CLARA's adaptive workflow management system

Reactive micro-services based data processing orchestration.

V. Gyurjyan

gurjyan@jlab.org

Jefferson Lab

U.S. DEPARTMENT OF ENERGY | Office of Science

JSA

Will address

- Heterogeneous data-processing optimization with CLARA's adaptive workload orchestration
- NUMA-aware workflow management system

Outline

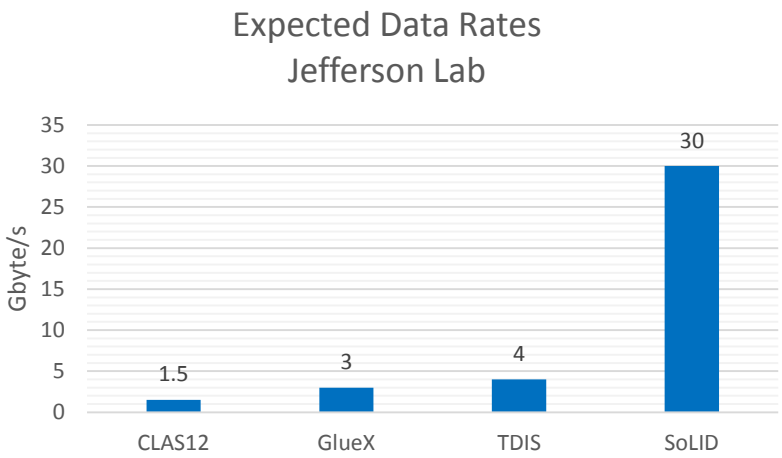
- Problem statement
- Micro-services vs Monolithic architecture
- Flow-based programming paradigm
  - Passive vs Reactive programming
  - Event vs message driven communication
- CLARA reactive micro-services based data-stream processing framework.
- Framework level workflow orchestration
- Data-processing performance optimization across diverse hardware and software infrastructures.

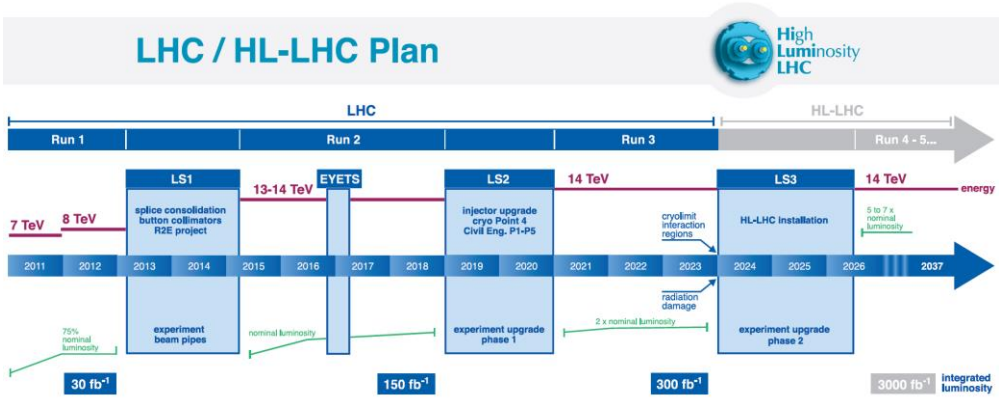
JLAB CLAS12

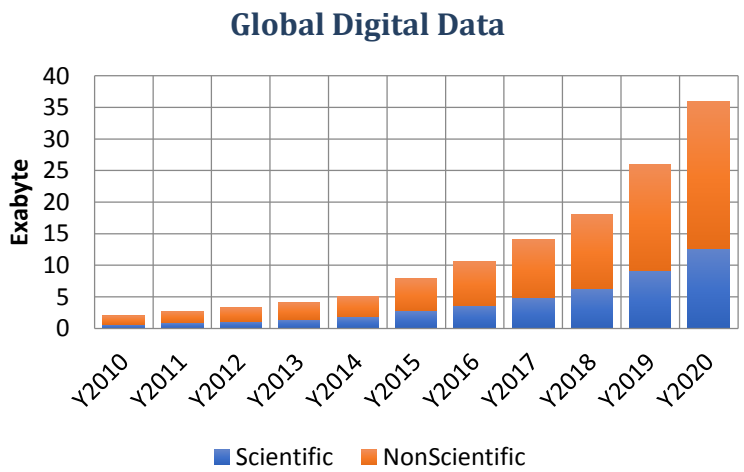
- Thomas Jefferson National Accelerator Facility (TJNAF), commonly known as Jefferson Lab or JLab, is a U.S. national laboratory located in Newport News, Virginia
- Superconducting RF technology based accelerator provides 12 GeV continuous electron beam with a bunch length of less than 1 picosecond.
- Nuclear physics experiments in 4 end- stations (A,B,C,D)
- CLAS12 is a large acceptance spectrometer installed in Hall B to study
  - Quark-gluon interactions with nuclei
  - Nucleon-nucleon correlations
  - Nucleon quark structure imaging,
  - etc.

