

Practical! Scientific Computing & Future Trends

*“Where we are & a glimpse
of where we’re going.”*

Brad Sawatzky (JLab)



First up:
A Couple Quick Tricks
to make your
Computing Work Suck Less

How to find information

• JLab's web search is . . . not good . . .

→ *Still* working on improving this...

» Baby steps: [ServiceNow SciComp Portal](#) “Knowledge Base”

» [Getting Started](#) and [Experimental Physics User's Guide](#) pages are updated

– Info still not widely searchable...

Getting better as of this week!

→ **Search trick: do this in Firefox:**

» Go to www.google.com and search for 'site:jlab.org foo'

» Right click on the bookmark and choose 'Properties'

– Give it a good name

– Give it a short 'keyword' like 'jj'

– Clean up the Location as shown, replace 'foo' with %s

→ **Now type 'jj ifarm' in URL bar**

» %s in 'Location' string is replaced with text following Keyword

» 'site:jlab.org' is google-fu to restrict search to jlab.org domain

» 'site:jlab.servicenowservices.com' can find (current) KB articles now

Name:	[jj] JLab Search
Location:	http://www.google.com/search?hl=en&q=site:jlab.org%20%s&btnG=Search
Tags:	Separate tags with commas
Keyword:	jj

How to find information

- Trick works great for many things
 - **JLab staff page** (<https://misportal.jlab.org/mis/staff/staff.cfm>)
 - » Keyword: 'page'
 - » Location (can extract from search on 'smith' above):
 - » `https://misportal.jlab.org/mis/staff/staff.cfm?field=all&name=%s&Search.x=36&Search.y=11&Search=Search&field=all`
 - **ROOT / G4**
 - » Keyword: 'gr'
 - » Location:
 - `https://www.google.com/search?hl=en&btnG=Search&q=site:cern.ch%20%`
 - **Stackoverflow.com**
 - **JLab Logbook (a little trickier, but you can work it out)**
 - ...

How to work from Offsite

- How to work from offsite without tearing your eyes out because, holy hell, the graphics and menus are just so slow...

- VNC + ssh tunnel to the rescue

→ VNC: Virtual Network Computing

→ ssh used to securely move VNC traffic through jlab firewall



- Computer Center How-to
 - <https://cc.jlab.org/accessingvnc>
- Old 'howto' I wrote for my collaboration
 - adapt to machine you use
 - Search: 'jj vnc session'
 - » Pick: [Using a VNC Server/Client](#)

How to work from Offsite

- How to work from offsite without tearing your eyes out because, holy hell, the graphics and menus are just so slow...

- Virtual Desktop Environment (VDI)

→ <https://vdi.jlab.org>

→ Fewer “hoops” than VNC, but...

- » limited number of ‘slots’ available
- » logins are not as persistent



- Computer Center How-to

→ <https://cc.jlab.org/remotearchive>

- Use screen, tmux for terminal sessions

- » Maintains ‘state’ if you disconnect.
- » Can reconnect from anywhere

Offline Analysis Farm Usage / General JLab Computing

Nuts to the Farm, I analyze on my Desktop

- Simple tasks, some analysis OK on the desktop, BUT!!
 - Thou shalt backup your code!
 - Thou shalt backup your results!
 - Who among us has done
 - % rm -rf stuff/
 - » Followed by !@#\$?
 - Don't keep only copies on your laptop
 - Don't keep only copies on your desktop's hard drive
 - Do use git for all code and scripts!
 - Commit early, commit often
 - 'git push' often too!
 - » It's a backup!
 - Hard drives die and the data are gone.
 - Drives are large and cheap
 - But reliability on consumer drives is worse that it used to be!
 - SSDs are (weirdly) no better!
- IF your hard drive died today, how long would it take to recover?
 - » a day, a week,
 - » a month???

JLab Systems Exist to Support You

- **/home, /group** are automatically backed up
 - They are snapshotted hourly!

```
% cd .snapshot/  
% ls -lrt
```
 - Longer term backups are on tape
- **/mss**
 - “Index” of what is on tape (not actual files)
- **/cache**
 - Actually access files on the tape system
- **/work, /volatile** are on heavily redundant filesystems
 - NOT backed up
 - » Use tape
 - More on this later...
- **NOTE:** Your JLab RHEL system can mount these directories if needed
 - Talk to me if this would help

File Systems: Where do I put my stuff?

- JLab SciComp/IT provides
 - /group – a space for groups to put software and some files, backup up by CST
 - /home – your home directory, backed up by CST
 - /cache – ‘mirrors’ files backed by tape system so you can use them
 - /volatile – acts as a scratch space for large output
 - /work – unmanaged outside of quotas & reservations; no backups; bigger and faster than /group

The JLab Farm • Power at your Fingertips

- Farm has many pieces
 - ~30000 compute cores
 - ~6 PB Lustre
 - ~5 PB NFS/XRootD (ZFS)
 - ~100+ PB of Tape
 - Consumes ~400kW of power!
- Growth is \$\$\$ and based on projections from Halls
 - Expenditures often switch between storage + CPU every other year



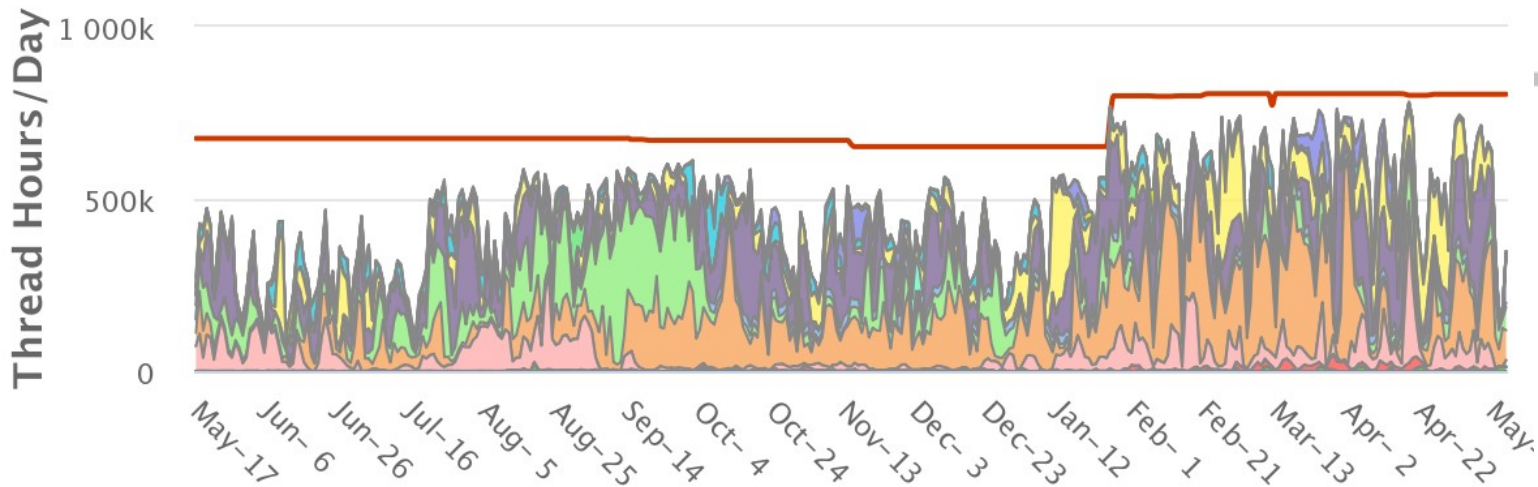
The JLab Farm • Batch Computing

- The Farm: Batch Computing
 - No direct access to these machines
 - » Use “Interactive” farm nodes for testing
 - ie. ifarm1802
 - DB and other network access (git, http, etc) generally constrained
 - Jobs controlled by automated system called “slurm”
 - You submit a job via slurm or swif2 and slurm schedules it to run
- All about trade offs:
 - “Latency” can be high (hours+ from submission to job execution)
 - » BUT!
 - Throughput is enormous
 - » 100s (1000s) of jobs can run simultaneously
 - » High bandwidth access to fast storage
 - A full replay (100s of runs) can be completed in the time it would take 2–3 runs to complete in series on your desktop.

