

CLAS12 remote data-stream processing using the ERSAP framework

Παντα ρει

V. Gyurjyan
gurjyan@jlab.org

Jefferson Lab



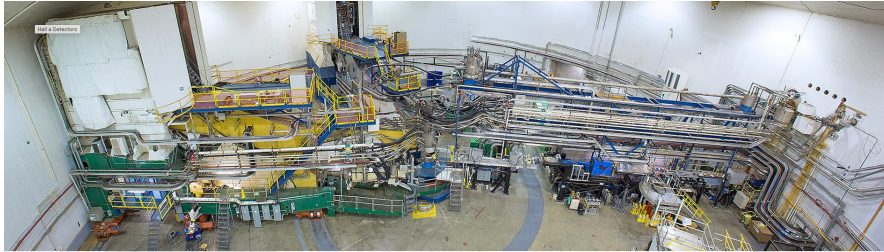
U.S. DEPARTMENT OF
ENERGY

Office of
Science

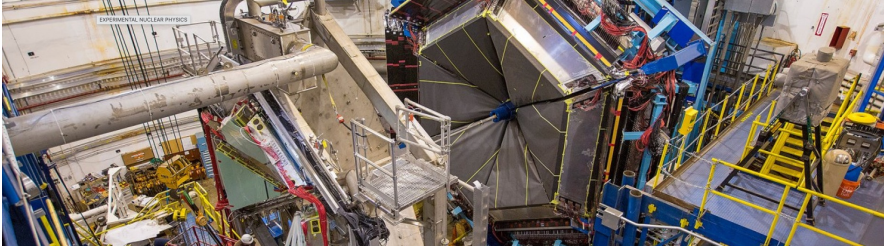


JLAB Experimental Halls

A



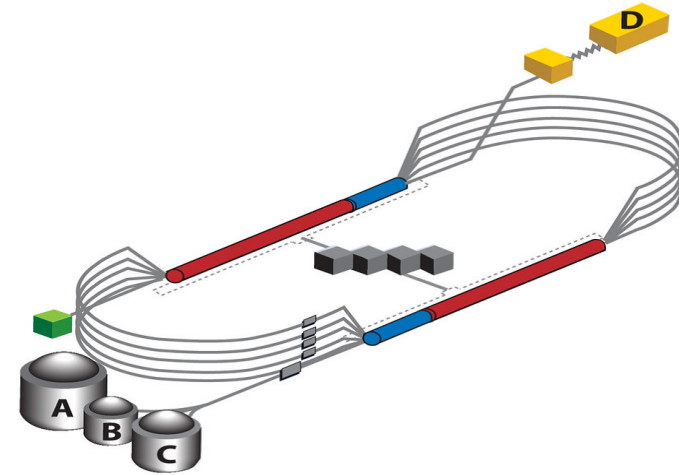
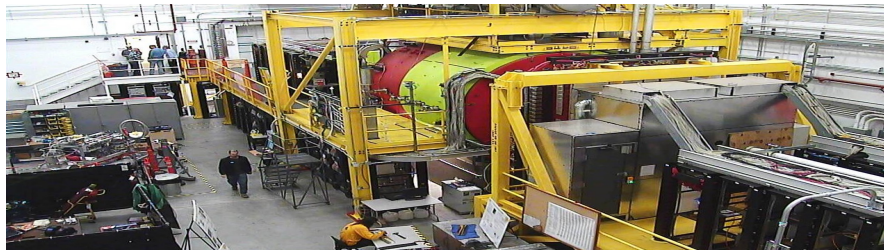
B



C



D

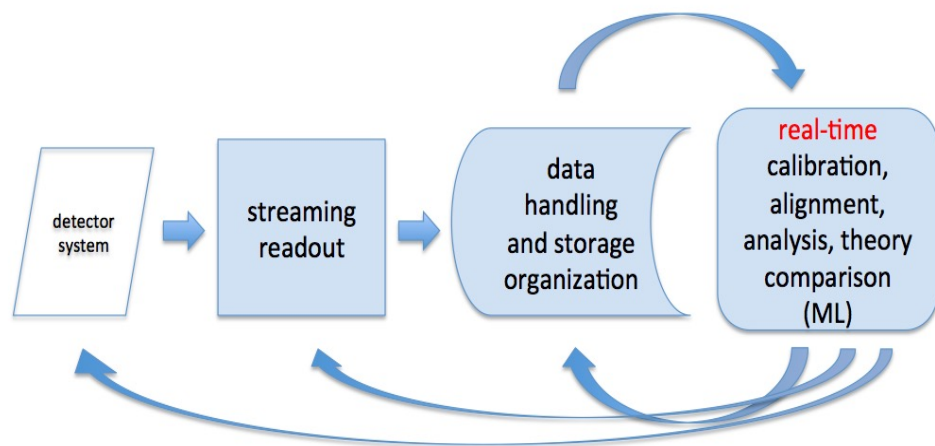


- Four experimental end-stations with different experimental equipment.
- Current and upcoming experiments require increased data acquisition, driving the demand for streaming technology.

