CLARA

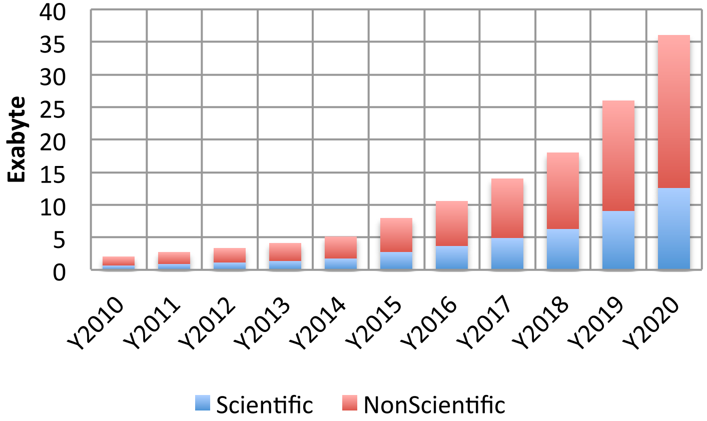
Component Based Dataflow Processing Framework

V. Gyurjyan, A. Bartle, C. Lukashin, S. Mancilla, R. Oyarzun, A. Vakhnin

Science Data 3V Expansion

- Unprecedented growth of data
  - volumes
    - Data at rest is growing exponentially.
      - EOS: 1Exabyte in 10 years.
      - LHC: 12-14 Petabyte/year. In 2023 400Petabyte/year.
      - SKA: 22Exabyte/year in 2023
  - Velocities
    - Data acquisition rates, as well as new data producing devices are in rise.
      - LHC: 1.5GByte/sec
      - SKA: 700Terabyte/sec
  - Varieties
    - Plethora of data formats, data structures and data types

Global Digital Data



This Is a Good News

- We need more data to confirm and/or generate a new knowledge.
- We need more diverse data.
  - Correlating multiple data sources can lead to interesting insights of all.
- We need more/easy access to the data.
  - "Two heads are better than one". We need to put data and humans together to get more science from it.

3V Challenge in Science

- Prevent data pollution
  - Unprocessed data is worse than garbage data
- We need to keep up with ever growing data production rates.
- Existing scientific data processing architectures will have difficulties handling future data volumes.
- We find commercial Big data processing solutions not well suited for our data processing needs.

Clas12 Reconstruction and Analytics

- Stream processing engine for scientific applications.
- Increase data processing speed.
  - The faster we analyze our data, the greater will be it's predictive power.
- Decrease application design, customization and maintenance effort.
  - Agile application design and maintenance is critical for embracing future technological advances.

Increase Data Processing Speed

- Move away from batch processing to near real-time data stream processing.
- Minimize data persistency.
  - Input data must be processed without being physically stored.
- Divide and concur.
  - Data divided into events that are processed in parallel in a horizontally scaled compute infrastructure.
  - Data processing becomes a streamline processing of defined events

Data Quanta or an Event

- Data acquisition trigger is used to readout sensor data
  - Hardware
  - Software
- Trigger defines a data quanta or an experimental event
- Data quanta (event) contains data fragments from multiple sensors or sources, related to one another with a defined science criteria.
- Processing streams of events defines scientific data analytics.

