

# Analysis at Jefferson Lab





# About Jefferson Lab

- CEBAF12-GeV continuous electron beam accelerator
- Experiments are carried out in 4 Experimental Halls (A,B,C & D)
- Aim to run 2/3 of the year with a >60% duty factor.
- A and C are currently high beam intensity but low solid angle detectors.
  - High trigger rate. > 100 kHz
  - Simple, small events.
  - Modest data rates less than 100 MB/s
- B contains CLAS12 – general purpose detector with many detector elements and large solid angle.
  - ~10-100kHz < 1 Gbyte/s
- D contains GLUEX – large solid angle high rate.
  - 200 kHz 1.5-2 Gbyte/s





# Hall A

## General

### Current Experiments

- Analysis of parity experiments PREX/CREX completed/ongoing.
- 13.5 TB of raw data.

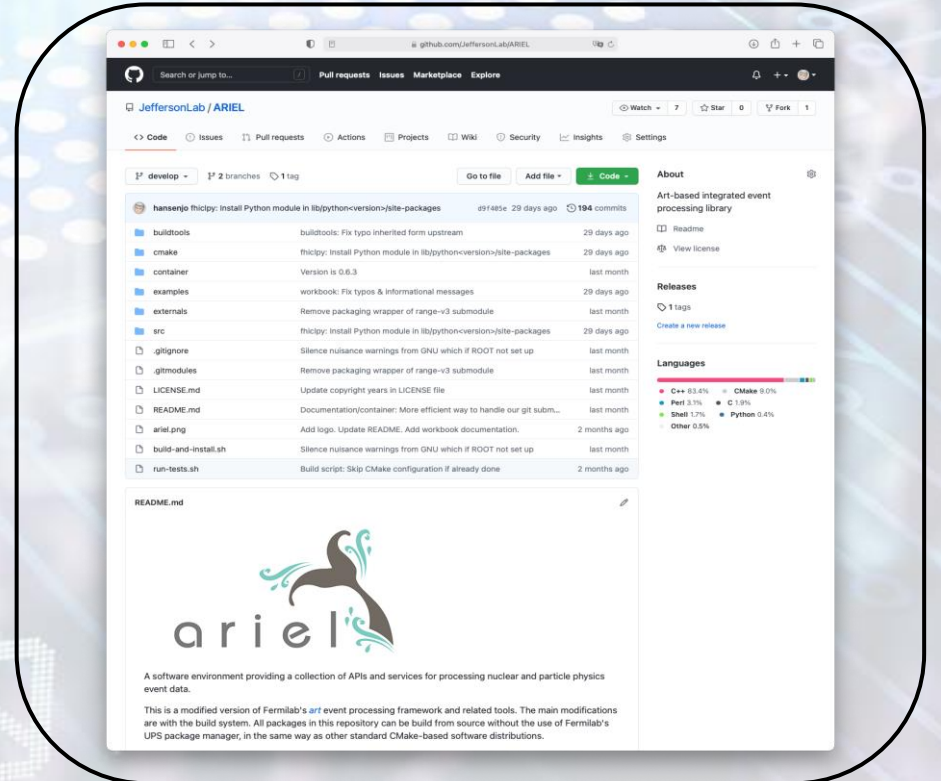
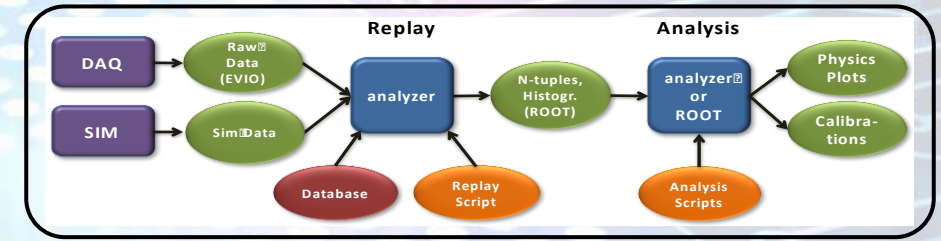
### SuperBigBite Spectrometer (SBS)

- Preparing for first data taking in Fall 2021.
  - Anticipating DAQ rates of up to 1 GB/s and ultimately several PB of raw data.
- Running extensive simulations, using in-house g4sbs package.
  - Offline reconstruction code, based on Hall A analyzer, under development: <https://github.com/JeffersonLab/SBS-offline>
- Multi-threaded version of Hall A analyzer (“Podd”) under development.
  - Will benefit SBS and Hall C.

### SoLID

- High rate experiment – tens to hundred GB/s
- Preparing for upcoming science review. Simulations ongoing, using Hall B’s GEMC framework.
- Prototype reconstruction framework: ARIEL (based on art)  
<https://github.com/JeffersonLab/ARIEL>
- Experimenting with DD4hep and track reconstruction libraries.

