - **Near(ish) term specific examples:**
    - **Multi-threaded Geant4 in production on HPCs**
    - **Reconstruction, derivation workflows on HPCs?**
      - **With what priority? To what practical gain?**
    - **Start playing with fast chain on HPCs as soon as there's something to play with**
    - **Distributed training service development over the next year**

HPC remarks

- Why do we pay so much attention to them?
  - Agency mandates, scale, compute-hungry culture
- Why are they so difficult to use? So expensive in development effort and labor?
  - Not all of them are like this. European ones aren't, NSF ones aren't.
  - The DOE builds very big (too big to ignore) very complex machines that weren't designed to be readily usable for our computing
- With our ATLAS culture of harvesting any and all resources we can find, and on the foundation of distributed computing that can cope with the heterogeneity, we've made ourselves experts at leveraging HPCs, despite the difficulty
- This has positioned us well for the "mandate" part, which is intensifying dramatically
- Now we have to become experts at using HPCs **with accelerators**

HPCs in HEP: US DOE view

What We've Learned So Far

- HPC architectures will continue to evolve, but moving to vectorized, multithreaded codes tailored to I/O-bound systems will result in higher efficiency codes
  - Engaging HPC experts to analyze code has helped identify algorithm alternatives and data flow bottlenecks, in some cases resulting in spectacular speedups (e.g. 600x). Continued engagement is therefore essential!
  - Need to identify which codes could benefit the most
- Using Exascale machines badly (e.g. by ignoring the GPU/accelerator) will result in a factor-of-40 penalty in performance that will not be tolerated. HEP will lose its allocations if it does this.
  - Engaging Exascale Computing Project (ECP) experts early and often will result in faster adoption of best practices for exascale machines, and influence ECP design choices to HEP's benefit. HEP needs a coordinated interface to both ECP & the Leadership Computing Facilities.
  - Need to identify which codes could benefit the most
- LQCD regularly rewrites its code, has reaped significant speedup benefits every time
- Reinforced that multiyear NERSC allocations & better metrics for pledges are needed
- End-to-end network data flow models are needed to support tradeoff analysis of storage vs. CPU vs. network bandwidth on a system-wide and program-wide basis
  - Greater sharing of the underlying data management software layer may also be beneficial

U.S. DEPARTMENT OF ENERGY | Office of Science

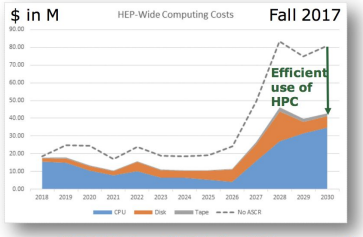
DOE HEP Status at HEPAP - May 2018 28

We must use them properly
(use the accelerators)

We must use them heavily

Updated HEP Computing Model

- In preparation for the Inventory Roundtable, the largest HEP experiments from all three frontiers were asked to provide a **more detailed estimate** of their expected computing needs
  - CPU, storage, network, personnel, and HPC portability
- Cost estimates for all experimental frontiers:
  - "Business as usual" (minimal additional HPC use): **$600M ± 150M**
  - With effective use of HPC resources this reduces to: **$275M ± 70M**
- By 2030 cost share by frontier is estimated to be:
  - ½ Energy Frontier
  - ¼ Intensity Frontier
  - ¼ Cosmic Frontier
- **A strategy encompassing all HEP computing needs is required!**



U.S. DEPARTMENT OF ENERGY | Office of Science

DOE HEP Status at HEPAP - May 2018 29

Jim Siegrist, HEPAP meeting, May 2018

Accelerator utilization

- DOE mandate drives an increased level of urgency to this, since
- **We're not in a position today to use accelerators at large scale in offline**
  - ATLAS has no offline production applications today that utilize GPUs
  - Similarly for other LHC experiments (we're all working towards it)
- The DOE position and the reality of GPU-rich machines prompted US ATLAS action
- In June 2018 a new "HL-LHC Computing" activity area was created in US ATLAS
- Its first action was a workshop in July that brought together HPC experts from BNL's Computational Science Initiative (CSI) and a team of senior ATLAS software developer / physicists
  - To look for **GPU and ML applications for exascale** and identify projects
- Then a followup meeting at UTA last month

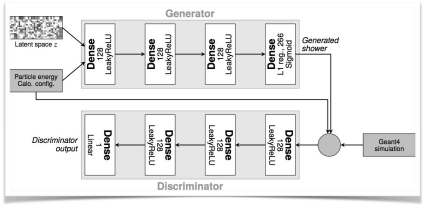
Leveraging Exascale + accelerators

- The workshop concluded that a promising route for ATLAS to exploit exascale in 2021 -- including, crucially, the use of accelerators -- is via ML applications, in particular
  - **Fast simulation**, and particularly **fast chain** (fast all the way to analysis outputs)
  - Tracking, in which there are a number of ML efforts, but not soon
- And, **scaling ML applications** to utilize large scale resources in order to minimize turnaround time in network development and tuning
  - **Distributed training** to achieve fast turnaround
    - Presents the possibility of bringing ATLAS workload management tools to bear (PanDA, pun not intended)
    - Large scale orchestration of parallel processing, with management of associated data flows and metadata
- Accordingly, the workshop convened fast simulation, distributed training (and tracking, but not active) working groups now underway