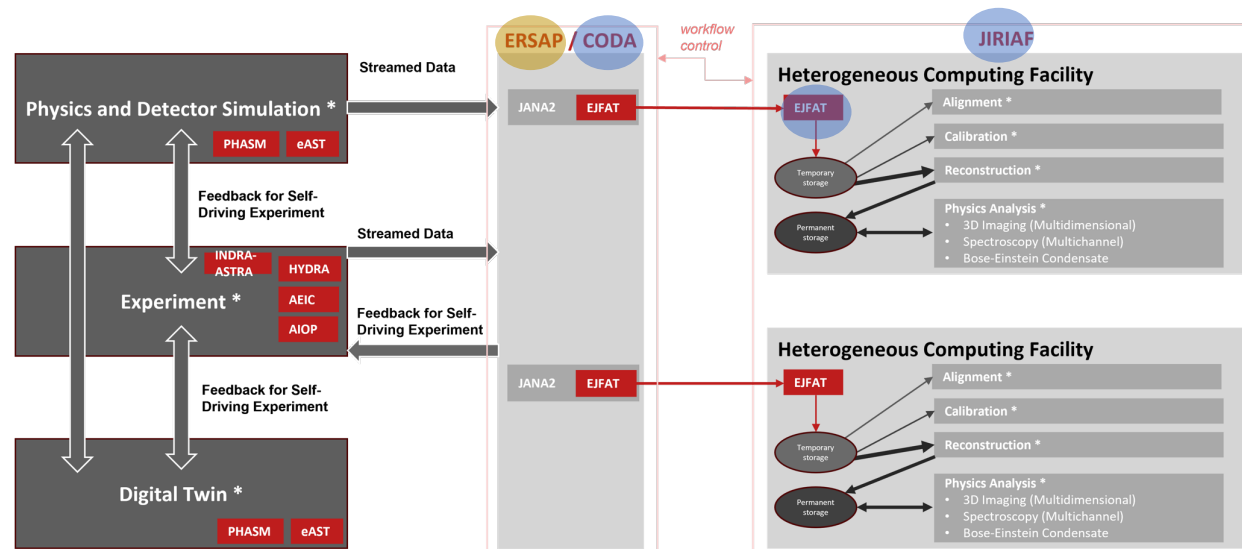
“Enable full offline analysis chains to be ported into real-time, and develop frameworks that allow non-expert offline analysis to design and deploy physics data processing systems.”

A Roadmap for HEP Software and Computing R&D for the 2020s. HEP Software Foundation, Feb. 2018

JLAB Grand Challenge in Readout and Analysis for Femtoscale Science



Courtesy of Amber Boehnlein, et al.

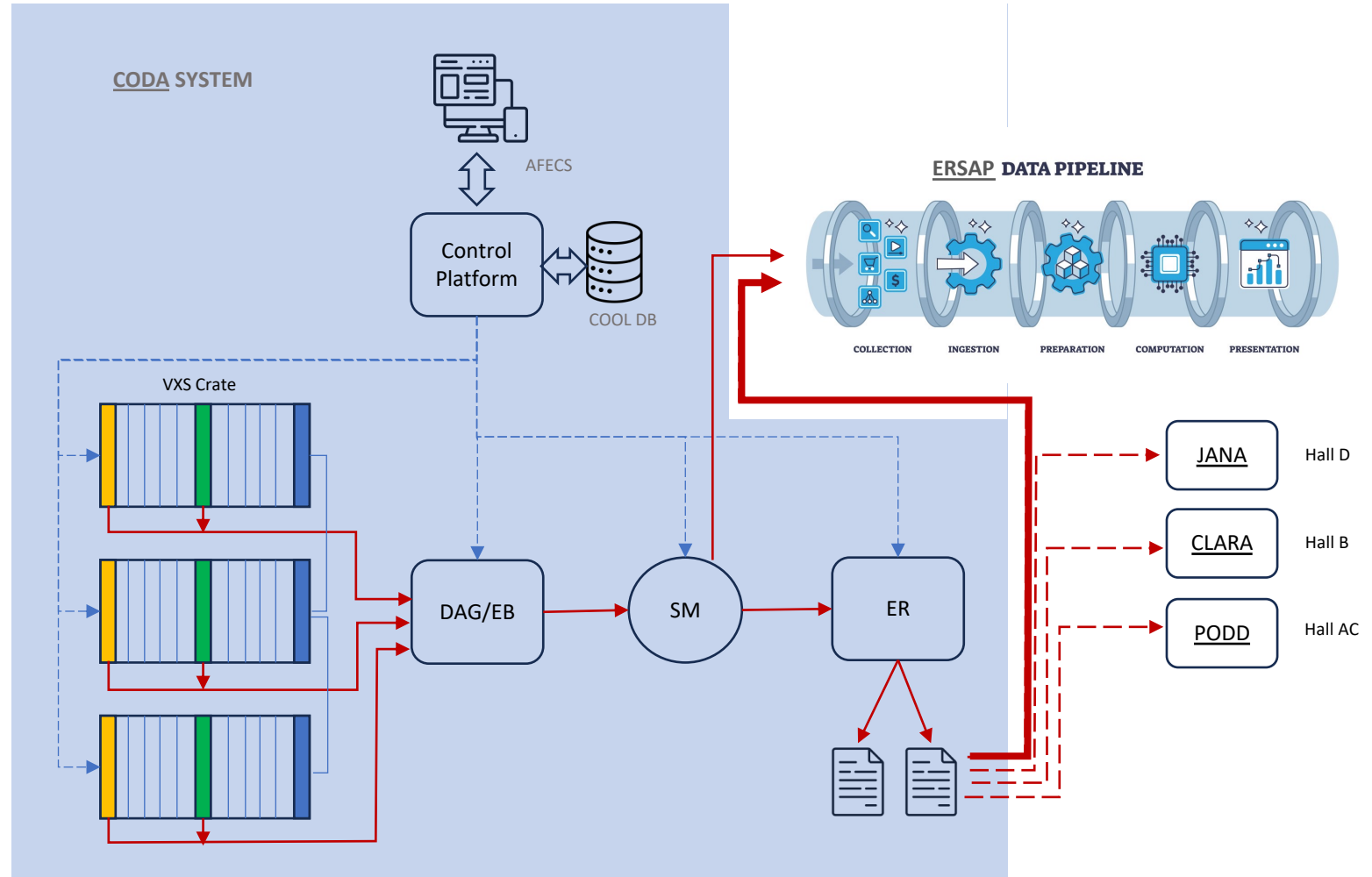


* AI/ML

Courtesy of David Lawrence

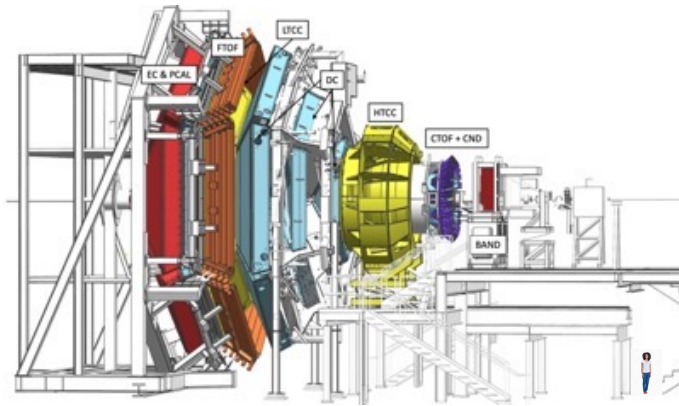
CEBAF Online Data Acquisition and Processing System