## Hall A Analyzer

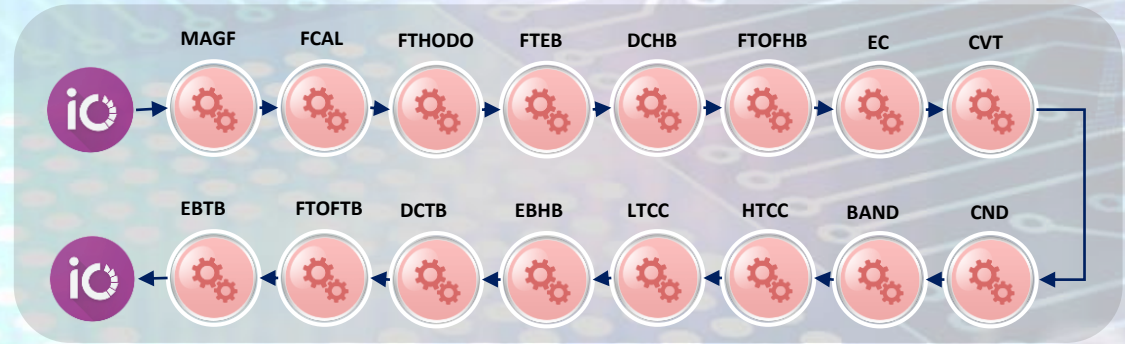


# Hall-B - CLAS12 Data Processing Framework, CLARA

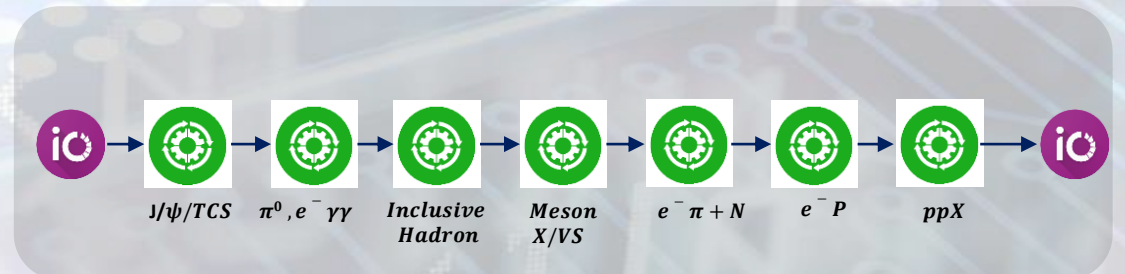
## Micro-services architecture based, reactive, event-driven data-stream processing framework

- Application is defined as a network of independent “black box” components, called micro-services. User composes an application using a simple YAML file.
- Micro-services can be deployed as threads within the container at run-time or as separate, distributed processes.
- Services communicate with each other by exchanging the data across predefined connections by message passing, where connections are specified externally to the services.
- User provided data processing single-threaded algorithms (engines) are presented as fully scalable micro-service in the framework.
- Loose coupling of services makes polyglot and heterogeneous data access and processing solutions possible, eliminating long term commitment to a single technology stack.
- Improves fault isolation and recovery process.

Event Reconstruction Application (example)



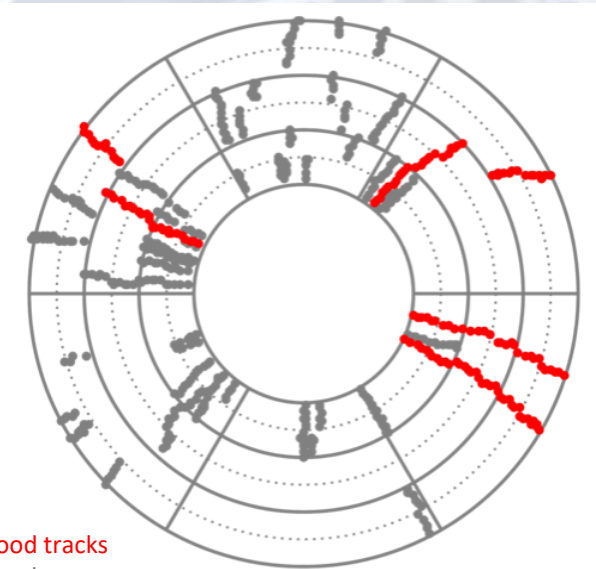
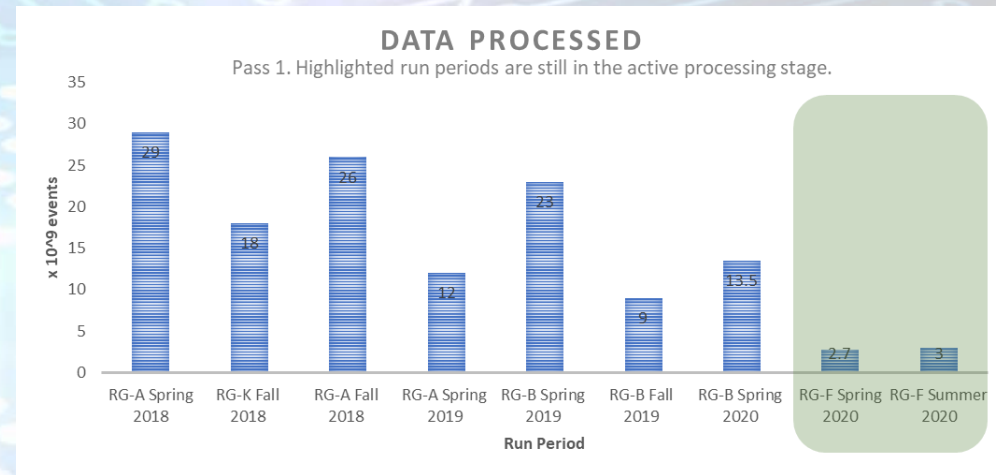
Physics Analysis Application (analysis train example)





