CLARA: Data-Stream Processing Framework

Framework for streaming readout

V. Gyurjyan for JLAB EPSCI Group

gurjyan@jlab.org

Jefferson Lab

U.S. DEPARTMENT OF ENERGY | Office of Science

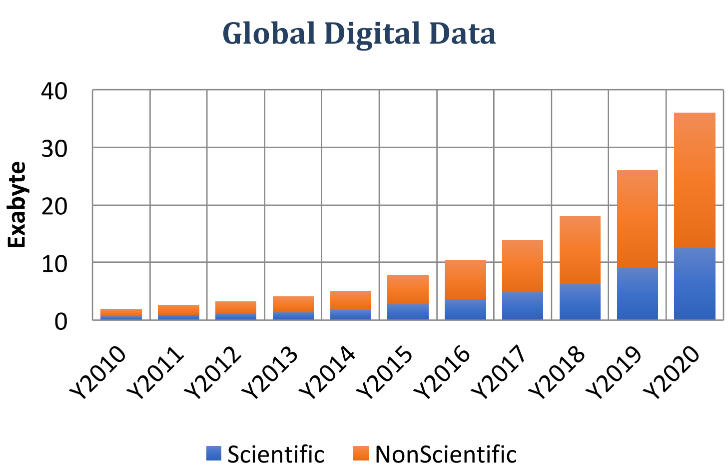
JSA

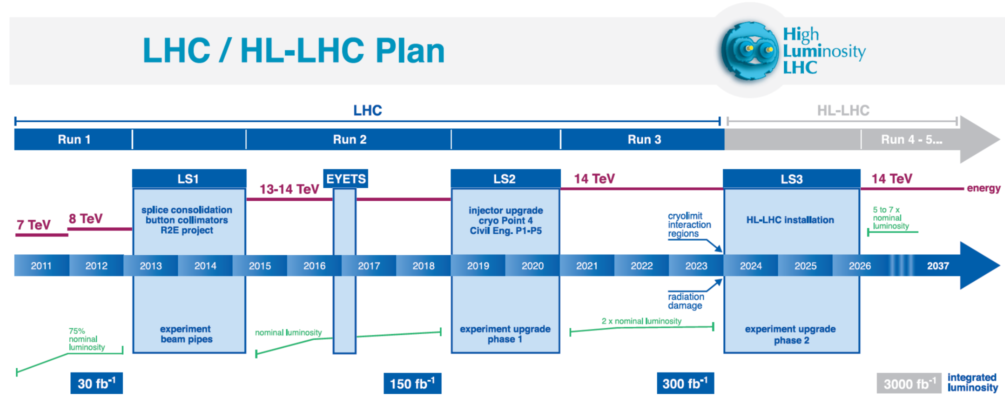
Outline

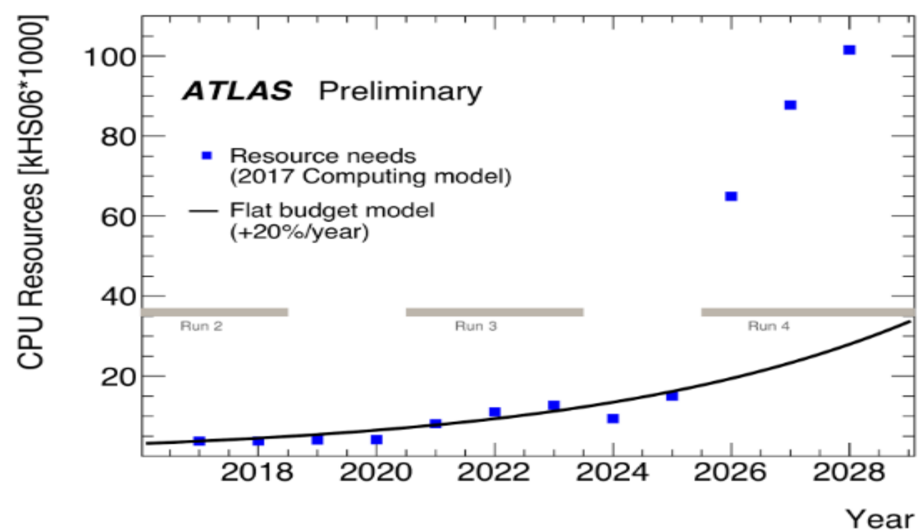
- Problem statement and possible solutions
  - Hardware diversification
  - Parallelization
  - Streaming
- Micro-services architecture
  - Micro-service vs monolith
- Flow based programming paradigm
  - Reactive communication
- CLARA: reactive data-stream processing framework that implements micro-services architecture and FBP
- Summary

Problem we face

Experiment	Conditions	Event Rate	Data Rate	Comments
Moller	Production/ integrated mode	1920Hz	130MB/s	Can be handled with the traditional DAQ.
EIC	$L=10^{34}$ $cm^{-2}s^{-1}$	450-550Hz Not included background noise rates.	20-25GB/s not included vertex tracker that will generate ~240GB/s	~10Kz/μb, track multiplicity = ~5 JLAB EIC detector design will have millions of channels. Only non-vertex detectors combined will have ~1M channels plus vertex detector: estimated 20-50M channels. In total ~1000 ROC's. Control nightmare (starting stopping a run). Streaming readout has less control requirements.
TIDIS		rTPC hit rates enormous (~800KHz/pad)	4GB/s	How to match up super Bigbyte detected electrons with rTPC detected spectator protons is a big question. Conventional triggered DAQ will be challenged.
SoLID	30 sector GEM		30GB/s	30 separate DAQ's each 1GB/s? How to combine GEM readout with other detectors? Handling GEM hits sharing adjacent sectors.
CLAS12	Phase 2	100KHz	5-7GB/s	

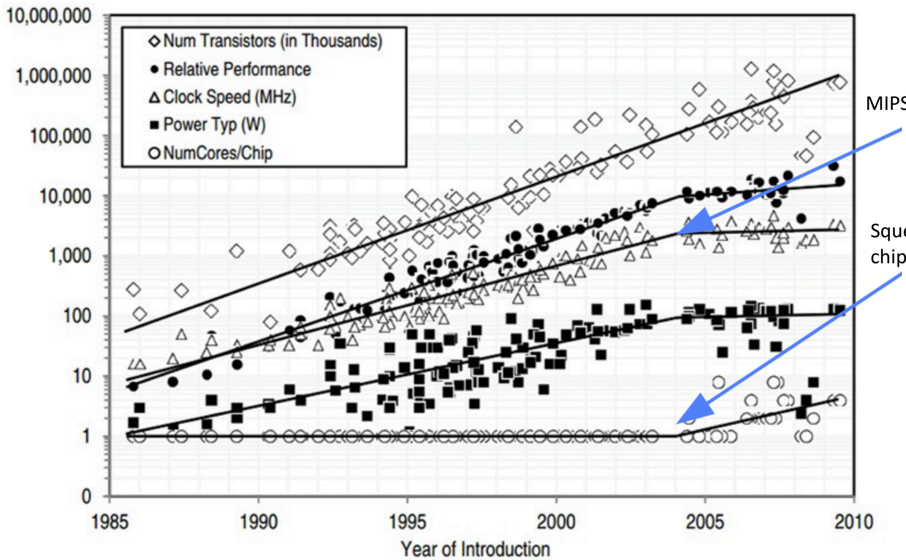






Jefferson Lab

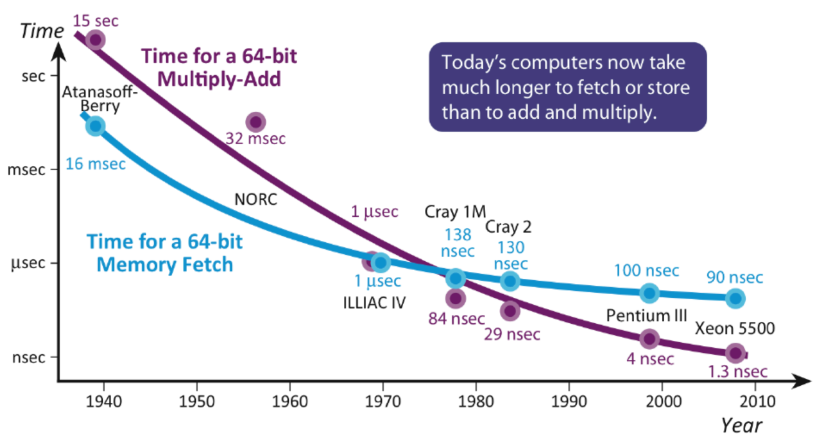
CPU based architecture limitations

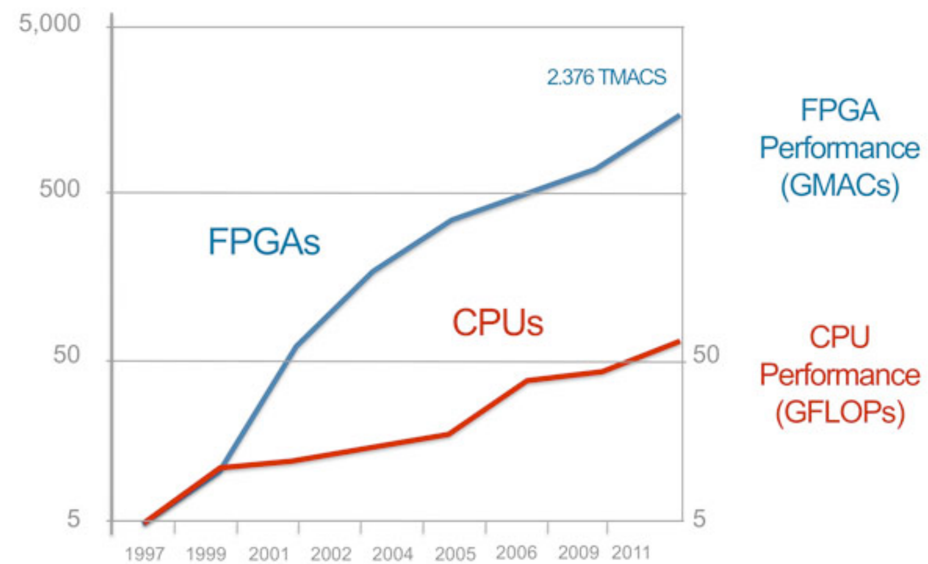


A Roadmap for HEP Software and Computing R&D for the 2020s. HEP Software Foundation, Feb. 2018