- CPU runs a software component ROC. It is responsible for payload board configuration and readout, as well as data formatting and passing it to the next stage.
- VTP relieves the ROC of all the “Readout” tasks and implements them in the FPGAs.
- Triggered or Streaming readout from ALL payload modules in parallel
- The Software ROC is now primarily responsible for configuring, controlling, and monitoring the VTP-based DAQ.
- TI Trigger interface card, responsible for trigger and clock distribution.

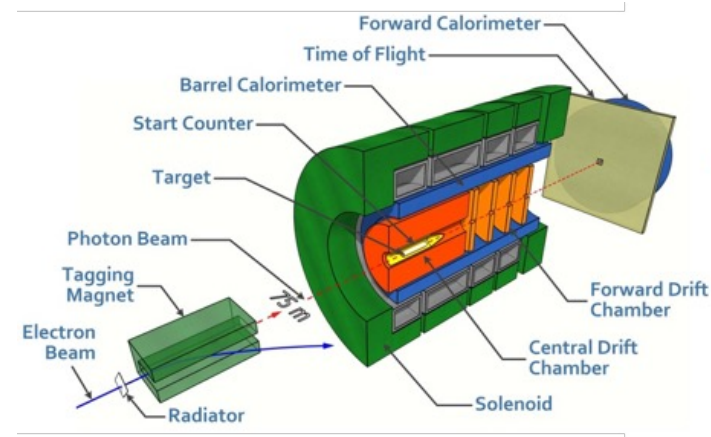


CLARA: CLAS12 Data Processing

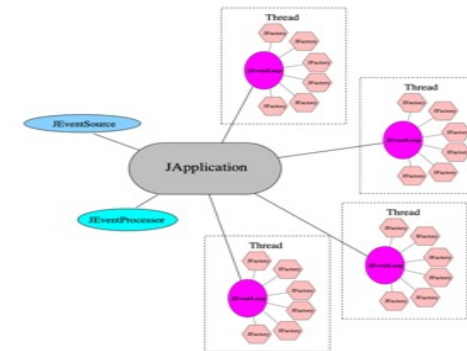
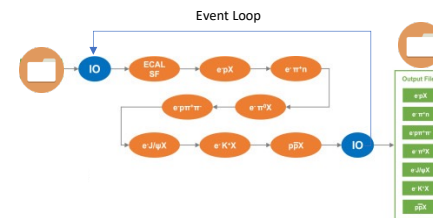
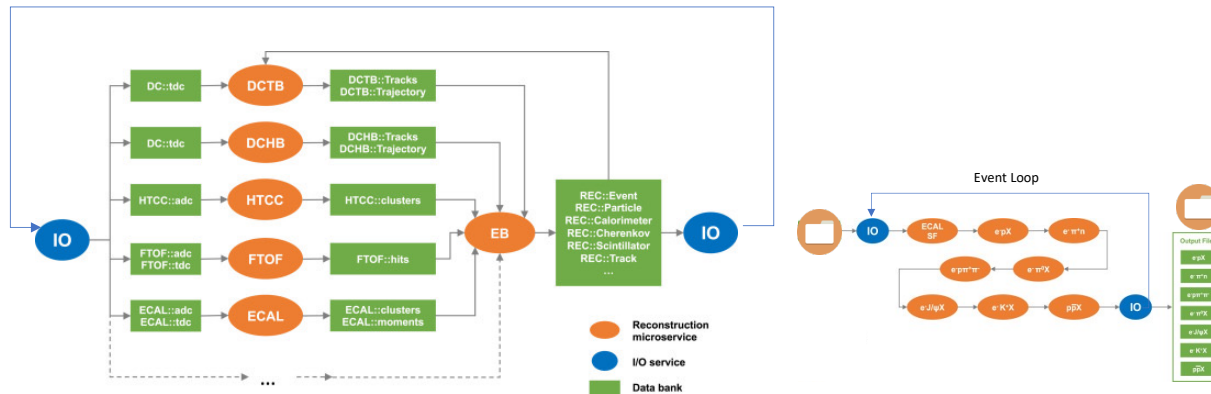
CLAS12 Detector