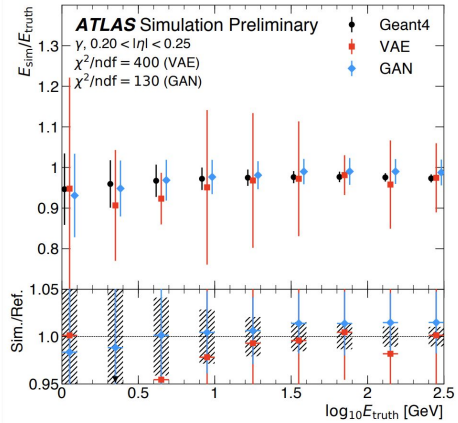
Working towards exascale

- Simulating events in ATLAS is the largest CPU consumer: about 50%
- ATLAS Run-3 objective is to **use fast simulation for most simulation needs**
  - Uses parameterised models of detector response (in particular calorimetry) to achieve a 10x speedup
  - **ML, particularly GANs**, well suited to developing high quality detector response models, with projects now in development, e.g. CaloGAN
- **Fast chain is the real target:** adds fast reco to fast simu to make the whole chain fast with reduced storage requirements and I/O intensity not much worse than full simu, ie HPC compatible
- **Developing, tuning and (re)training of networks for these applications** will be a compute intensive process that could be well-suited to exascale
  - Leverages the scale of the machine to minimize turnaround time
  - **Spiking for fast turnaround** rather than steady state for large throughput
  - *Will the demands of training be enough to benefit from exascale? Anecdotally, seems to be yes, for GANs in particular*
- Can we benefit from exascale for fast simulation proper as well as training?
  - *Will ML inference in a fast chain workflow use enough GPU to benefit from exascale? Would enable steady state, large throughput usage*

GAN

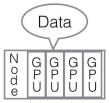


Early results

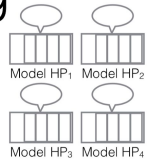


Distributed Training

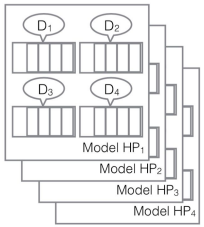
1. *Tensor operation parallelism*: GPUs, FPGA, and ASICs (Google's Tensor Processing Unit).
   - Note additional HN, Data, Model parallelism with multi-GPU

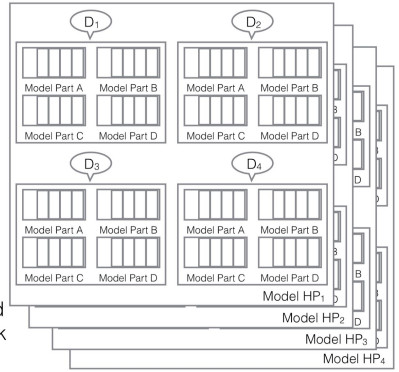


2. *Hyper-parameter scan*: simultaneously train multiple models. *e.g.* 1 model per GPU or node.



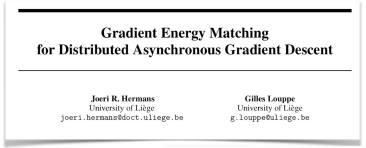
3. *Data Parallelism*: Each GPU or Node computes gradient on sub-set of data. Sync'ing gradients bottlenecked by bus or network.





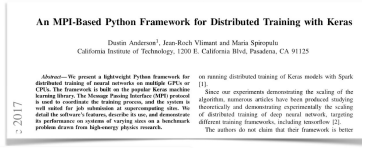
4. *Model Parallelism*: Large model spread over many GPUs or nodes. Less network traffic but only efficient for large models.

- Existing tools and techniques:
  - *Horovod:* from Uber.
  - *MPILearn:* from HEP colleagues at CalTech
  - From Giles' group: https://arxiv.org/pdf/1805.08469.pdf
  - From LBL/NERSC colleagues: https://arxiv.org/pdf/1708.05256.pdf
- In general, scaling to large number of nodes can be difficult.