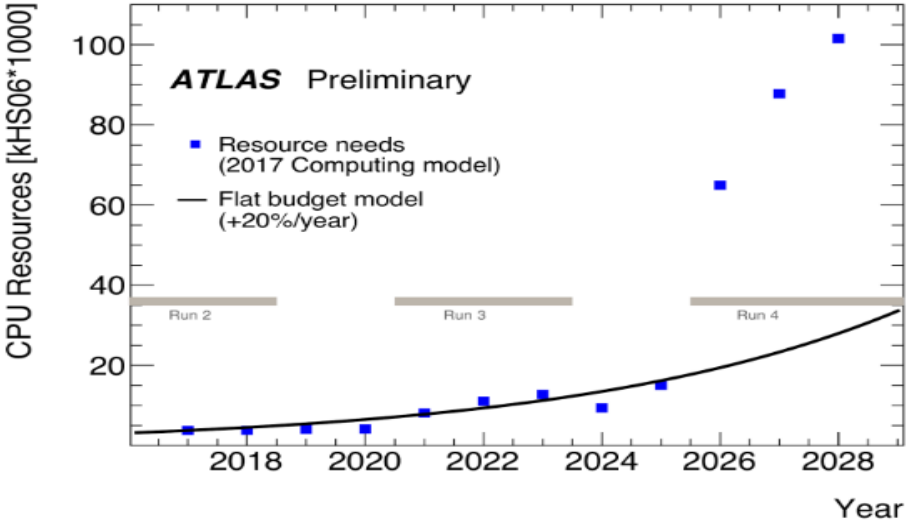
Jefferson Lab

Problem we face



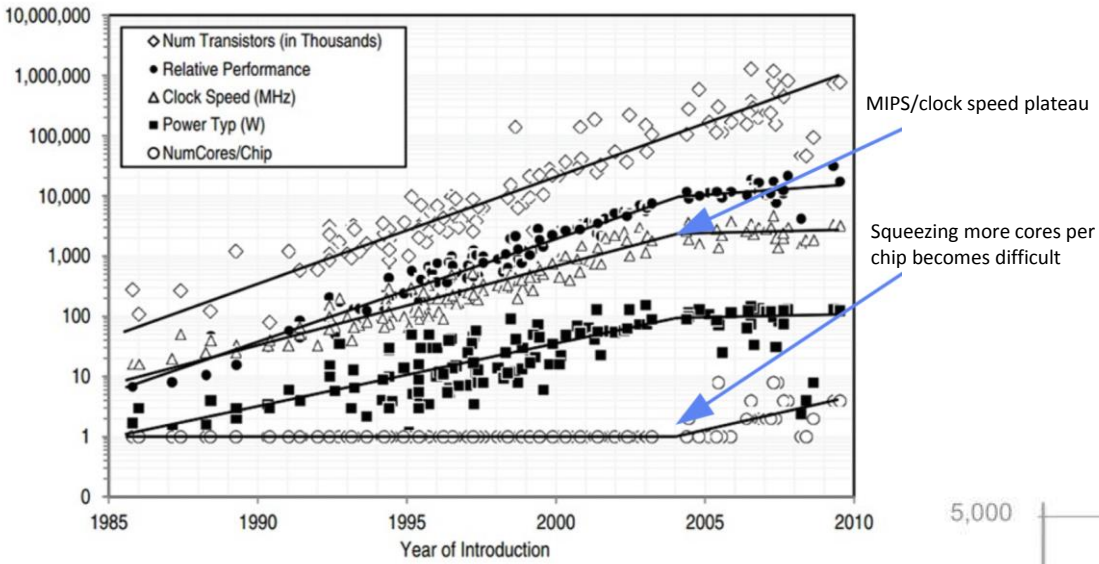




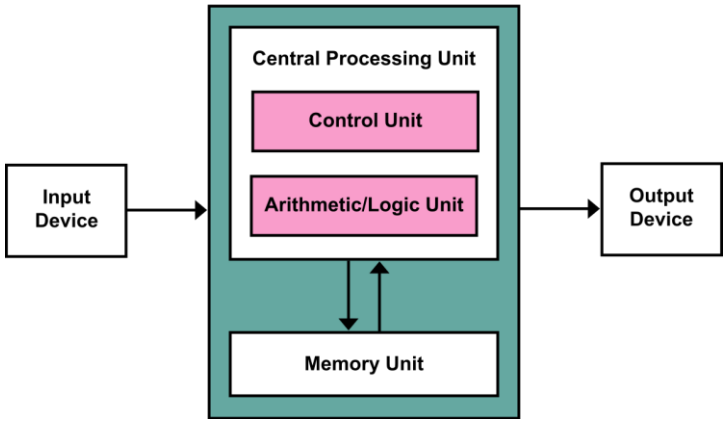


Jefferson Lab

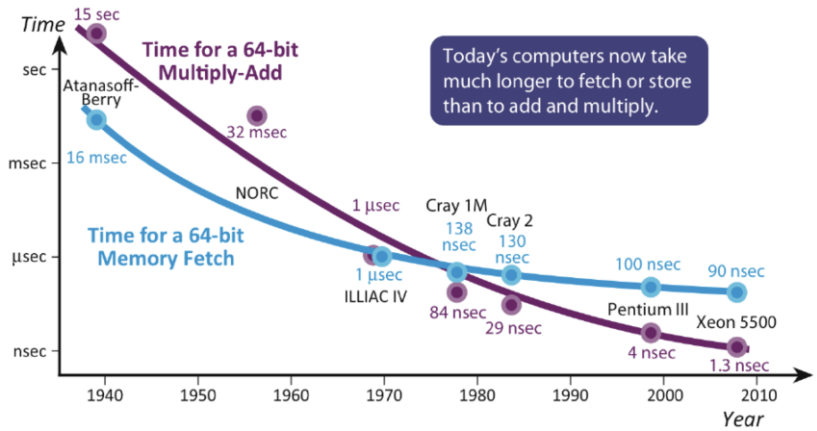
CPU based architecture limitations

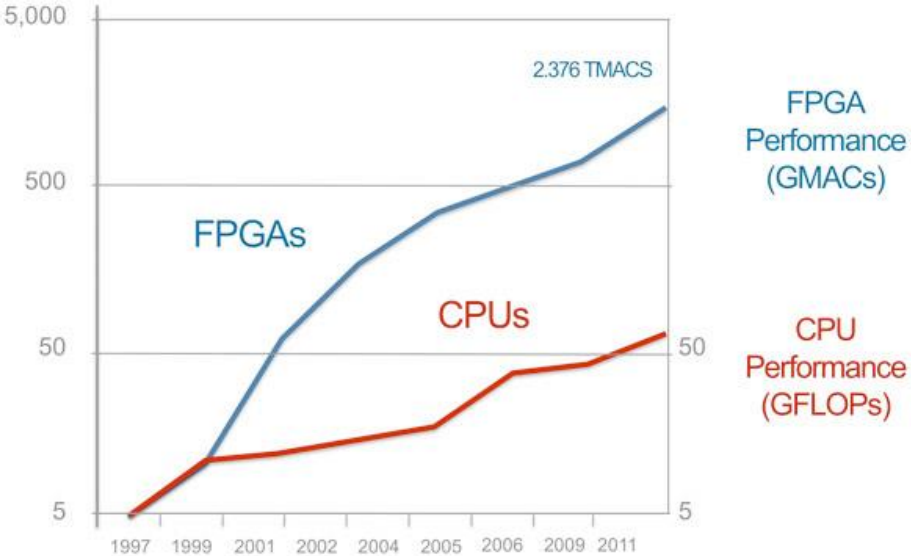


von Neumann Bottleneck



Memory Latency





Jefferson Lab

Only CPU based parallelism is not enough

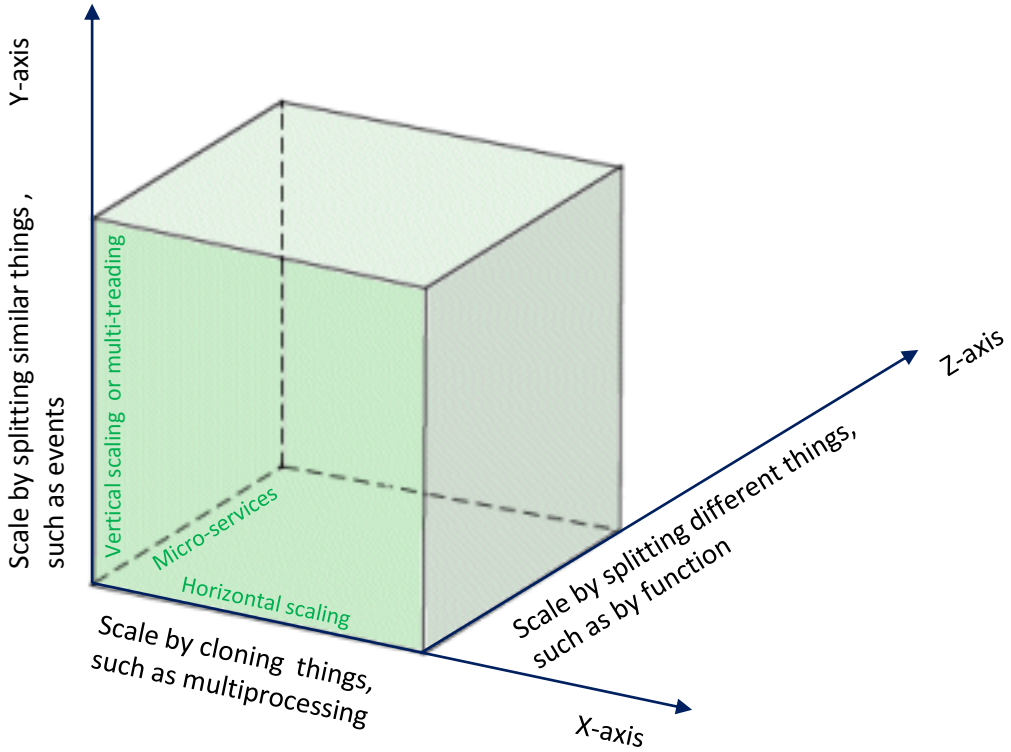
“Frameworks face the challenge of handling the massive parallelism and heterogeneity that will be present in future computing facilities, including multi-core and many-core systems, GPUs, Tensor Processing Units (TPUs), and tiered memory systems, each integrated with storage and high-speed network interconnections.”

“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

Jefferson Lab

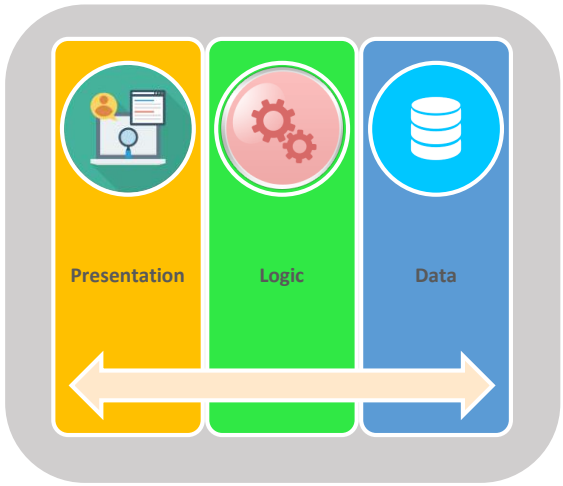
The Scale-Cube



The Art of Scalability. by Martin L. Abbott and Michael T. Fisher. **ISBN-13:** 978-0134032801

Jefferson Lab

Micro-services vs Monolithic architecture

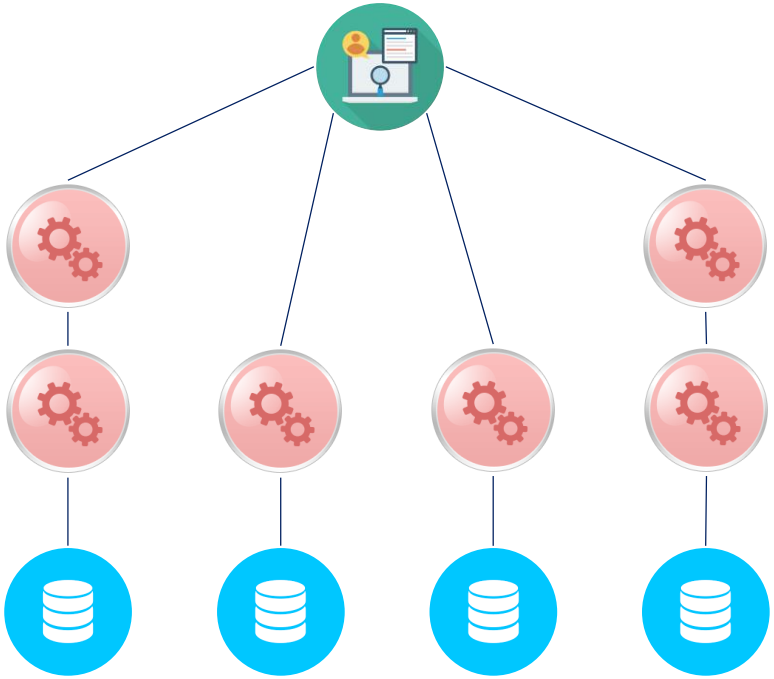


Pros

- Strong coupling, thus better performance
- Full control of your application

Cons

- No agility for isolating, compartmentalizing and decoupling data processing functionalities, suitable to run on diverse hardware/software infrastructures
- No agility for rapid development or scalability