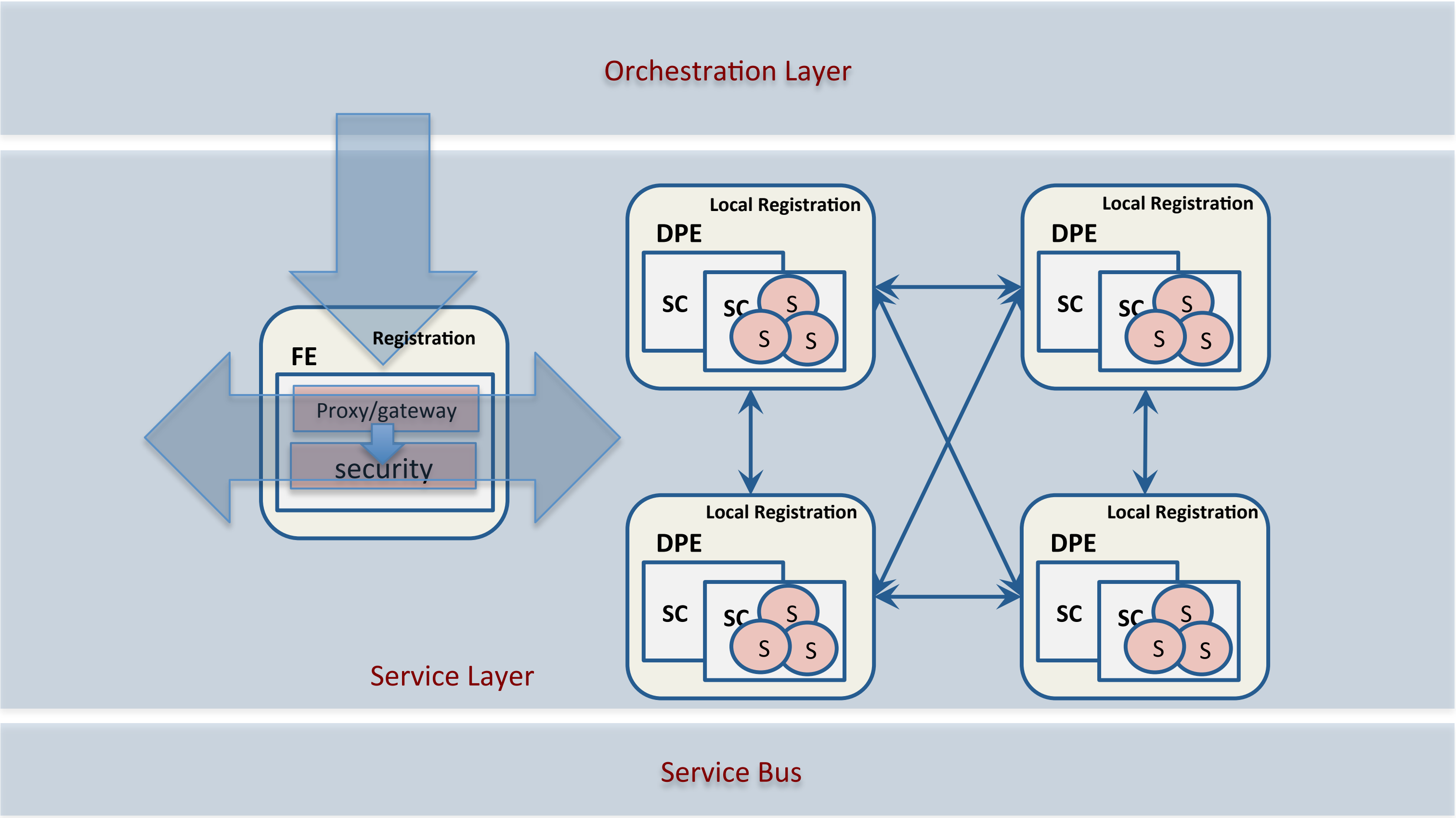
Agile Application Design

- Application is defined as a network of loosely coupled “black box” processes, called services (SOA / FGP implementation).
- Services communicate with each other by exchanging the data quanta.
  - Thus, services share the same understanding of the transient data, hence the only coupling between services.
- Services exchange data across predefined connections by message passing, where connections are specified externally to the services.
- Services can be requested from different data processing applications.
- Loose coupling of services makes polyglot data access and processing solutions possible.