# Hall B – CLAS12 analysis status

- First full-scale data processing of CLAS12 2018-2020 experimental runs on JLab farm with very good success
  - Workflows developed and throughput benchmarked in advance, performing as expected, 50 million CPU hours in 2020
  - Selected resulting physics analysis in pipeline for collaboration approval and publications (SIDIS, TCS, DVCS)
  - Many related presentations at DNP 2020 conference
- Started mid-year using JLab's OSG VO for supporting CLAS12 simulations
  - With dedicated web portal, priority system, and multiple collaboration institutions providing dedicated resources
  - Averaging few-100K CPU hours per week, 12.5 million in the past year
  - Using CVMFS for software distribution and XRootD for common data distribution, e.g., background files
- Machine-Learning developments
  - CLAS12 track-finding for speed improvement implemented and being incorporated into standard workflow
  - Cylindrical tracker noise-reduction, magnetic field, and RICH/ECAL actively being explored



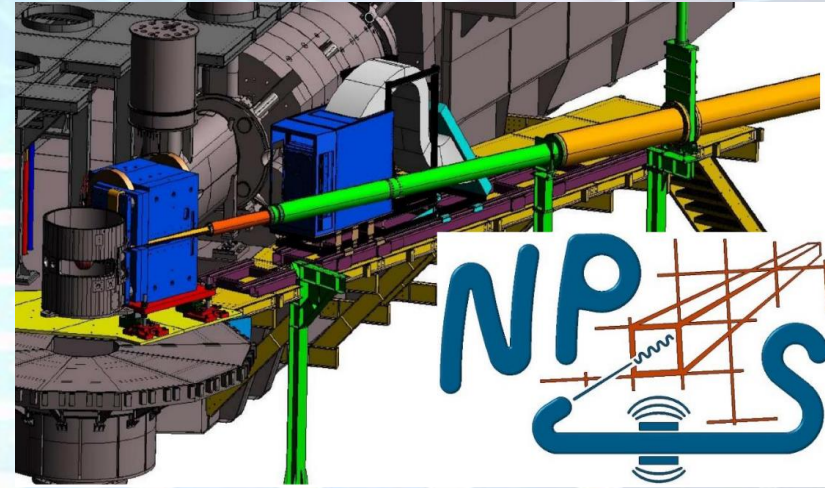
One tracker event, with **good tracks found by AI in red**, and noise/background in grey



# Hall C

## General

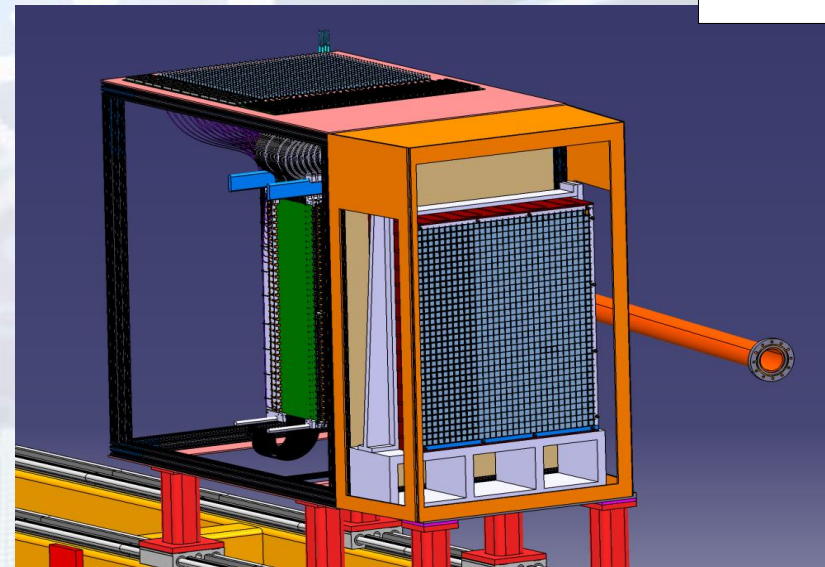
- No major upgrades needed to satisfy current program
  - System is very reliable, no issues
- Hall C “Remote Counting House” prototype developed during the Fall 2020 run. (Used to supplement standard on-site shift crew in this phase – worked well.)
  - This will be a 'production' option for selected external collaborators for upcoming slate of experiments beginning in Fall 2021



NPS in Hall C

## Neutral Particle Spectrometer (NPS)

- Future experiment that will put pressure on current Hall C DAQ/network systems (Earliest run period: 2022/23?)
  - O(100 MB/sec) data rates driven by waveform readout on 1080 crystal calorimeter array
  - Modest network/compute infrastructure upgrades are in progress to increase bandwidth and allow greater overhead
  - FE/DAQ group developing VTP + FADC firmware to sparsify readout based on updated clustering algorithm
  - Updates to Hall C Analyzer (based off Hall A Podd) to support new detector and data analysis requirements also in progress