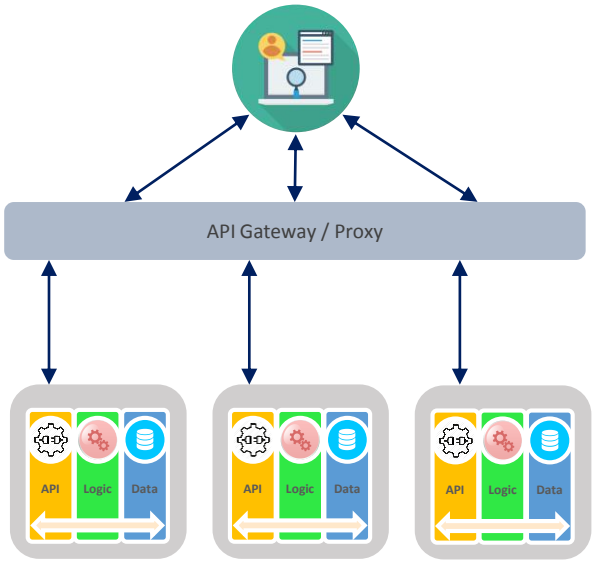
Pros

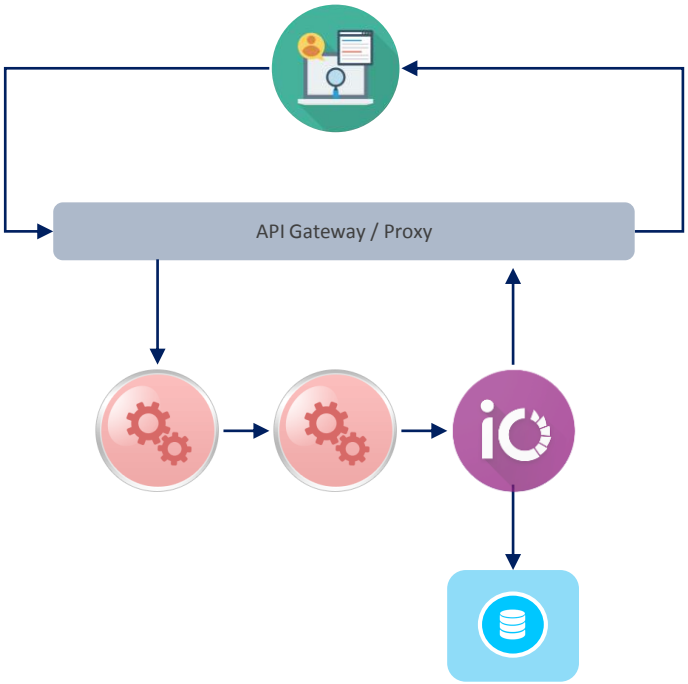
- Technology independent
- Fast iterations
- Small teams
- Fault isolation
- Scalable

Cons

- Complexity networking (distributed system)
- Requires administration and real-time orchestration

What is micro about a service?



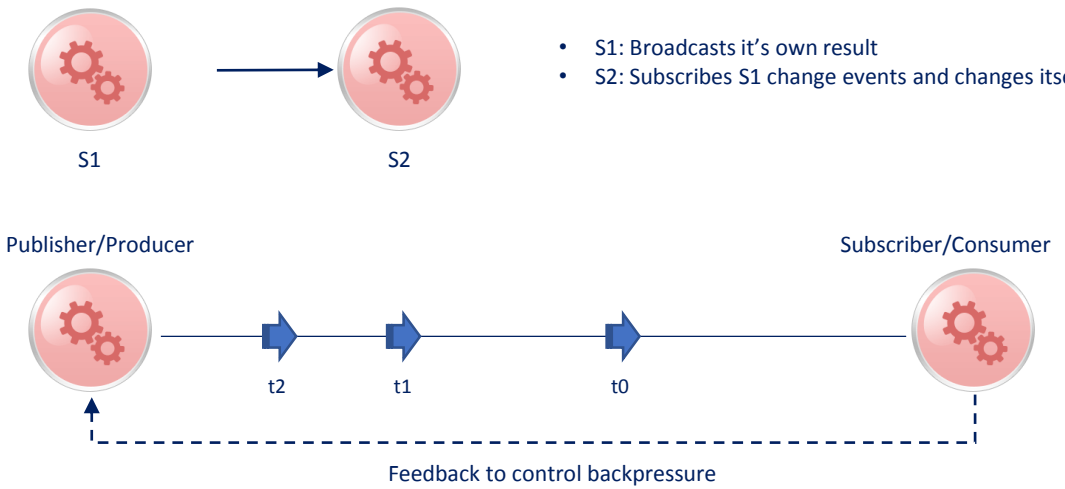


Jefferson Lab

Passive vs Reactive

- S1: Proactive, responsible for change in S2
- S2: Passive, unaware of the dependency

Passive programming



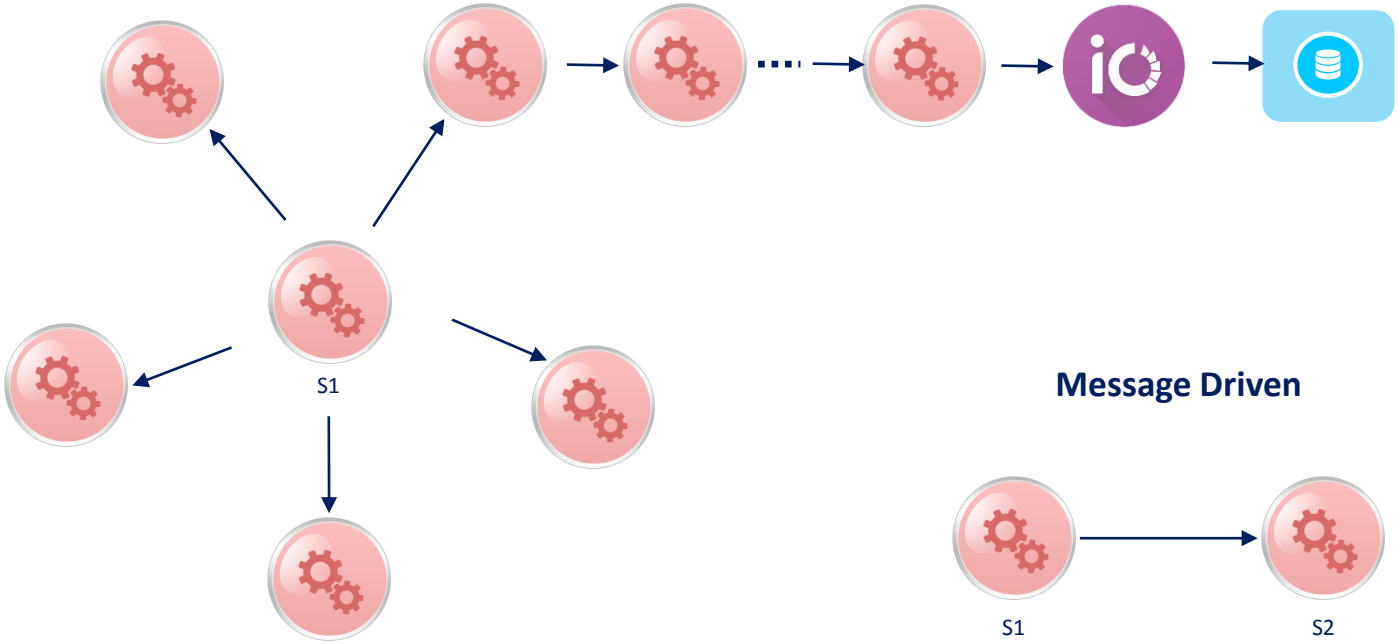
- S1: Broadcasts it's own result
- S2: Subscribes S1 change events and changes itself

Reactive programming
Enables event driven stream processing

Jefferson Lab

Event-Driven vs Message-Driven

Event Driven



S1 event broadcasting

Message Driven

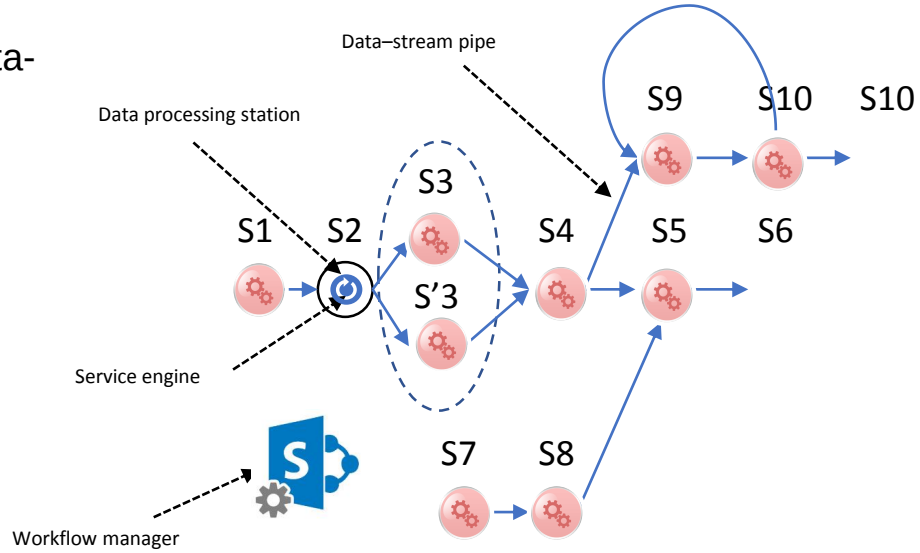
S1 message has a clear destination

Jefferson Lab

CLARA Framework

Reactive, event-driven data-stream processing framework that implements micro-services architecture and FBP