The JLab Farm • Scheduling

- The Farm is a Lab-wide shared resource
 - Each Hall's budget includes \$\$\$ to support their usage
 - *Rough* allocation:
 - » A: 9%, C: 9%
 - » B: 34%, D: 34%
 - » EIC: 14%
- Ruled by Slurm workflow manager
 - Allocations *not* written in stone and are adjusted based on needs
- The balance is trickier to manage than you may think...
 - Jobs take time to run (system doesn't know how long beforehand)
 - Upcoming job load is hard to predict
 - System balances allocations over a few days, not hours
- More documentation here:
 - <https://scicomp.jlab.org/>
 - <https://data.jlab.org/>

Farm Cluster Daily Usage by Account



Slurm Fairshare Setting/Usage Info

Farm Utilization

Farm Usage

GPU Usage

Monthly Report

Summary Report

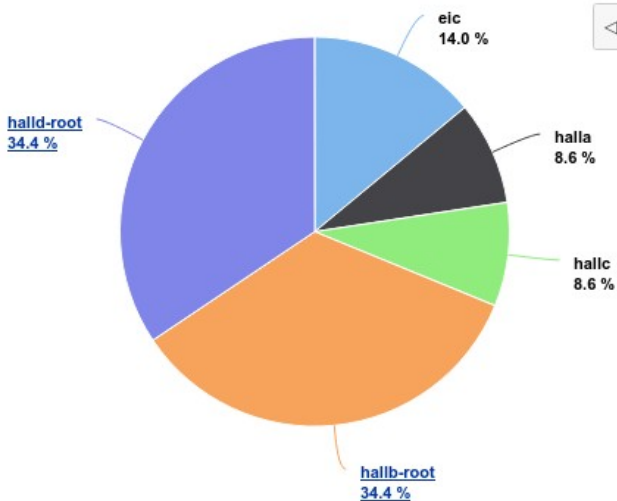
Fairshare

Summary Chart

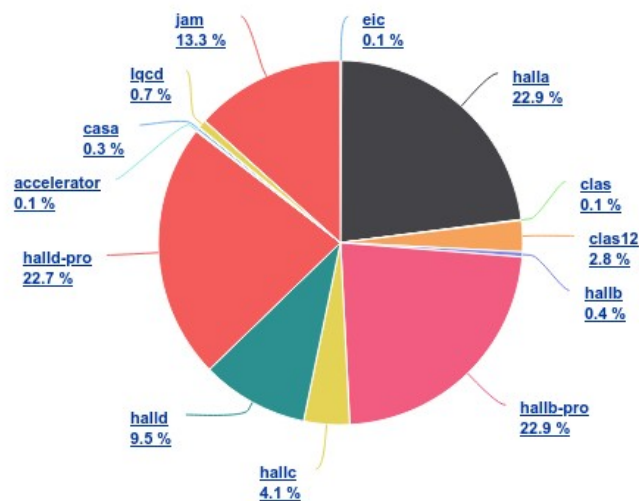
Select date Range

4/16/2023 - 5/17/2023

Slurm Fairshare Setting



Slurm Accounts Usage (CPU Hours)



Do use the Farm!

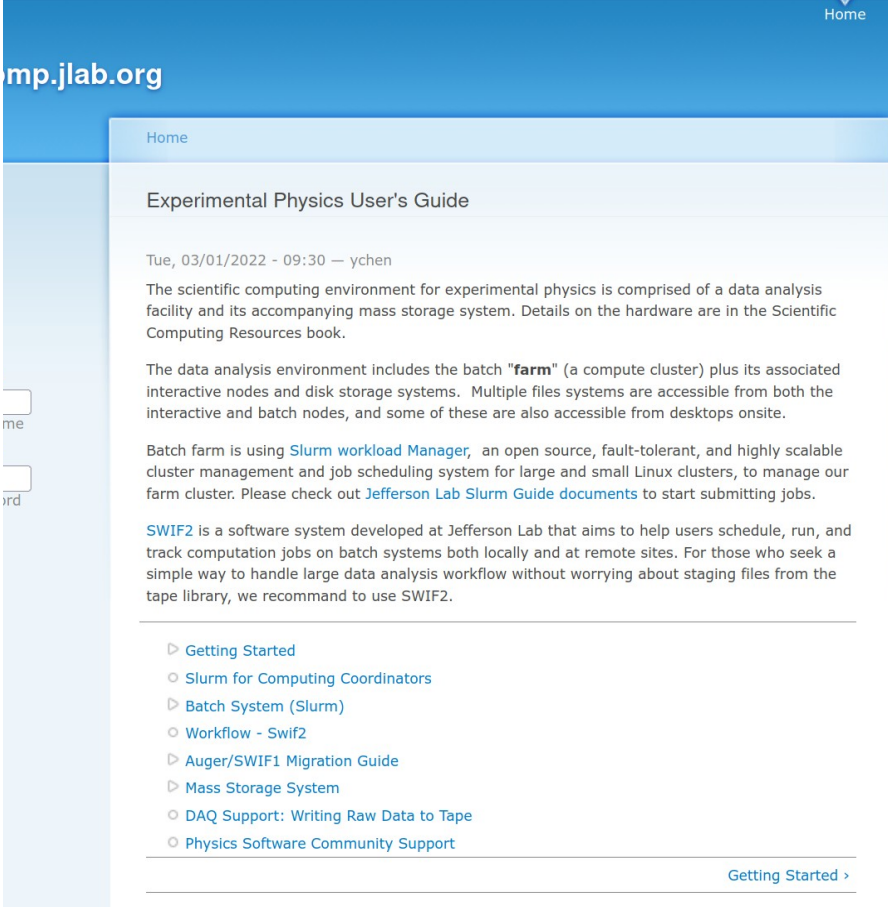
- The Farm is not your desktop
 - Best to plan, test, and fire off groups of jobs
- Test your job first!
 - Can it run reliably?
 - » If it doesn't run on ifarm180x, it won't run on the farm!
 - Is the output what you want?
 - » Check before firing off 100 jobs
- Simple tasks, some types of analysis can be done on small systems, BUT!!
 - Thou shalt back up your code!
 - Thou shalt back up your results!
 - IF your hard drive died today, how long would it take to recover?
- Don't keep only copies on your laptop
- Don't keep only copies on your desktop's hard drive



What's a “Job”?

- A 'Job' often maps to a shell script
 - It can do multiple things, but usually it executes a single instance of your software
 - » Analyze one run, or
 - » Simulate “1M” events,
 - » *etc...*
- **NOTE:** *Output that would normally go to a terminal goes to special file system:*
 - `/farm_out/$USER/job_id.out`
 - `/farm_out/$USER/job_id.err`

<https://scicomp.jlab.org/docs/FarmUsersGuide>



The screenshot shows the 'Experimental Physics User's Guide' page on the scicomp.jlab.org website. The page title is 'Experimental Physics User's Guide' and it was last updated on Tue, 03/01/2022 - 09:30 by ychen. The content describes the scientific computing environment, including the batch 'farm' cluster and the SWIF2 software system. A table of contents is provided at the bottom of the page.

mp.jlab.org

Home

Experimental Physics User's Guide

Tue, 03/01/2022 - 09:30 — ychen

The scientific computing environment for experimental physics is comprised of a data analysis facility and its accompanying mass storage system. Details on the hardware are in the Scientific Computing Resources book.

The data analysis environment includes the batch "farm" (a compute cluster) plus its associated interactive nodes and disk storage systems. Multiple files systems are accessible from both the interactive and batch nodes, and some of these are also accessible from desktops onsite.

Batch farm is using [Slurm workload Manager](#), an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters, to manage our farm cluster. Please check out [Jefferson Lab Slurm Guide documents](#) to start submitting jobs.

[SWIF2](#) is a software system developed at Jefferson Lab that aims to help users schedule, run, and track computation jobs on batch systems both locally and at remote sites. For those who seek a simple way to handle large data analysis workflow without worrying about staging files from the tape library, we recommend to use SWIF2.

- ▷ Getting Started
- Slurm for Computing Coordinators
- ▷ Batch System (Slurm)
- Workflow - Swif2
- ▷ Auger/SWIF1 Migration Guide
- ▷ Mass Storage System
- DAQ Support: Writing Raw Data to Tape
- Physics Software Community Support

Getting Started >

Debugging a job

- Generally want a single script that does everything!

→ Set up full environment

→ Use full paths

» /group/myExp/myscript.sh

» ~~./myscript.sh~~

- Testing your script:

→ 1st: Run on ifarm180x

→ 2nd: Submit job to Farm

- Test with the 'debug' Farm track

→ Max priority, fast sched.

→ Limited 4 hour runtime

→ Limited jobs/user

- Test on ifarm180x

```
% ssh you@ifarm1802
```

```
% /group/myExp/myscript.sh
```

→ Make sure it worked!

» check histos, report files

- Quick Test on Farm

```
% swif2 add-job -create \  
-track 'debug' \  
<other options> ... \  
/group/myExp/myscript.sh
```

→ Make sure it worked!

» check histos, files

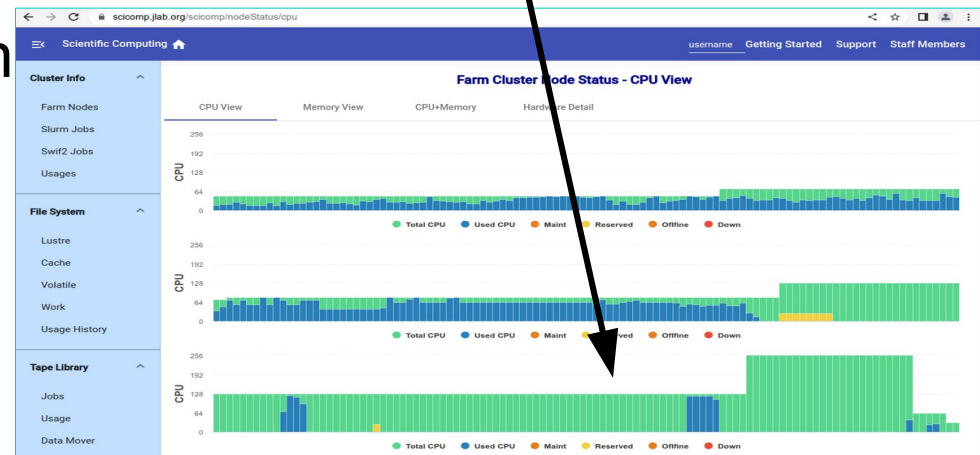
» check /farm_out/\$USER/

- Then submit full set!