# Hall-D, GlueX

	2017 (low intensity GlueX)	2018 (low intensity GlueX)	2019 (PrimEx)	2019 (high intensity GlueX)
Real Data	1.2PB	6.3PB	1.3PB	3.1PB
MC Data	0.1PB	0.38PB	0.16PB	0.3PB
<b>Total Data</b>	<b>1.3PB</b>	<b>6.6PB</b>	<b>1.4PB</b>	<b>3.4PB</b>
Real Data CPU	21.3Mhr	67.2Mhr	6.4Mhr	39.6Mhr
MC CPU	3.0Mhr	11.3Mhr	1.2Mhr	8.0Mhr
<b>Total CPU</b>	<b>24.3PB</b>	<b>78.4Mhr</b>	<b>7.6Mhr</b>	<b>47.5Mhr</b>

Projection for out-years  
of GlueX High Intensity  
running at 32 weeks/year

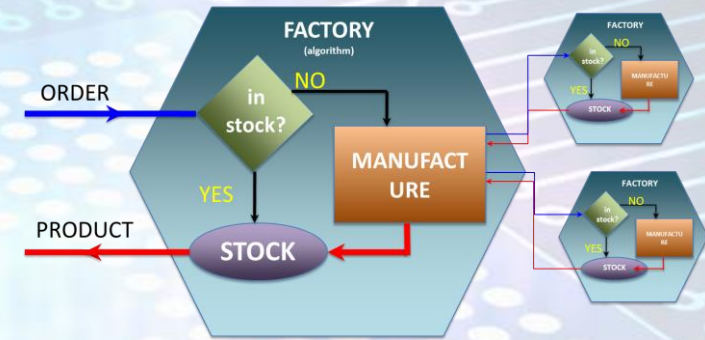
	Out – years (high intensity GlueX)
Real Data	16.2PB
MC Data	1.4PB
<b>Total Data</b>	<b>17.6PB</b>
Real Data CPU	125.6Mhr
MC CPU	36.5Mhr
<b>Total CPU</b>	<b>162.1Mhr</b>

**Event size:**  
12-13kB

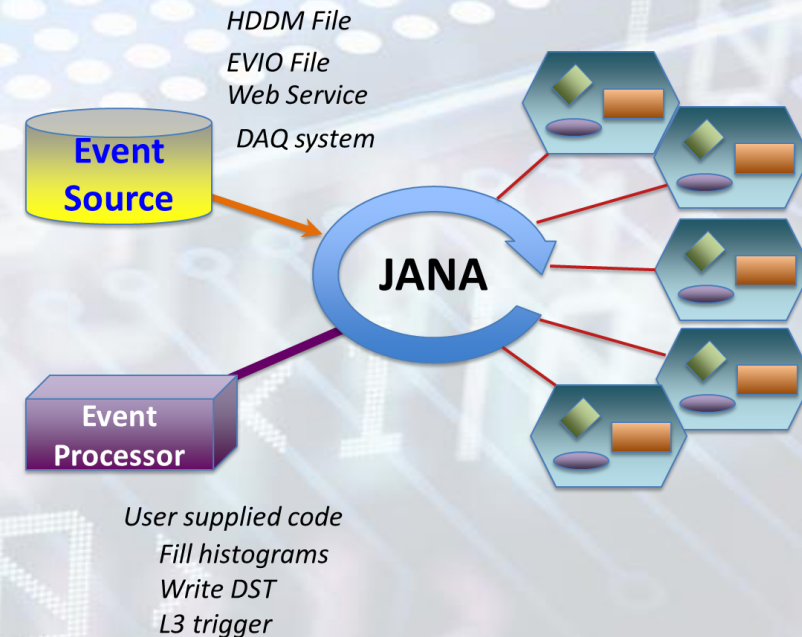


# GLUEX Event Reconstruction - JANA

- Reconstruction framework based on a “factory model.”
  - Data on demand, don't create if not needed
  - Reuse data already created



- Framework has a layer that directs object requests to the factory that completes it
- Multiple algorithms (factories) may exist in the same program that produce the same type of data objects.
- This allows the framework to easily redirect requests to alternate algorithms specified by the user at run time





# Hall D – GLUEX Analysis

- Off-Site data processing expansion

- Sites:

- NERSC
- Pittsburgh Supercomputing Center
- Big Red at Indiana University
- Compute Canada
- Open Science Grid

- Raw data reconstruction as well as simulation

- Multi-threaded Geant4-based simulation (HDGeant4)

- Containerization progress

- OSG's Oasis (CVMFS) as a software distribution mechanism
- Routine use of Singularity for infrastructure development on future platforms

- Hall D Data Model (HDDM) instrumented with option for HDF5 formatting and I/O

- Hall D Online Skim System (HOSS)