Benefits

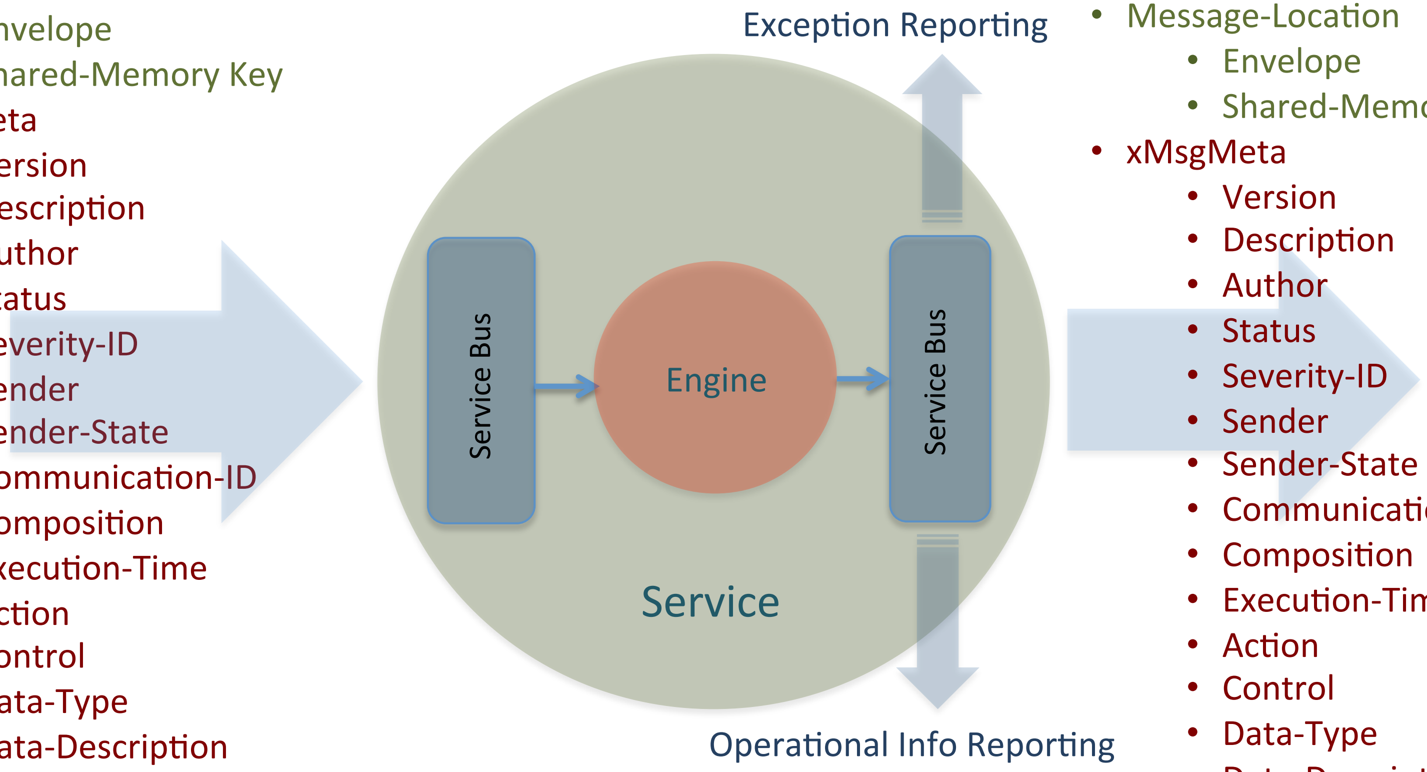
- Data processing application design is simple as graphically wiring services together.
- Inherently concurrent suited for multi-core hardware systems.
- Simplifies application customization, testing, monitoring and logging.
  - Just wire the data stream to a logging, debugging services or branch to a parallel processing service chain for comparative analyses.

Architecture



Transient Data and SaaS

- Topic
- Message-Location
  - Envelope
  - Shared-Memory Key
- xMsgMeta
  - Version
  - Description
  - Author
  - Status
  - Severity-ID
  - Sender
  - Sender-State
  - Communication-ID
  - Composition
  - Execution-Time
  - Action
  - Control
  - Data-Type
  - Data-Description
  - Reply-To
  - Byte-Order
- xMsgData-Object
- Byte-Array



- Topic
- Message-Location
  - Envelope
  - Shared-Memory Key
- xMsgMeta
  - Version
  - Description
  - Author
  - Status
  - Severity-ID
  - Sender
  - Sender-State
  - Communication-ID
  - Composition
  - Execution-Time
  - Action
  - Control
  - Data-Type
  - Data-Description
  - Reply-To
  - Byte-Order
- xMsgData-Object
- Byte-Array

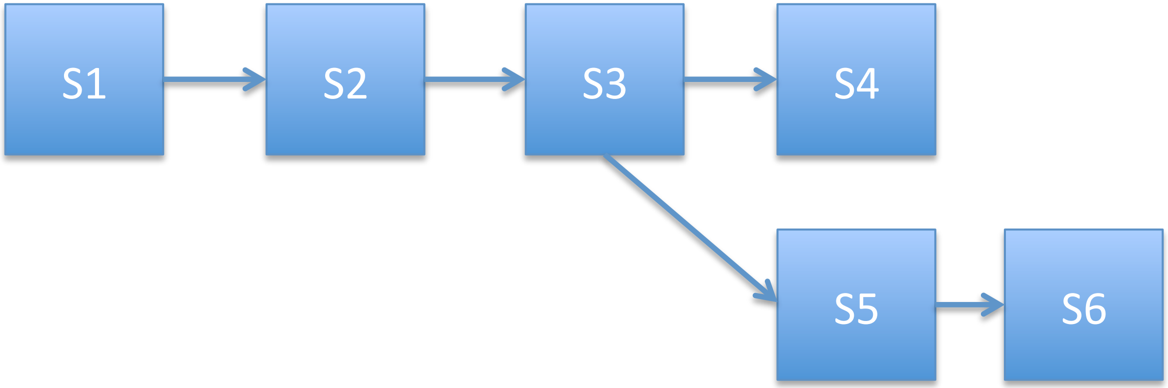
Types of Services

- Operational types
  - Event building and data provisioning (stream source)
    - Minimize raw data migration: EB services operate close to the data
    - Memory mapped raw data files
  - Event processing
- Conceptual types
  - Entity
    - Utility: legacy code as a service
  - Composite
    - Task
    - Orchestrated task