GlueX Detector



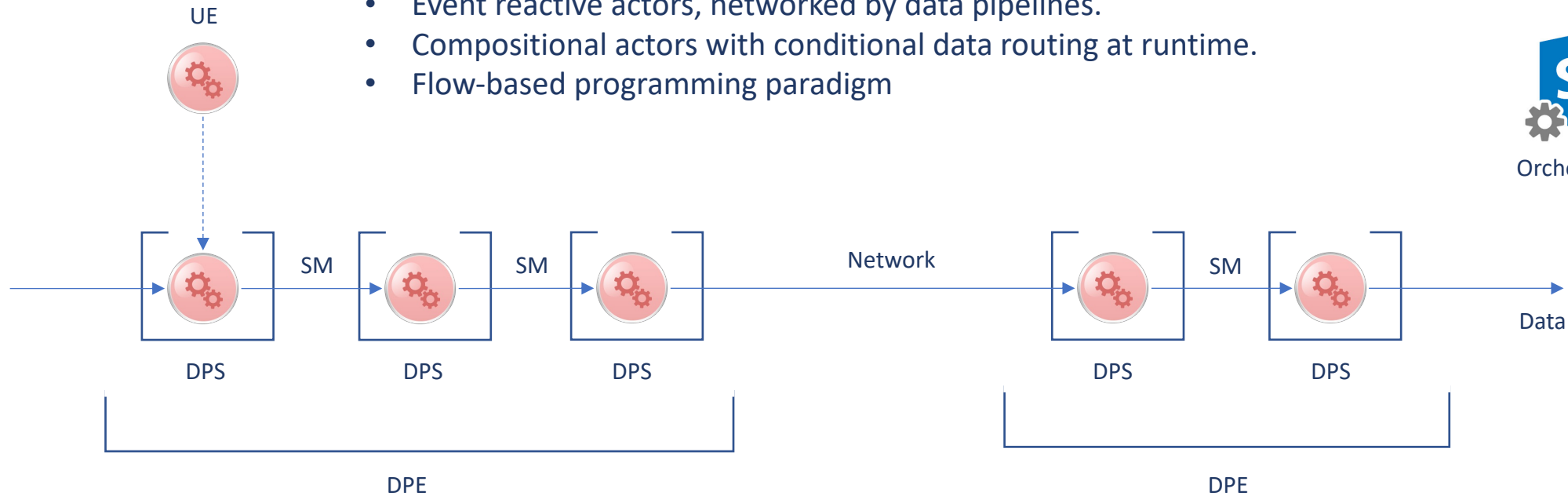
Event Loop



The event loop is part of the application.
They are deployed as a monolith.

ERSAP: Environment for Real-time Streaming, Acquisition, and Processing Framework

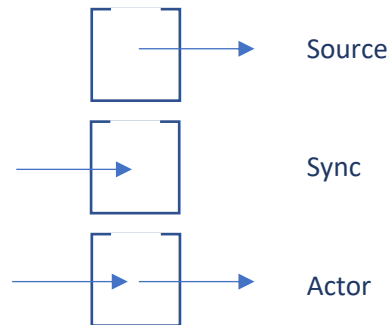
- Event reactive actors, networked by data pipelines.
- Compositional actors with conditional data routing at runtime.
- Flow-based programming paradigm



DPE : Data Processing Environment

SM : Shared Memory

DPS : Data Processing Station



UE : User Engine

Init(Object O)

Object process(Object O)

Data processing station: actor

- User *engine* run-time environment.
- Engine follows data-in/data-out interface.
- Engine gets JSON object for run-time configuration.

UE : User Engine Interface

init(JSON O)

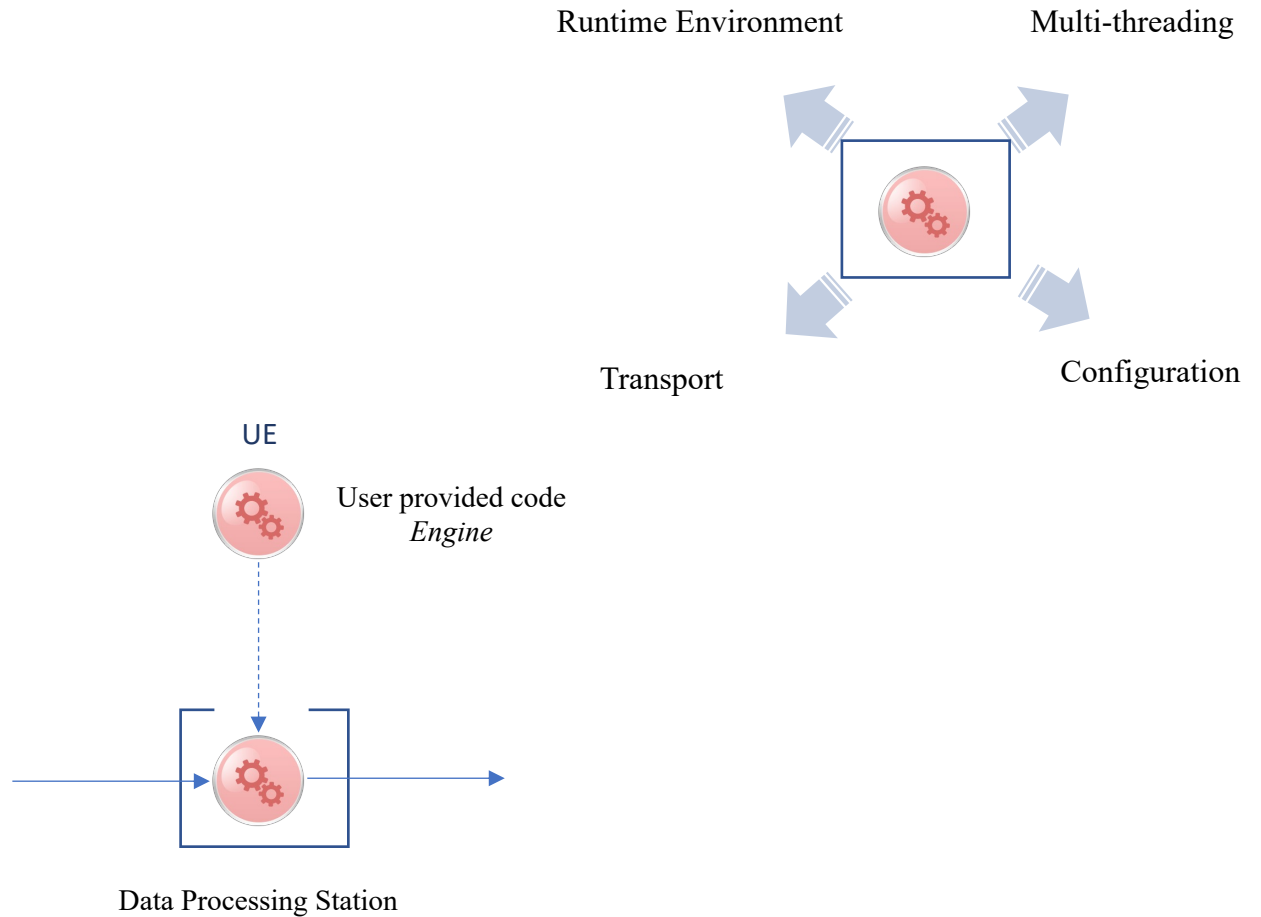
Object *process*(Object O)