→ SWIF2!

Make your jobs run faster!

- Scheduling jobs takes many things into account
 - File availability from tape
 - Memory request
 - CPU/core request
 - » >1 is useless for podd/hcana
 - 'Fairshare' metric
 - » Average Hall utilization
 - » Hall Usage can be subdivided further
- Details
 - [Fairshare Web Page](#)
- If a Hall / Project is not using 'their' fraction, then those Farm resources are available to anyone on a first-come, first-serve, basis!
 - If the Farm is idle, you can take advantage!
 - » Like now!



Make your jobs run faster!

- Common Bottlenecks/ Mistakes

- CPU count

- » use 1 core only (unless you know the job can multi-thread efficiently!)
- » your job gets 'billed' for requested cores whether you use them effectively or not

- Memory allocation

- » < 2GB is best!
- » Smaller → Faster scheduling!

- Insufficient debugging/ cross checks

- » Fire off 100s of jobs with bad config, buggy code
- » Waste compute time on the Farm waiting for trash to complete, then have to resubmit and wait again after you notice the issue.



Small I/O Problems

- Small read/write operations are very inefficient
 - Old/legacy code defaults can be very small (~4kB)
 - Should be closer to 4MB chunks for decent performance
 - Buffered IO can bridge the gap if needed
 - » Common errors:
 - 'Debugging' output
 - » `stderr << "got here" << endl;`
 - » `fprintf(stderr, "event %d\n", eventNum);`
 - Opening/closing files very frequently
 - **Frequent** random I/O
 - » ie. searching through a file for a parameter every event
- Workflows / procedures that may work on desktops or older systems do not scale well on modern systems (100s or 1000s of simultaneous jobs)
 - **Can take down / degrade system-wide filesystems**
 - Always be mindful you are on a large-scale shared system, not a personal desktop

Check Job Status

← → ↻ 🌟 📄 🔒 🔍 https://scicomp.jlab.org/scicomp/swif/active

Scientific Computing 🏠 username Getting Started Support Staff Members

Cluster Info ^

- Farm Nodes
- Slurm Jobs
- Swif2 Jobs

File System ^

Active Workflows

Active Workflows Dormant Workflows Workflow Summary File Queue Globus Status

Filter

Id	Site Name	Workflow Name
54180	jlab/enp	offmon_2023-01_ver03_post
54179	jlab/enp	analysis_2017-01_ver64_batch01_merge
54178	jlab/enp	rgc-dra-dst_sqlite3-16327

Workflow Summary

Active Workflows Dormant Workflows Workflow Summary File Queue Globus Status

Choose a user: jjaegle Choose a workflow: wf-RunPeriod-2019-01-target-nobfield-primex-eta-full-ver...

swif Job Id	Slurm Job Id	Attempt Id	Job Name	Problem Code	Node	Resolution	Complete Time
16255661	64052684	22256631	wf-RunPeriod-2019-01-target-nobfield-primex-eta-full-ver-17052023-skim-eta2g-skim_061378	SLURM_FAILED	farm140211		May 17, 2023 7:32:30 PM
16255663	64052688	22256633	wf-RunPeriod-2019-01-target-nobfield-primex-eta-full-ver-17052023-skim-eta2g-skim_061391	SLURM_FAILED	farm140209		May 17, 2023 7:27:32 PM
16255665	64052686	22256635	wf-RunPeriod-2019-01-target-nobfield-primex-eta-full-ver-17052023-skim-eta2g-skim_061435	SLURM_FAILED	farm140122		May 17, 2023 7:14:59 PM
16255667	64052692	22256637	wf-RunPeriod-2019-01-target-nobfield-primex-eta-full-ver-17052023-skim-eta2g-skim_061437	SLURM_FAILED	farm140127		May 17, 2023 7:18:26 PM

- <https://scicomp.jlab.org/scicomp/swif/active>
- Workflow Summary tab can help you find information how jobs ran (or didn't run...)
 - ie. Memory usage!
 - See also: `/farm_out/$USER/*`

Jefferson Lab Thomas Jefferson H

Questions on the “Practical Scientific Computing” bit?

CHEP 2023

- 26th International Conference on Computing in High Energy & Nuclear Physics (CHEP 2023)
 - Hosted by JLab in Norfolk, VA
 - May 8–12, 2023
 - » Pre-Conference Workshop
May 6–7
- ~600 Registrants
 - 450+ Oral Presentations
 - 140+ Posters
- 20 Plenaries
 - Plenaries recorded; online soon
- DEI Roundtable Discussion
- I am leaning on the excellent work of the Program Committee and Track Conveners!
 - Track Summaries
 - Scientific Program (Indico)



Data and Metadata Organization, Management and Access



- XRootD, dCache, Rucio
- Consolidation of access protocols and APIs
- Networks
 - need to view as scarce resource
 - direct impact on compute models
 - can *not* treat as a black box anymore
- Caching, Data *meta-data* transparency/visualization critical
- Cloud Storage increasingly important
 - Rucio ‘speaks’ AWS, GCS, SEAL, etc
 - Authentication is complicated

Erasure Coding / Data protection in XRootD

Originally developed for EOS, extended to work with any type of Xrootd storage

A simplified view of EC:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = P_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = P_2$$

$$\dots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = P_m$$

- Data: (x_1, x_2, \dots, x_n)
- Parity (P_1, P_2, \dots, P_m)
- Tricky to choose Vandermonde matrix A_{ij}
- Compare to RAID blocks, EC block sizes are usually much larger

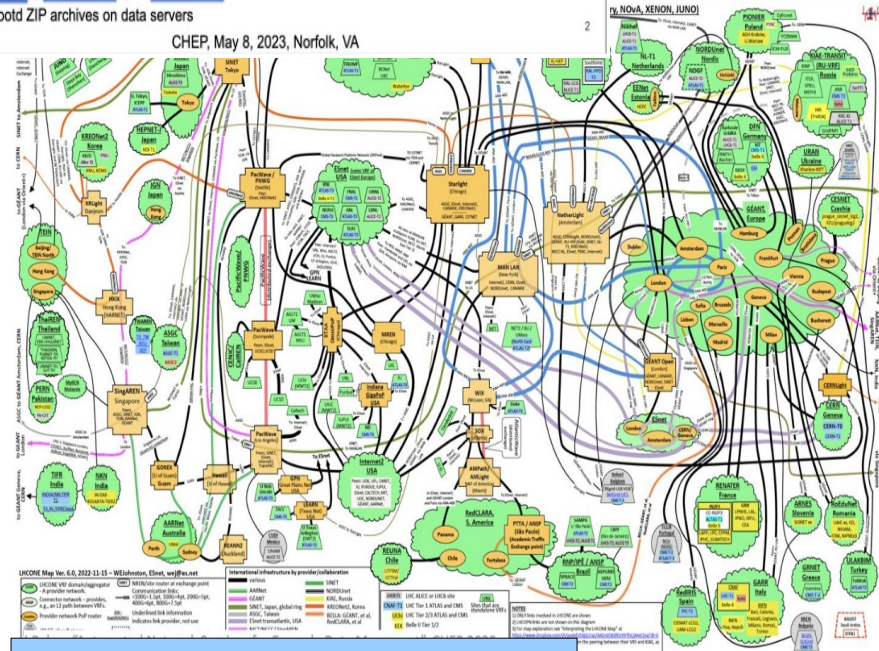
Writing:

- o A data block at client is divided into chunks
- o The chunks are erasure coded
 - EC is implemented in Xrootd client
 - Using Reed Solomon erasure coding from Intel® ISA-L
 - o Calculate crc32 of all chunks (data/parity)
 - o Spread chunks to Xrootd data servers, using ZIP archive to group individual chunks and crc32c