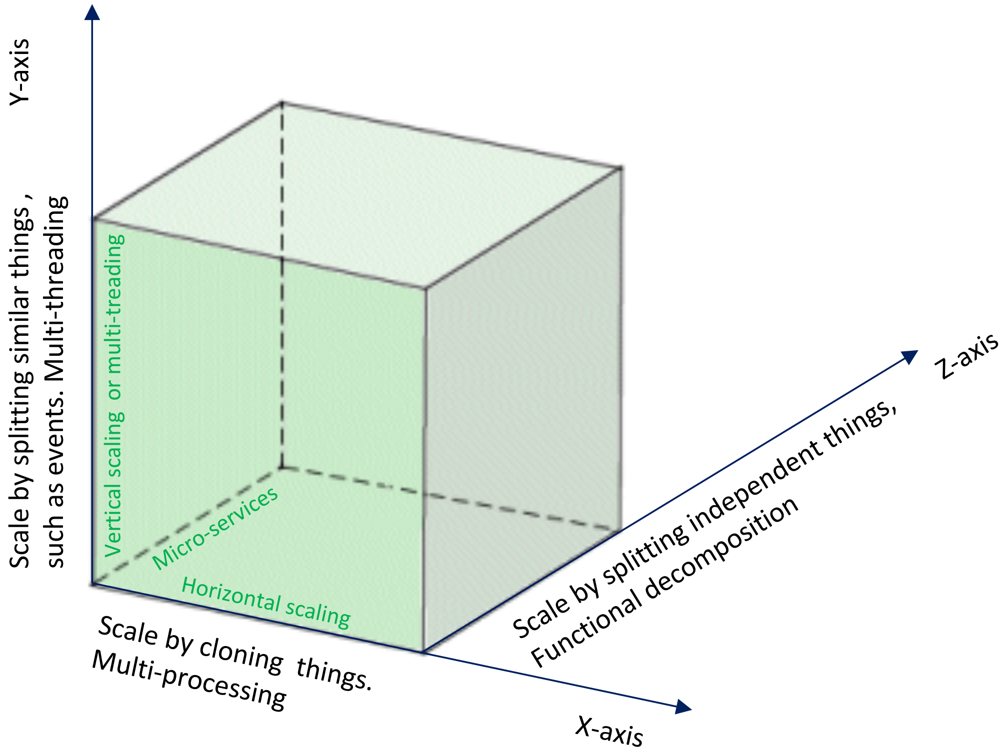
"Frameworks face the challenge of handling the massive parallelism and heterogeneity that will be present in future computing facilities, including multi-core and many-core systems, GPUs, Tensor Processing Units (TPUs), and tiered memory systems, each integrated with storage and high-speed network interconnections."

Memory Latency





The Scale-Cube



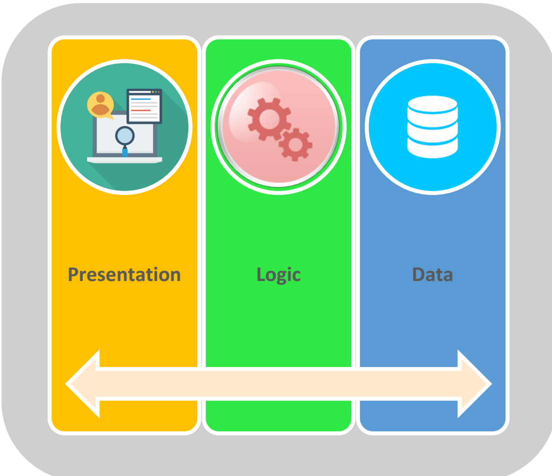
The Art of Scalability. by Martin L. Abbott and Michael T. Fisher. **ISBN-13:** 978-0134032801

Why decomposition into independent modules

- Smaller and independent code bases. Reinforce a maximum independence and isolation of functional components.
- Fault tolerant
- Overall micro-services based application evolves much faster
- No other dependencies other than data (loose coupling) can run on heterogeneous hardware and software infrastructures.
- Relatively easy evolution, due to
    - Requirement changes
    - Environment changes
    - Errors or security breaches
    - New equipment added or removed
    - Improvements to the system
- Encourages contribution and inclusion of new technologies

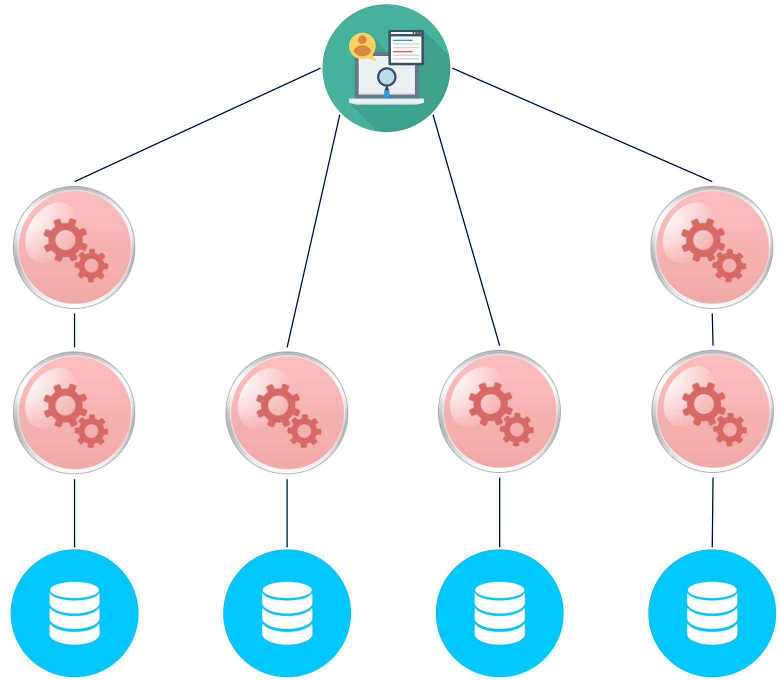
Jefferson Lab

Micro-services vs Monolithic architecture



Pros
- Strong coupling, network independent
- 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

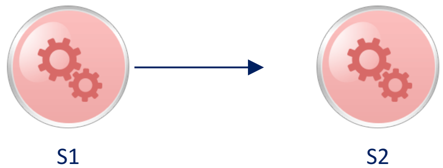


Pros
- Technology independent
- Fast iterations
- Small teams
- Fault isolation
- Scalable

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

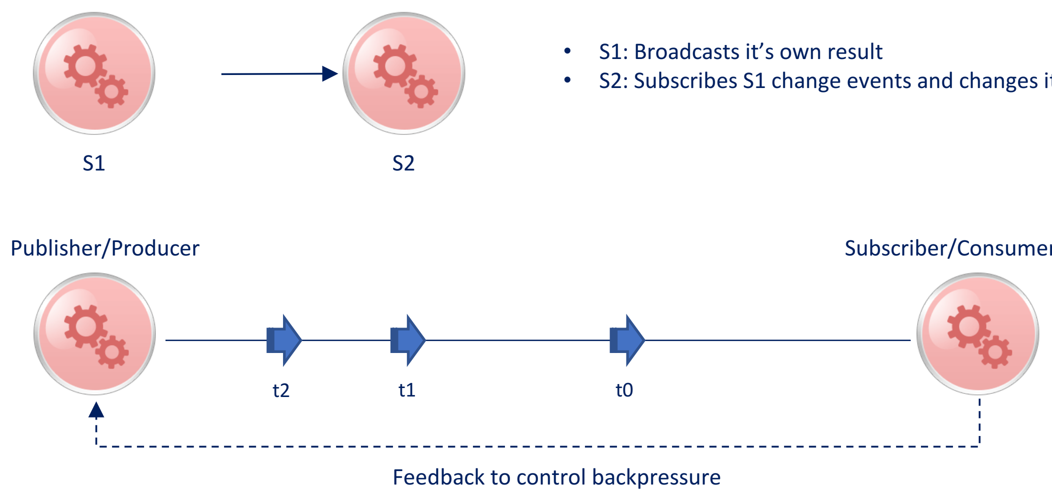
FBP paradigm and reactive programming

Flow based programming paradigm assumes reactive programming



- 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

CLARA Framework

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

- 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
- Defines streaming transient-data structure
- Supports C++, JAVA, Python languages

Authors and chronology

- V. Gyurjyan, S. Mancilla, R. Oyarzun, S. Paul, A. Rodrigues
- Design: 2009
- Betta release and first application: 2010
- 3 master theses

Publications

- *CLARA: A Contemporary Approach to Physics Data Processing*, 2011, J. Phys.: Conf. Ser. 331 032013 doi:10.1088/1742-6596/331/3/032013
- *Development of A Clara Service for Neutron Reconstruction*, 2011, APS: 2011APS..DNP.EA024C
- *Component Based Dataflow Processing Framework*, 2015, IEEE DOI: 10.1109/BigData.2015.7363971, ISBN: 978 1-4799-9926-2
- *Earth Science Data Fusion with Event Building Approach*, 2015, IEEE DOI: 10.1109/BigData.2015.7363972, ISBN: 978 1-4799-9926-2
- *CLARA: The CLAS12 Reconstruction and Analysis framework*, 2016, J. Phys.: Conf. Ser. 762 012009 doi:10.1088/1742-6596/762/1/012009

