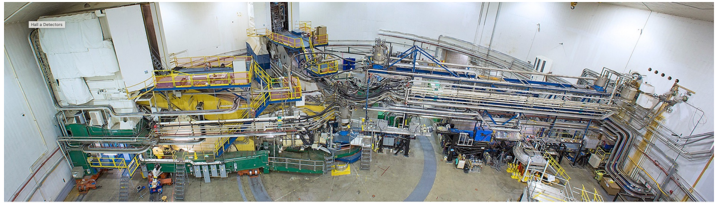
Data Stream Processing & Integrated Research Infrastructure Across Facilities

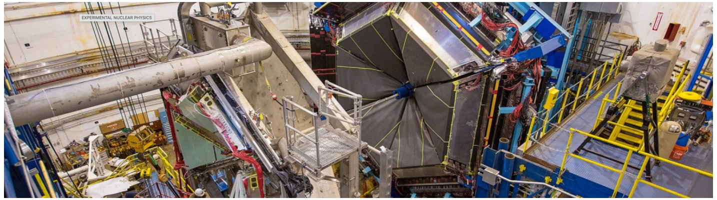
Jeng-Yuan Tsai

JLAB Experimental Halls

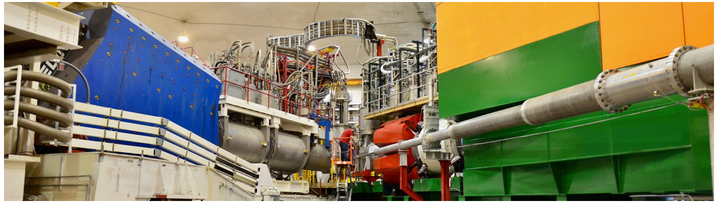
A



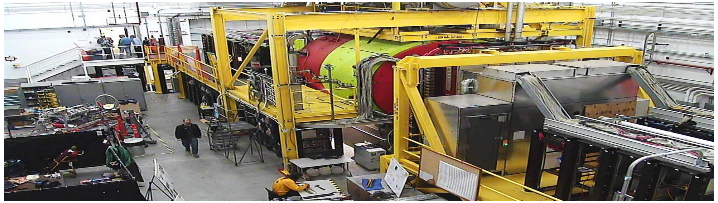
B

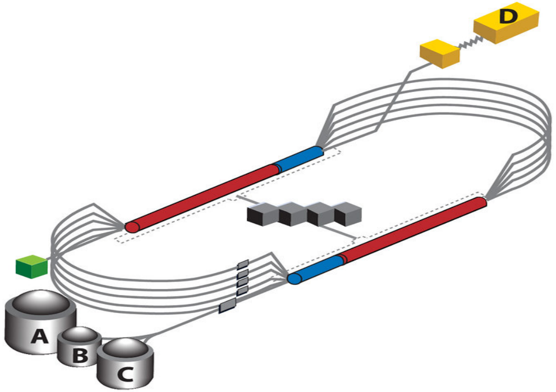


C



D





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

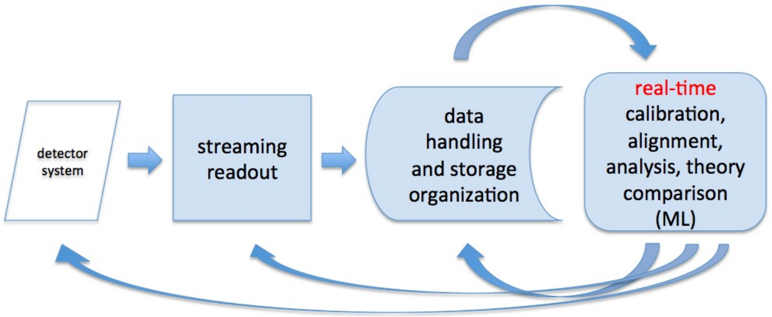
JLAB Grand Challenge in Readout and Analysis for Femtoscale Science

Grand Challenge: Big data generated by NP experiments

- CLAS12/GlueX detectors:
- Data produced ~ several PB/year.

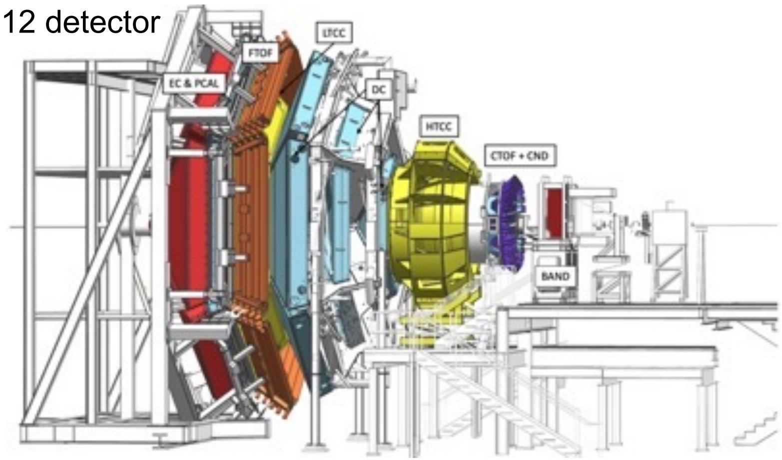
Aim: Remove the separation of data readout and analysis