Application design

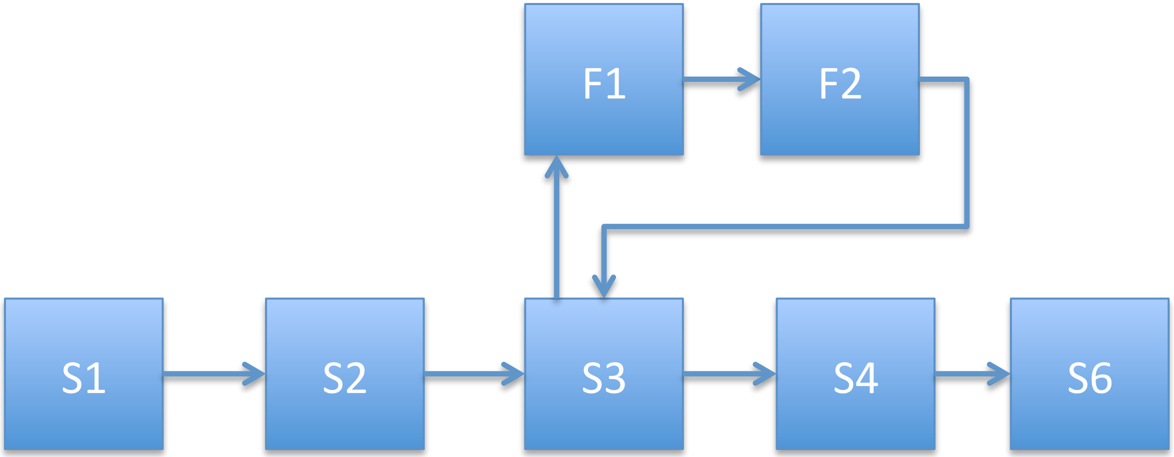
- Data driven, data centric design.
  - The focus is on transient data modifications. Advantage over algorithm driven design is that a much greater ignorance of the data processing code is allowed (loose coupling).
- Design = service composition + data-routing.
  - Self routing (no routing scheduler)
- Data routing graph defines application algorithm
- Syntactic and graphical representation of the application design