Rewards

Research Opportunities in Space and Earth Sciences (ROSES) 2015

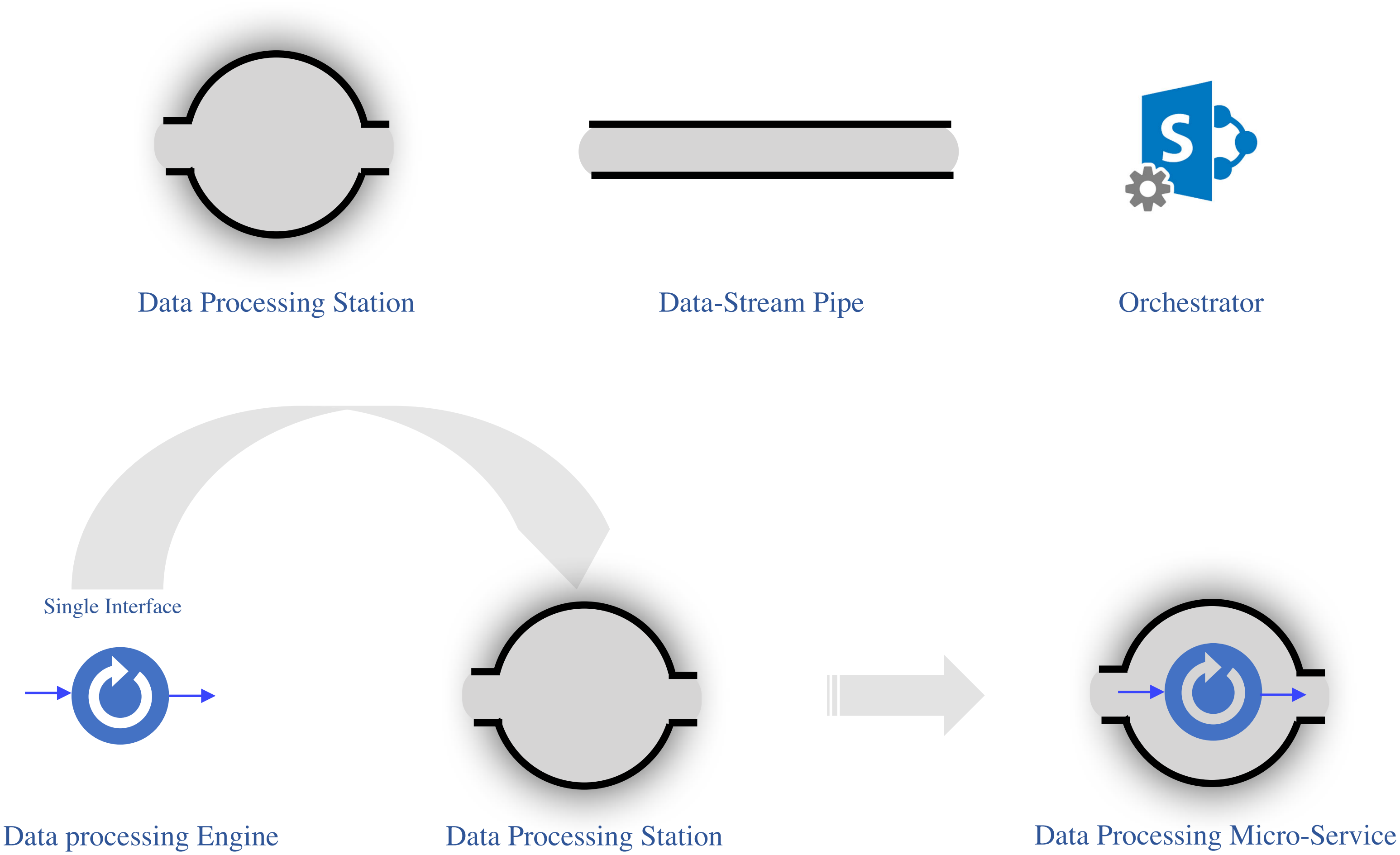
Funding for 3 years by the NASA's Earth Science Technology Office (ESTO) and the Advanced Information Systems Technology (AIST) Program.
NAIADS Project ID: AIST-14-0014.
SRB Project ID:LARC-14-0014-2

Users

Documentation

http://claraweb.jlab.org

Basic components and a user code interface



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

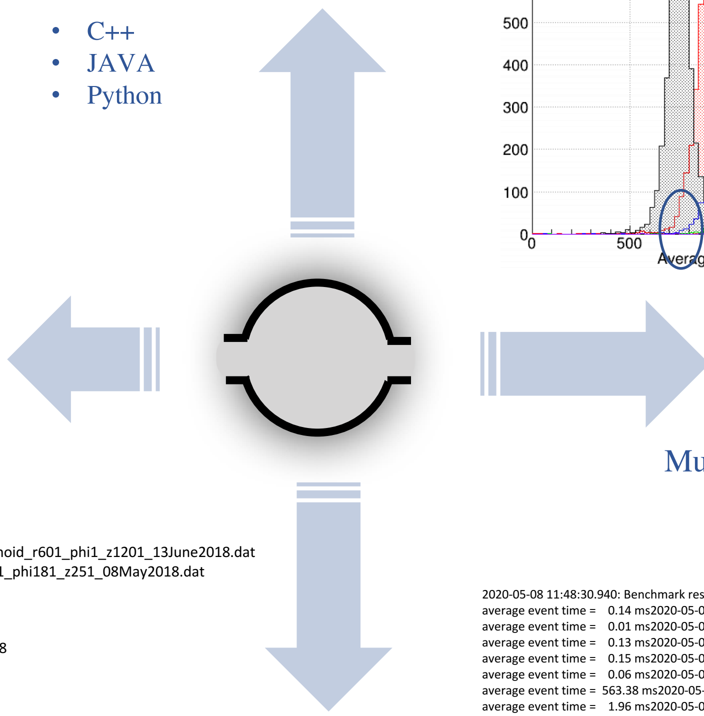
- C++
- JAVA
- Python

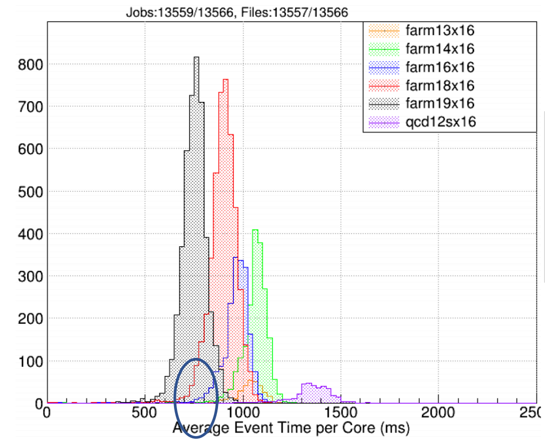
Configuration

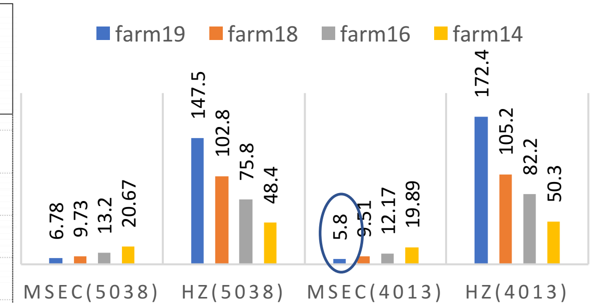
Multi-threading

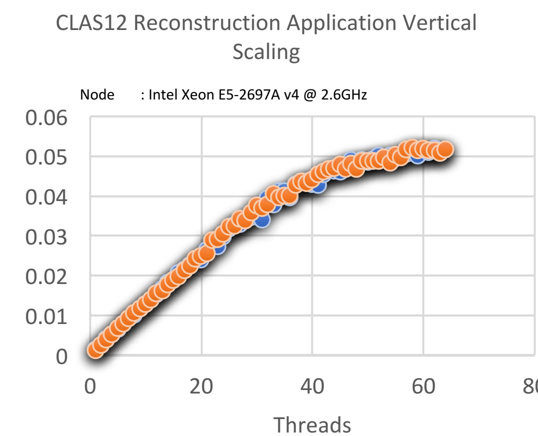
Communication

- 0MQ
- POSIX-SHM
- In-memory Data Grid (IDG)

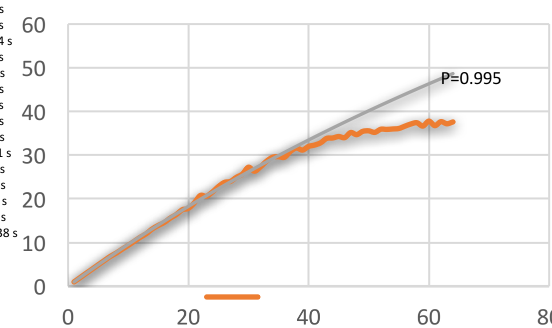








Amdahl's Law Curve Fit

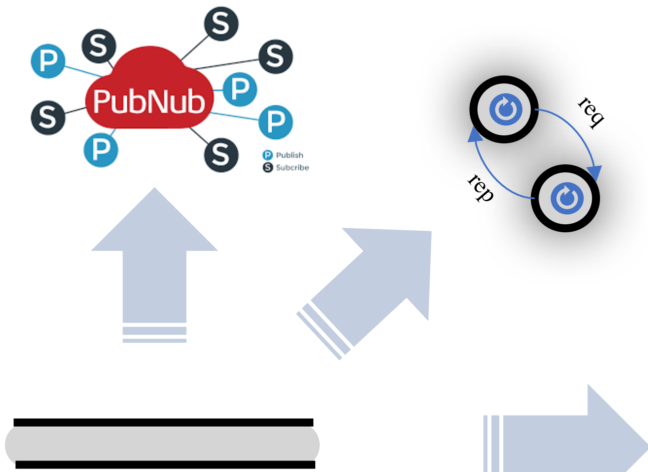



```
configuration:
  io-services:
    writer:
      compression: 2
  services:
     MAGFIELDS:
       magfieldSolenoidMap: Symm_solenoid_r601_phi1_z1201_13June2018.dat
       magfieldTorusMap: Full_torus_r251_phi181_z251_08May2018.dat
       variation: rga_fall2018
     DCHB:
        variation: rga_fall2018
         dcGeometryVariation: rga_fall2018
         dcT2DFunc: "Polynomial"
      DCTB:
         variation: rga_fall2018
         dcGeometryVariation: rga_fall2018
      EC:
          variation: rga_fall2018
mime-types:  - binary/data-hipo
```