Gradient Energy Matching for Distributed Asynchronous Gradient Descent

Joeri R. Hermans, University of Liège, joeri.hermans@doct.uliege.be
Gilles Louppe, University of Liège, g.louppe@uliege.be



An MPI-Based Python Framework for Distributed Training with Keras

Dustin Anderson, Jean-Roch Vlimant and Maria Spiropulu
California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125

Deep Learning at 15PF: Supervised and Semi-Supervised Classification for Scientific Data

Thorsten Kurth, Jian Zhang, Nadathur Satish, Ioannis Mitliagkas, Evan Racah, Mostofa Ali Patwary, Tareq Malas, Narayanan Sundaram, Wahid Bhimji, Mikhail Smorkalov, Jack Deslippe, Mikhail Shiryaev, Srinivas Sridharan, Prabhat, Pradeep Dubey

A Vision

- User sets up a training session in similar manner as current GRID submissions.
  - Define resources required. e.g. CPUs / GPUs per training job.
  - Define training data samples. Use Distributed Data Management system (i.e. Rucio).
- Hyper-parameters and measured optimization metrics on test/validation samples are book-kept and reported to PANDA via appropriate API.
- Hyper-parameter optimization can be either provided as part of service or run externally via the API.
- Processing performance metrics (e.g. time per epoch) are also reported to the system via API and monitored.
- Trained Models and any results (e.g. plots), are stored in DDM.

Short Term Plan

- Identify 2 types of problem:
  1. *Quick training:* few hours training times where large scale Hyper-parameter optimization is warranted.
     - Build out system for training submission, hyper-parameter management, and monitoring.
  2. *Long training:* days/weeks of training where distributed training is warranted.
     - Study scaling / speed up and integrate into system.

R&D activities in our current WBS: 2.4.3

- **2.4.3 Distributed computing development**
  - **Also fully synergistic; basically no daylight between 2.4 and 2.2**
  - **Building a data delivery infrastructure from tape to disk to data lake CDN to caching hierarchy to consuming client**
    - **Data carousel as a joint Facilities/DDM/WFMS system**
      - **Should have a robust production at-scale version of data carousel by Run-3**
    - **Data transformations during tape => disk to optimize layout/format for client delivery**
    - **Data (event) streaming, interaction with caches**
    - **What is the CDN?? What does it look like, what does it do, how do clients interact with it?**
    - **Event streaming service at data (server) and client (pilot) ends**
  - **Data flows to/from HPCs**
  - **New/extended metadata and information services in support of new workflows, services, platforms**
    - **e.g. in support of distributed training service**
    - **Metrics, monitoring & analytics**

Finally

- Many synergistic activities “now”
- They should involve everyone with interest and expertise
  - The team is the same, share the fun
- Broad themes:
  - Algorithms & framework
    - Synergy mostly in data layout, I/O aspects particularly as needed for data streaming and HPCs
    - Out of the box thinking: standalone GPU algorithm crunching
  - Porting to new platforms
    - Expanding to new workflows on existing machines and to new machines
  - Distributed software
    - Building the data delivery system, with tight interplay between DDM and WFM, from tape drive to event streaming client consumer
      - Build a full prototype data lake and play with it
    - Extending the capabilities of our distributed software to new applications
      - e.g. distributed training service

Sorry to not be there at ANL as I’d planned… talk to many of you next week!

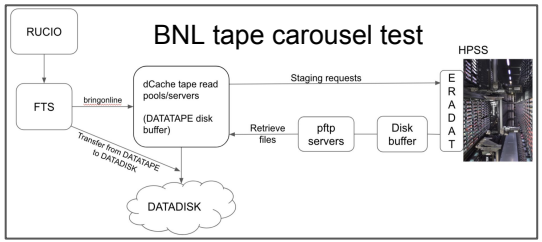
Some related activities & materials

- GPU hackathon series latest one this week at BNL (DOE sponsored)
- ANL Aurora A21 early science program, ANL HEP selected for participation
- ATLAS / CSI workshop on development towards exascale, BNL, July 2018
  - Simulation software: fast and full, Heather Gray
  - Proposals, Amir Farbin
  - Scaling DNNs using HPCs, Abid Malik
- Data intensive science at LCFs, Jack Wells (ORNL), June 2018
- BigPanDA for Titan and Summit early science program, A. Klimentov, July 2018
- Connecting the Dots workshop series on advanced tracking
- Kaggle TrackML ML tracking challenge
- IRIS-HEP kickoff workshop