- DAQ (CODA) system writes raw data to a large RAM disk on a buffer server
- HOSS distributes raw data from buffer to one of several RAID servers
- HOSS sends copy of raw data to one of several farm nodes to produce skim files

compute | calcul  
canada | canada



Open Science Grid



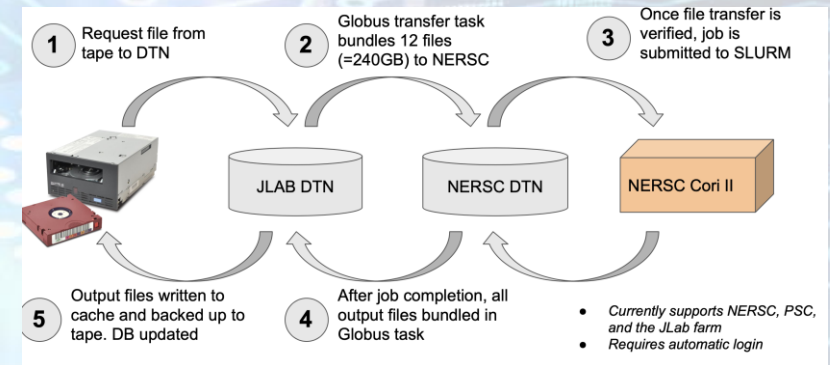
# Common environment using SPACK and CVMFS

- Project goal
  - ENP software requires a lot of underlying packages with various versions.
  - In the past the various packages were supported by an ad-hoc group of staff and users.
  - Use SPACK and CVMFS to simplify software distribution.
  - Scientific Computing group is coordinating this.
    - [https://wiki.jlab.org/epsciwiki/index.php/SPACK\\_Mirror\\_on\\_JLab\\_CUE](https://wiki.jlab.org/epsciwiki/index.php/SPACK_Mirror_on_JLab_CUE)
- Use cases for end users (pre-built binaries)
  - Working with local disk access to software packages.
  - Working offsite on system with access to CVMFS. e.g., OSG or NERSC.
  - Working offsite on system without CVMFS. e.g., counting house or work from home.
- An ideal setup allows single set of builds to support all of these.
  - The preferred method of using and building packages is via a Singularity container.
    - The container provides a static OS that is independent of OS installed on machine.
    - Once set up the environment is usable by many people wherever they are.
- This gives us the needed stepping stone to a user environment that is site agnostic.

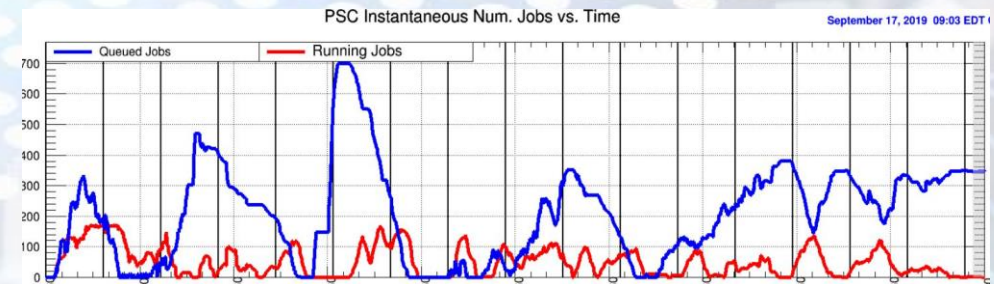


# SWIF Workflow manager

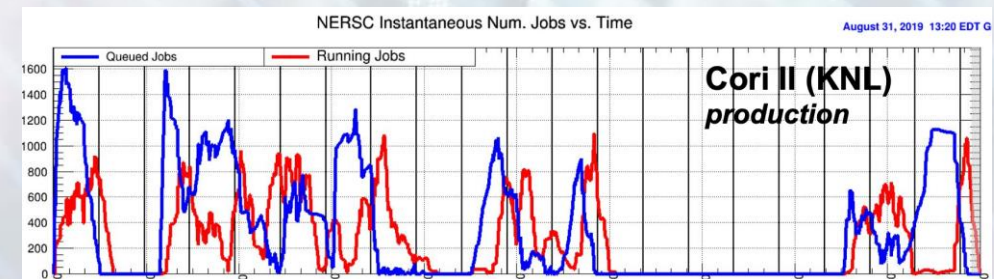
- Manages file transfer and job submission for nontrivial workflows.
- Was around before this year but now seeing major production use by CLAS12 and GLUEx.
- User defines a workflow:
  - Inputs.
  - Code.
  - Output destination.
  - Retry handling.
- In use by both CLAS12 and GlueX.
  - GlueX successfully running at supercomputer centers.
  - CLAS12 used SWIF locally to improve reconstruction usage by retrying failed jobs.
- Works well but is a “JLab only” product – area where help and input needed



## GlueX at Pittsburgh (XSEDE)



## GlueX at NERSC Cori II



# Facilities : ENP Hardware

- ENP Cluster

- Capacity growth over recent years to meet program goals.
- Recent focus to improve utilization and uptime, former focus was on performance.
  - CGROUP kernel segmentation
  - ~ uptime at 95.6%.
- Added GPU component (40 GPUs) to support AI/ML development