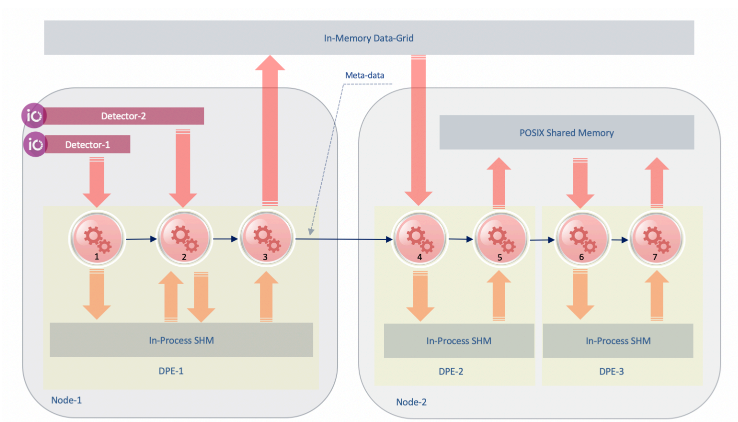
```
2020-05-08 11:48:30.940: Benchmark results:2020-05-08 11:48:30.941:
average event time =   0.14 ms2020-05-08 11:48:30.943:  MAGFIELDS   2000 events   total time =    0.02 s
average event time =   0.01 ms2020-05-08 11:48:30.945:  FTCAL       2000 events   total time =    0.26 s
average event time =   0.13 ms2020-05-08 11:48:30.946:  FTHODO      2000 events   total time =    0.29 s
average event time =   0.15 ms2020-05-08 11:48:30.948:  FTEB        2000 events   total time =    0.13 s
average event time =   0.06 ms2020-05-08 11:48:30.949:  DCHB        2000 events   total time = 1126.76 s
average event time = 563.38 ms2020-05-08 11:48:30.951:  FTOFHB      2000 events   total time =    3.93 s
average event time =   1.96 ms2020-05-08 11:48:30.952:  EC          2000 events   total time =    1.87 s
average event time =   0.94 ms2020-05-08 11:48:30.953:  CVT         2000 events   total time =  150.14 s
average event time =  75.07 ms2020-05-08 11:48:30.955:  CTOF        2000 events   total time =    4.75 s
average event time =   2.37 ms2020-05-08 11:48:30.956:  CND         2000 events   total time =    1.49 s
average event time =   0.74 ms2020-05-08 11:48:30.957:  BAND        2000 events   total time =    0.02 s
average event time =   0.01 ms2020-05-08 11:48:30.959:  HTCC        2000 events   total time =    0.11 s
average event time =   0.05 ms2020-05-08 11:48:30.960:  LTCC        2000 events   total time =    0.05 s
average event time =   0.03 ms2020-05-08 11:48:30.961:  EBHB        2000 events   total time =    1.60 s
average event time =   0.80 ms2020-05-08 11:48:30.963:  DCTB        2000 events   total time =  988.61 s
average event time = 494.30 ms2020-05-08 11:48:30.964:  FTOFTB      2000 events   total time =    3.98 s
average event time =   1.99 ms2020-05-08 11:48:30.965:  EBTB        2000 events   total time =    2.86 s
average event time =   1.43 ms2020-05-08 11:48:30.966:  RICH        2000 events   total time =    2.15 s
average event time =   1.07 ms2020-05-08 11:48:30.967:  WRITER      2000 events   total time =    7.09 s
average event time =   3.55 ms2020-05-08 11:48:30.968:  TOTAL       2000 events   total time = 2296.38 s
average event time = 1148.19 ms
```

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

Data Stream Pipe

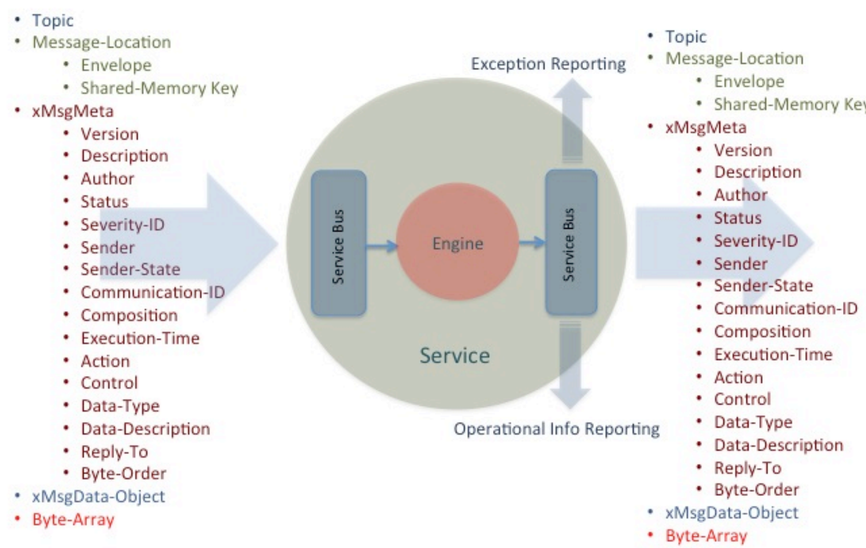


OMQ / POSIX_SHM / IDG



Transient Stream Unit
Google ProtoBuf

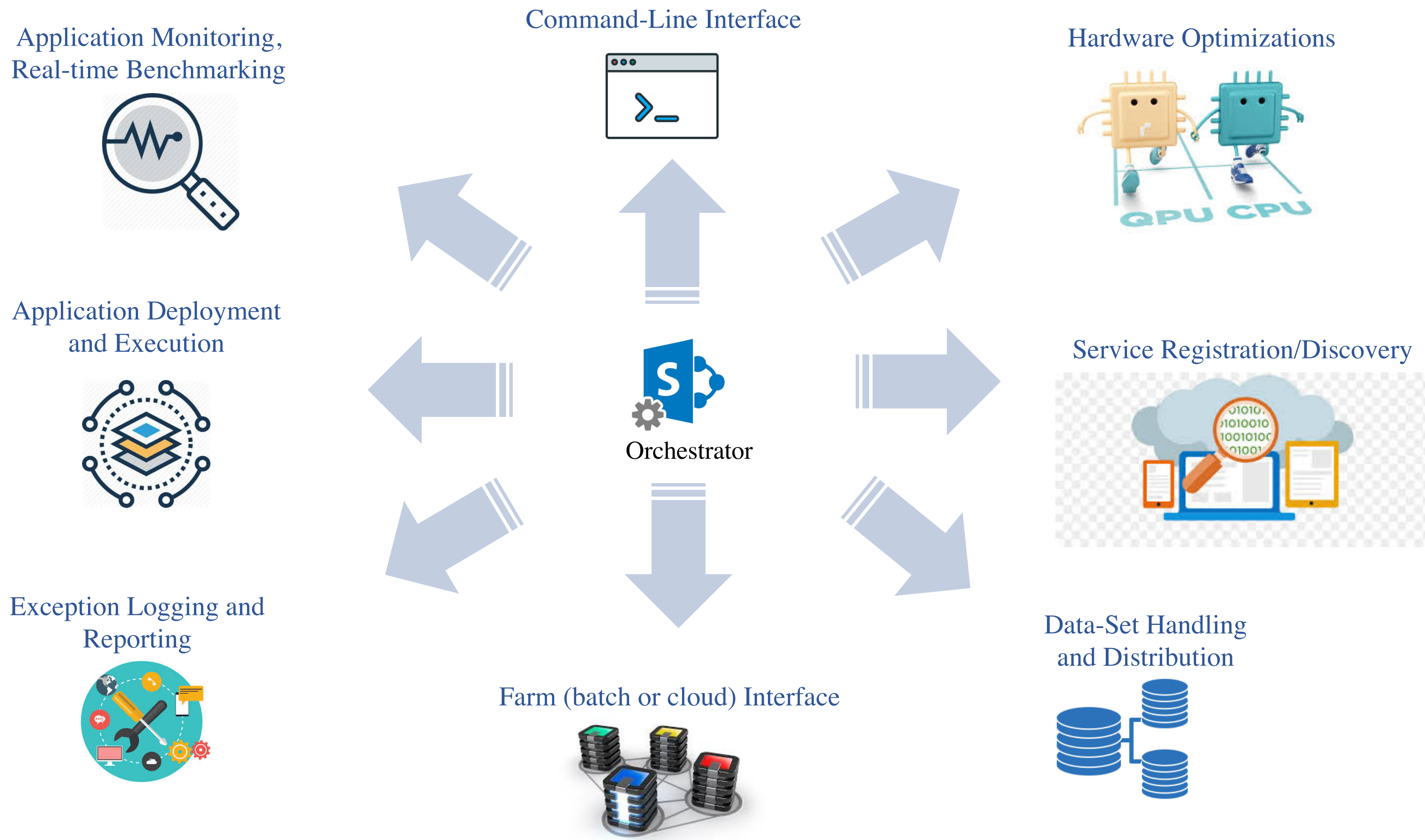
- Meta-data
- Serialization
- Encryption