- Taking advantage of streaming readout and processing

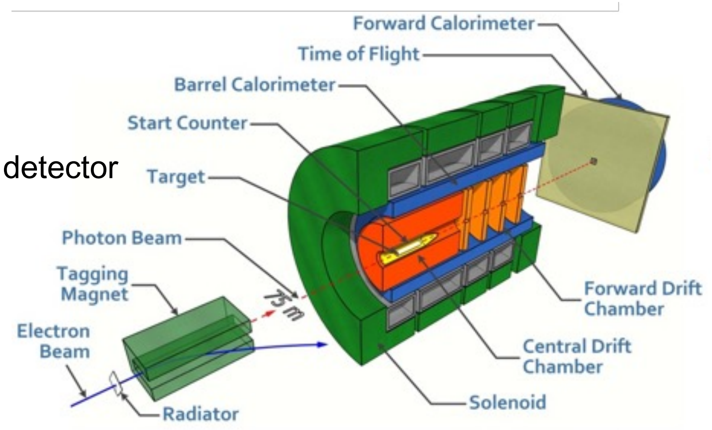


Courtesy of : Amber Boehnlein, Markus Diefenthaler, Rolf Ent, Graham Heyes, Cynthia Keppel, David Lawrence, Brad Sawatzky

CLAS12 detector



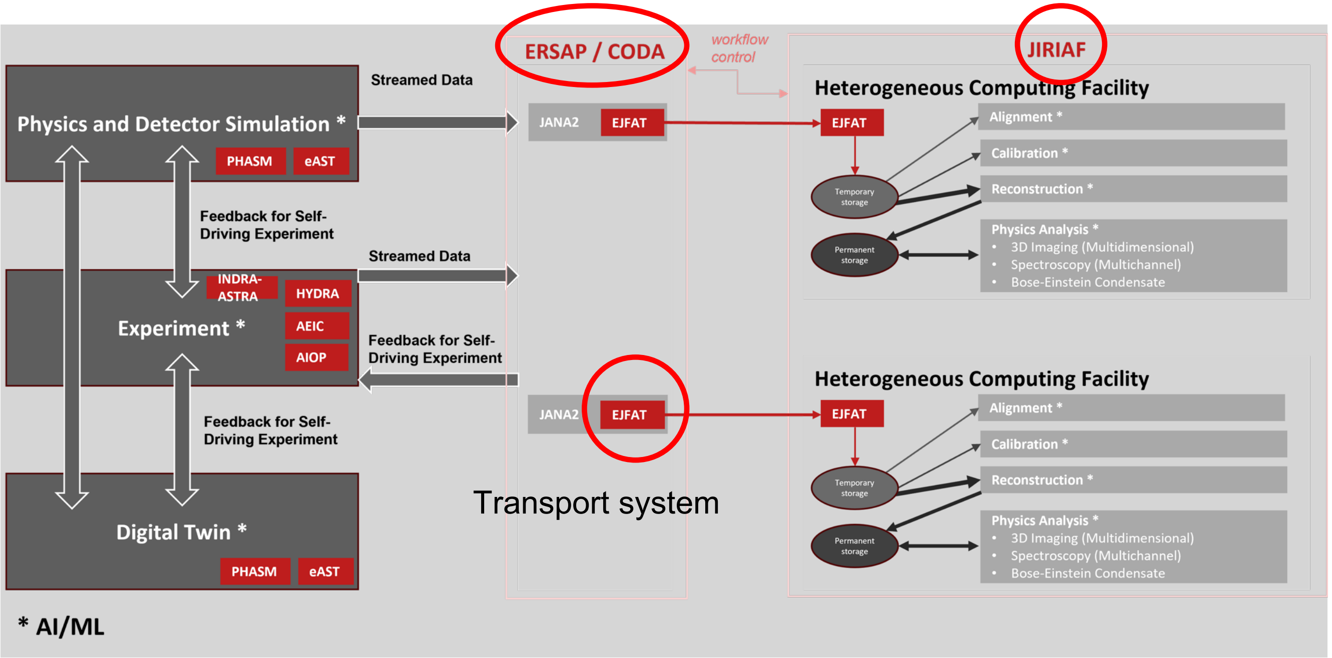
GlueX detector



Today's Talk…

Data acquisition and processing

Computational workflow management



Transport system

Courtesy of : David Lawrence

Traditional DAQ vs. Streaming Readout; Batching vs. Stream Processing

	Traditional Data Acquisition	Streaming Readout
Data Collection	Collects data based on specific triggers	Continuously collects data without interruption
Data Retention	Records only data related to triggered events	Stores more original data for holistic analysis

Fast transport — EJFAT

	Batch Processing	Stream Processing
Approach	Processes data all at once	Processes data continuously as it flows
I/O	I/O operations occur when reading and writing data	Constantly processes the incoming flow of data
Scalability	Challenging due to the need for large memory and disk resources	Inherently scales well due to its real-time nature