Application Examples

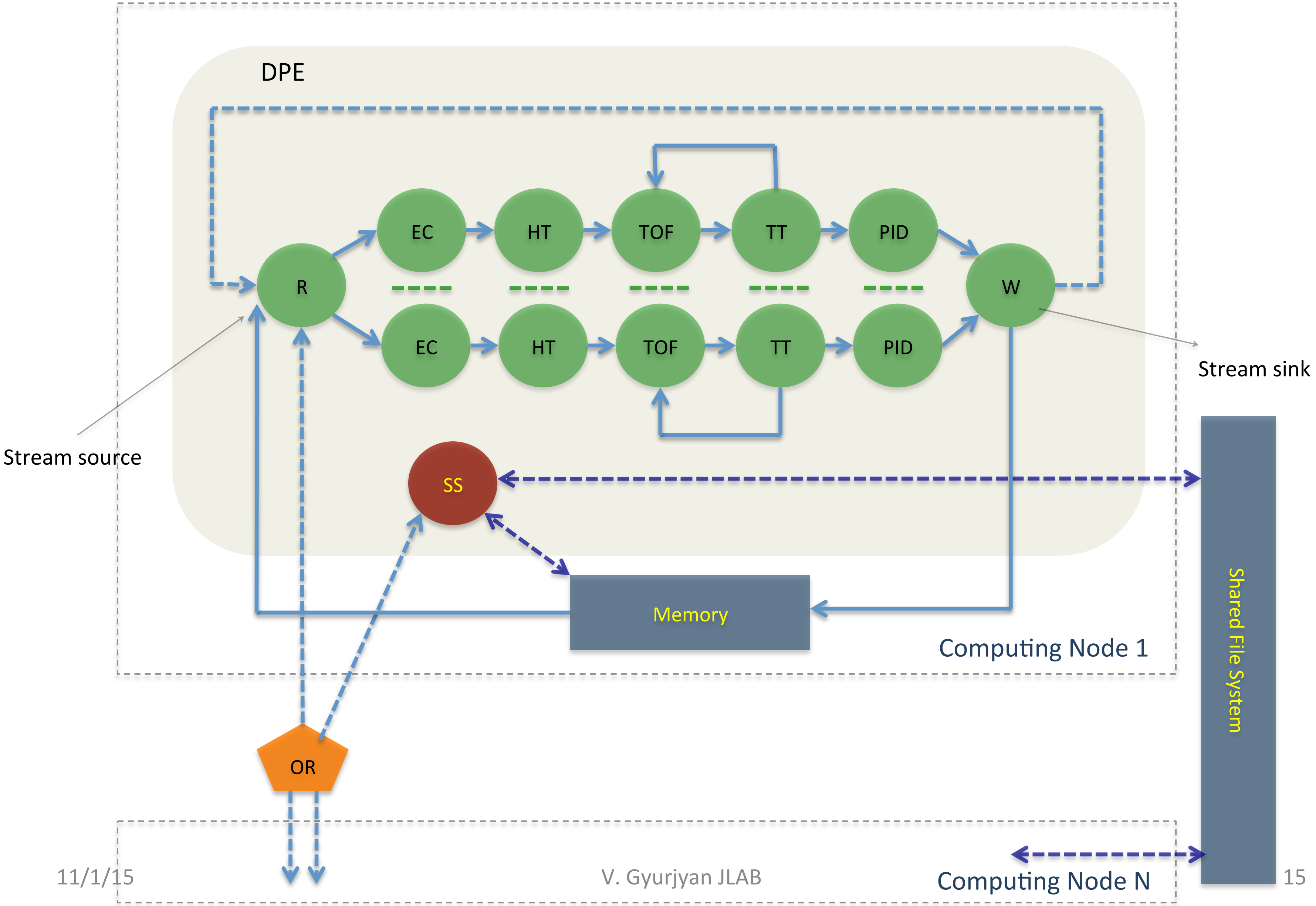


```
S1 + S2 + S3 + S4;
S3 + S5 + S6;
```

```
S1 + S2 + S3;
while( S3 == "xyz") {
   F1 + F2 + S3;
}
  S3 + S4 + S6;
```