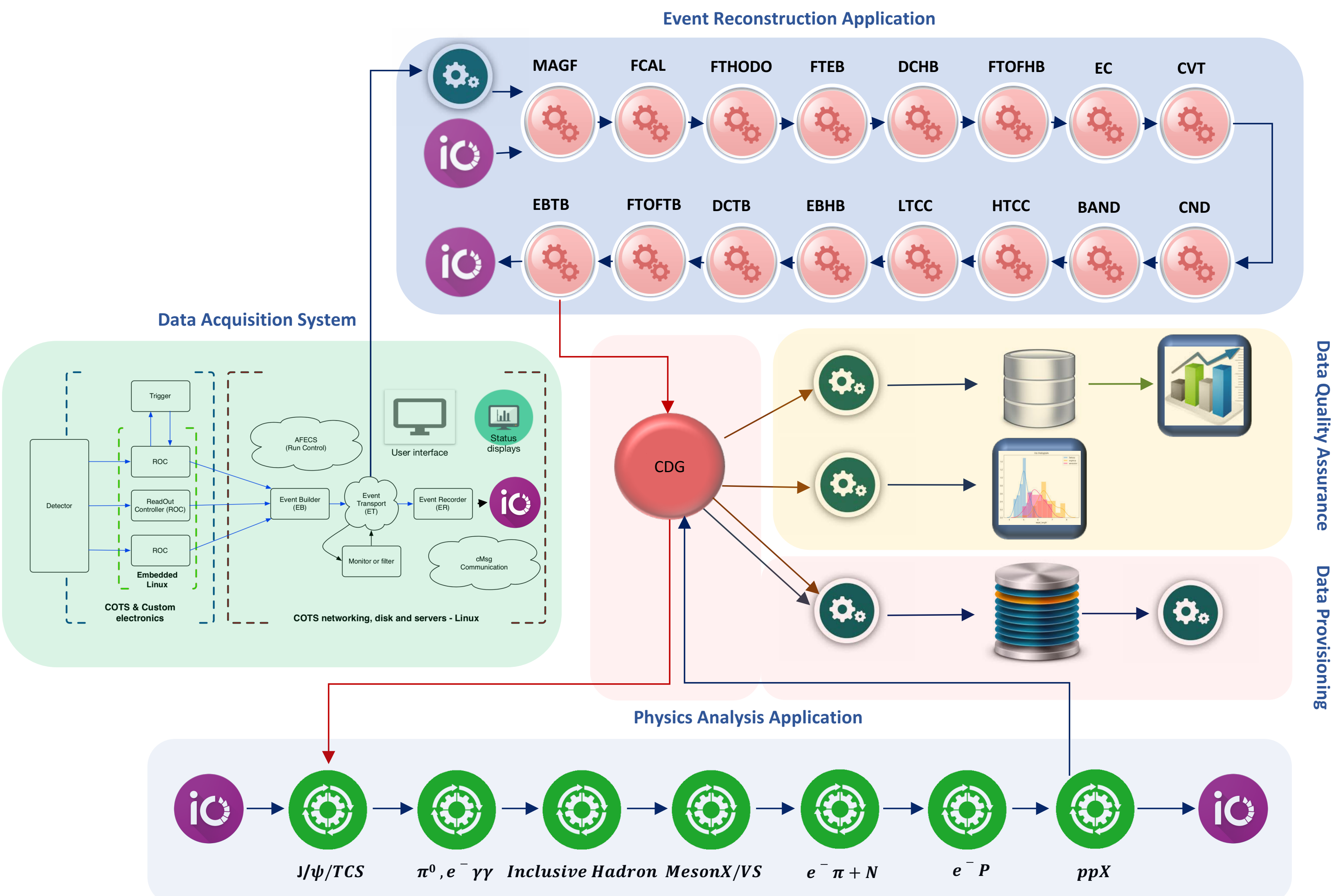
Language Bindings

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

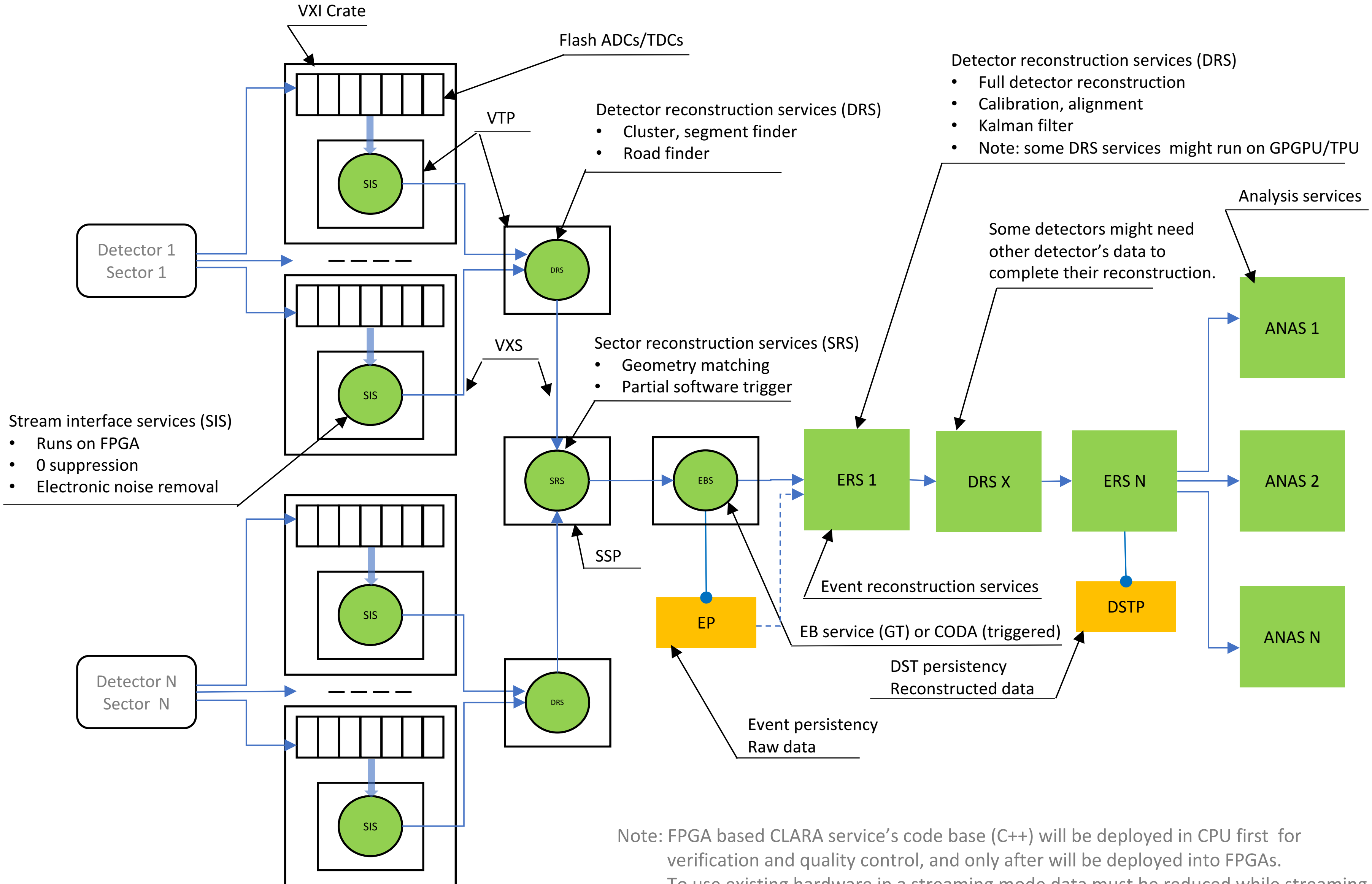
Workflow orchestrator



CLAS12 Data Processing Applications

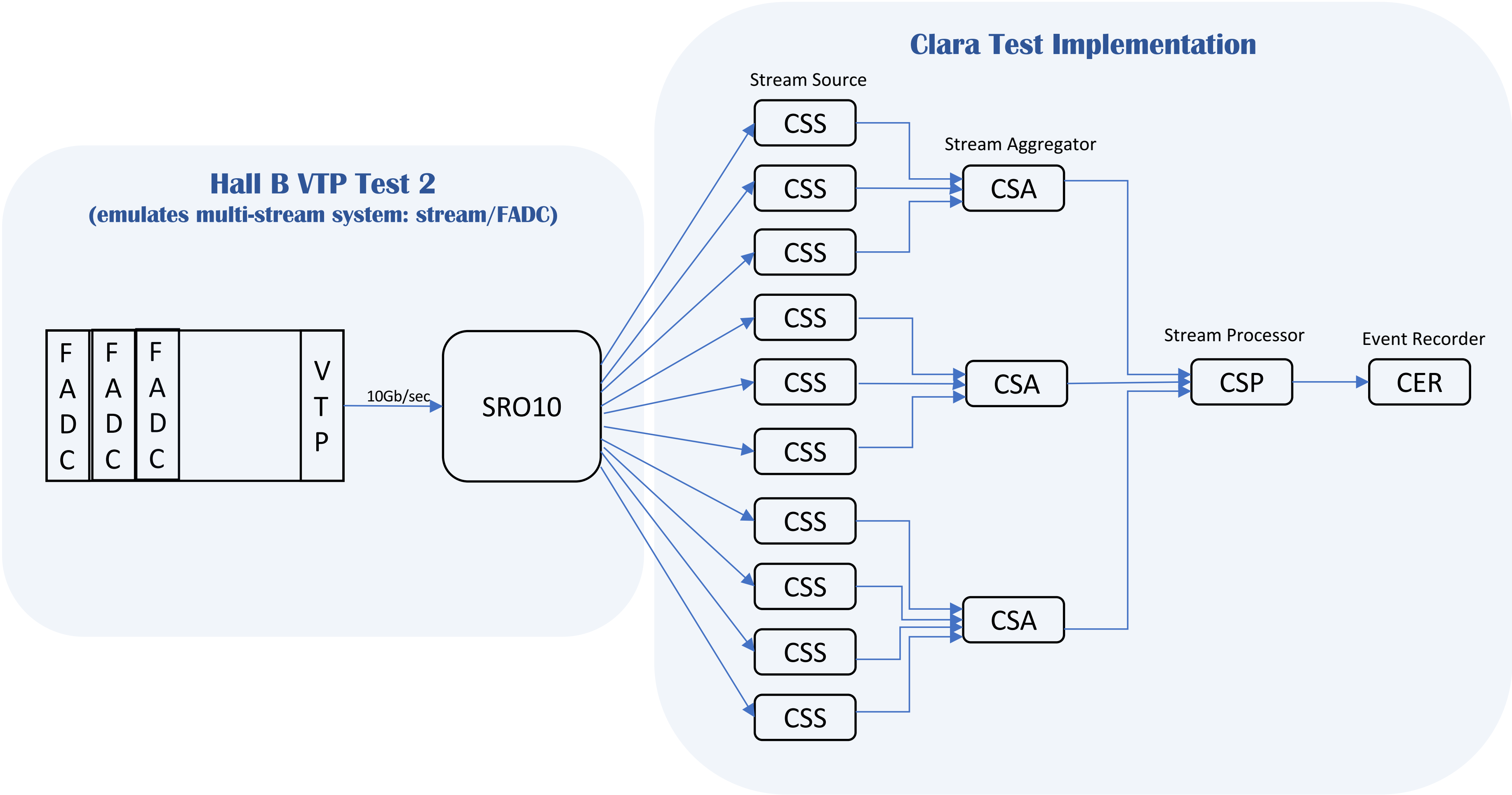


Heterogeneous data-stream processing (LDRD-2018)



Note: FPGA based CLARA service's code base (C++) will be deployed in CPU first for verification and quality control, and only after will be deployed into FPGAs.
To use existing hardware in a streaming mode data must be reduced while streaming.

Hall-B VTP Test-setup 2



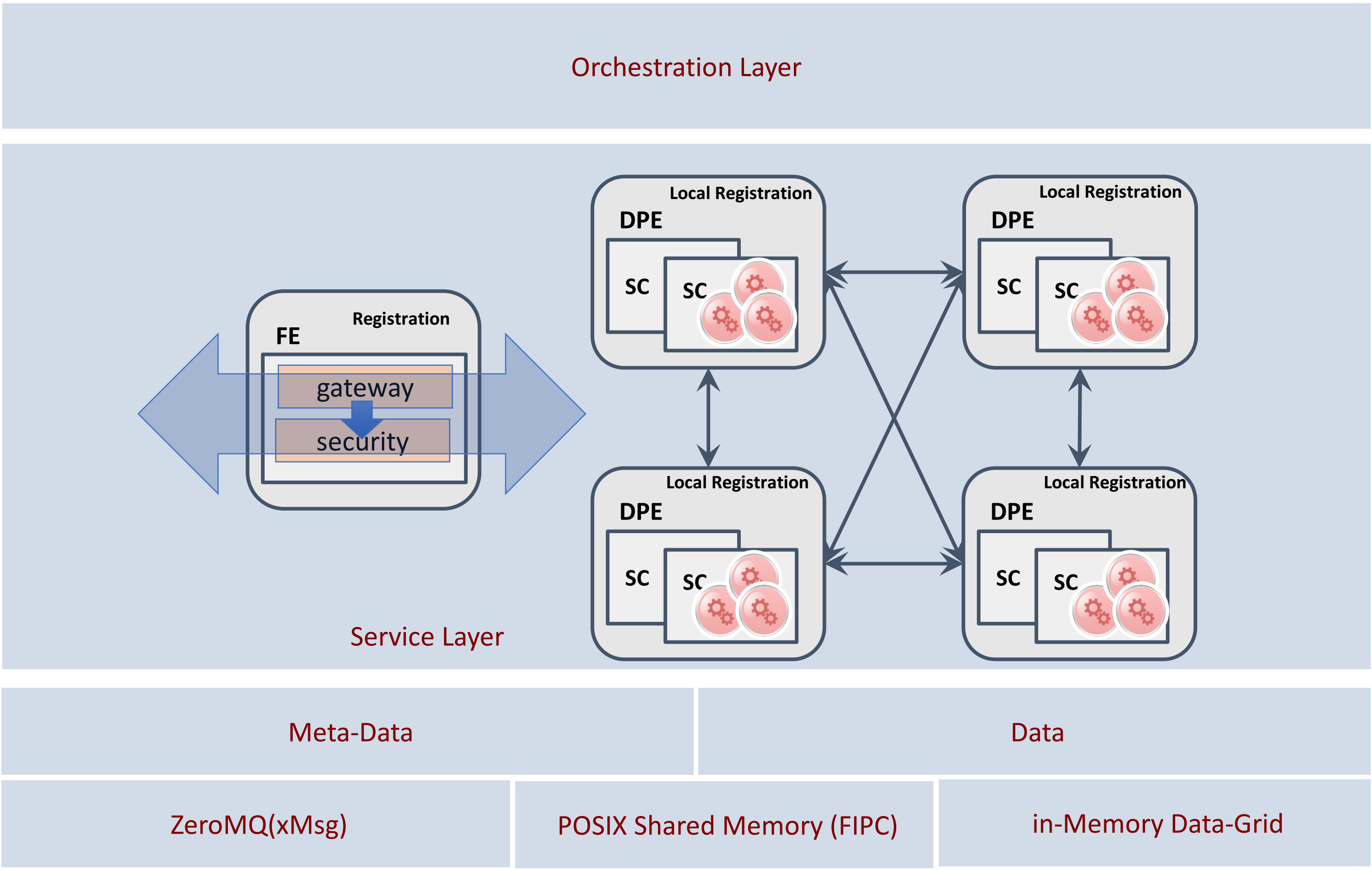
Summary

- To address scientific data 3V expansion we need to design frameworks capable of leveraging data streams, as well as massive parallelism and heterogeneity of feature computing facilities.
- CLARA is a mature data stream processing framework that utilizes micro-services architecture and flow-based programming paradigm, currently in production-use at JLAB and NASA Langley.
- CLARA together with JANA are being tested on the Hall-B SRO test-setup 2 for evaluation, and setting up a foundation for an integrated data processing framework for future experiments at home and elsewhere.