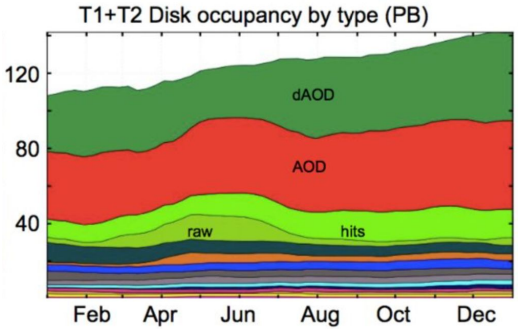
Supplemental

Scaling to HL-LHC: Storage

- ATLAS disk is already ‘always full’ before facing the HL-LHC shortfall
- ‘Opportunistic storage’ basically doesn’t exist
- Working on format size reductions, but hard to achieve large gains
  - 30% reduction is target of current study group
- Replica counts already squeezed, hard to achieve large gains
- **Storage is our biggest cost component and biggest challenge. We need new approaches!**
- ATLAS disk usage is ⅔ analysis formats and ⅓ everything else
- A way to **dramatically reduce our storage footprint** is to **grow the use of tape** (it looks like our cheap storage will remain tape)
  - Use a ‘**tape carousel**’ approach for the analysis formats
  - A moving window of say ~10% staged to disk at any one time
- **This is hard:** tape is slow and complicates workflow orchestration
  - Analysis workflows are time critical and already complex
- Tape is **geographically limited**, while processing happens everywhere
  - Remote processing must be efficient
- **Fertile ground for R&D**, now underway



ATLAS disk usage today: ⅔ analysis formats



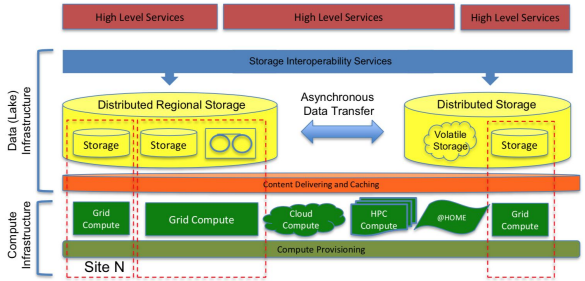
Storage-directed R&D

- (US) ATLAS is a leader in advancing R&D to reduce storage needs
- Storage-directed R&D activities underway:
  - Tape carousel workflows serving data from tape
    - BNL Tier-1 has longstanding expertise in this from RHIC
    - Leverages our workload manager PanDA's tight coupling with data management (Rucio) to orchestrate complex workflows
  - Advanced xrootd based caching for efficiently distributing hot data
    - US ATLAS co-leads the WLCG R&D in this area
    - xrootd's creator and project leader is in US ATLAS
  - Event streaming service for fine-grained, optimized data delivery
    - Next step in the development of the ATLAS event service
    - Make full use of the network to economize storage
      - Send only the data the consuming client needs
    - Process data with WAN latency hiding to efficiently process data being streamed from far away
  - Leveraging commercial hot/cold data storage in a Google R&D collaboration

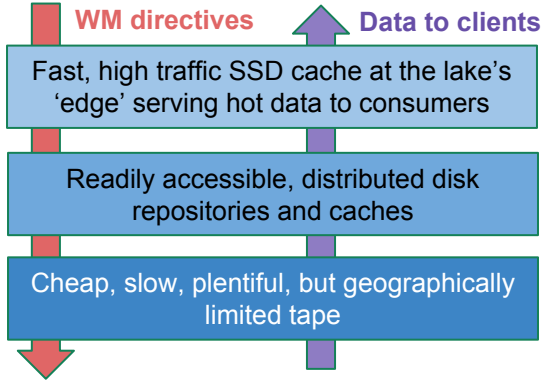
The storage R&D goal: “data lakes”

- Our sites are linked with (ever higher) high-bandwidth networking
  - Networking is growing faster than our requirements; we must make the most of it
- **Data lakes**: integrated consolidation of distributed storage (and compute) facilities, leveraging high-bandwidth networks
- Data lake encompasses facilities with several levels of storage
  - **Tape**, at a relatively limited number of sites
  - **Standard disk**, at large storage repositories and smaller caches
  - Fast SSD ‘**edge cache**’ for the hottest data
  - Should be able to **place data optimally** based on (dynamic) need
- Workload management knows the hot popular data in use
  - Use that knowledge to drive preparing data in the lake, asynchronously to the processing, e.g.
    - tape staging in a **carousel workflow**
    - placing hot data in SSD cache **‘close’ to available CPU**
    - **transforming/marshaling data** optimally for client delivery
  - Requires ‘QoS’ APIs supporting WM directives
- **Instead of O(1) replicas on disk today, manage dynamic availability of actively used data to achieve replica count <<1**