ERSAP

JIRIAF

Outline

- **Data Stream Processing**

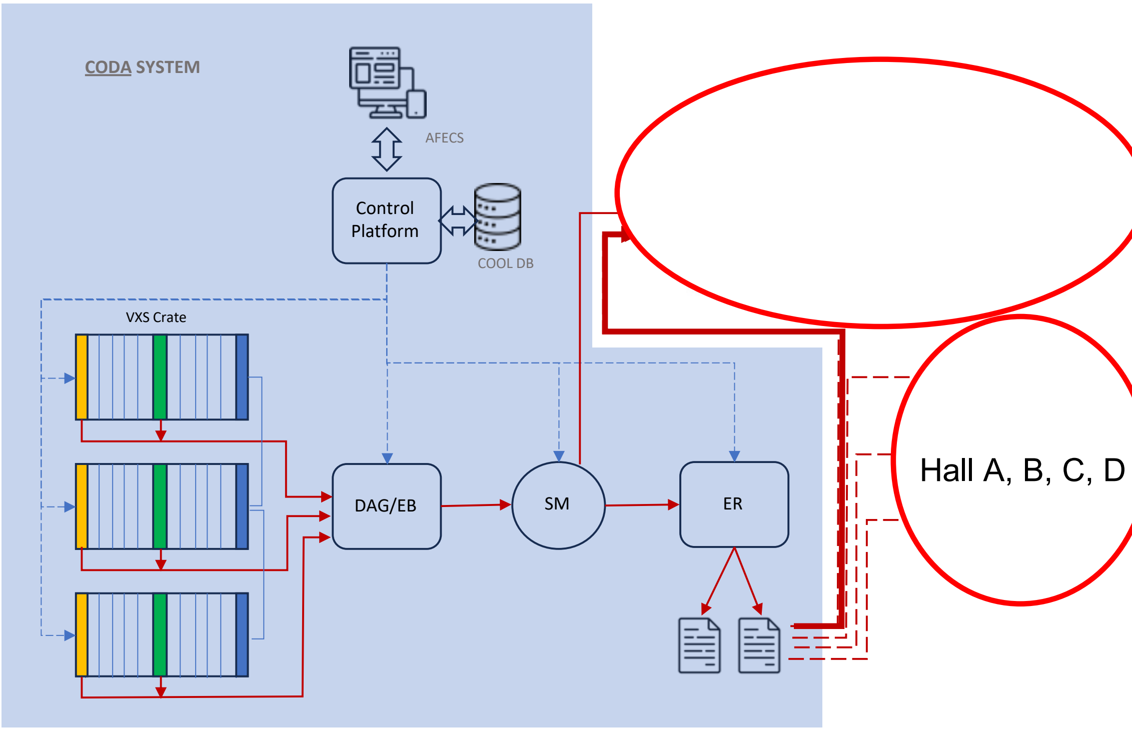
  Environment for Real-time Streaming, Acquisition, and Processing Framework: ERSAP
  ESnet-JLab FPGA Accelerated Transport System: EJFAT

- **Integrated Research Infrastructure Across Facilities**

  Leveraging Computational Resources for Scientific Advancement

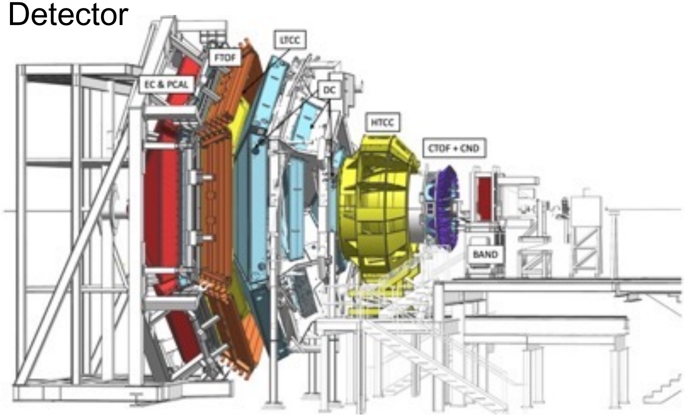
CEBAF Online Data Acquisition (CODA) and Data Processing Frameworks

- CODA as a versatile toolkit—a set of building blocks for data acquisition (DAQ).
- CODA is designed so that it can simultaneously work as a triggered and streaming system.
- Customers of CODA are data processing frameworks.
- A data-stream processing framework called ERSAP, which is to transform how we process physics data, making it seamless, efficient, and unified.