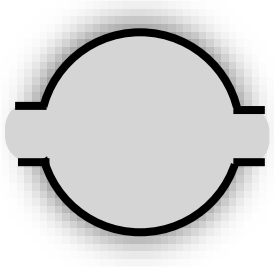
- Defines streaming transient-data structure
- Provides service abstraction (data processing station) to present user algorithm (engine) as an independent service.
- Defines service communication channel (data-stream pipe) outside of the user engine.
- Stream-unit level workflow management system and API
- Supports C++, JAVA, Python languages



http://claraweb.jlab.org
https://claraweb.jlab.org/clara/docs/clas/hands-on.html

Jefferson Lab

Basic components and a user code interface

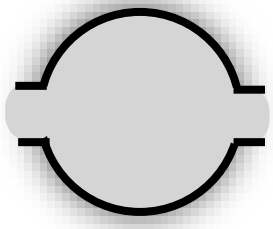


Data Processing Station

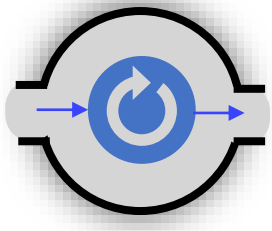
Data-Stream Pipe

Orchestrator

Data processing Engine



Data Processing Station



Data Processing Micro-Service

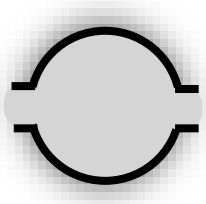
Engine Tutorials

- https://claraweb.jlab.org/clara/docs/quickstart/java.html
- https://claraweb.jlab.org/clara/docs/quickstart/cpp.html
- https://claraweb.jlab.org/clara/docs/quickstart/python.html

Data Processing Station

Runtime Environment

Multi-threading



Data Processing Station

Communication

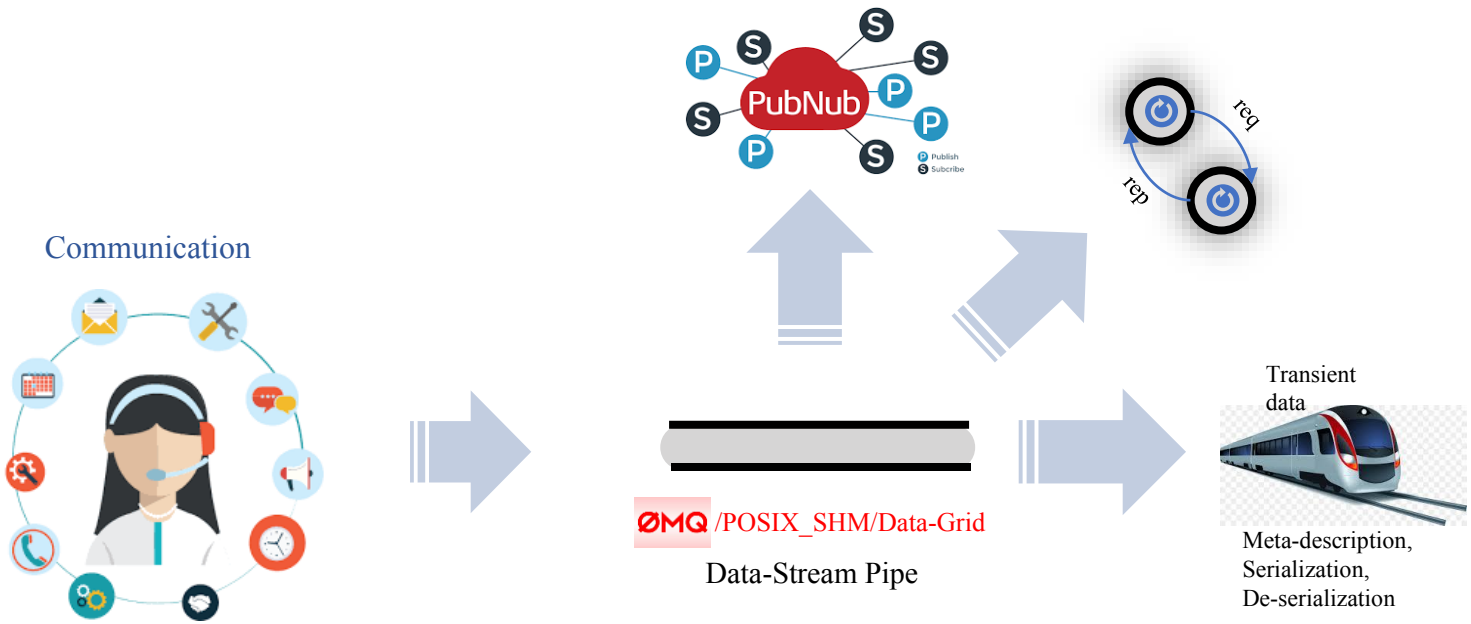
Configuration

Language Bindings

- https://github.com/JeffersonLab/clara-java.git
- https://github.com/JeffersonLab/clara-cpp.git
- https://github.com/JeffersonLab/clara-python.git

Jefferson Lab

Data Stream Pipe

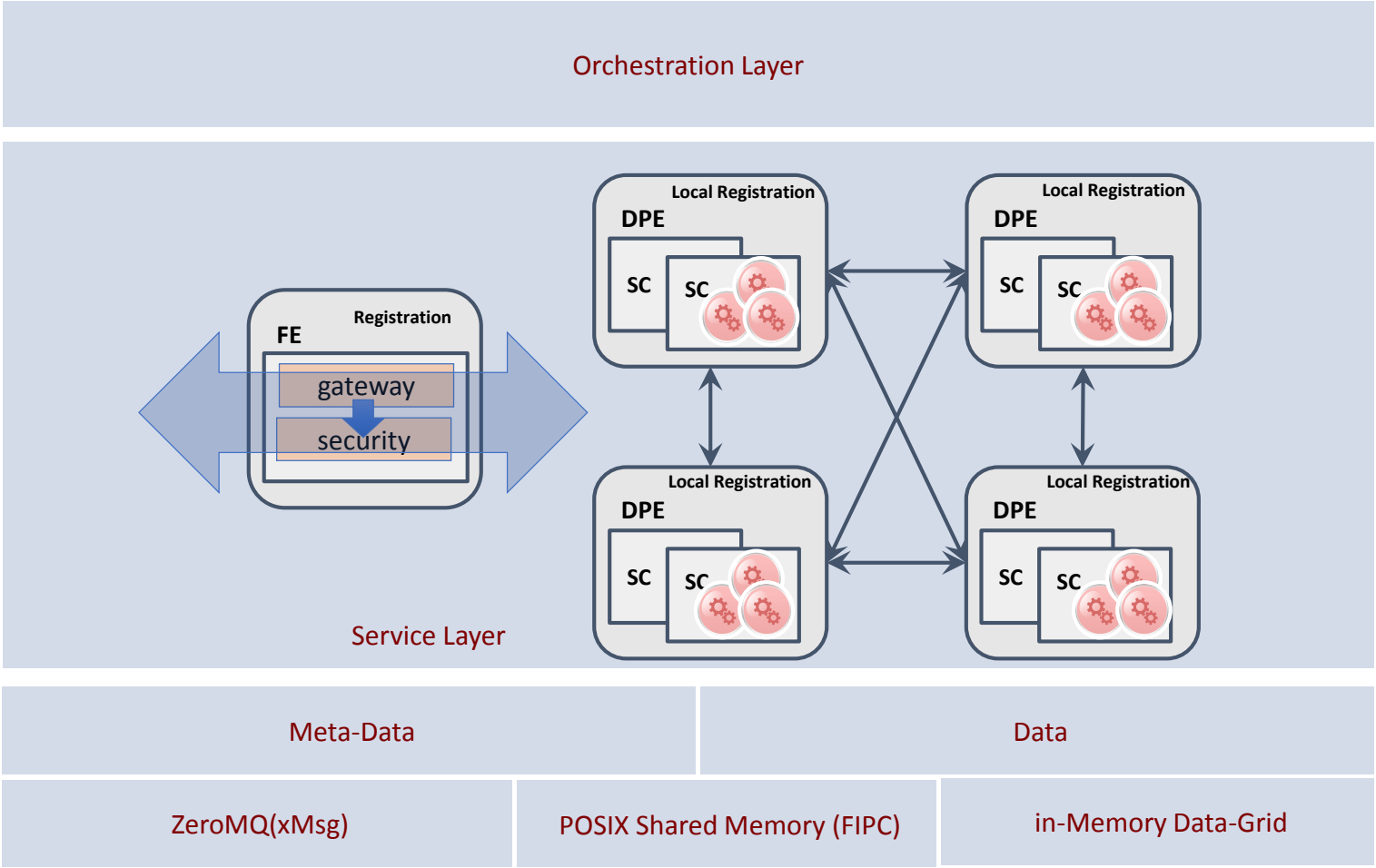


Language Bindings

- https://github.com/JeffersonLab/xmsg-java.git
- https://github.com/JeffersonLab/xmsg-cpp.git
- https://github.com/JeffersonLab/xmsg-python.git