Data lake schematic

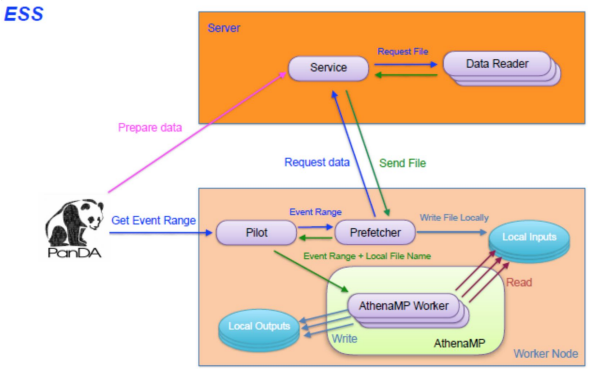


Data lake interactions



Streaming data from the lake

- ATLAS is in the vanguard of developing innovative new *fine-grained streaming* approaches to event processing
  - Very good fit for the data lake model
- The first phase, the Event Service (ES), is in early production for ATLAS simulation
- The next phase, the Event Streaming Service (ESS), is in early R&D
- They are built on the solid foundations of PanDA, XRootD and Rucio
- They allow agile, dynamic and automatic processing and data flows
  - Work goes to the optimal locations of the moment
- They insulate processing from the latencies of the WAN
  - Fetching data asynchronously and as-needed in near real time
- They enable the full and efficient utilization of opportunistic resources and HPCs
  - Fine-grained processing is “the sand to fill the processing cracks”
- They also open the door to the ultimate storage saver (at a CPU cost), “virtual data”
  - Don’t save it, just (re)generate it when you need it
  - Feasible for “fast chain” simulation & reconstruction



Implementing streaming: granular event processing

- Event service + event streaming service = granular processing several benefits
  - Fine grained work assignments: **utilize CPU fully by 'filling the cracks'**
    - Keep all multiprocessing cores busy all the time
    - Keep a ramping-down grid site busy until it's offline,
  - Fine grained inputs: **stream remote inputs asynchronously**
    - Without the up-front latency and complexity of pre-staging big files
    - Without WAN latency impacting the processing
  - Fine grained outputs: **stream outputs asynchronously in quasi real time**
    - Avoid losing good data on a resource that vanishes (spot market cloud, preemptible grid queue, BOINC)
    - Hide WAN latencies, consolidate distributed outputs to one destination
- **Streaming = copying small files**, not a remote open and read across WAN
  - Files are robust, cacheable, easier to diagnose when problems appear, allows pre-marshaling of input data at the source

R&D prospects

- **Event service:** experience has proven the concept, flexible granularity is worthwhile, but more work is needed to complete a clean implementation
  - Goal: one ATLAS workflow with tunable granularity to optimize
  - Currently implemented over a largely coarse-grained foundation
  - Plan to re-engineer to fine-grained all the way down
    - While preserving conventional mode
  - Also enabling fine-grained mode **for use beyond ATLAS??**
- **Event streaming service:** intimately coupled to data lake (and I/O) R&D
  - Very early stage in ATLAS: **good moment to team up**
  - Ideas outlined here nicely expanded in Data Lakes white paper from US ATLAS (R. Gardner et al)
- **Data management via Rucio:** close ally in all this work. Intimate coupling between workload management and data management
  - **Much commonality and collaboration in R&D**

R&D prospects

- **Metadata management**
  - ATLAS Data Characterization and Curation (DCC) group consolidates event metadata activities, with a substantial R&D component
    - Bringing together established systems (e.g. AMI) and new developments (e.g. Data Knowledge Base R&D)
  - New R&D project important to granular processing is the **Event Whiteboard**
    - Database of event (more accurately, object) records with user defined metadata, and collections of them
    - Pursue as basis for metadata needs of event service, ESS, …
    - **Ripe for commonality** (avoid repeat of 7 different file catalogs 15 years ago :-)

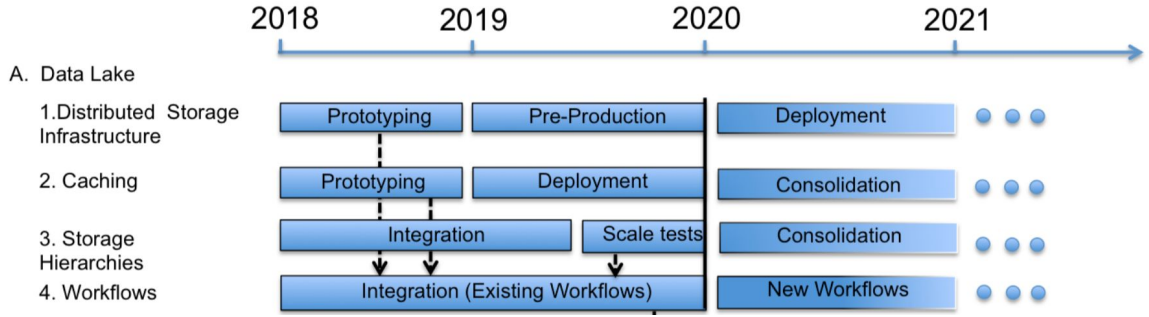
R&D prospects

- Understand **optimal granularity** for data management and access for all storage tiers and workloads. What are the **elementary data objects**, how are they aggregated, moved between tiers, served to the client.
- Related to ESS: storage-side **intelligent data marshalling**, informed by the scientific content of the data and the client's actual needs. What **transformations are applied** to data, how, and where. How do they relate to compression, and effective use of cache hierarchies. Virtual data support.
- In latency hiding, data marshaling & packaging, compression management, caching, … **how much do we do via ROOT**
- Tape based **data carousels** with strong couplings between workload management, DDM, storage for intelligent automation. How fast can they go.
- **Data lake API and service design** supporting workload management automation and thoroughly isolating clients from special knowledge of storage organization. A smooth, intelligent global CDN (or Data Delivery Network)