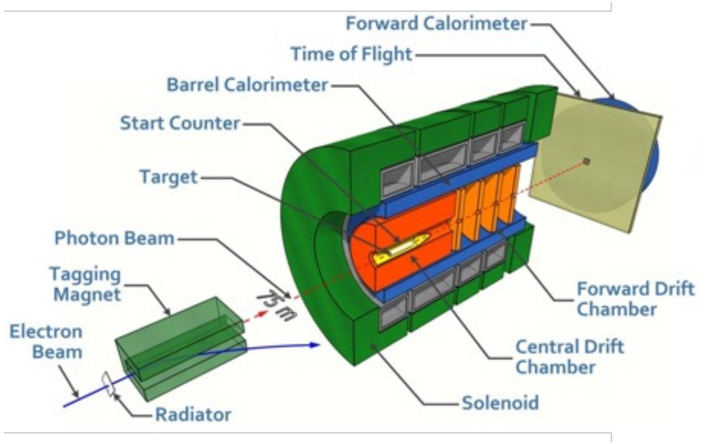


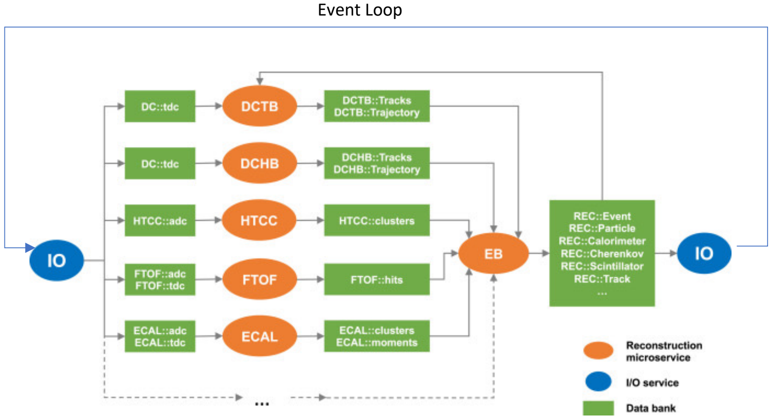
Data Processing in CLAS12 and GlueX

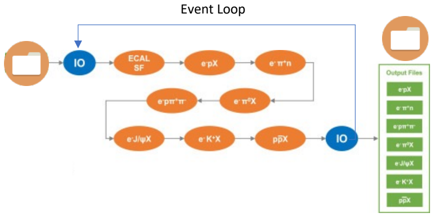
CLAS12 Detector



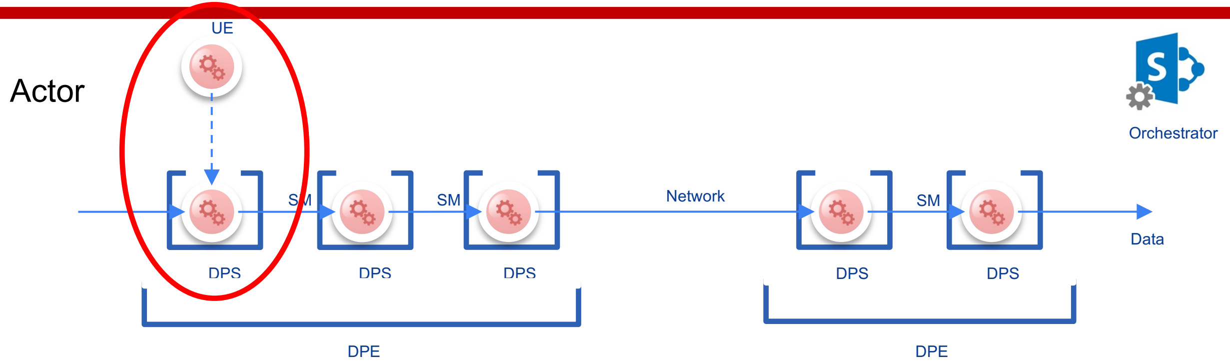
GlueX Detector







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

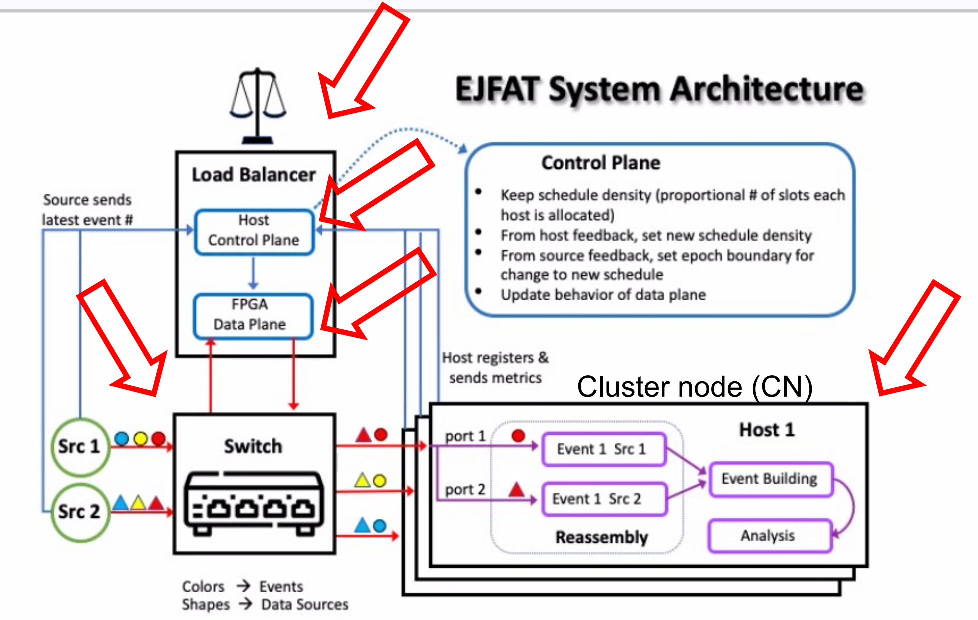


- A reactive, actor-model, and FBP (Flow Based Programing)-based framework designed for data-stream processing in nuclear physics experiments.