Reading

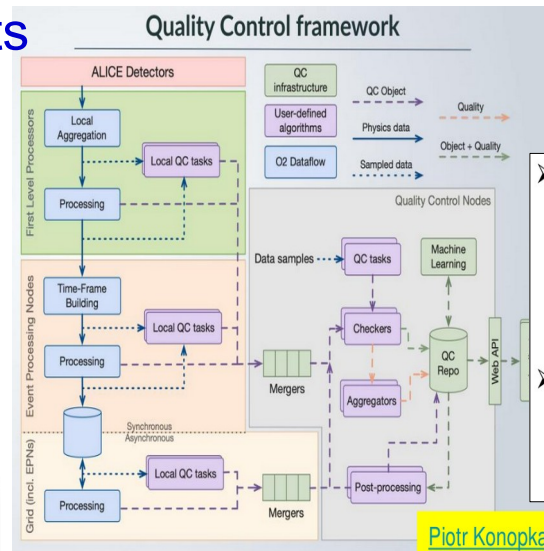
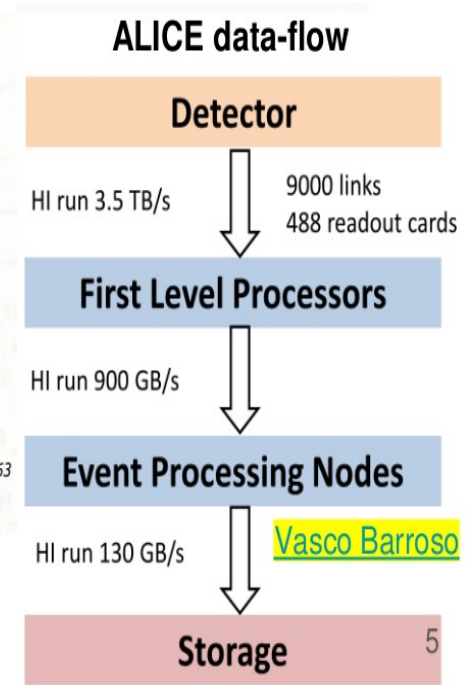
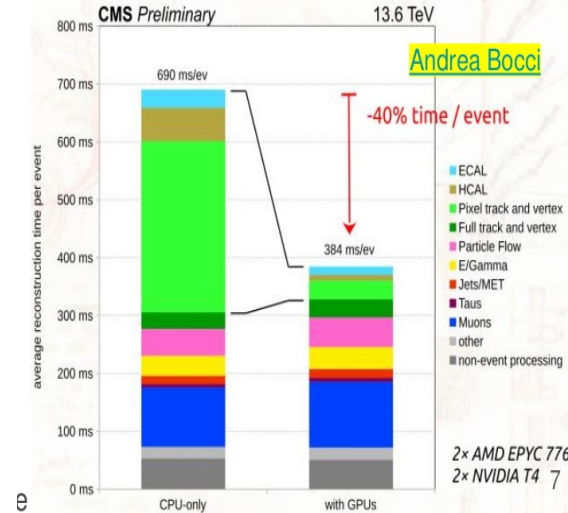
- o Only read data chunks, unless reconstruction / error correction

Xrootd ZIP archives on data servers



Online Computing

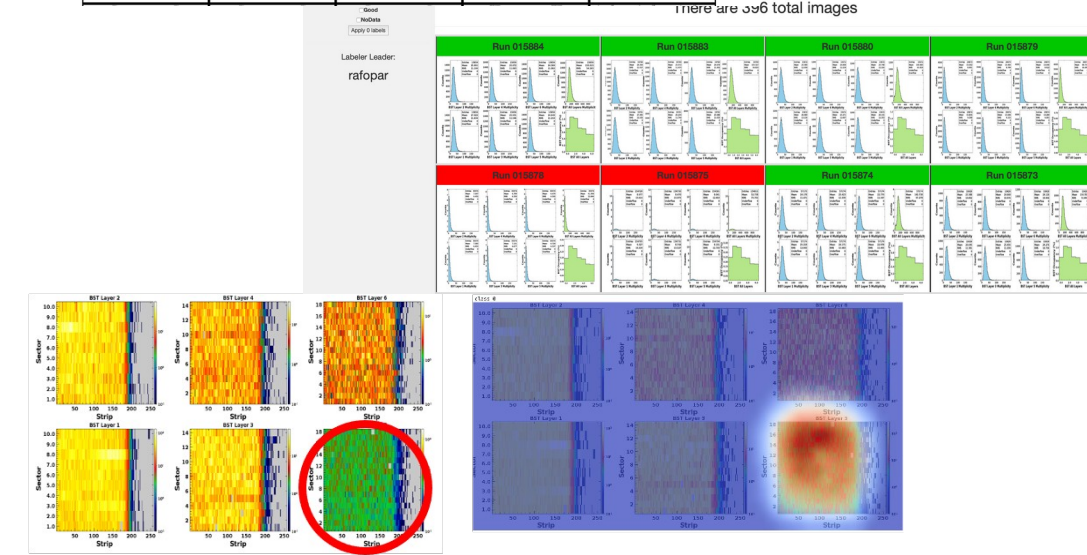
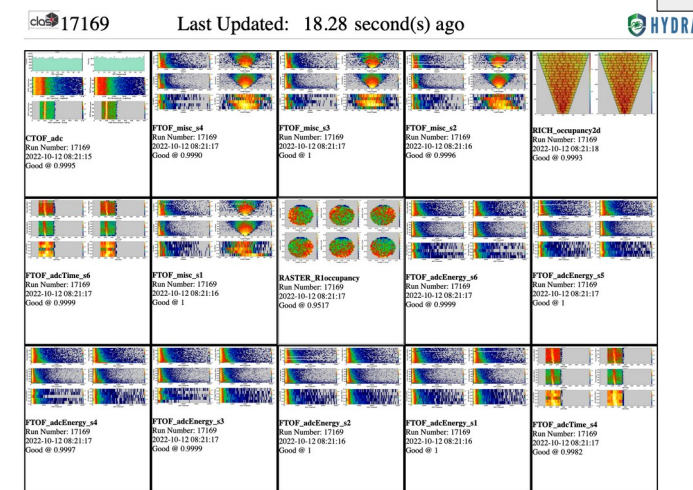
- Streaming Readout development
 - LHCb/Run3 proof-of-principle
 - ALICE
- CMS Run 3: 40% reduction in reconstruction time using GPUs
- JLab HYDRA and AIEC efforts called out as highlights
- Containerization of DAQ systems combined with Kubernetes orchestration
 - Each DAQ application is containerized and handled by Kubernetes(container orchestration tool)
 - Remove the dependency on OS/Library
 - Easier deployment of DAQ application over hundreds of nodes.
 - Study is ongoing in DUNE and CMS.
 - To what extent can Kubernetes control the DAQ system?



- Triggerless DAQ in LHCb
 - The system has been successfully used for the first part of Run3!
 - Development : Implementing trigger for Long-lived particle detection (short track and displaced vertex)
- New ALICE DAQ system (O2/FLP) for Run3
 - Reconstruct TPC data in continuous readout in combination with triggered detectors.
 - Excellent initial performance, quite promising for Run 3

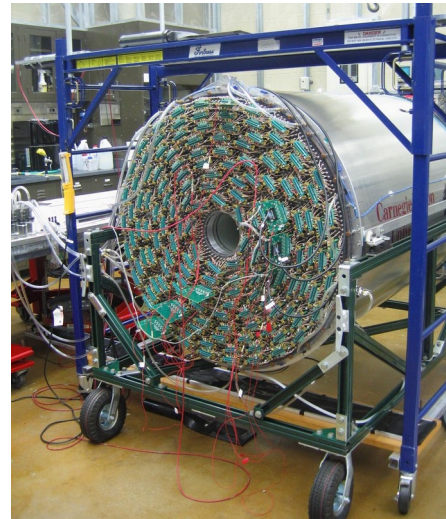
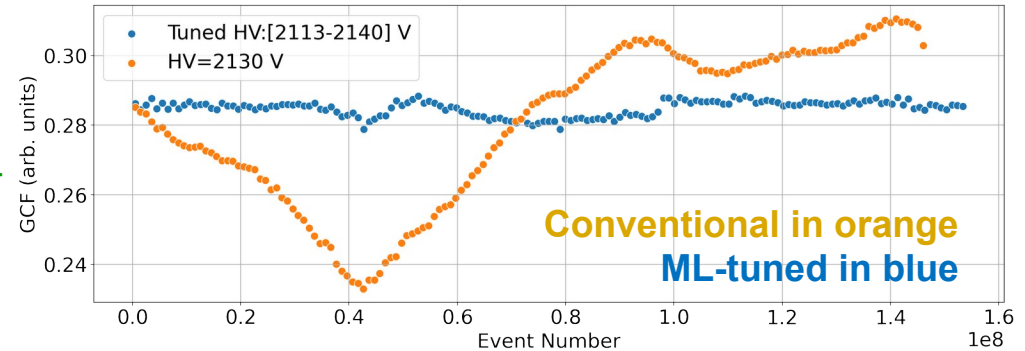
Thomas Britton, David Lawrence, Torri Jeske, Kishan Rajput and Nathan Baltzell

- AI based detector monitoring
 - Do the anomaly detection that is critical for smooth data taking, but is tedious and exhausting for humans
- Web-based UI
 - Tagging/Labeling
 - Anomaly monitoring
- Extensible Framework
 - All JLab Halls have deployed HYDRA
- Future development
 - Continue to streamline UI and generalize software deployment
 - Determine and flag data regions that triggered anomaly for users
 - » ie. reveal *why* Hydra labeled something 'bad'