R&D prospects

- Breaking co-location boundaries. How much will latency hiding & caching allow us to **relax co-location**? Commonality in more relaxed brokerage?
  - Continued value in preferentially processing close to data
- Workload management for **cloud based analysis**
- Mine deep data on system operation for **intelligent automation via ML**
- **Good metrics** to measure performance, efficiency and drive improvements
- Scalable, uniform **resource provisioning** among dynamic diverse resources

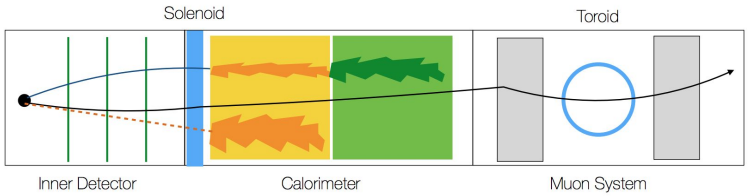
For data lakes (among other things), Ian Bird and Simone Campana have started to lay out an R&D timeline towards writing an informed HL-LHC computing TDR in 2020



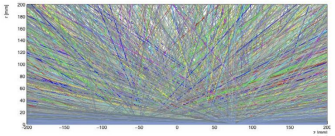
Near term benefits from long-view R&D

- Fine grained processing
  - Full and efficient utilization of **opportunistic resources**
  - **Relaxing co-location**, simplifying data placement and brokerage
- Data carousels
  - **Lessen pressure on storage** resources
- Data lake
  - Use prototypes to provide relatively low-volume but high-value functions like hosting **distributed analysis outputs** in a robust, location-agnostic way
- Metadata
  - More powerful **tools for analysts** to follow and annotate the course of analysis work
- Next-gen resource provisioning
  - Better control and knowledge flow between WM and resource to better **match resources to tasks** and their requirements

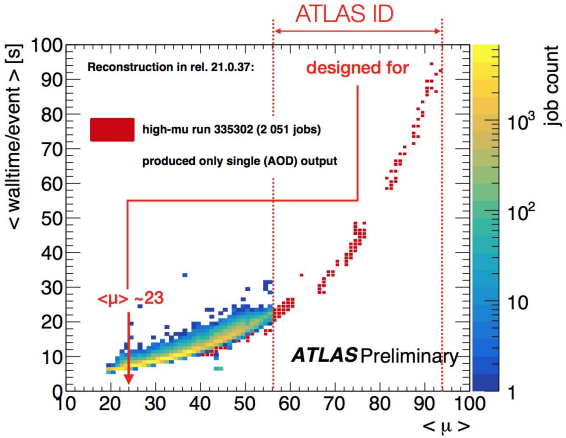
Scaling to HL-LHC: Processing



- Tracking presents the largest CPU challenge algorithmically for HL-LHC
  - Combinatorics of up to 200 overlaid pileup tracks
  - Major R&D activity: fast, accurate tracking at high pileup
- Calorimeter simulation is today the dominant CPU component
  - Major R&D activity: fast calorimeter simulation
- Common denominator to much of the R&D: **machine learning**
- Other improvements underway:
  - Performance optimization
  - Further reconstruction optimizations
  - Use of truth info in MC track reconstruction
  - Pre-mixing of pileup events, reducing digitization time
- Expect **substantial use of HPCs**, with *data intensive* requirements from our use cases
  - ATLAS apps are good HPC citizens: fully parallel MPI applications, keeping every core busy for the full job duration
  - *But*, demands reengineering sufficient to utilize them efficiently, *including their accelerators*