Component	Description
User Engines (UE)	Entities that interact with the ERSAP framework.
Shared Memory (SM)	A memory space accessible by multiple processes or actors.
Data Processing Stations (DPS)	Stations responsible for processing data within the framework.
Orchestrator	Coordinates and manages interactions between different components.
Network	Represents the communication channels connecting the framework components.
Data	The information flowing through the system.

ESnet-Jefferson Lab FPGA Accelerated Transport: EJFAT

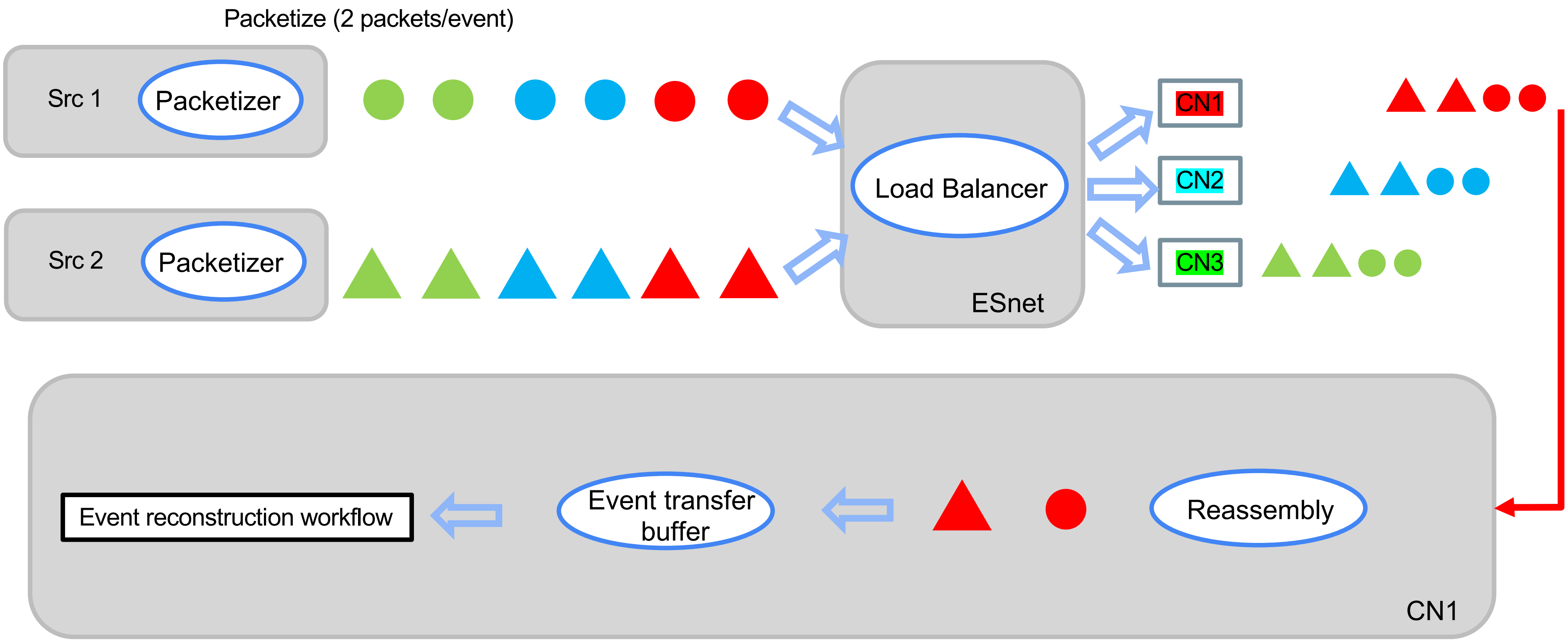
- EJFAT Load Balancer (LB) consists of Control Plane (CP) and Data Plane (DP)
- CP: Dynamically balances CN workloads via CN feedback
- DP: Redirects Data Event UDP packets to Cluster Nodes (CN)
- The use of FPGA technology enhances performance and scalability.



Slide courtesy of Mike Goodrich

Type	Function
DP	Redirects Data Event UDP packets to Cluster Nodes (CN) via Network Address Translation (NAT) with fixed, low latency. Directs Data Channels within an Event to CN ports for parallel processing.
CP	Receives subscriptions from CNs. Dynamically balances CN workloads via CN feedback.

How Data Streams Directed Towards Worker Nodes via EJFAT



Summary

Real-time Data Processing

ERSAP is a reactive, actor-model, and FBP (Flow Based Programming)-based framework designed for data-stream processing in nuclear physics experiments.

Heart of Transmission

The EJFAT makes data transport seamless, ensuring high throughput and minimal delay.