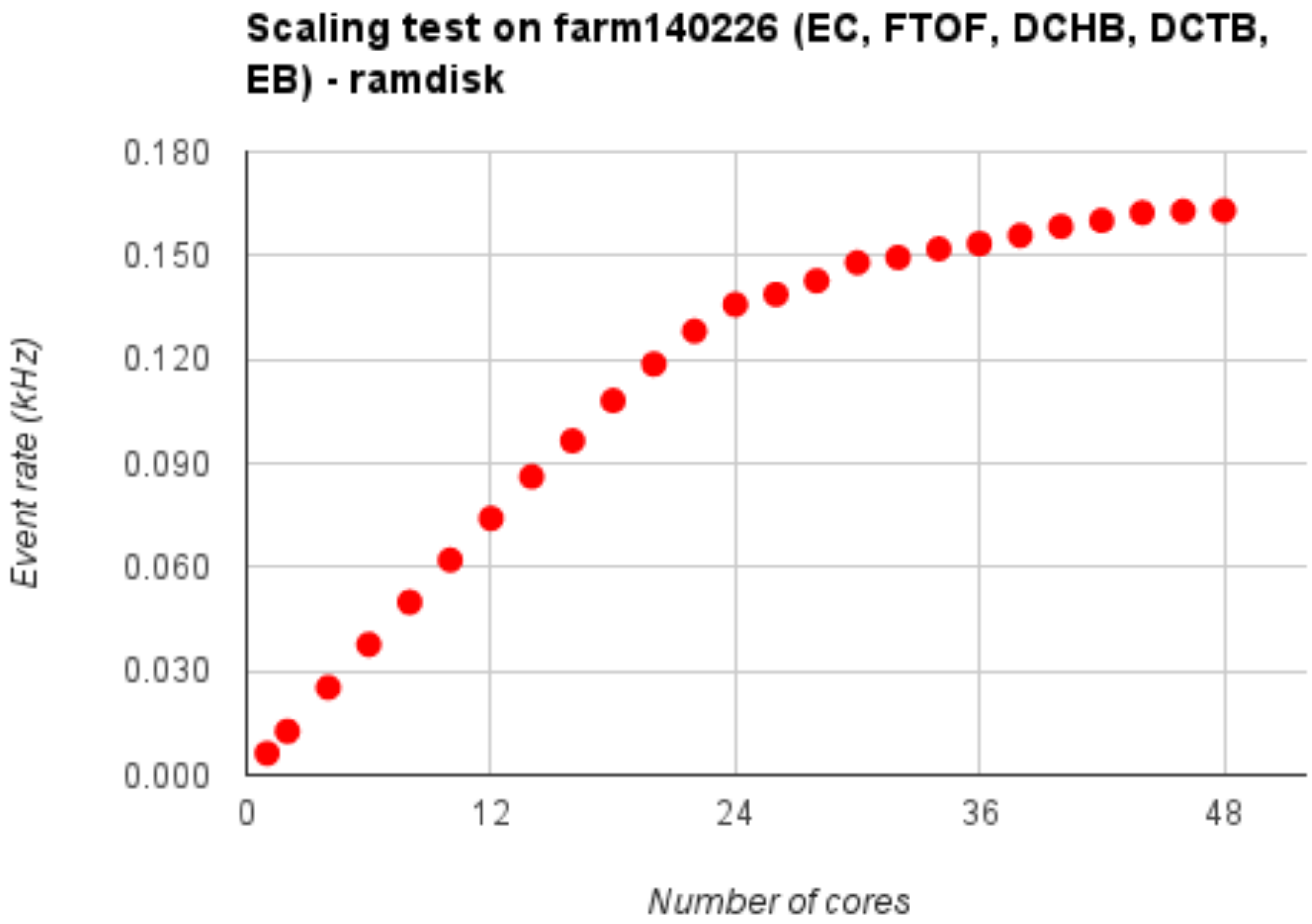


Clas12 Reconstruction Application



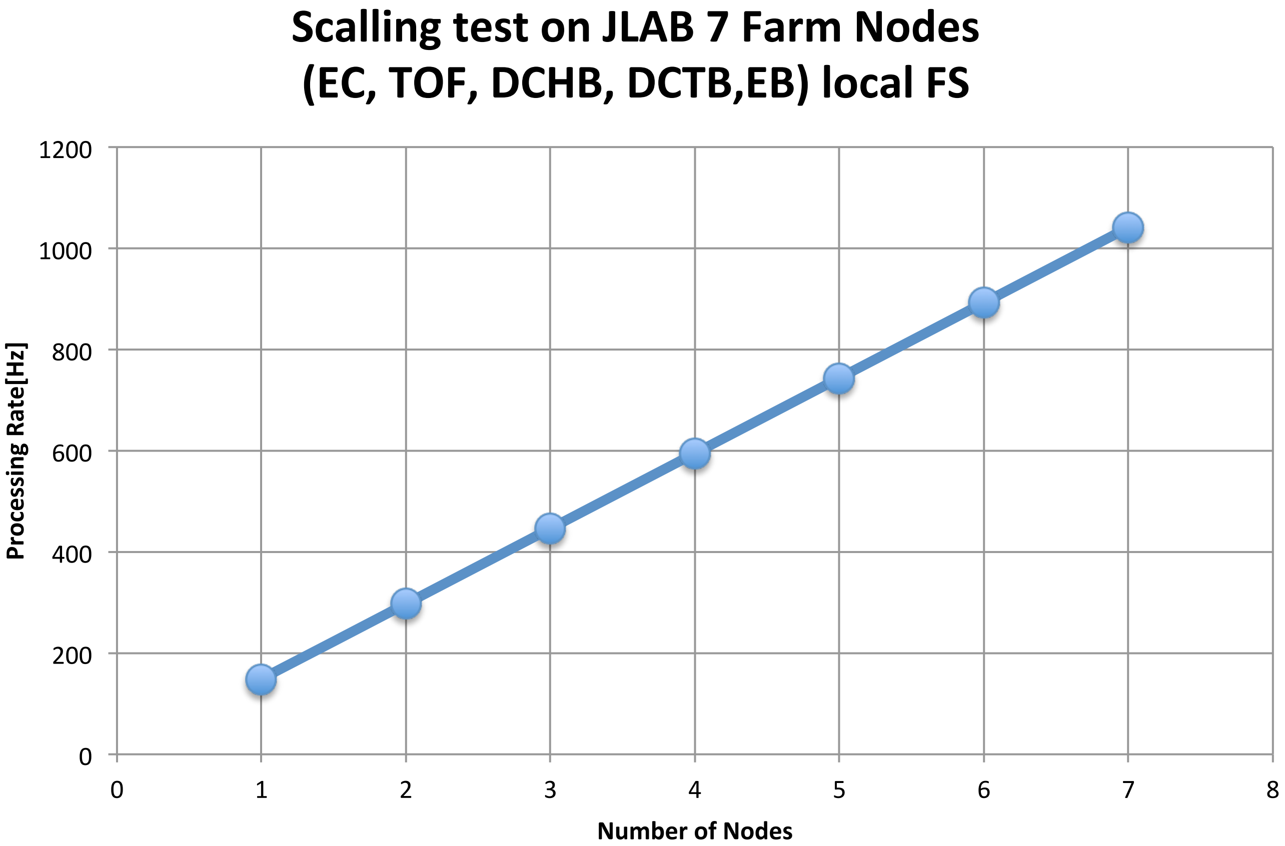
Scaling Within a Single Node

Intel Haswell 24/48 core system



Scaling Within Multiple Node

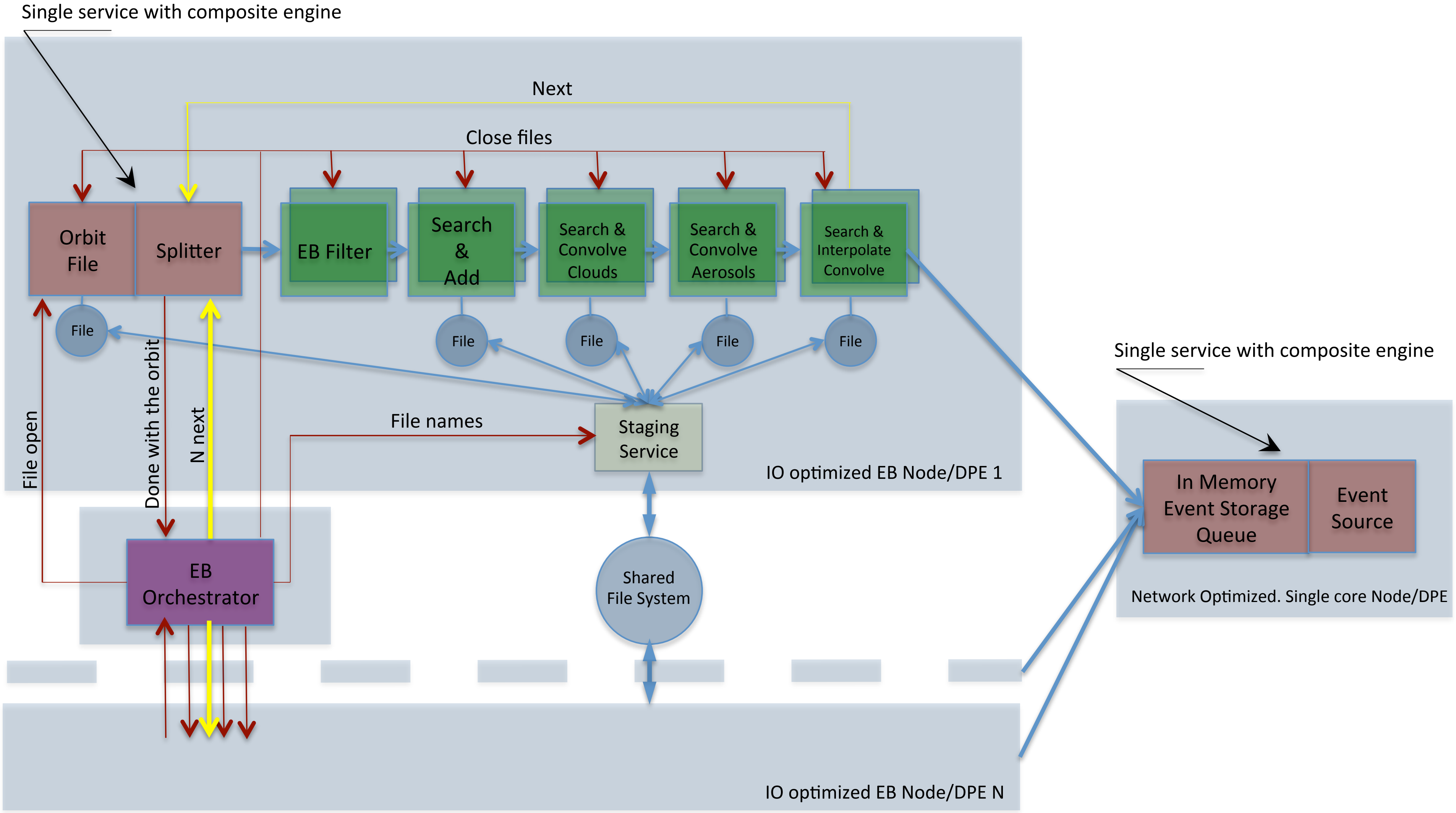
Intel Haswell 24/48 core system