- LQCD clusters

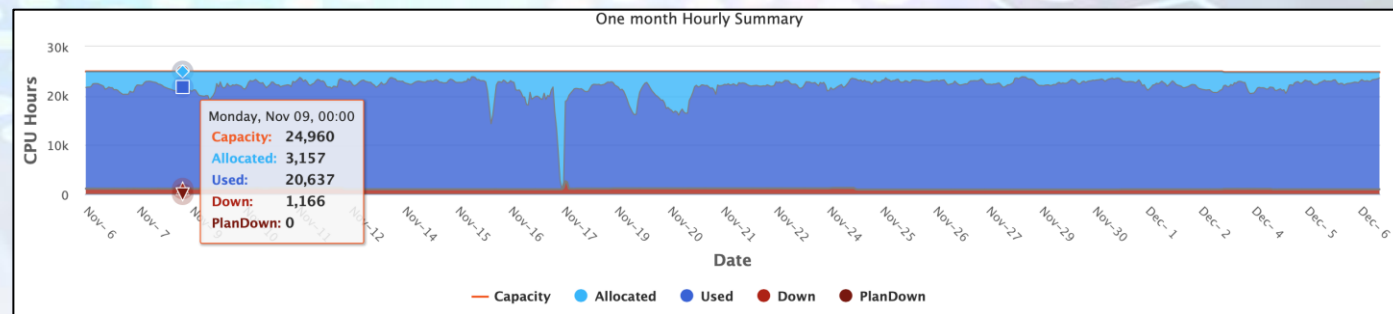
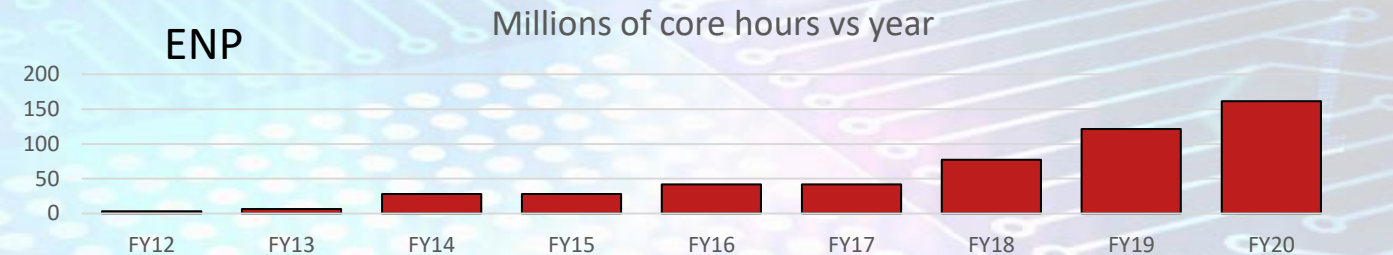
- Currently two clusters KNL and GPU
- New 128 AMD-GPU cluster coming in 2021.
- USQCD Allocated

- Both clusters have SLURM batch systems.

- Nagios for monitoring and Puppet for configuration control.

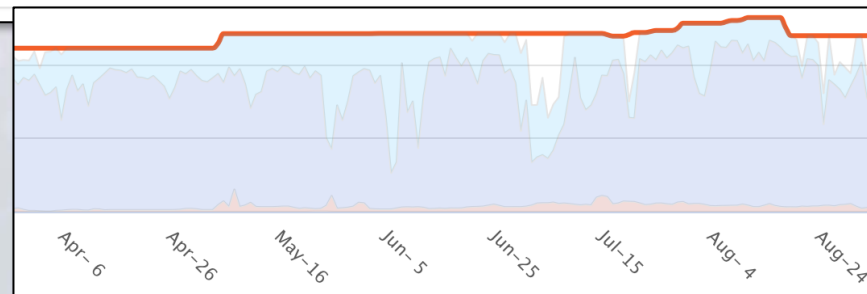
- Storage

- Added Solid State Cache disk.
- LUSTRE Storage system.
  - 3.9PB ENP + 1.3PB LQCD
- IBM Tape Library ~10PB stored
  - Raw data is duplicated and the copy ejected.



Farm Computing Nodes Detail

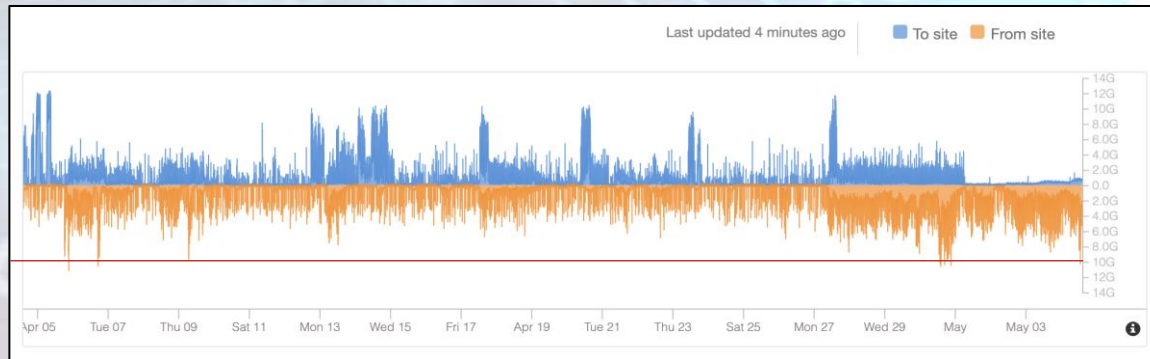
Id	Year	OrderCnt	ProdCnt	Type	Memory	Disk	IB
farm19	2019-2020	76	75	dual 32 core AMD EPYC 7502 (Rome)	256 GB	1 TB SSD	FDR IB
scim19	2019	3	3	Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz	188 GB	1 TB nvme drives	FDR IB
farm18	2018	88	86	dual 20 core 2.4 GHz Xeon (Skylake)	96 GB	480 GB SSD	FDR IB
farm16	2016	44	40	dual 18 core 2.3 GHz Xeon E5-2697V4 (Broadwell)	32 GB	dual 1 TB HDDs	FDR IB
farm14	2014	104	97	dual 12 core 2.3 GHz Xeon (Haswell)	32 GB	dual 1 TB HDDs	QDR IB
farm13	2013	24	19	dual 8 core 2.0 GHz Xeon (Sandy Bridge)	32 GB	dual 500 GB HDDs	QDR IB





# Facility: Distributed Computing

- In 2019, exceeded 10 Gb/s Wide Area Network capacity - ESnet doubled the connectivity
  - ESnet 6 hardware install soon > 100 Gb/s.