Object *process*(Object[] O)

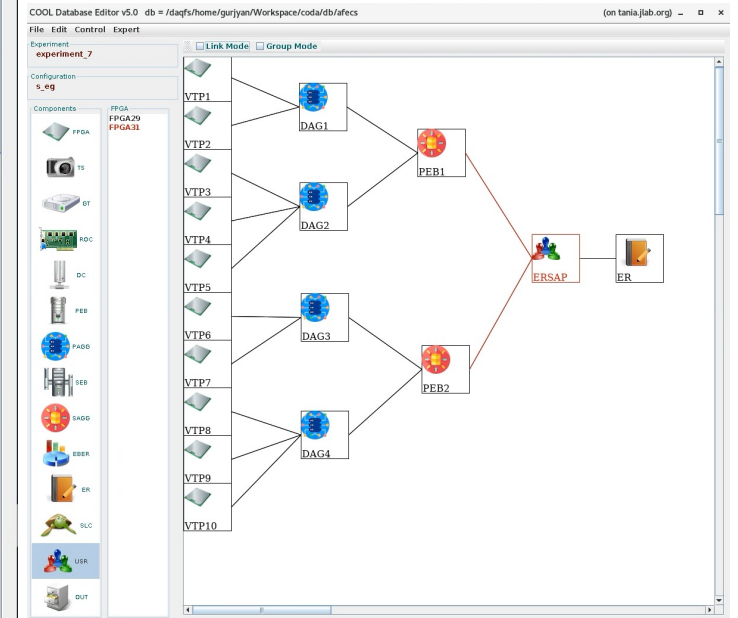
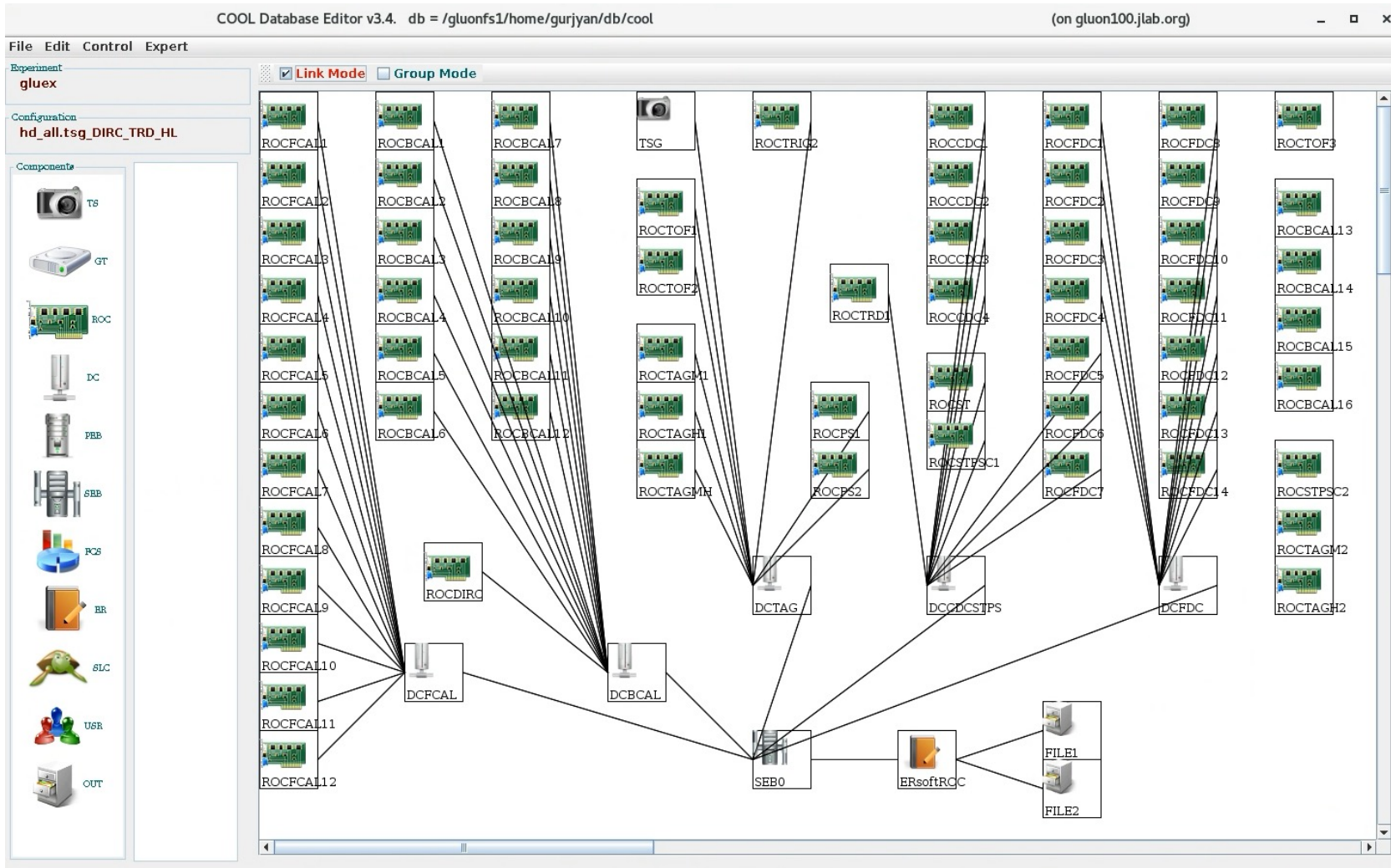
Object *process*(Map<String, Object> O)