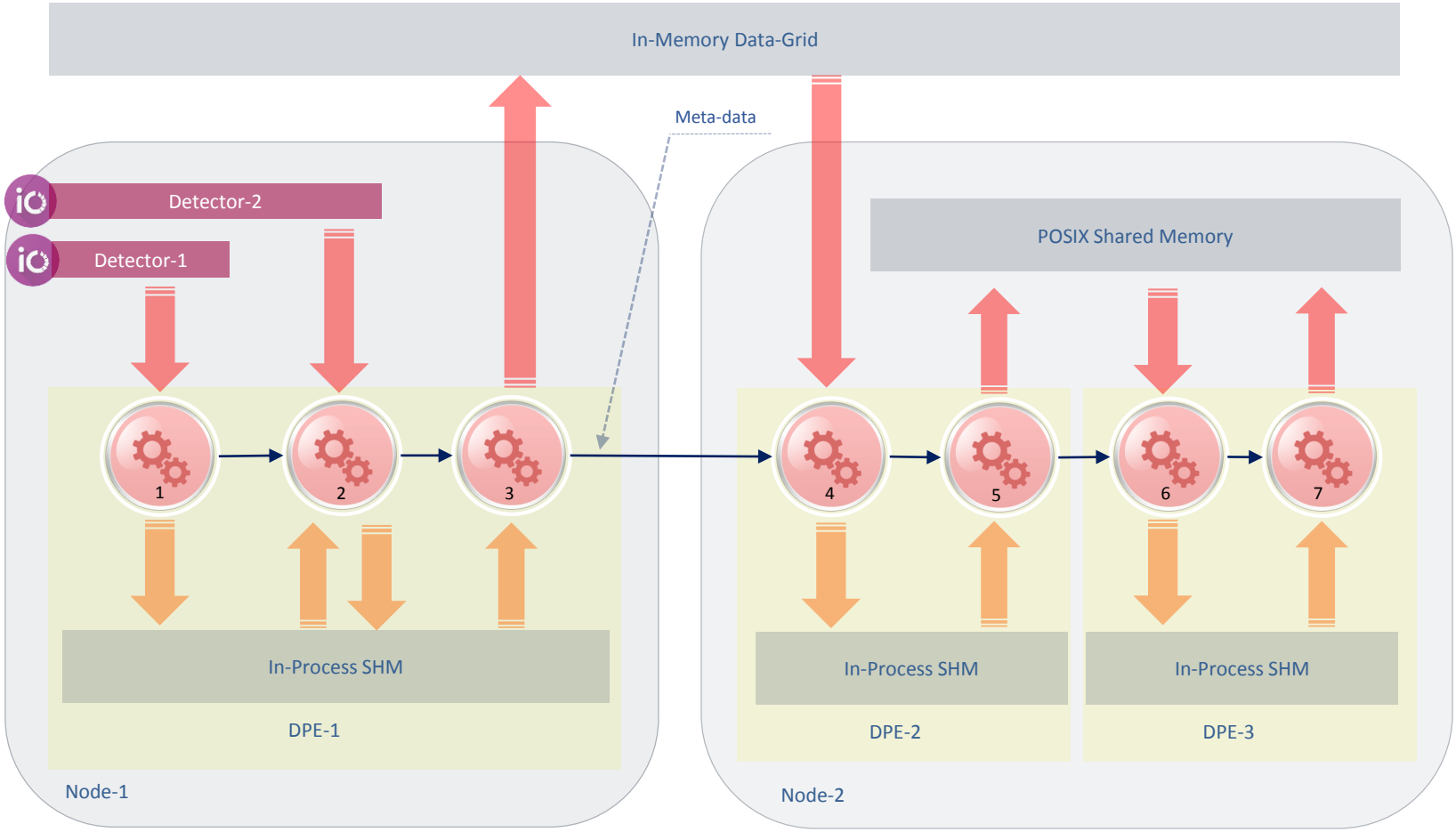
Jefferson Lab

Structure



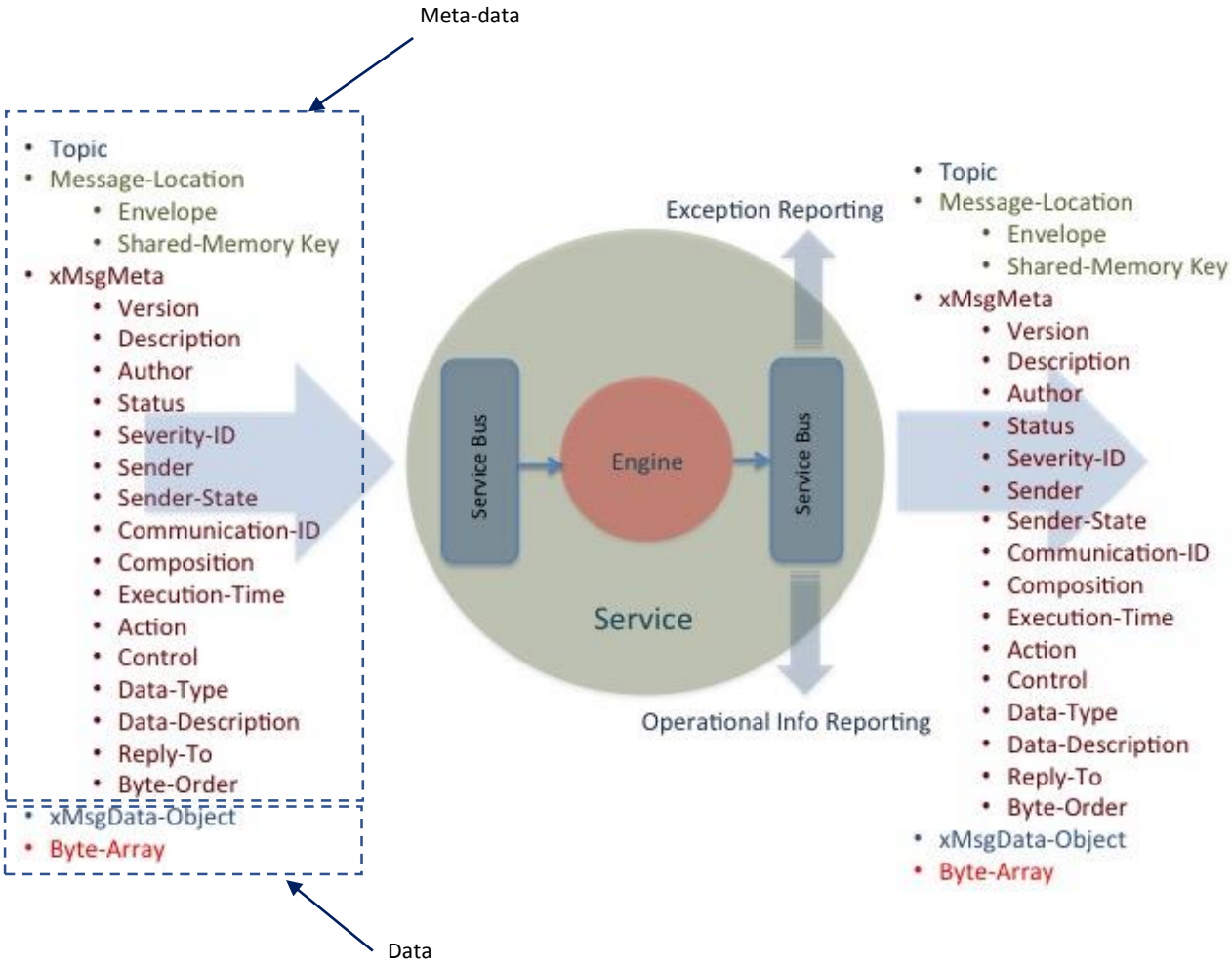
Jefferson Lab

Streaming Data-Flow Processing



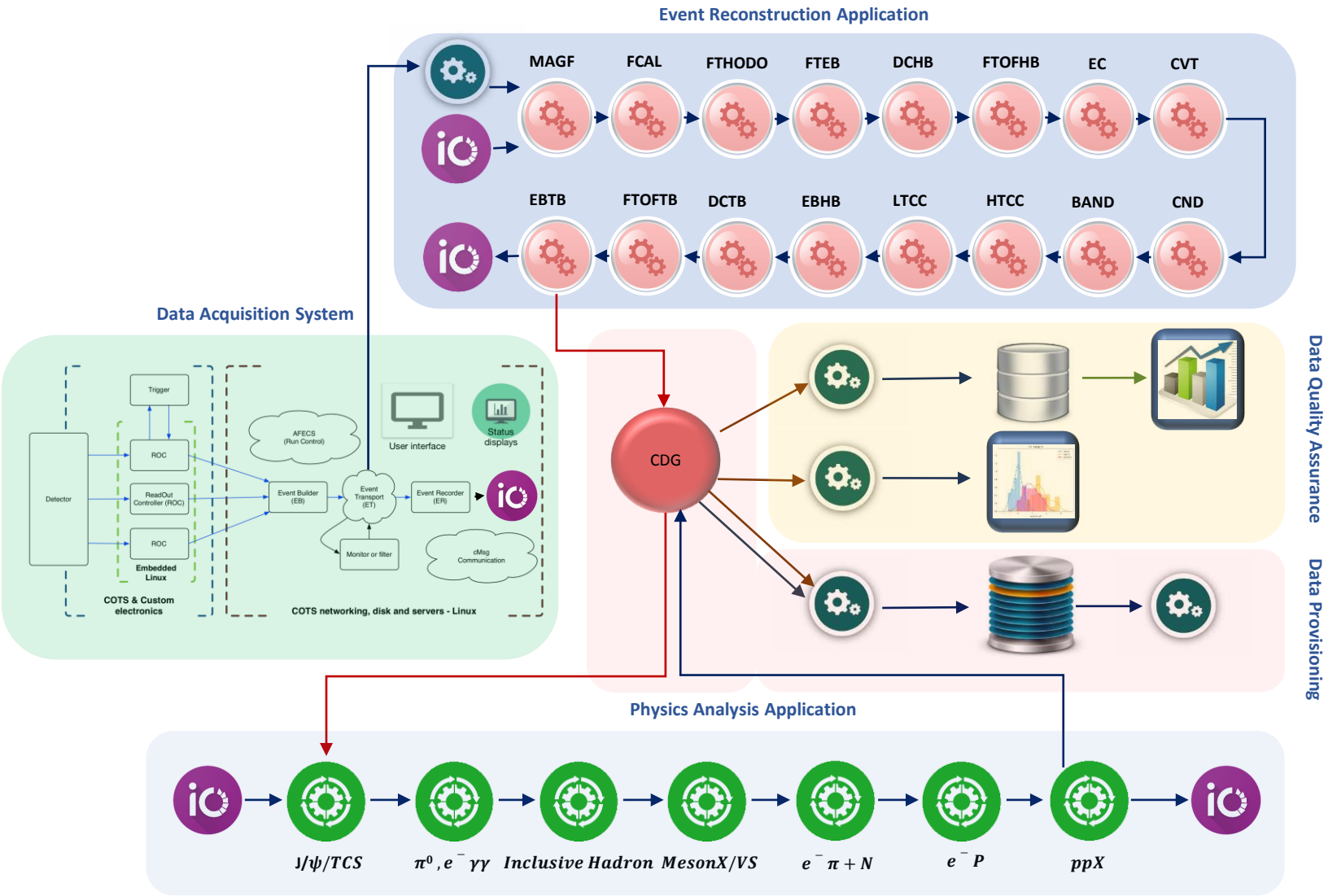
Jefferson Lab