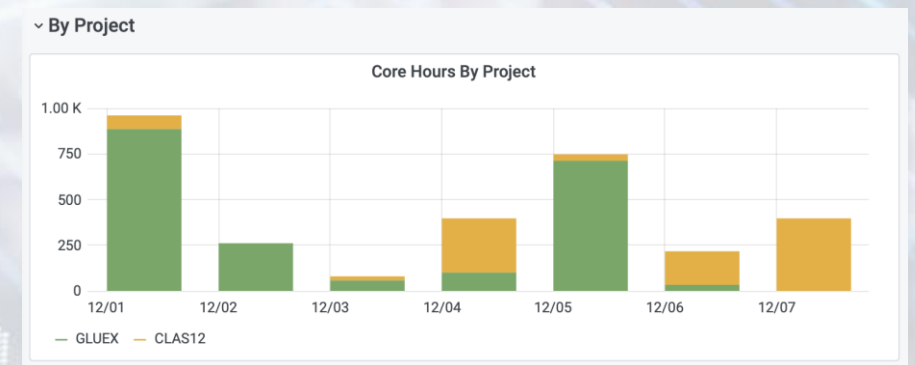
GlueX on OSG

osg	113.7 Mi
nova	19.0 Mi
dune	18.6 Mi
icecube	17.4 Mi
gm2	15.8 Mi
microboone	15.6 Mi
cms	10.27 Mi
gluex	5.81 Mi
ligo	4.35 Mi
minerva	4.33 Mi

- Jlab runs an OSG job submission server.
  - We now routinely move thousands of jobs per day to the Open Science Grid.
- MCWrapper- web form that automatically produces and monitors a workflow to fulfill the request.
  - To Date: 13 Billion simulated events in 2.1 Million jobs and over 4 million core hours.
- GLUEX is running opportunistically on OSG and has a NERSC allocation.
  - Also making use of Big Red at Indiana University.
- CLAS12 is running via OSG and dedicated CLAS12 collaborating sites.
  - GRIDPP at Glasgow, UK.
  - INFN.
  - Orsay.

# Facility: OSG Computing details

- Jlab also has the infrastructure in place to run OSG jobs on the local cluster.
- Jobs submitted in the CLAS12, GLUEX, or EIC VOs will land on local nodes if they are free.
  - Much more powerful accounting and job control.
  - Jobs that run locally run using local authentication.
    - Short term solution - we made three JLab user accounts: osg-gluex, osg-clas12, and osg-eic.
    - So, it's a jlab account per VO at this point and since it all runs in a container, it works.
    - Each of those accounts participates in the fair share for the respective account.
  - Jobs count against local SLURM Fairshare settings – not a back door.
- Allows EIC users at BNL to submit a job and use Jlab resources allocated to EIC.
- Eventual goal is to have:
  - One way to submit jobs.
  - Where a job runs become less visible.
  - One way to package up a job for submission
    - Containerized with code on CVMFS – see later
- Still need to work out how we want to use this.
  - Big step forward in 2020 though!





# Artificial Intelligence and Machine Learning Coordination

- AI Town Hall for JLab ENP+CST on Aug. 8, 2020
  - 24 projects presented:

<https://docs.google.com/spreadsheets/d/1bAsHq4Zp4pTUMqwUn3600jT3fUrFGIhN5EdCJqn2fr8/edit?usp=sharing>

- Most are current projects, a few recent, and a few planned
- Broad categories:
  - Detector Design
  - Monitoring
  - Triggering
  - Charged Particle Tracking
  - PID
  - Simulation
  - plus a few more ...
- Weekly, informal meeting
  - Every Wednesday at noon:
  - [https://www.jlab.org/AI/lunch\\_series/2020](https://www.jlab.org/AI/lunch_series/2020)