Object[] *process*(Object[] O)

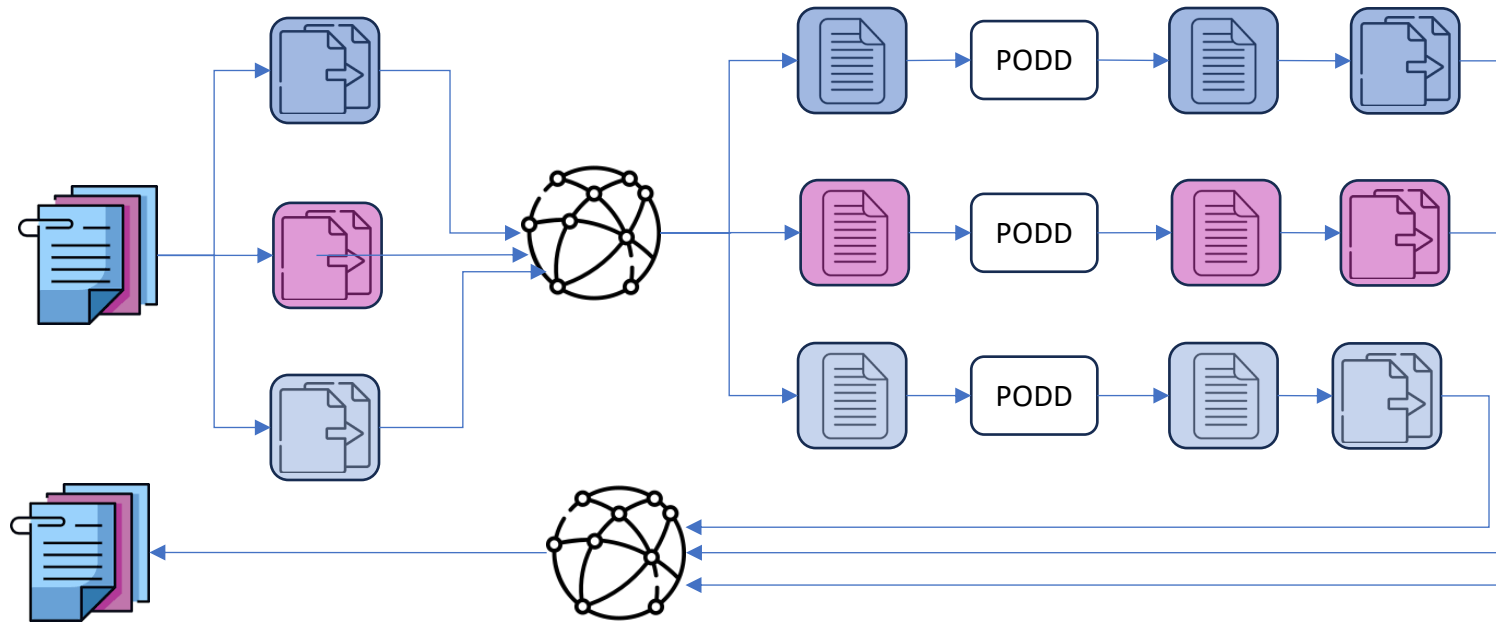
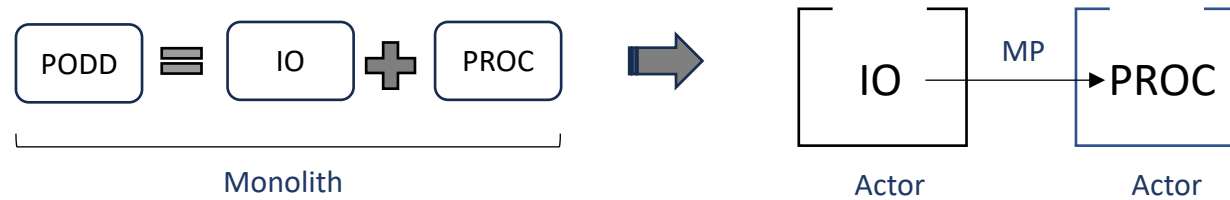
Map<String, Object> *process*(Map<String, Object> O)