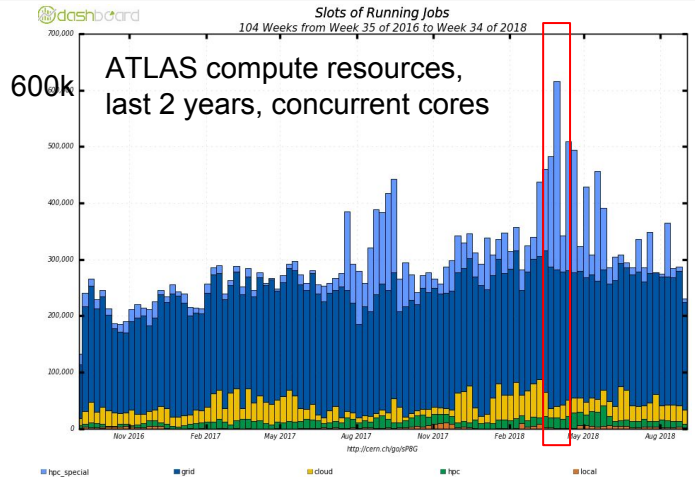


2026, <mu>=200

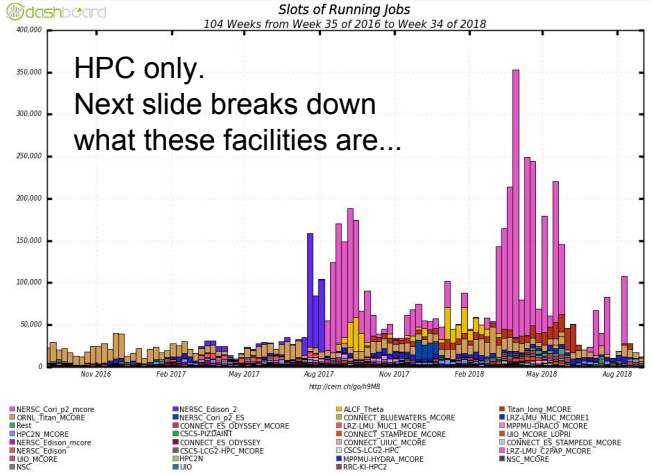


ATLAS reconstruction time dependence on pileup (mu), as measured in data, showing exponential rise due to inner detector (ID) tracking combinatorics

HPCs in ATLAS: deep experience & capability



A long history but a new era in the last year: very large facilities, so far in the US



Light blue: "special" HPCs, where special means big, difficult to use, US DOE

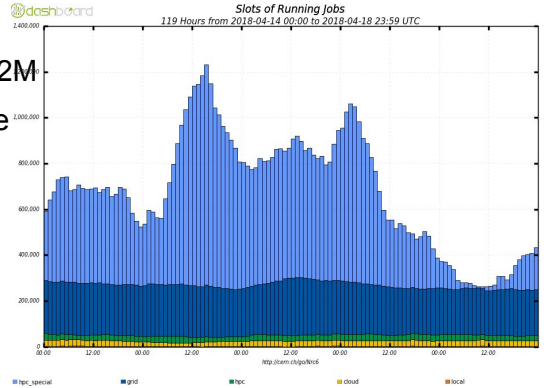
Dark blue: the grid

Yellow: cloud resources including (dominantly) HLT

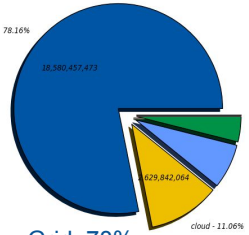
Green: "regular" HPCs, meaning easier to use, European or US NSF

Zoom showing full size of scaling peak: 1.2M concurrent cores.

Our workload management system is highly scalable!



CPU HS06 shares, last year



Grid: 78%

Cloud, HLT: 11%

HPC special: 7%

HPC regular: 4%

Software performance - experience so far

- Compiler/code/build based improvements
  - Investigating (typically unused) advanced compiler features
  - Reducing shared library overheads by building large libs instead of loading many small ones
  - AutoFDO feedback guided optimization
  - Looked at Geant4, reconstruction code, NLO generators (the newer slower ones)
  - Several 15-20% CPU improvements (not all can be combined)
    - HEP code very sensitive to compiler switches; requires substantial physics validation
- Code profiling shows up to 25% of time spent in memory management
  - Allocation and de-allocation of small, very short lived objects
    - 10% can be saved with refactorization
- Hardware counter based analysis: HEP code ~1 op/cycle, HPC code ~4, vector instructions up to 8
  - 100% improvement should be feasible, leveraging vector units, with considerable work: substantial code changes, high level skills
  - Overlap with GPU utilization work
  - ALICE HLT tracking code re-design for GPUs also produced huge gains on CPUs with vector units
- Overall estimate of potential sw performance gains without changing paradigms & algorithms: 200%
  - 50% at moderate cost

from Markus Schulz and the cost model working group

Every experiment is exploring ML in calo and tracking

Generative Models @ LHC

- Every Experiment is Exploring: ATLAS, CMS, LHCb, ALICE

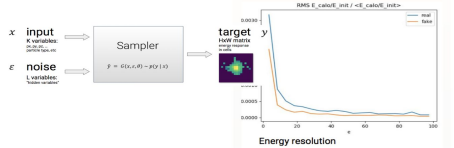
Generative models for fast cluster simulation @ALICE