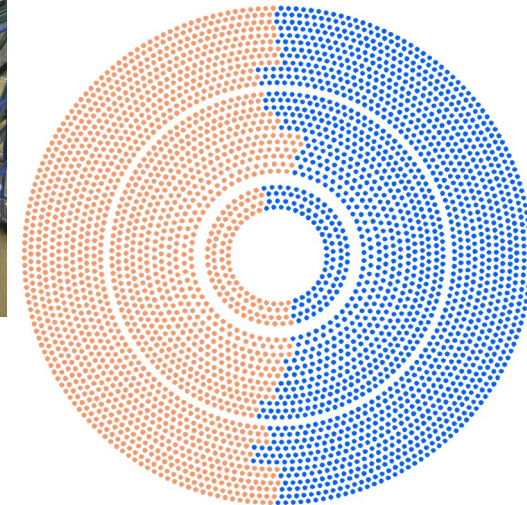


AI Experimental Calibration and Control (AEIC)

- Sensitive detectors need to be calibrated to obtain optimal resolution
 - Calibrations cause a delay between data collection and analysis (weeks-months)
 - Multiple iterations are needed to converge to final set of constants
 - Development platform
 - » GlueX Central Drift Chamber
 - » Gaussian process w/ 3 features:
 - atm. pressure, gas temp, HV board current draw
- Main Goal:
 - Dynamically adjust the controls of a sensitive detector to reduce/eliminate need for calibration
 - Key factor: “Guardrails”!!
 - » Do not allow the system to move outside reasonable boundaries (*because it will!*)



Thomas Britton[‡], Michael Goodrich[‡],
Naomi Jarvis^{*}, Torri Jeske[‡], Nikhil
Kalra[‡], David Lawrence[‡], Diana
McSpadden[‡], Kishansingh Rajput[‡]



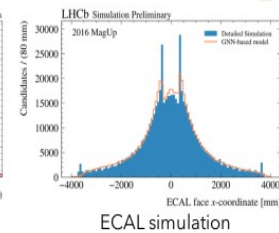
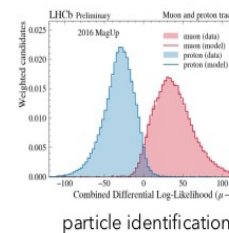
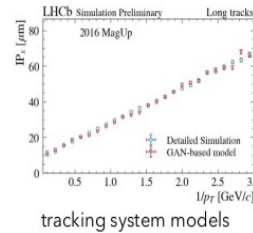
Offline Computing

- Simulation, Reconstruction
- ML is pervasive
 - reconstruction
 - anomaly detection
 - surrogate models for fast simulation
 - » ie. PHASM
- GPU / heterogeneous architectures are the future
 - Simulation (incl. G4)

Ultra-Fast Simulation: Lamarr

LHCb **Ultra-Fast Simulation** strategies replace Geant4 with parameterizations able to transform generator-level particles into analysis-level reconstructed objects [1].

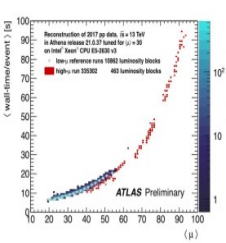
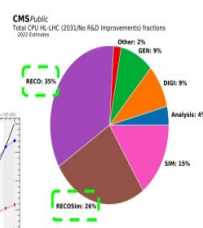
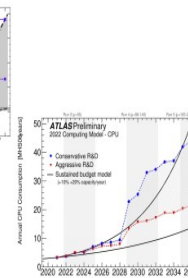
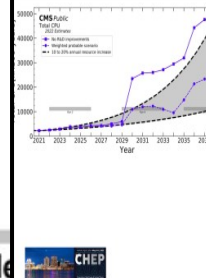
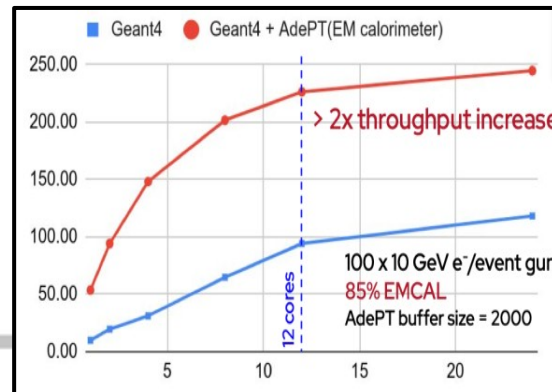
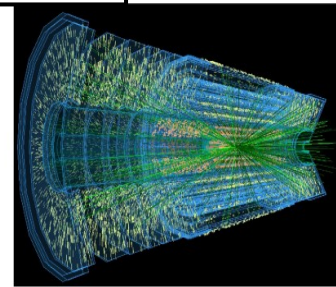
Lamarr consists of a **pipeline of (ML-based) modular parameterizations** designed to replace both the simulation and reconstruction steps



Computing challenges for HL-LHC

Focus is on speeding up Simulation and Reconstruction:

- Need for increased MonteCarlo simulation samples
- Reconstruction scales badly with pileup



Parallel Hardware via Surrogate Models (PHASM)

- What is PHASM

- LDRD project at JLab
- 1 year old; 2–3 developers
- Proof of concept

- Basic Idea

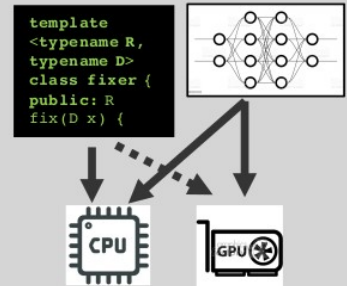
- Simplify training a neural net surrogate model to mimic and replace an arbitrary piece of existing numerical code.
- Systemize and formalize the process from analysis to deployment.

- Perspective Shift

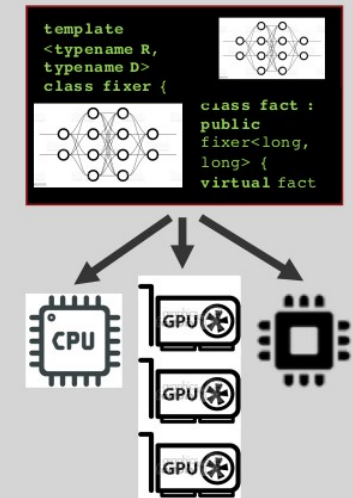
- A neural net surrogate model of an algorithm is a *transformation* of that algorithm.
- Eventually, classical numerical methods and their data-driven analogues will be understood under a unified theory.

Nathan Brei
Xinxin Mei
David Lawrence

Present situation



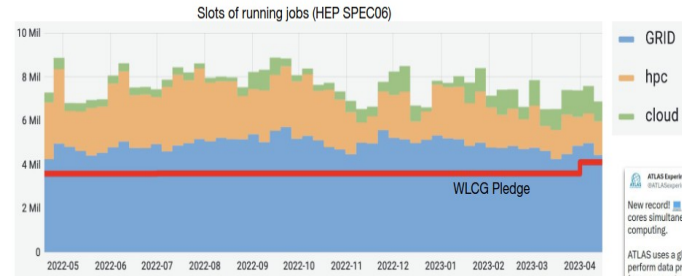
Future situation



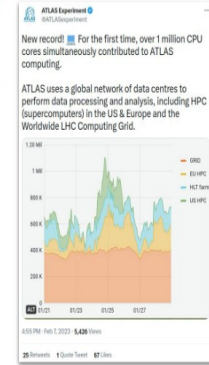
Distributed Computing

- Discussions of a variety of analysis workflows ATLAS, CMS, Astro, Grav.
 - All working to use Grid, HPC and cloud
- Central Orchestration frameworks, Workload management
 - HEP-"born" systems being reused/repurposed within other fields (Astro, NP)
 - PanDA, DIRAC, Rucio, etc
- Monitoring and Analytics
 - Tracking and Visualizing job performance is becoming critical
 - Analysis Grand Challenge (HEP)
 - » Study scalability and feature-completeness of scikit-HEP tools

ATLAS Distributed Computing today



- Steady 600k+ running job slots, 24 hours a day, 365 days a year
 - Variety of job types, depending on the current focus of ATLAS activities
 - Mostly running as 8 cores per job, 2GB/core
- As well as the grid, which is consistently over pledge, ATLAS is also



TECHNOLOGIES

- HTCondor for job management
- Rucio for (most) data management
- CVMFS + StashCache (xrootd) for data distribution
- CVMFS for software distribution
- GitLab + Conda + cmake for code management
- Apache Kafka for low-latency data exchange
- Kubernetes for service deployment on cloud resources
- Collaboration operations: Federated identity (IAM)