Data Acquisition and Processing Pipeline Designer



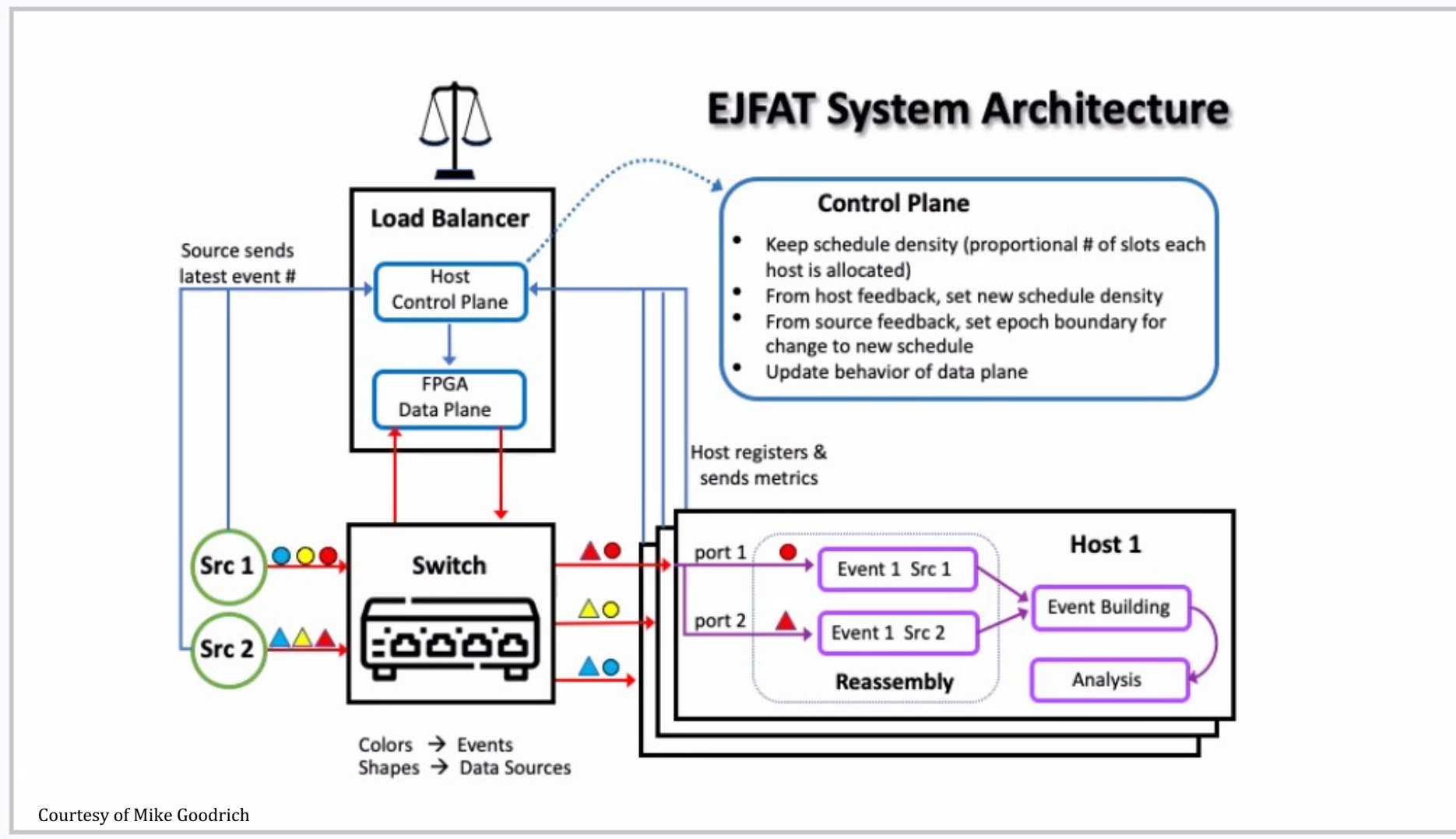
ERSAP Remote Data Provisioning and Orchestration



What do we need for successful data stream processing?

1. A framework capable of designing distributed data processing applications.
2. Low latency, reliable data-stream transport, and load-balancing system.
3. Elastic remote resource allocation, workflow deployment, and orchestration system.

EJFAT: ESnet FPGA Accelerated Transport System



JIRIAF: JLAB Integrated Research Infrastructure Across Facilities

- Elastic, distributed Kubernetes cluster based on opportunistic resources from various computing facilities that function independently of provider-specific setup requirements.
- Seamlessly deploys and scales user workloads over multiple computing facilities to ensure the best use of resources.
- Proactive resource provisioning based on ML models.
- Workflow-facility digital twin based on agent-based Bayesian probabilistic graph model.

Optimizing Resource Provisioning Across Diverse Computing Facilities with Virtual Kubelet Integration

13 Mar 2024, 16:15

30m

Charles B. Wang Center, Stony Brook University

Poster

Track 1: Computing ...

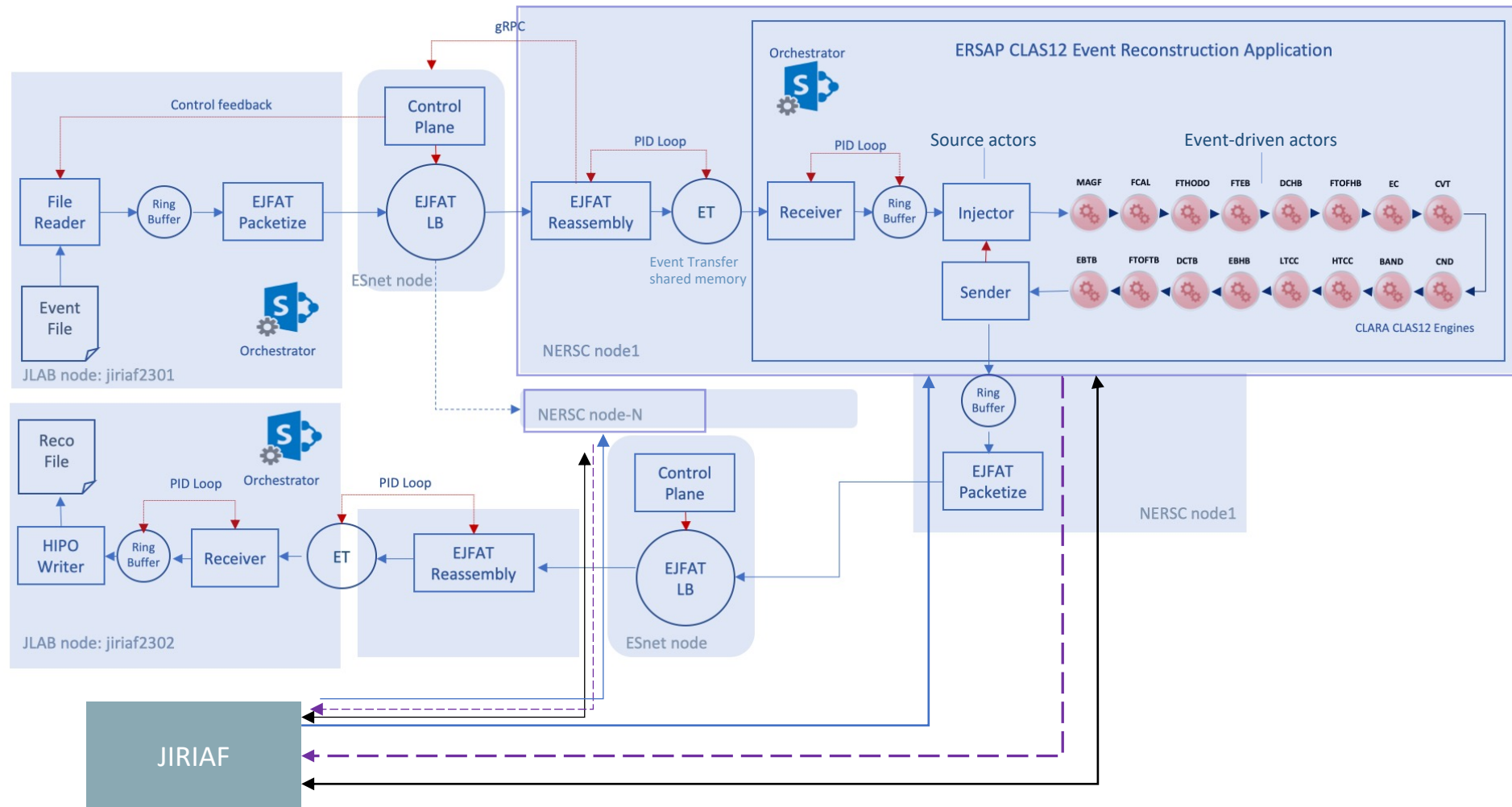
Poster session with co...