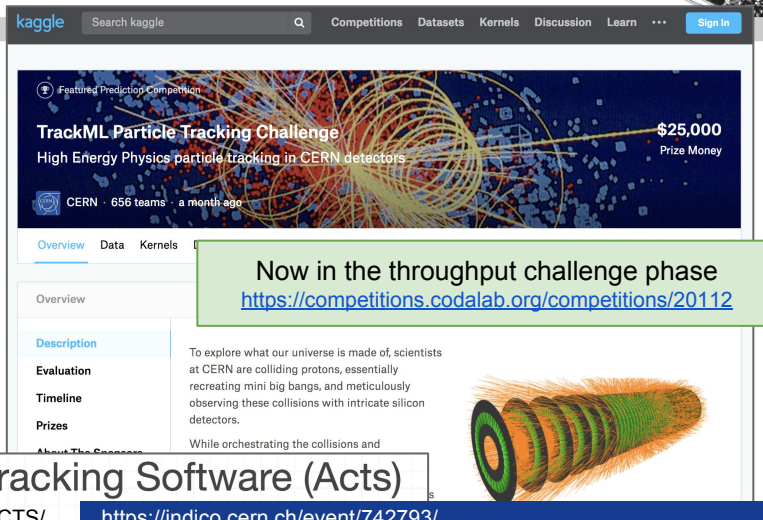
Amir Farbin, July 2018

Most computational expensive step in simulation is the particle propagation ⇒ avoiding the step using generative models

Method	MSE(mm)	speedup
GEANT3	0.085	1
Random (estimated)	166.155	N/A
GAN-MLP	55.385	10^4
GAN-LSTM	54.395	10^4
VAE	37.415	10^4
DCGAN	26.18	10^2
cVAE	13.33	10
proGAN	0.88	30

Fast calorimeter simulation @ LHCb





Now in the throughput challenge phase
https://competitions.codalab.org/competitions/20112

HEP.TrkX

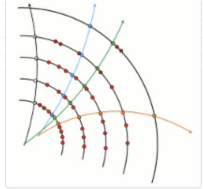
Cross experiment, DOE supported
https://heptrkx.github.io/

about | HEP advanced tracking algorithms with cross-cutting applications (Project HEP.TrkX)

summary | This is an HEP/ASCR DOE pilot project to evaluate and broaden the range of computational techniques and algorithms utilized in addressing HEP tracking challenges. Specifically the project will provide a framework to develop and evaluate new algorithms for track finding and classification, that will be demonstrated by applying advanced pattern recognition techniques to track candidate formation. For example, an optimized track formation algorithm that scales linearly with LHC luminosity, rather than quadratically or worse, may lead by itself to an order of magnitude improvement in the track processing throughput without affecting the track identification performance, hence maintaining the physics performance intact in the LHC upgrades.

A Common Tracking Software (Acts)

http://acts.web.cern.ch/ACTS/



https://indico.cern.ch/event/742793/

CTD/WIT 2019
Connecting the Dots and Workshop on Intelligent Trackers
IFIC, València, Spain
2nd - 5th April 2019

Connecting The Dots / Intelligent Trackers 2019

Common ground for collaboration:

- IML machine learning forum across the LHC experiments
- Community wide HSF software forum

Community collaboration

- No single experiment can go it alone in evolving HEP S&C for the future
- HEP Software Foundation (HSF) created in 2015 to facilitate cooperation and common efforts, particularly in concurrency and reengineering
- HSF coordinated the writing of a community roadmap (arXiv 1712.06982) and series of white papers for HEP S&C that identified R&D directions across the field
- Informed by the community roadmap…
  - The Worldwide LHC Computing Grid (WLCG) is coordinating development of an HL-LHC computing plan, and organizing common R&D activities in distributed computing
  - US NSF has established a far-sighted, broadly scoped project, IRIS-HEP, to pursue R&D directions in innovative algorithms, data management and analysis tools & approaches
- Other such initiatives hopefully coming!

HSF
HEP Software Foundation

WLCG

Institute for Research and Innovation in Software in High Energy Physics (IRIS-HEP)

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

HSF-CWP-2017-01
December 15, 2017

- HSF established in 2015 to facilitate *coordination* and *common efforts* in software and computing across HEP in general
- Charged by WLCG to address R&D for the next decade
- 70 page document on arXiv (1712.06982)
- **13 topical sections** summarising R&D in a variety of technical areas for HEP Software and Computing
  - Backed by topical papers with more details also (e.g. 50-page detailed review about Detector Simulation)
- **1 section on Training and Careers**
- **310 authors** (signers) from 124 HEP-related institutions
- Feature article in CERN Courier
- More details on the HSF web site

Contents

HL-LHC R&D Collaborations: Google

- Google-ATLAS Proof of Concept demonstration project
  - Data transfers and PanDA jobs shown to work transparently between Google cloud and WLCG sites
  - Results presented at NEXT 2018, CHEP 2018 and many talks at CERN
- Expanded R&D projects started in 5 new working groups
  - Track 1: Data Management across Hot/Cold storage
  - Track 2: Machine learning and quantum computing
  - Track 3: Optimized I/O and data formats
  - Track 4: Worldwide distributed analysis
  - Track 5: Elastic computing for WLCG facilities
- All 5 groups co-led by US ATLAS and Google members
- Active interest and participation from international partners
  - CERN IT, CERN OpenLab, WLCG, Tokyo U, UK & EU institutions...