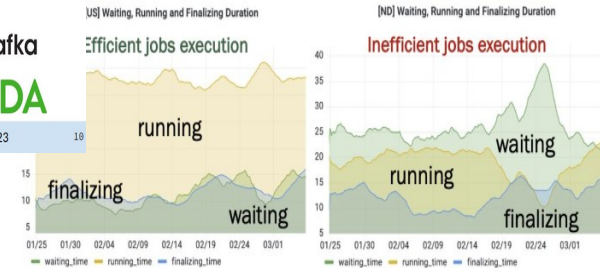


The Ligo-Virgo-KAGRA Computing Infrastructure for Gravitational-wave Research - F. Legger, CHEP23

lorfolk, VA, USA, May 2023

3

○ New metrics can help improve utilization



Operational Analytics Studies for ATLAS (Klimentov)

Sustainable and Collaborative Software Engineering



- CI & Build Infrastructure automation
 - Kubernetes, Nomad
 - GitLab / GitHub actions
 - Jenkins popular
- Interesting Research on *how* physics analysis gets done
 - Global analyses of code within ATLAS git repos
 - » libraries in use, version uptake, function-call pattern, languages
 - Tools/Studies of I/O workflows
- Julia came up in a few contexts as “sweet spot between C++ and Python” wrt to performance vs. coding efficiency
- User Centered Design for EIC called out as a highlight (Markus, Wouter)

Sustainable Analysis

Coffea = user interface for columnar analysis:

- `dask_awkward` fundamentally changes how we can describe analysis
- `dask_awkward` based analyses, via dask task graphs, are rendered into a general, complete, declarative analysis description language (ADL)
- Represents the culmination of ~4 years of R&D

Visualization	Coffea	matplotlib	lumis	
Algorithms	Scipy	Numba	Coffea	
Array API	ARROW	NumPy	Awkward Array	
Data ingestion	Laurelin	ServiceX	Uproot	
Task scheduler	Spark	DASK	Striped	ParSL
Resource provisioning	kubernetes	HTCCondor	slurm	etc.

Law = Luigi Analysis Workflow, in Python

- Large Scale End-to-End Analysis Automation over Distributed Resources
- Designed to fully decouple these 3 aspects:



SYSTEM ARCHITECTURE SWITCHING TO NOMAD FUTURE WORK

ARCHITECTURE OVERVIEW

- resources: 600 CPU cores + 1.7 TiB memory
- Nomad, Consul, Vault from Hashicorp, designed to complement each other
- Nomad:** allocates jobs to machines; resource accounting
 - ▶ long-running jobs: release builders, custom CI
 - ▶ web services: user account admin, tarball servers
 - ▶ scheduled jobs: repository maintenance and cleanup
- Consul:** generic key/value store and DNS
 - ▶ job discovery: `*,service.consul` DNS
 - ▶ `Traefik` auto-config for web access
 - ▶ job monitoring: simple health checks
- Vault** stores secrets, using Consul as backend
- metrics of the whole cluster stored and visualised



Track 6: Physics Analysis Tools



Stat. Inference & Fitting

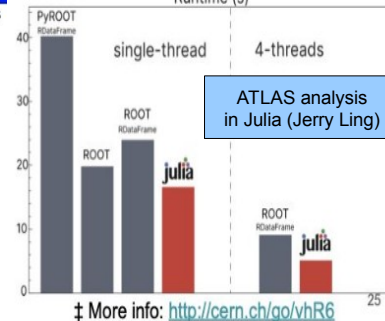
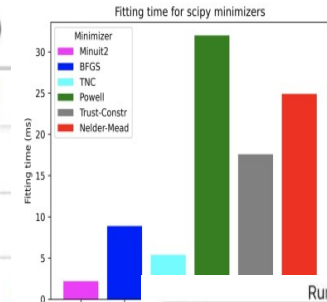
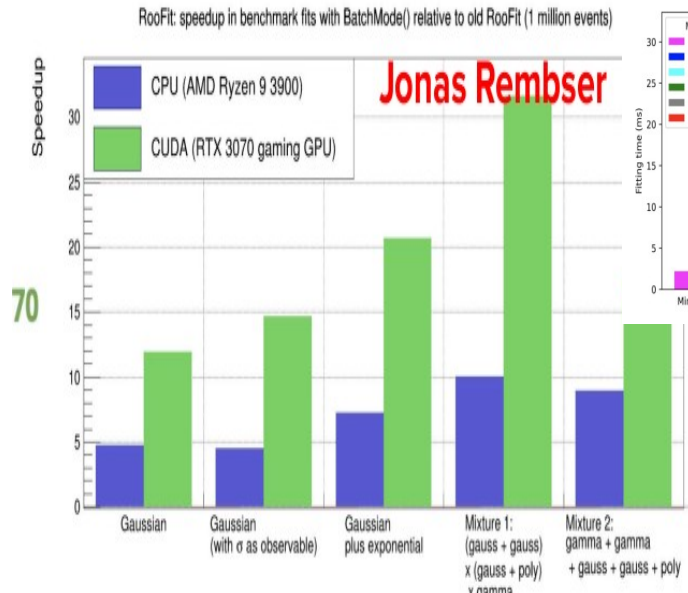
- RooFit GPU backend speedup: 25–40x!
- Minuit2 going strong after 50 years. Still one of the best performers
- Parallel Bayesian work: pyhf

I/O and Data Formats

- ROOT RNTuple
 - » smaller files, faster I/O vs TTree
- Multi-thread ROOT I/O performance (P. Canal)
- » RDataFrame, Thread safe statics

Analysis Workflows

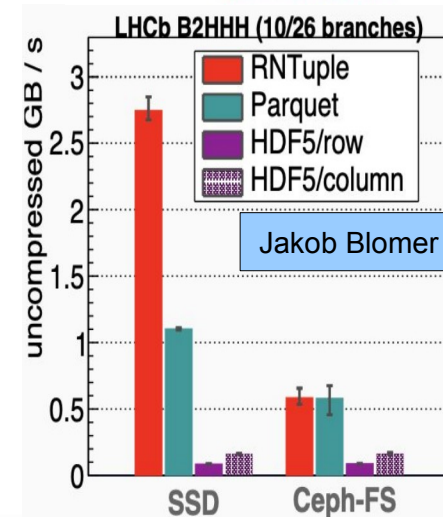
- Tools/frameworks to “make analysis easier/faster”
 - » Developing data structures and supporting toolkits that simplify/accelerate analysis
 - » “Awkward” Ecosystem, PHYSLITE
- Automate calibration, finding files, schedule jobs, etc



Speeding-up TFile Philippe Canal

- 8x reduction in elapsed time in a RDF benchmark reading one column from 4000 files with 1M entries and using 256 threads
 - New `TFile::Open` option to skip global registration, RDF uses this option by default
 - `TFile::Open` no longer reprocess identical `TStreamerInfo`.
- Another 2x by improving `TFile::Open`'s plugin mechanism
 - Increase pre-calculation (pay upfront, avoid synchronization later)
 - Increase caching to avoid calls to locking checks
 - Use local mutex rather than global lock (required attention to avoid dead lock)
- Yet another 2x Skip registration of `TFile`'s UUIDs
 - Breaks the very rare case where a `TRef` points to the TFile object
 - Cpu usage from 1557% to 14271%

2x in time
9x in CPU usage
Bottleneck switches from mutex to spin locks



May 8, 2023

ROOT I/O — CHEP 2023



Facilities and Virtualization

- Containers abound

- How to find them, verify them, distribute them

- Abstract the underpinning infrastructure

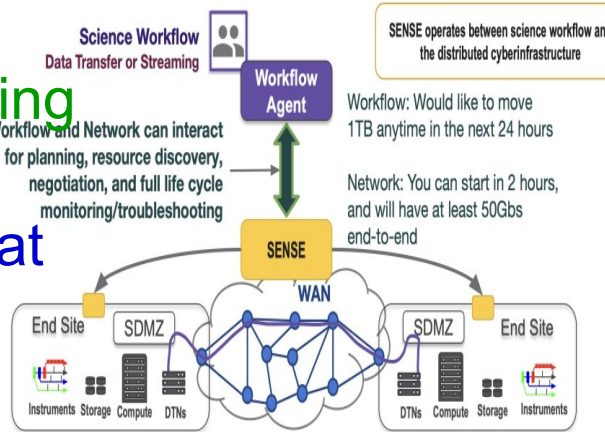
- Automation and Efficiency at Facilities

- Global resource coordination

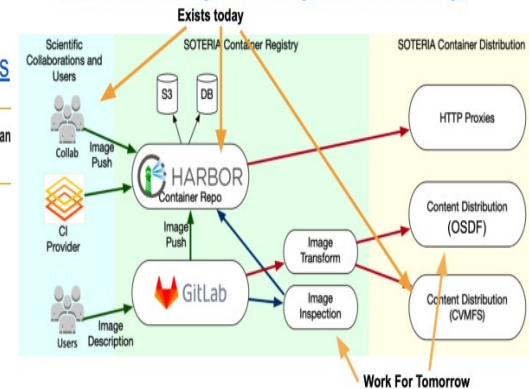
- » file caching and compute resources

- Cloud Resource Integration

- Authentication is a challenge

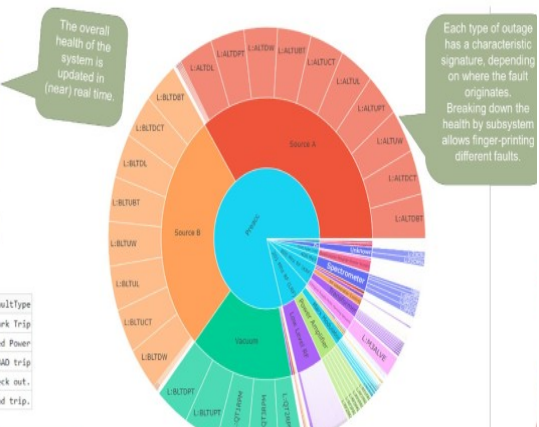


SOTERIA for container registry, discoverability, visibility & traceability



Date_Time	System	Duration(in Minutes)	FaultType
2021-06-26T00:35:00	RF	1.02	LBF2 Spark Trip
2021-06-25T12:30:00	RF	4.98	LRF Driver-trip and KRF 2 Reflected Power
2021-06-25T07:48:00	RF	1.98	LIAC KRFS/6 PFN ACEN BAD trip
2021-06-23T13:15:00	Blag/Inst	4.98	Line's BLH ITRN check out.
2021-06-23T21:31:00	RF	1.98	KRF7 Reflected Power Rad trip.