Transient data unit (meta-data + data)



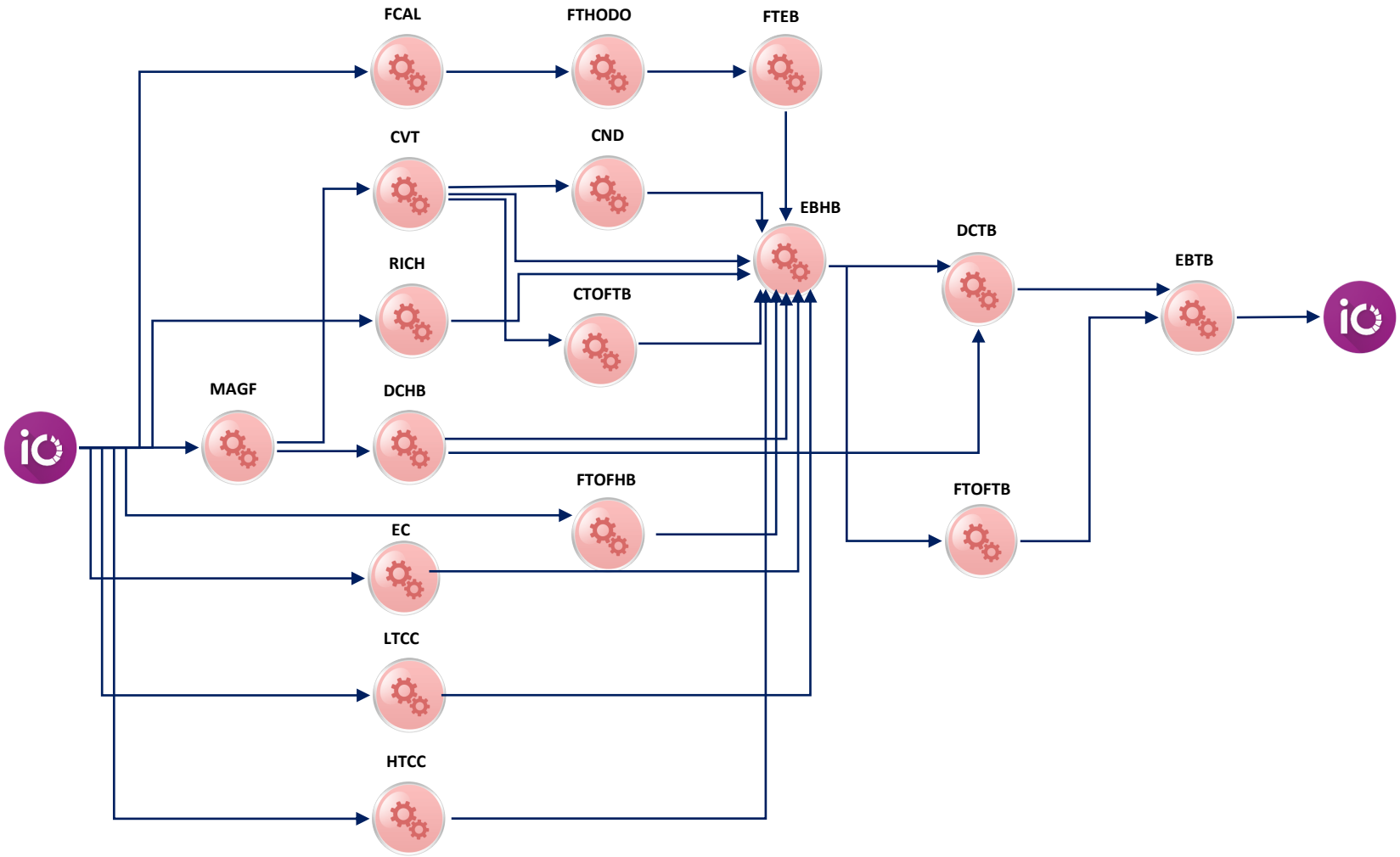
Jefferson Lab

CLAS12 Data Processing Applications



Jefferson Lab

Event Reconstruction Application (sub-event level parallelization)