Outline

- **Data Stream Processing**

  Environment for Real-time Streaming, Acquisition, and Processing Framework: ERSAP
  Esnet/JLab FPGA Accelerated Transport System: EJFAT

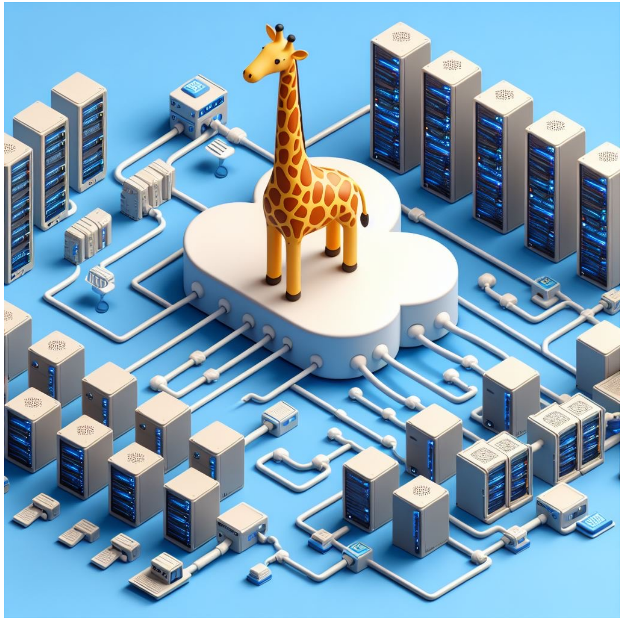
- **Integrated Research Infrastructure Across Facilities**

  Leveraging Computational Resources for Scientific Advancement

Motivation

Importance of Computing Infrastructure Integration

- Online migration and management of workloads from control plane.
- Sharing Data: Feasible to share data across multiple locations.
- Cost Reduction: Reduce costs by reducing the need for duplicate equipment.



Advantages and Challenges in Computing Integration

Advantages

- Central control of workloads across distributed resources.
- Feasible to share data across multiple locations, facilitating collaboration.
- Reduces costs by eliminating the need for duplicate equipment.

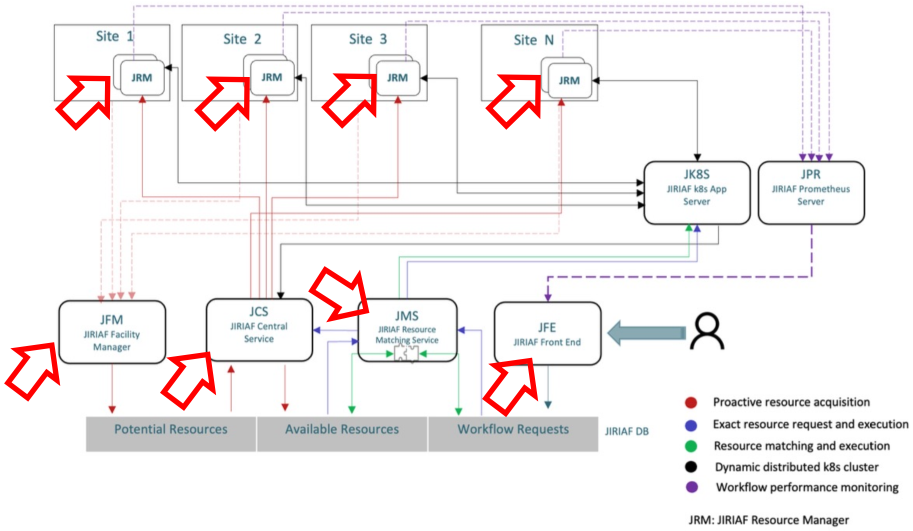
Challenges

- Integrating different computing resources into a unified infrastructure.
- Hardware heterogeneity poses additional difficulties.
- Requires careful planning and coordination to maximize data processing and science output.

JIRIAF Architecture

Components of JIRIAF

- JFM (JIRIAF facility manager) updates the resource pool by scraping resource data from each computing facility periodically.
- JCS (JIRIAF central service) initiates pilot jobs via JRM (JIRIAF resource manager) by leasing resources reported by JFM.
- After JRMs execute, JSC updates the available resource DB table to match user request table using JMS (JIRIAF matching service algorithm).
- User requests are managed through the JFE (JIRIAF front end) component, populating the user workflow request table.

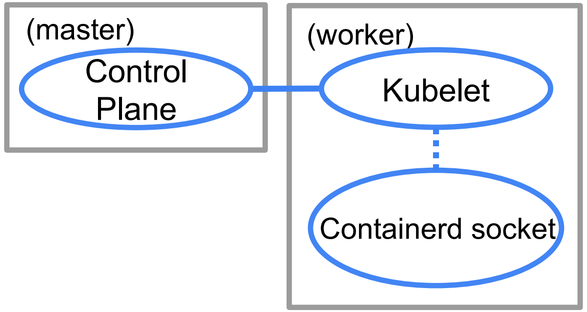


JRM leveraging Kubernetes (K8s)

Brief introduction to k8s

- Control plane is the master node that controls k8s.
- Kubelet is installed in worker nodes and connects to containerd socket.
- **Due to the connection to containerd, installing kubelet requires root credential.**

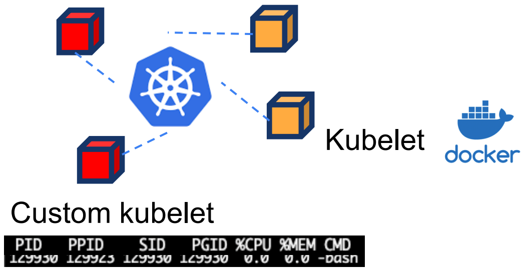
Components in K8s



JRM – custom kubelet backed by custom services

- JRM is a kubelet backed by BASH commands, operating in userspace
- JRM translates objects in regular K8s into BASH commands for JIRIAF:
- Lifecycle of pods and containers is available for JIRIAF.
- Monitoring pods and nodes is available by using regular metrics server in K8s.
- Workload specific monitoring is available with Prometheus server.