Outages are automatically assigned labels and the most recent ones are displayed.



AI for Improved Facilities Operation (Milan Jain)

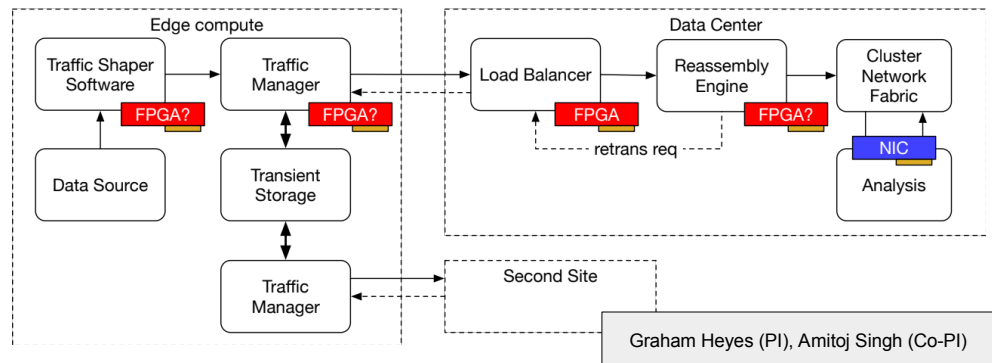
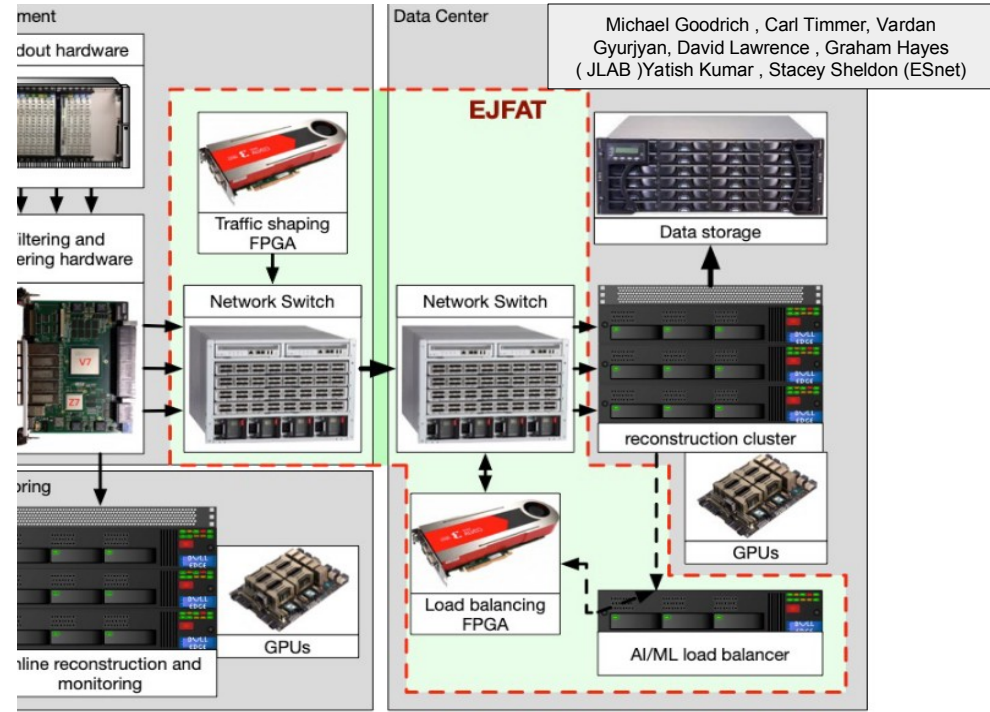
ESnet JLab FPGA Accelerated Transport (EJFAT) / Integrated Research Infrastructure Architecture (IRIAD)

Demonstrate dynamic steering of streaming data

- Data format contains metadata describing content
- Using standard IP based network, all traffic is directed to an FPGA device
- Firmware modifies packet headers to reroute data based on **data type**, and what kind of destination it should stream to
- Route data based on the kind of data it is, where it came from, and where it needs to go
- Reroute data in-flight when system architecture reconfigures
- Remote data sources can be agnostic of the destination hardware configuration

Goal is dynamic, intelligent steering of data

- Improve resilience and fault tolerance
- automated failover between two data center sites
- All core tech for future time-critical science use cases!



AI and ML

- It's everywhere...

- Reconstruction

- » tracking, cluster/vertex finding

- Anomaly detection

- » at physics level, detector level, and hardware level

- Simulation / Fast Parametrizations

- Detector Design (ie. ePIC)

- Concerns about 'Black Box' systems are well recognized and being aggressively addressed among the experts

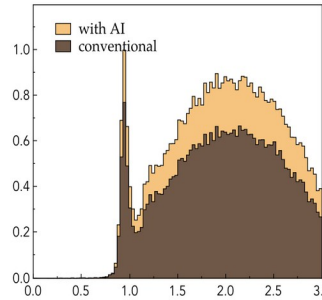
- Uncertainty modelling

- Interpretability

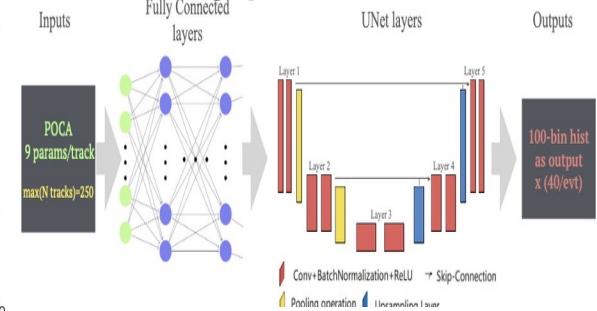
- » How is a model making a decision/classification?

- » What neurons are active, what has the system found to be important.

Track ID for CLAS12 using AI
 $ep \rightarrow e' \pi^+ \pi^- (X)$



Improved Primary Vertex finding @ATLAS and LHCb

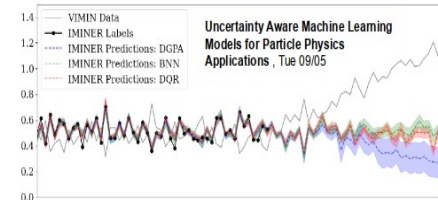


ML shouldn't be a black box!

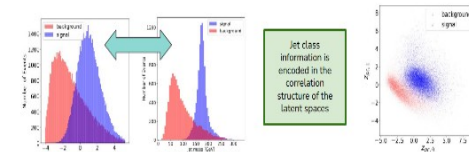
Interpretability

Helps to improve performance
 □ Analyse eg. feature importance, neuron activation patterns, latent space distributions

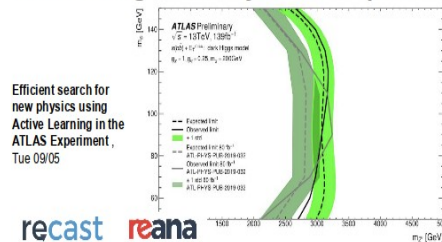
Uncertainty modelling



Interpretability Inspires: Explainable AI for DNN Top Taggers, Tue 09/05

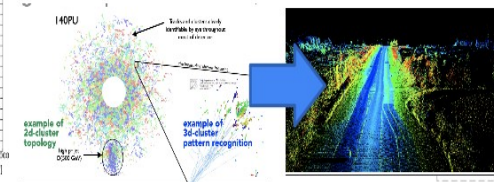


Limit setting and analysis reinterpretation



FAIR: Findability, Accessibility, Interoperability, and Reuse of digital assets

FAIR4HEP: Fair AI models in High Energy Physics, Thu 11/5



Thank You

Useful links:

[JLab Scientific Computing](#)

[JLab EPSCI Group](#)

[Software & Computing Round Table](#)

[Future Trends in Nuclear Physics Computing](#)



Yet more on
Where do I put my stuff?