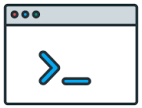


Jefferson Lab

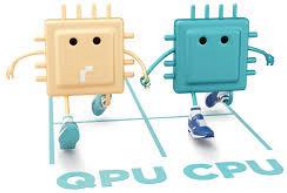
Workflow orchestrator

Application Monitoring,
Real-time Benchmarking

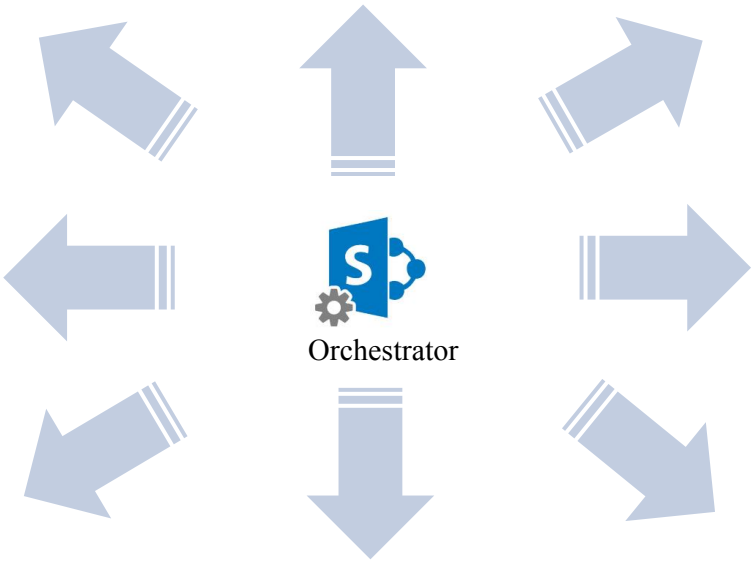
Command-Line Interface



Hardware Optimizations



Application Deployment
and Execution



Service Registration/Discovery

Exception Logging and
Reporting

Farm (batch or cloud) Interface

Data-Set Handling
and Distribution

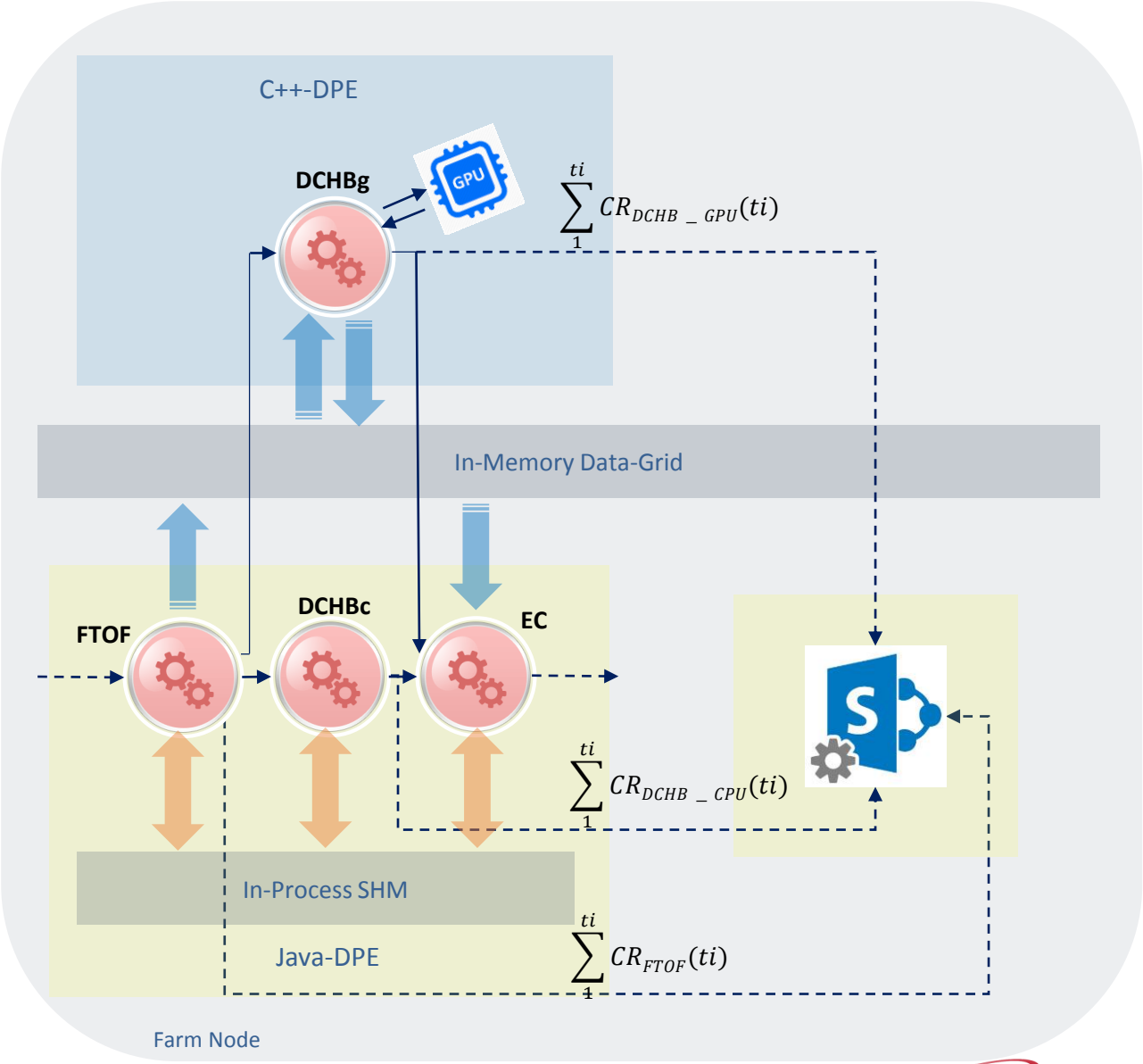
Jefferson Lab

Heterogeneous deployment algorithm

$$P_g = \frac{\sum_1^{ti} CR_{FTOF}(ti)}{\sum_1^{ti} CR_{DCHB\_GPU}(ti)}$$

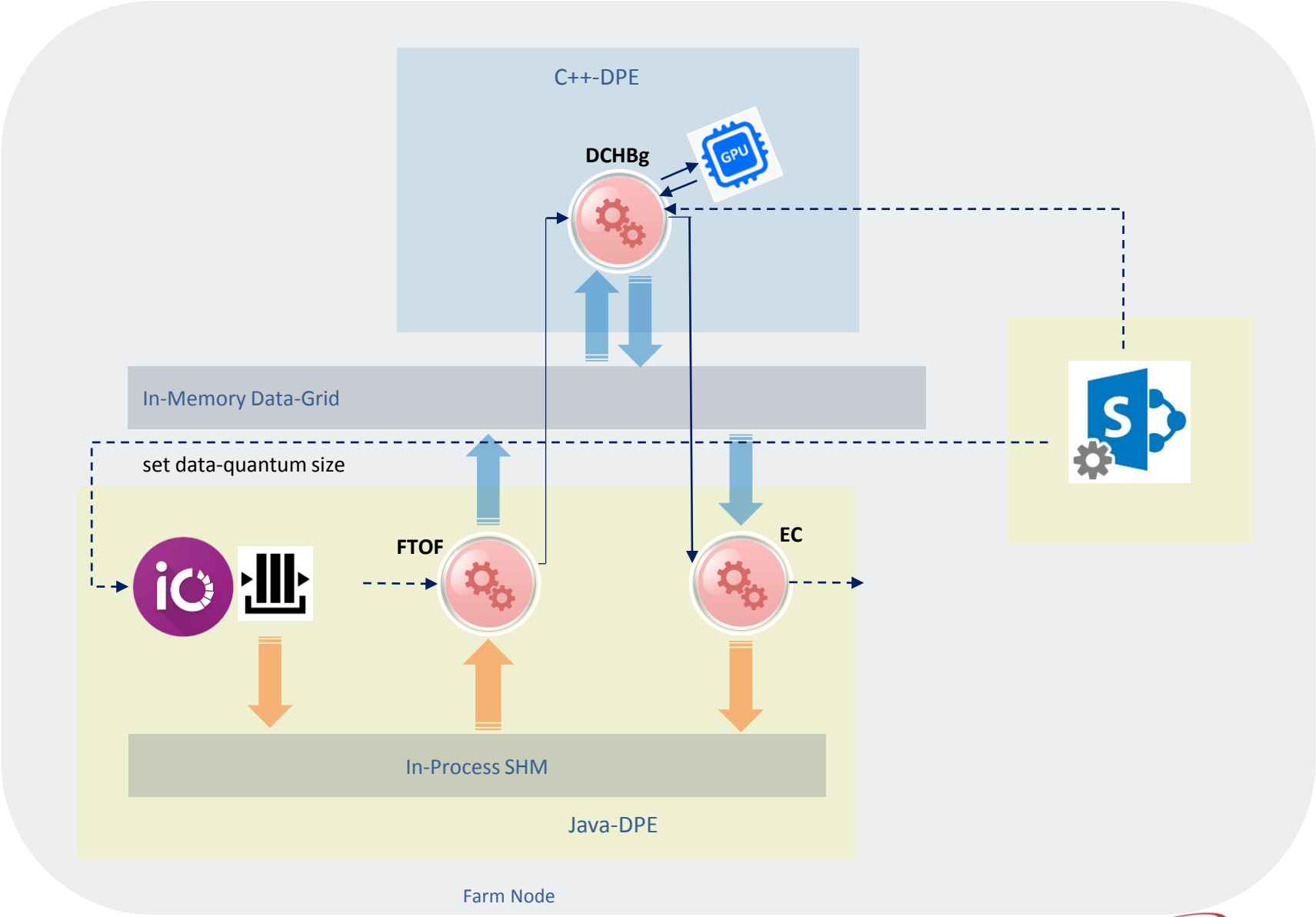
$$P_c = \frac{\sum_1^{ti} CR_{FTOF}(ti)}{\sum_1^{ti} CR_{DCHB\_CPU}(ti)}$$