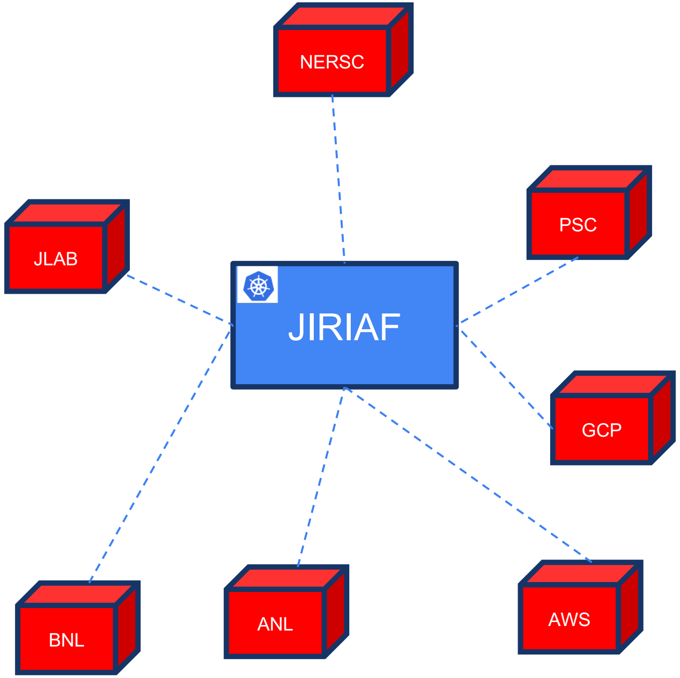
K8s of regular and custom kubelets



Benefit of Using JIRIAF

Controllability and Agile

- JIRIAF framework with K8s flexibility brings us:
  - With control plane of K8s , monitor and control of distributed resources and workloads become feasible. (Distinct from submitting jobs via SLURM)
  - With JRM available in userspace, JIRIAF can utilize wide range of resources.



➢ **JIRIAF is an elastic K8s cluster.**

Project Development Focus - Computing Environments

At the current stage of project development, we concentrate on two subprojects.

1. The first subproject optimizes resource allocation and utilization using statistical/AI methods

2. The second subproject focuses on seamless workflow migration and data stream processing – Demo: Concept validation experiment of remote stream processing

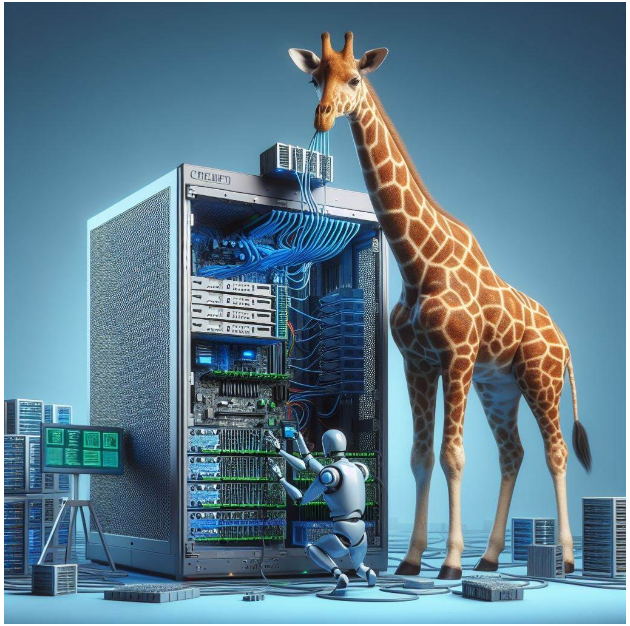
Server Resources

Allocation Overview

- Server resources are essential for efficient performance and reliability.
- Effective allocation ensures optimal utilization of resources.

Management Overview

- Metrics of server resources include CPU, memory, storage, and network bandwidth.
- Proper management of resources improves scalability and response time.



JIRIAF, serving as a central resource control, ensures efficient allocation and effective management

Resource Optimization

Insights:

- Accurate resource estimates improve resource planning.
- Historic utilization data optimizes resource allocation.
- Analyzing the gap between estimates and utilization informs JIRIAF’s future decisions.

Challenges:

- Inaccurate resource estimates can cause overallocation or underutilization.
- Lack of utilization data hampers efficient resource management.
- Closing the gap demands ongoing monitoring and adjustment.

Optimizing Task Execution

1. Efficiently running multiple tasks can greatly improve productivity and reduce wasted time.

2. By prioritizing tasks and utilizing time management techniques, you can ensure important tasks are completed on time.

3. Automation and workflow optimization tools can also help streamline task execution and increase efficiency.

Statistical/AI Approaches to Performance

Benefits of Analyzing Historic Utilization Data

Using data for maintenance

Utilizing stats/AI methods to analyze performance data of critical tasks for effective maintenance and improvement.