File Systems: Where do I put my stuff?

- JLab SciComp/IT provides
 - /group – a space for groups to put software and some files, backup up by CST
 - /home – your home directory, backed up by CST
 - /cache – ‘mirrors’ files backed by tape system so you can use them
 - /volatile – acts as a scratch space for large output
 - /work – unmanaged outside of quotas & reservations; no backups; bigger and faster than /group

Where do I put my stuff?

- `/home/<you>/`
 - hourly snapshots
 - » `cd .snapshot/`
 - personal, non-analysis files
 - » papers, notes, thesis, etc...
 - analysis scripts: ~OK
 - » use git!
 - source code: ~OK
 - » /work better
 - NEVER store ROOT files or CODA files in /home
- Your laptop / desktop
 - Should **really** be just a front-end for working on JLab systems
 - Everybody plans to do backups, but almost no one actually does backups until **after** they've lost data...



Where do I put my stuff?

- /group

- Think “/home” for work groups

- » papers, thesis, etc

- hourly snapshots

- » `cd .snapshot/`

- analysis scripts: YES

- » use git!

- source code: ~OK

- » /work is better

- papers, thesis, etc in user subdirs is great

- /work

- Tuned for speed, small files

- » ie. source, binaries, etc.

- NOT backed up

- » but is resilient

- » snapshots available under `.zfs/snapshot/`

- Source code: YES

- » use git!

- ROOT output: ~ick (don't)

- CODA data: No

- YOU must backup to tape

- » `tar + jput` (*more on this soon*)

Where do I put my stuff?

- /group

- Think “/home” for work groups

- » papers, thesis, etc

- hourly snapshots

- » `cd .snapshot/`

- analysis scripts: YES

- » use git!

- source code: ~OK

- » /work is better

- papers, thesis, etc in user subdirs is great

- /work

- Tuned for speed, small files

- » ie. source, binaries, etc.

- NOT backed up

- » but is resilient

- » snapshots available under `.zfs/snapshot/`

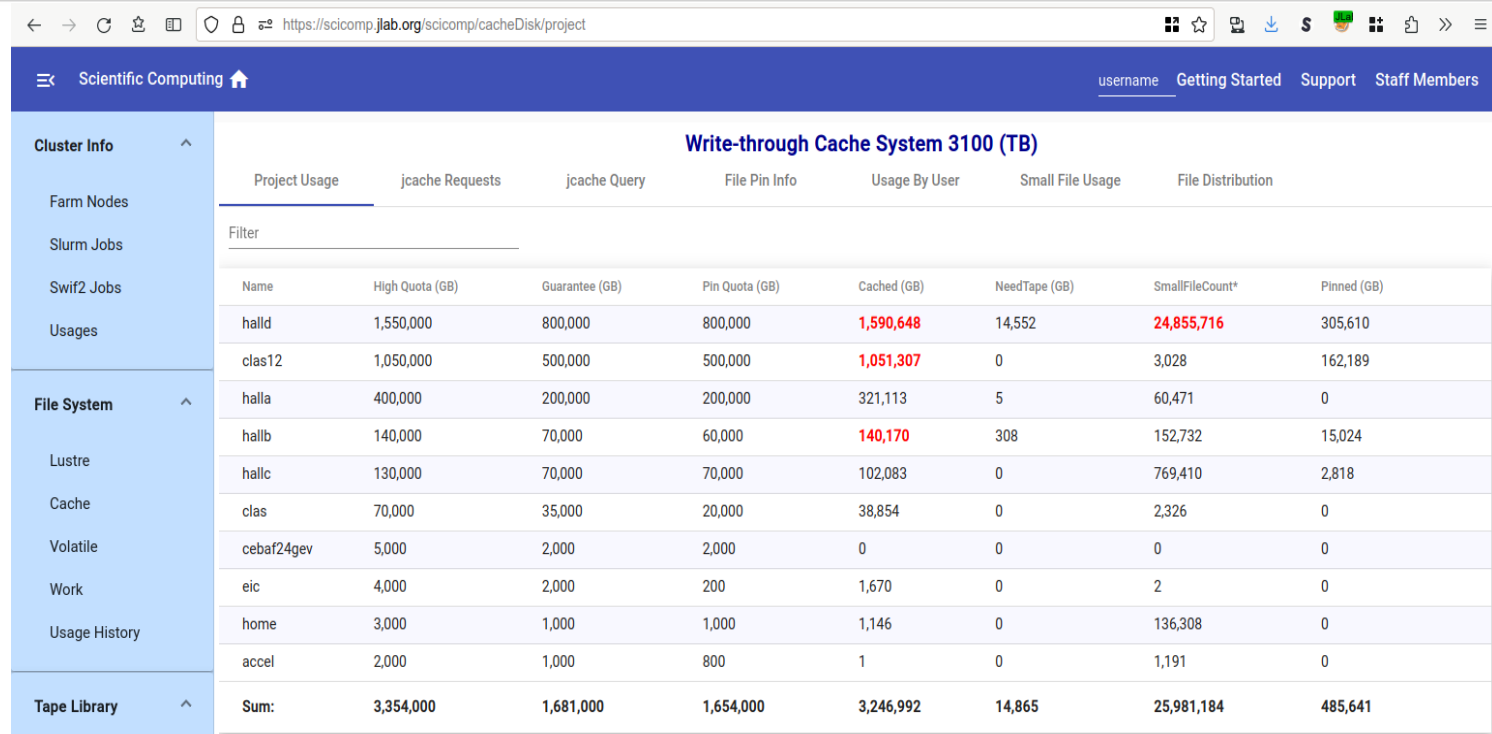
PSA: /work snapshots can be a pain because they count towards the quota for that space! (But you can't see them.)

- Generate big files, fill quota, whoops!
 - `rm -rf <all the big files>`
- quota still full!?!
 - Talk to helpdesk... (nothing you can do)

Where do I put my stuff?

- /volatile
 - Largest file system
 - » Petabyte scale
 - High performance for large files
 - » ie. ROOT output
 - NOT backed up
 - Files auto-cleaned based on quota/ reservation/ and filesystem pressure
 - » https://scicomp.jlab.org/docs/volatile_disk_pool
 - Analysis output goes here!
 - » Check, then push to tape if good!
- Tape System
 - Even bigger
 - » 100+ PB and growing
 - /mss/hallX/...
 - » Stubs: shows what is in the tape system!
 - » not the actual files
 - /cache/hallX/...
 - » actual files
 - » auto-clean up in play
 - next slide

File duration in /cache



Scientific Computing [username](#) [Getting Started](#) [Support](#) [Staff Members](#)

Write-through Cache System 3100 (TB)

Project Usage | jcache Requests | jcache Query | File Pin Info | Usage By User | Small File Usage | File Distribution

Filter

Name	High Quota (GB)	Guarantee (GB)	Pin Quota (GB)	Cached (GB)	NeedTape (GB)	SmallFileCount*	Pinned (GB)
halld	1,550,000	800,000	800,000	1,590,648	14,552	24,855,716	305,610
clas12	1,050,000	500,000	500,000	1,051,307	0	3,028	162,189
halla	400,000	200,000	200,000	321,113	5	60,471	0
hallb	140,000	70,000	60,000	140,170	308	152,732	15,024
hallc	130,000	70,000	70,000	102,083	0	769,410	2,818
clas	70,000	35,000	20,000	38,854	0	2,326	0
cebaf24gev	5,000	2,000	2,000	0	0	0	0
eic	4,000	2,000	200	1,670	0	2	0
home	3,000	1,000	1,000	1,146	0	136,308	0
accel	2,000	1,000	800	1	0	1,191	0
Sum:	3,354,000	1,681,000	1,654,000	3,246,992	14,865	25,981,184	485,641

- Files auto-cleaned based on quota and system pressure on /cache
 - Clean up least-recently-used files first
 - Can 'pin' files to keep them stable
 - » Shared resource, don't abuse!

Accessing files from Tape

- Retrieving files from tape
 - `jcache get /mss/.../foo.dat`
 - » Manual pull from tape to `/cache/.../foo.dat`
 - » **Never** call this (or `jget`) in a farm script!
 - Let SWIF2 do it!
 - » List needed files as `<Input>` tag(s)
 - » Backend will prestage them for you in advance
 - `jget /mss/.../foo.dat $PWD/`
 - » pull file from tape to any filesystem
 - » generally **not** the right tool

Copying files to Tape

- Storing files on tape
 - `jput file /mss/.../`
 - » 'jput -h'
 - » [Online Docs](#)
 - 'write-through cache' ([Online Docs](#))
 - » write large file output directly to /cache/hallX/...
 - no 'staging' on /volatile
 - » automagically backed up to tape after a few days
 - guaranteed to be safe on tape **before** /cache auto-removal kicks in
 - » **Gotchas:**
 - small files (<1MB) not backed up to tape
 - avoid pathname collisions with files already on tape
 - » ie. 'overwriting' files with same name, etc