Thank you

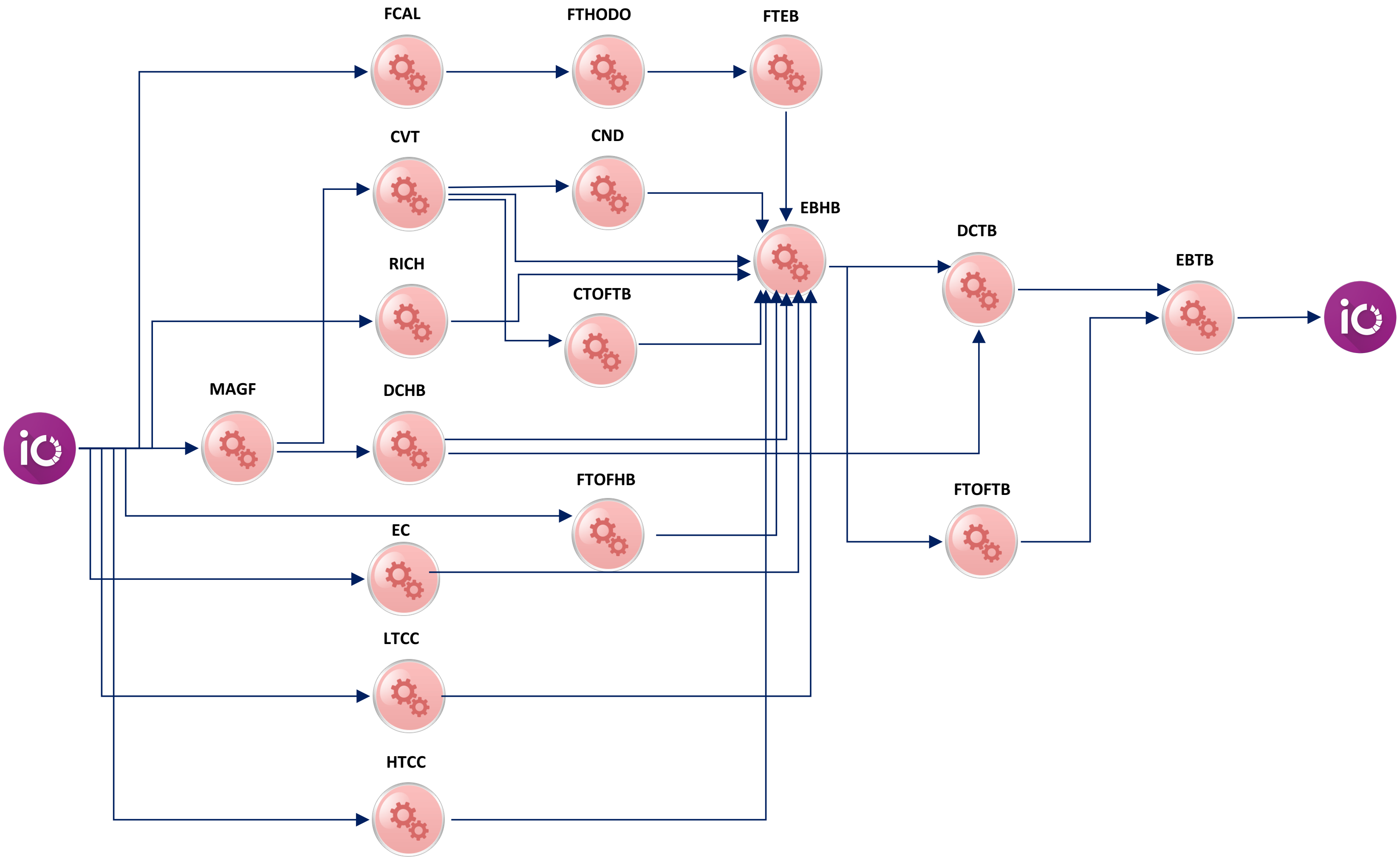
Backups

Structure



Orchestration Layer
Registration
FE
gateway
security
Local Registration
DPE
SC
SC
Local Registration
DPE
SC
SC
Local Registration
DPE
SC
SC
Local Registration
DPE
SC
SC
Service Layer
Meta-Data
Data
ZeroMQ(xMsg)
POSIX Shared Memory (FIPC)
in-Memory Data-Grid

Event Reconstruction Application (sub-event level parallelization)

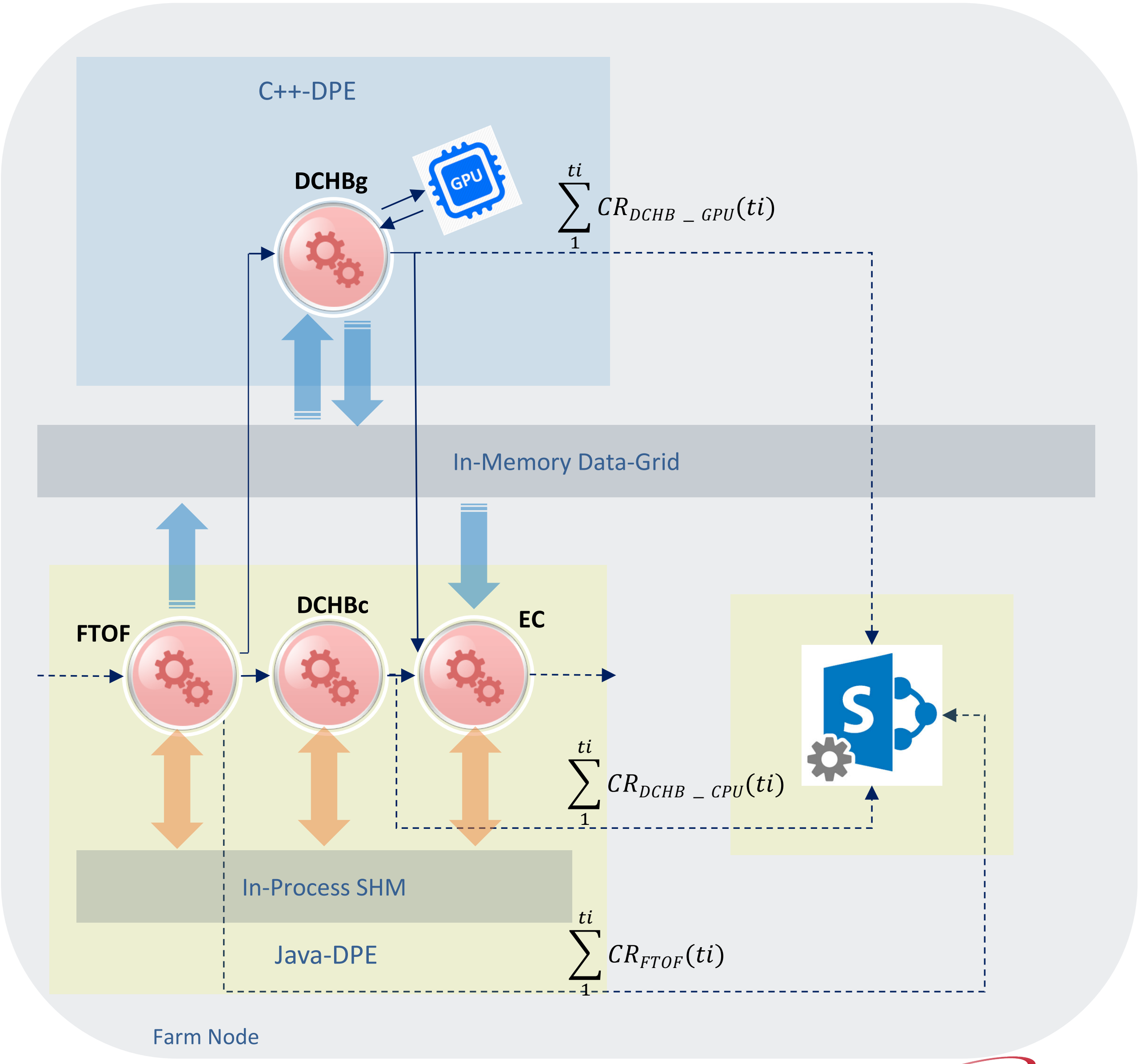


Heterogeneous deployment algorithm

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

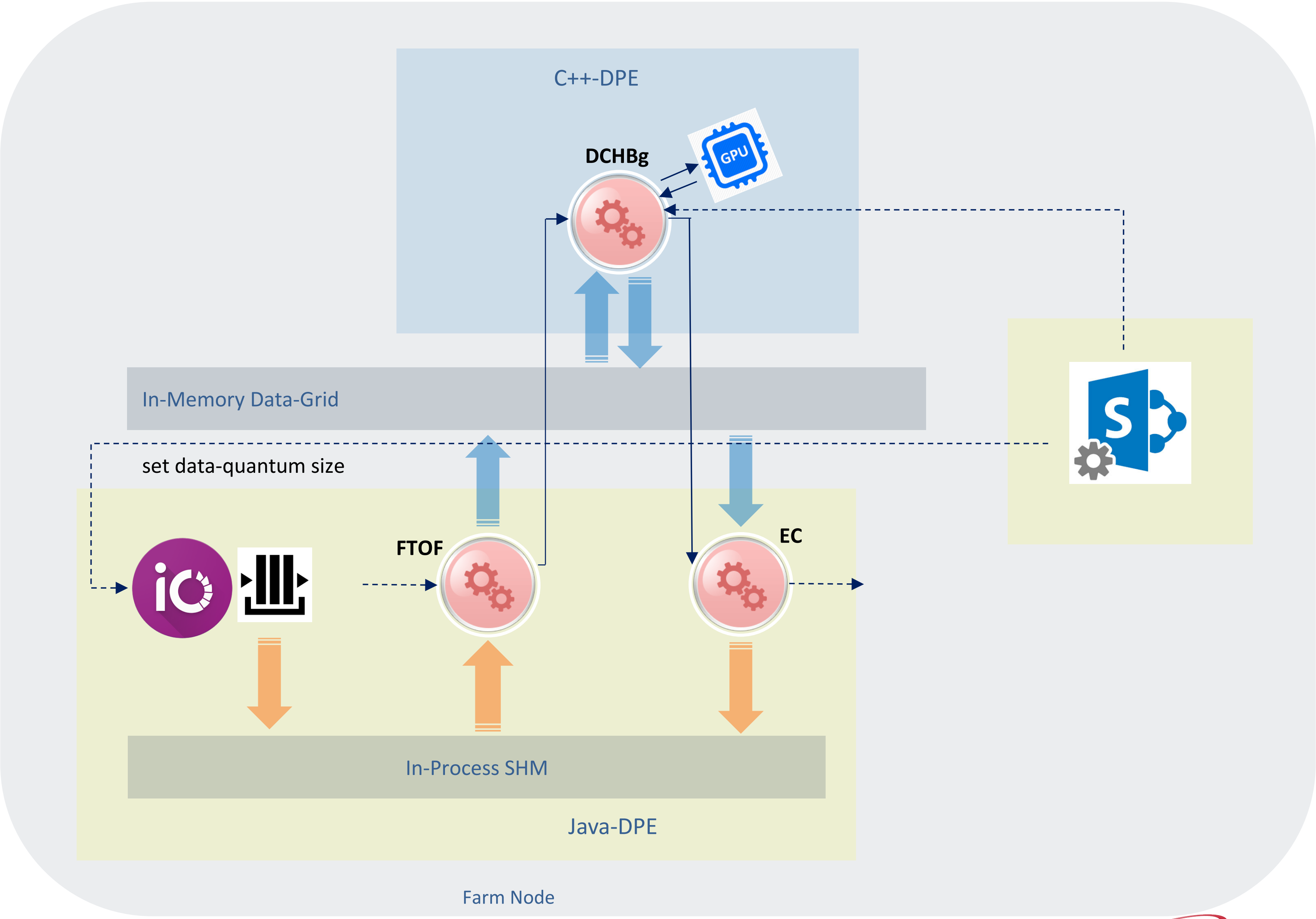
$$P_c = \frac{\sum_1^{ti} CR_{FTOF}(ti)}{\sum_1^{ti} CR_{DCHB\_CPU}(ti)}$$