Identifying patterns and trends

Leveraging stats/AI approaches to identify patterns and trends that can help maintain consistent performance.

Optimizing task performance

Applying stats/AI techniques to optimize performance and ensure the efficiency of critical tasks.

Remote Data Stream Processing via EJFAT, ERSAP, and JIRIAF

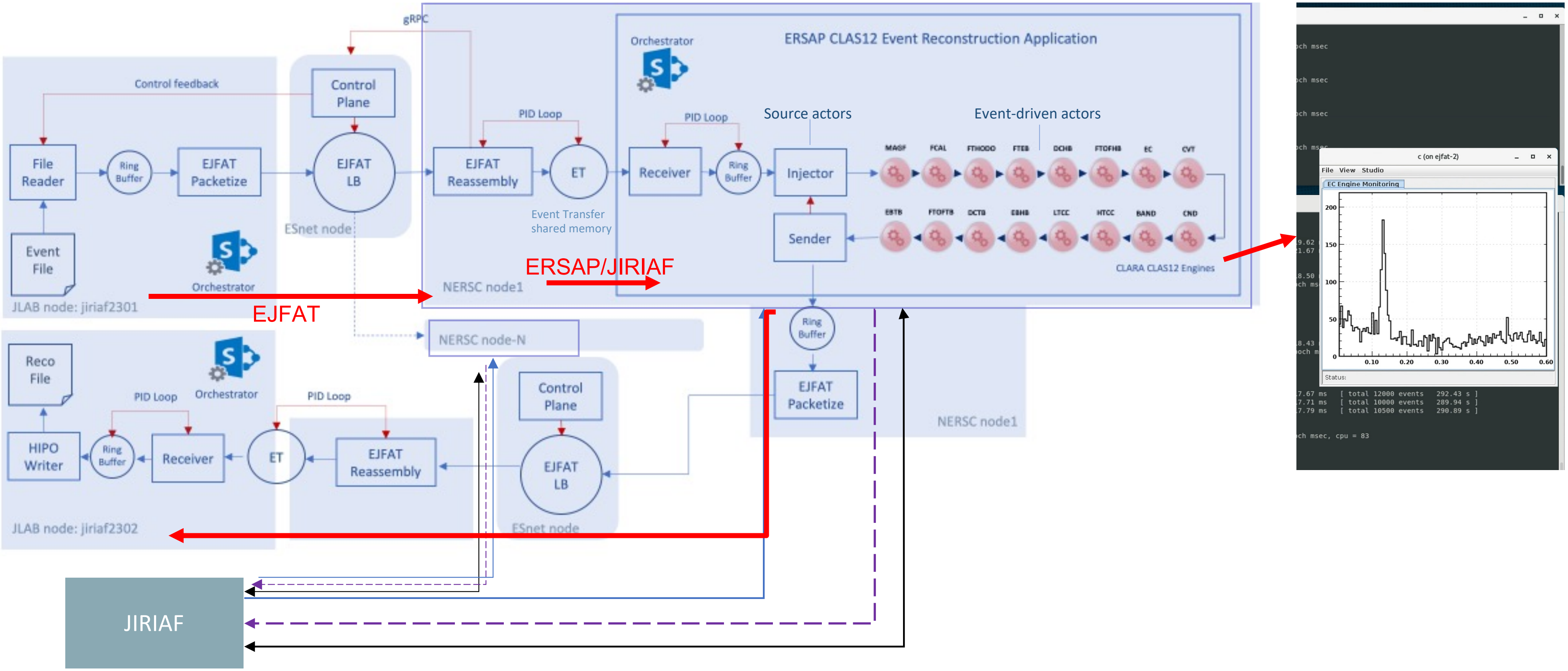
Data Processing Collaboration

- Real-time data streams from JLAB to NERSC via EJFAT load balancer, reducing latency.
- The EJFAT makes data transfer seamless, ensuring high throughput and minimal delay.
- Data is processed at the NERSC Perlmutter HPC cluster, a hub for computational analysis.
- A continuous loop of data streaming supports effective event reconstruction in experiments.

Workflow Deployment

- The ERSAP-based workflow is being tailored for the Perlmutter HPC cluster environment.
- JIRIAF deployment tool plays a crucial role in implementing the event reconstruction services.
- Efficient result streaming back to JLAB is enabled, optimizing the experiment's cycle.

Concept Validation Experiment: CLAS12 Data-Stream processing at NERSC



Summary

Feasibility of Workload Rollovers

This project demonstrates the feasibility of workload rollovers across DOE computing facilities, providing operational resilience and load balancing during peak times.

Bringing Computing Facilities Together

The project mandates uniform data movement, data processing API unification, and resource sharing, bringing science-oriented computing facilities together.

Increasing Science Rate

By implementing static resource provisioning and carving, the science rate will increase, ensuring efficient utilization of resources.

Main Takeaways

1. ERSAP, a reactive, actor-model, and FBP (Flow Based Programming)-based framework, is designed for data-stream processing in nuclear physics experiments.

2. EJFAT integrates smart load balancing and leverages FPGA technology for optimizing data transport and processing.

3. JIRIAF leveraging Kubernetes framework provides a solution for computing infrastructure integration.

Thank you!