Courtesy of Jeng Tsai

Concept Validation Experiment

CLAS12 Data-Stream processing at NERSC



CLAS12 Stream Event Reconstruction: JLAB – Esnet - NERSC

JLAB gurjyan@tania:~

```

Events:      476.2 Hz,  475.6 Avg, total 9788220
Packets:    1780 Hz,   1771 Avg, time: diff = 4000138 usec, abs = 1328662276 epoch msec
Data (+hdrs): 13.8 (13.83) MB/s, 13.69 (13.73) Avg
Events:      475 Hz,  475.6 Avg, total 9790120

Packets:    1754 Hz,   1771 Avg, time: diff = 4000120 usec, abs = 1328666276 epoch msec
Data (+hdrs): 13.6 (13.64) MB/s, 13.69 (13.73) Avg
Events:      475.5 Hz,  475.6 Avg, total 9792022

Packets:    1774 Hz,   1771 Avg, time: diff = 4000119 usec, abs = 1328670276 epoch msec
Data (+hdrs): 13.73 (13.76) MB/s, 13.69 (13.73) Avg
Events:      476 Hz,  475.6 Avg, total 9793926

Packets:    1788 Hz,   1771 Avg, time: diff = 4000119 usec, abs = 1328674276 epoch msec
Data (+hdrs): 13.82 (13.85) MB/s, 13.69 (13.73) Avg
Events:      475 Hz,  475.6 Avg, total 9795826
                
```

NERSC gurjyan@login08:~

```

Events:      475.4 Hz,  313.5 Avg, total 96574
Dropped: evts: 0, 0 total, pkts: 0, 0 total

2023-11-09 17:44:58.951: Processed 500 events in 9.81 s average event time = 19.62
2023-11-09 17:44:59.166: Processed 500 events in 10.83 s average event time = 21.67
Fifo level 950 Avg: 818.27, 86.13%, pid err -1.299115

2023-11-09 17:45:00.215: Processed 500 events in 9.25 s average event time = 18.50
Packets:    1759 Hz,   1175 Avg, time: diff = 4000100 usec, abs = 1561582856 epoch msec
Data (+hdrs): 13.67 (13.7) MB/s, 9.087 (9.11) Avg
Events:      476.2 Hz,  315.6 Avg, total 98479
Dropped: evts: 0, 0 total, pkts: 0, 0 total

Fifo level 826 Avg: 920.73, 96.92%, pid err -1.380536

2023-11-09 17:45:05.476: Processed 500 events in 9.22 s average event time = 18.43
Packets:    186.5 Hz,   1163 Avg, time: diff = 4000110 usec, abs = 1561586856 epoch msec
Data (+hdrs): 1.452 (1.456) MB/s, 8.99 (9.013) Avg
Events:      50.25 Hz,  312.2 Avg, total 98680
Dropped: evts: 0, 0 total, pkts: 0, 0 total

2023-11-09 17:45:07.785: Processed 500 events in 8.83 s average event time = 17.67 ms [ total 12000 events 292.43 s ]
2023-11-09 17:45:08.021: Processed 500 events in 8.86 s average event time = 17.71 ms [ total 10000 events 289.94 s ]
2023-11-09 17:45:09.108: Processed 500 events in 8.89 s average event time = 17.79 ms [ total 10500 events 290.89 s ]
Fifo level 0 Avg: 0.04, 0.00%, pid err -0.891927

Packets:    1018 Hz,   1161 Avg, time: diff = 4000159 usec, abs = 1561590856 epoch msec, cpu = 83
Data (+hdrs): 7.86 (7.88) MB/s, 8.976 (8.999) Avg
Events:      270 Hz,  311.7 Avg, total 99760
Dropped: evts: 1, 1 total, pkts: 4, 4 total
                
```

c (on ejfat-2)

File View Studio

EC Engine Monitoring

Status:

JIRIAF Dashboard at JLAB



Summary

- An Event-driven reactive actor-based framework is under development at Jefferson Lab
- Along with simplifying user application development (e.g. data transport and multithreading) it provides distributed workflow orchestration, simplifying user application migration and deployment.
- It can adopt traditional data processing applications into streaming by suggesting user application decomposition followed by reactive actor representation.
- For applications that are difficult to modify, it provides remote provisioning for data files that are opaque to users.
- Using ERSAP, EJFAT, and JIRIAF, we successfully demonstrated remote data-stream processing for the first time.

Thank You