if $Pg < Pc$
route data–stream through DCHBg



Jefferson Lab

Data-quantum size and GPU occupancy

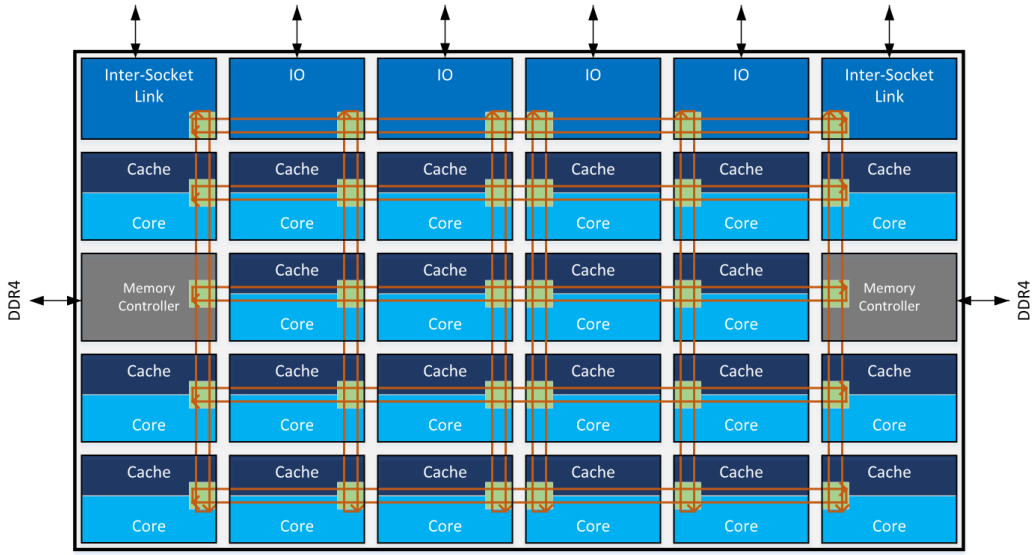


Jefferson Lab

Thread motion and DVFS

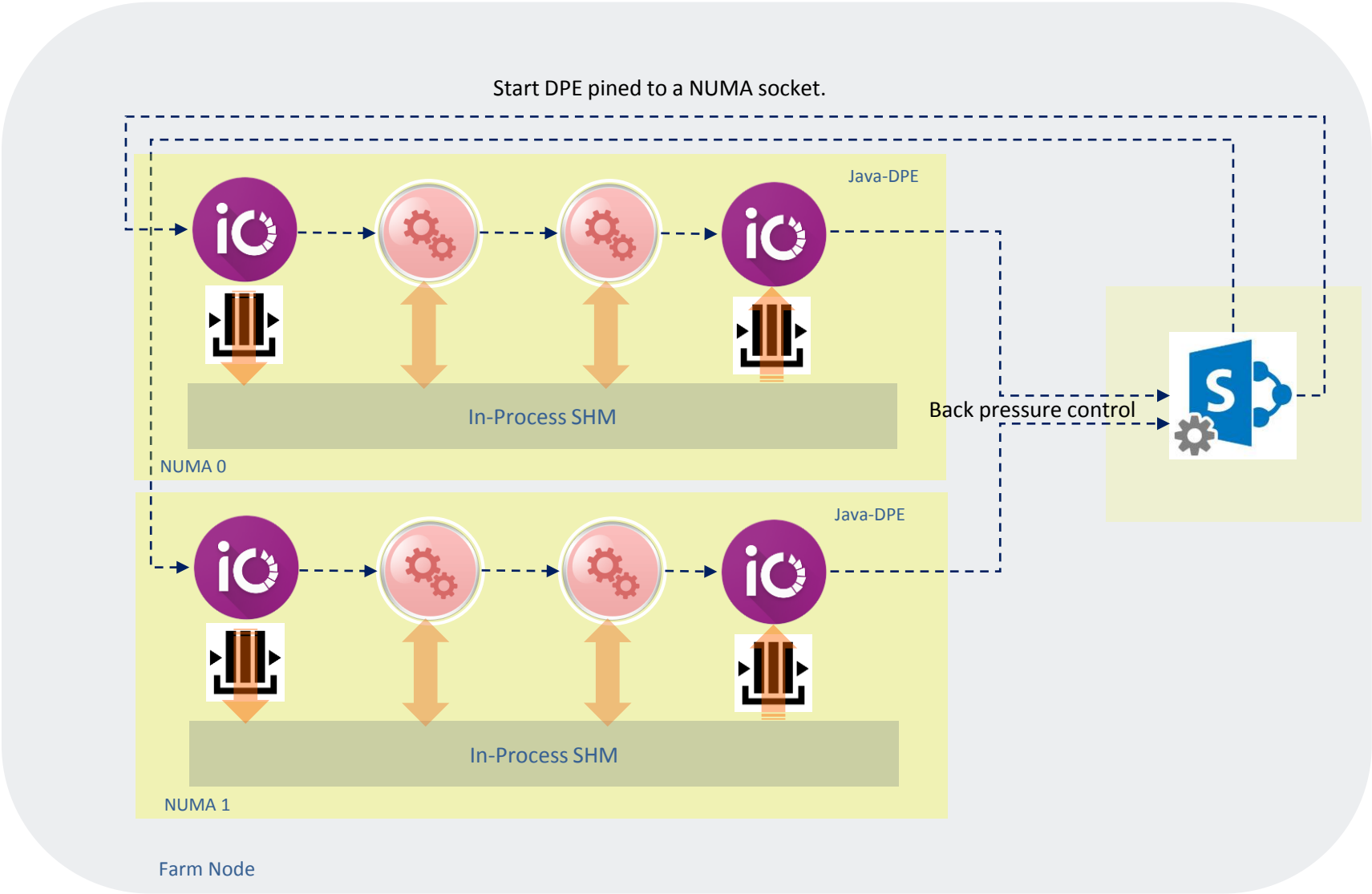
- Per-core, independent voltage control becomes impractical
- Limited number of independent DVFS systems for multicore systems
- Large core density systems are deploying a new power management technique that migrates threads from core to core to adjust power and performance to the time-varying needs of a running program.

Intel Mesh Interconnect Architecture



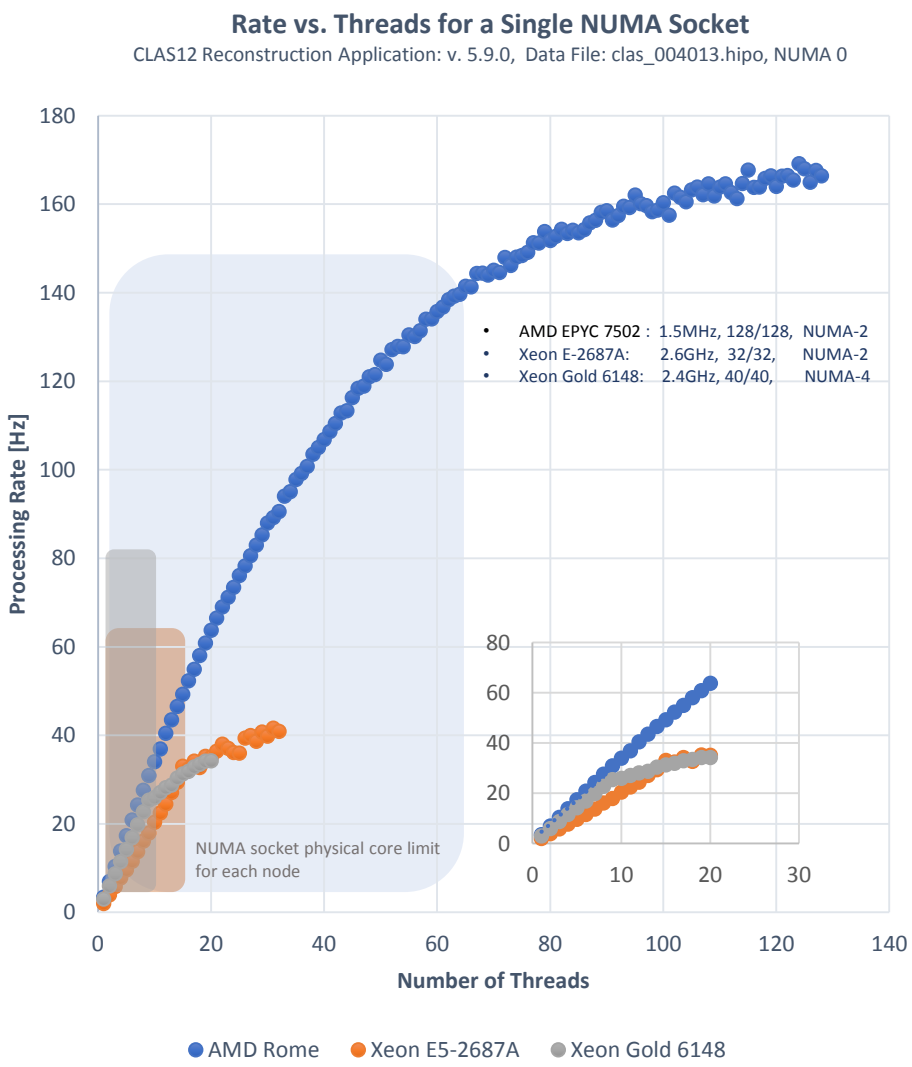
Jefferson Lab

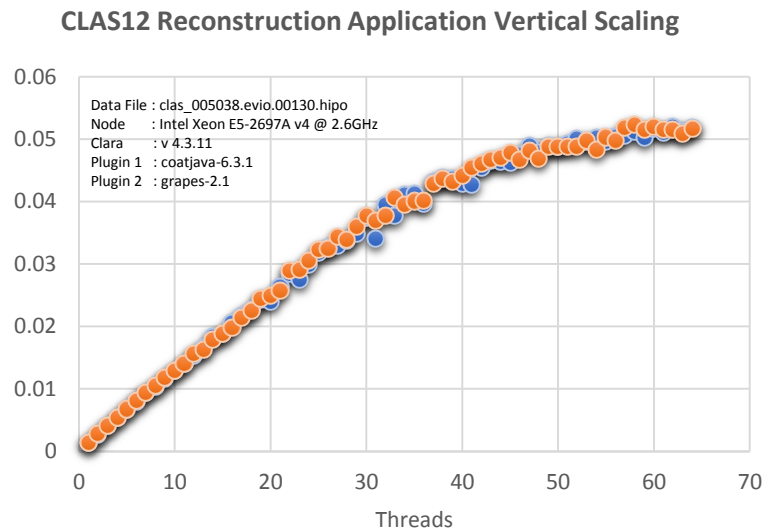
Data-processing chain per NUMA

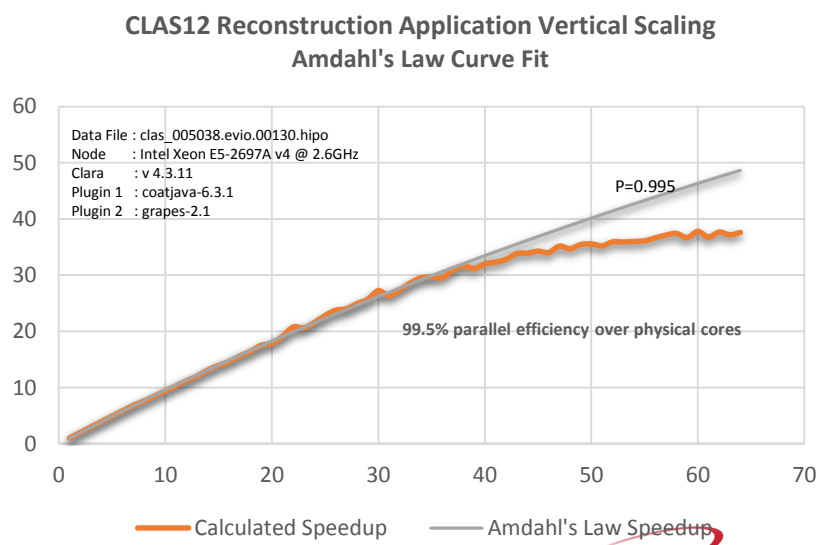


Jefferson Lab

Results







Jefferson Lab

Summary

- Frameworks based on the micro-services architecture are in a better position to address massive parallelism and heterogeneity of current and future computing facilities.
- CLARA is a mature, micro-services based, data stream processing framework in production-use at JLAB and NASA.
- Internal, stream-unit level workflow management system is designed with adaptive functionalities that guarantees maximum data processing performance across diverse hardware and software infrastructures.