$if\ Pg < Pc$
$route\ data\text{-}stream\ through\ DCHBg$

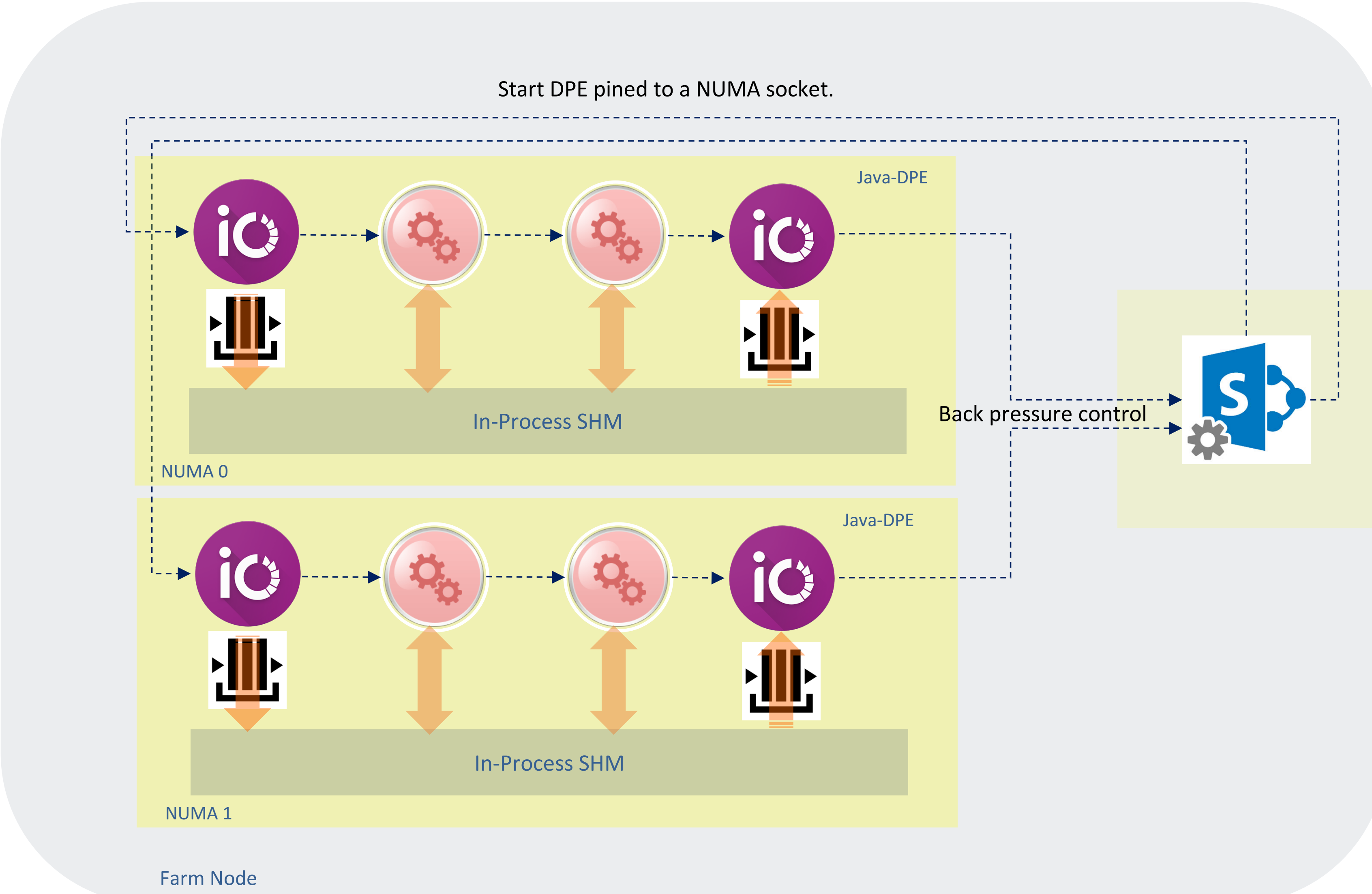


Jefferson Lab

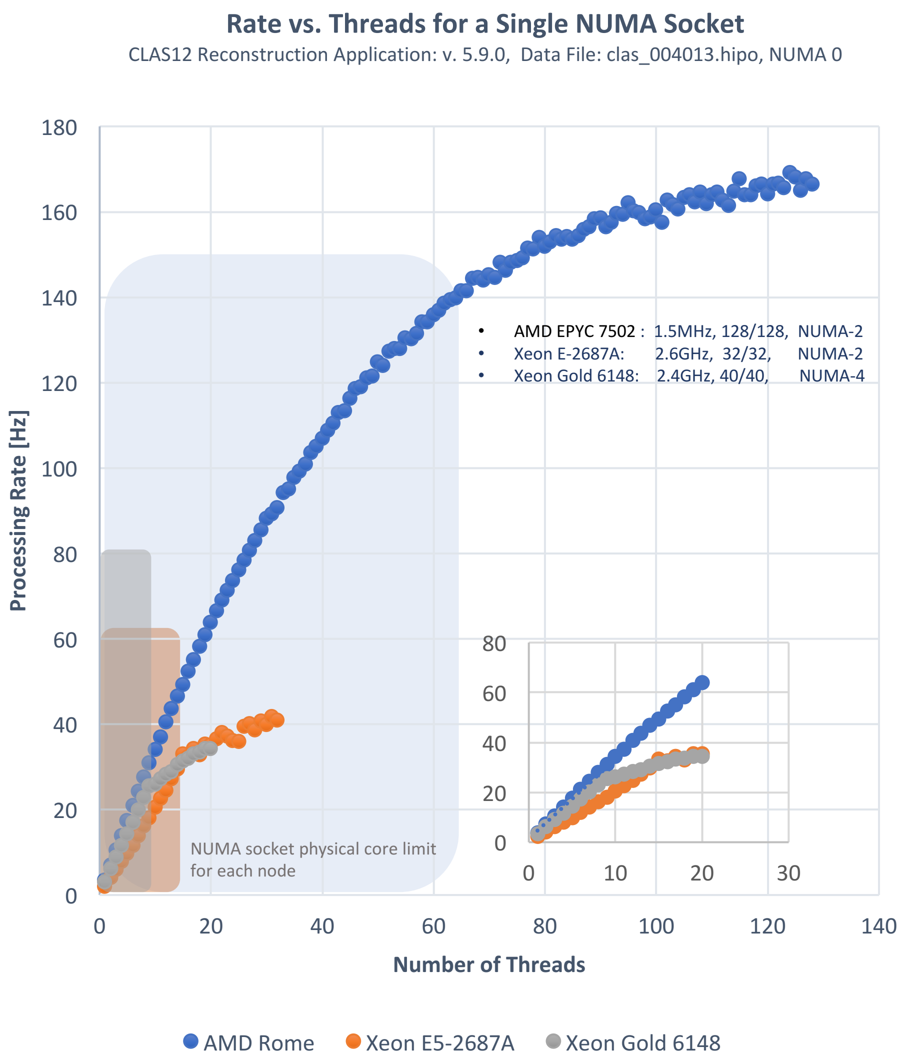
Data-quantum size and GPU occupancy

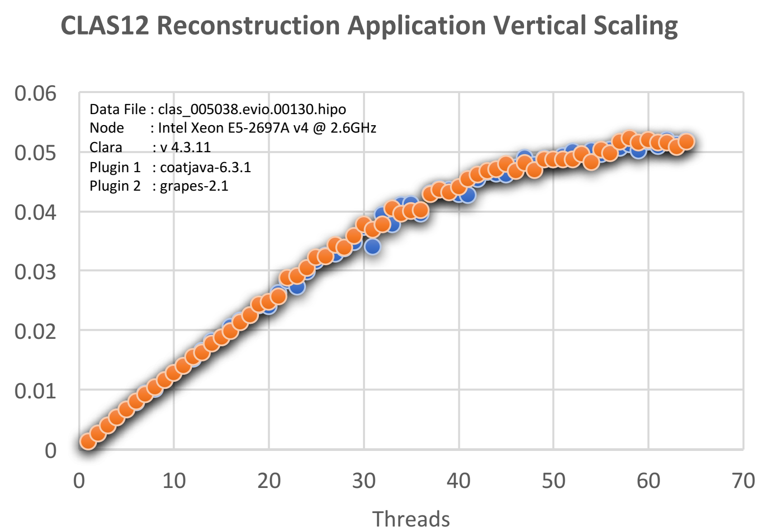


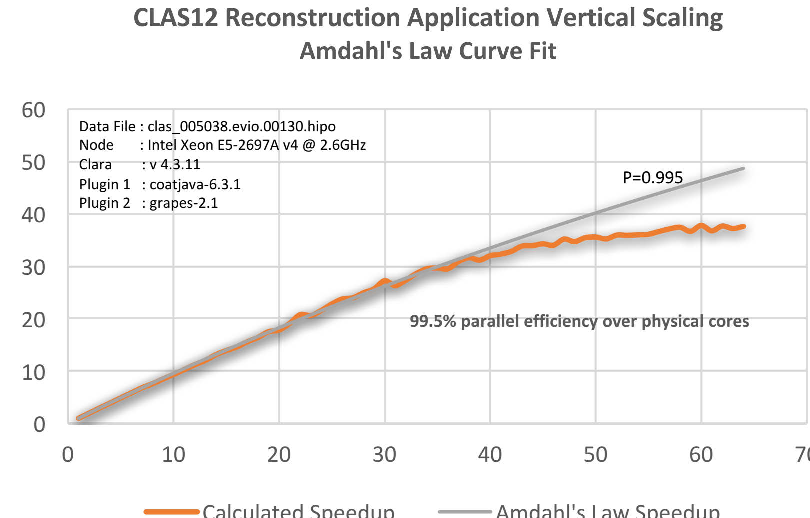
Data-processing chain per NUMA



Results







Jefferson Lab