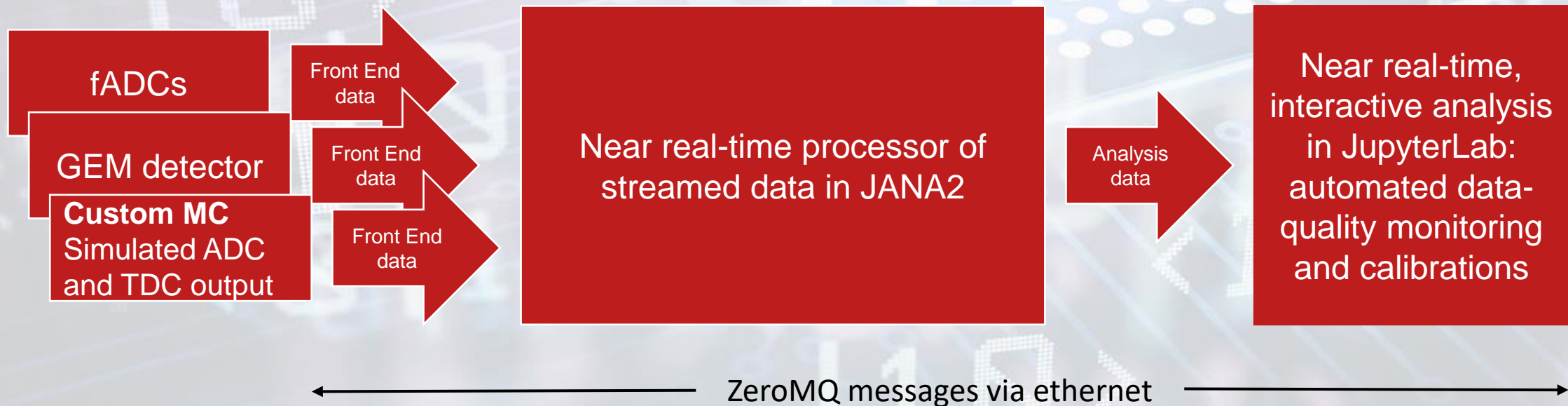


# LDRD: INDRA-ASTRA

## LDRD goal

**prototype components of streaming readout at NP experiments**  
→ integrated start to end system from detector read out through analysis  
→ comprehensive view: no problems pushed into the interfaces

**prototype near real-time analysis of NP data**  
→ inform design of new NP experiments





# EIC: Jefferson Lab's role in Software & Computing for the EIC

**EIC Computing** coordinated by Graham Heyes (JLAB) and Jerome Lauret (BNL)

## EIC Software



Emphasizes the close connection of EIC Science to 12 GeV CEBAF science program

**Coordinating role for EIC Software** in close collaboration with BNL

- EIC Software Consortium (ESC), NAS study, **EICUG Software Working Group (SWG)**

Strong involvement in Yellow Report Initiative: **ESCalate with g4e**, modern MCEGs



Experimental Physics Software and Computing Infrastructure

led by David Lawrence

### JANA 2 framework development

- C++ event processing framework
- **factory model** on demand interface, user-centered design
- multi-threaded with > 10 years experience
- **plugin support** provide mechanism for many physicists to contribute, multi-threading external to contributed code (parallelizer)

### Streaming Readout

- Environment for **Real-time Streaming Acquisition and Processing (ERSAP)**
- hardware/software system that will process unbounded streams of continuous data at scale over distributed heterogeneous resources in (near) real-time
- currently tested at CLAS12

### Artificial Intelligence

- dynamic control/calibration of experimental detector systems
- automated data quality monitoring
- automated calibrations
- heterogeneous hardware support in reconstruction

## Last words

- There is clearly a lot going on.
- We are making extensive use of many third party packages.
- My feeling is that JLab still relies too much on home grown software.
- That said, in some areas continuing existing “products” make sense
  - JANA2
  - GEMC
  - CLARA
- For a small lab the temptation is to be so busy “working on things” to look at the landscape.
  - Taking any advantage to be involved in the larger community
  - And here we are.



**The last slide was the end the next slides are extra.**

# System Administrator: Using CVMFS for publishing software

*software packages are built locally and stored in this directory.*

**/scigroup/cvmfs**

*Automatically copied to cvmfs for remote use.*

*software packages are available globally using this path.*

**/cvmfs/oasis.opensciencegrid.org/jlab**

*Global namespace*

*We are using the OSG stratum 0 server*

*our namespace*

**Example:**

**/scigroup/cvmfs/epsci/centos/7.7.1908**

*Pushed to global CVMFS every 4 hours*

**/cvmfs/oasis.opensciencegrid.org/jlab/epsci/centos/7.7.1908**



## Publishing binaries to the web

`/scigroup/spack/mirror`



`https://spack.jlab.org/mirror`

Using the web gives access to the list of available pre-built binary packages (example home use).

Individual packages can then be ***“installed”*** locally and then ***“loaded”***

Software administrators must take one extra step to create the *buildcache* files from the binaries installed in CVMFS

e.g.

```
spack mirror add jlab-public https://spack.jlab.org/mirror
```

# Building packages using Singularity to build software

- The preferred method of building packages is via a Singularity container. Why?
  - The container provides a static OS that is independent of OS installed on machine.
  - Once set up the environment is usable by many people wherever they are.
  - Using a singularity container to build on interactive node :

```
$ singularity shell \  
-B /scigroup/cvmfs:/cvmfs/oasis.opensciencegrid.org/jlab \  
-B /scigroup:/scigroup \  
/scigroup/spack/mirror/singularity/images/epsci-centos-8.3.2011.img
```

*mounts build destination r/w with path as it will appear to user code over cvmfs*

*mounts /scigroup r/w for updating buildcache*

*container image*

```
Singularity> source /cvmfs/oasis.opensciencegrid.org/jlab/epsci/centos/8.3.2011/share/spack/setup-env.sh
```

- The end user runs the job using the same container image but without mounting /scigroup