50 node cluster is capable of keeping up with CLAS12 DAQ rates

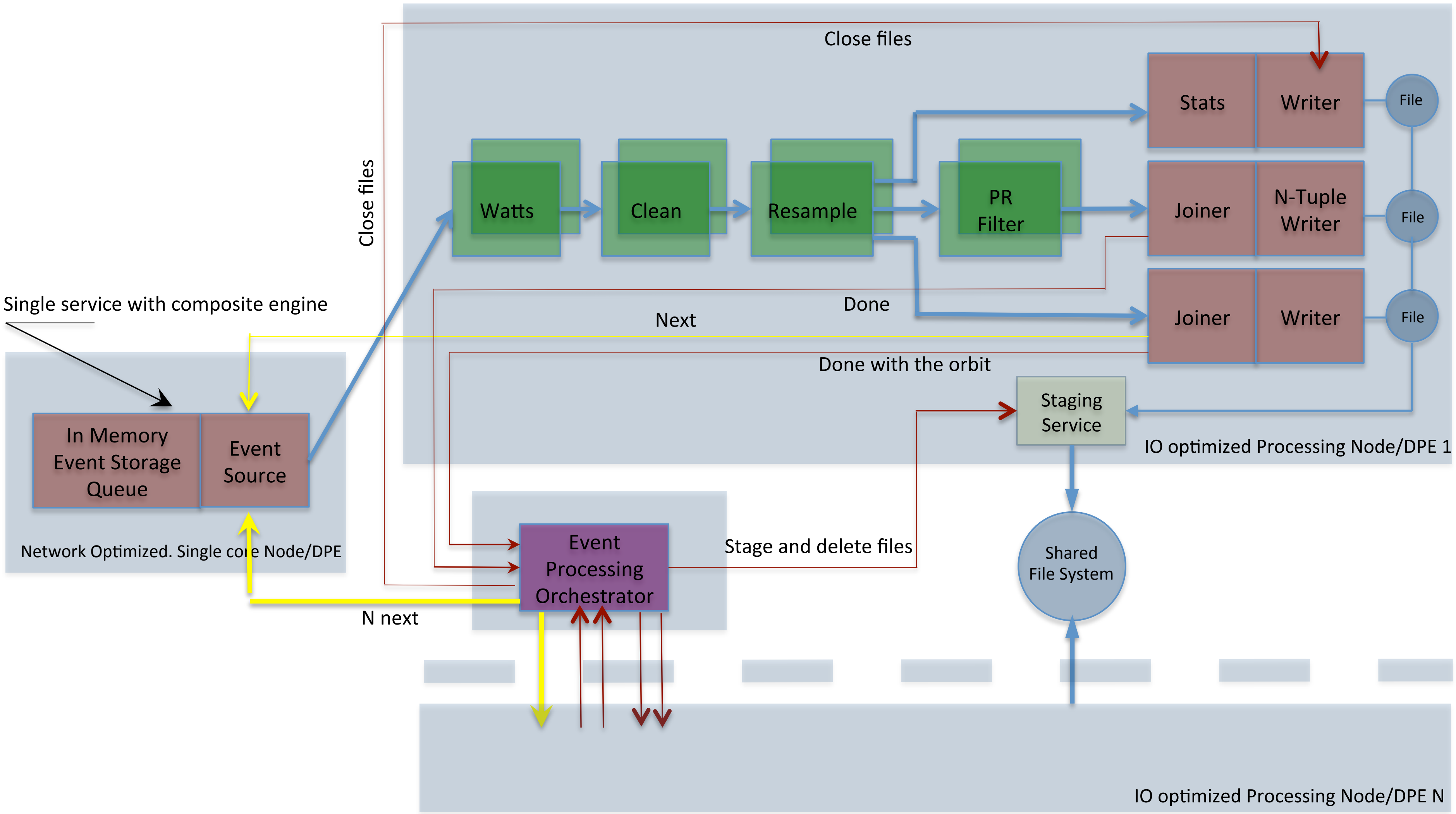
NAIADS Parallel Event Building Application

(Stream builder)



Single service with composite engine
Next
Close files
Orbit File
Splitter
EB Filter
Search & Add
Search & Convolve Clouds
Search & Convolve Aerosols
Search & Interpolate Convolve
File
File
File
File
File
File open
Done with the orbit
N next
File names
Staging Service
IO optimized EB Node/DPE 1
EB Orchestrator
Shared File System
IO optimized EB Node/DPE N
Single service with composite engine
In Memory Event Storage Queue
Event Source
Network Optimized. Single core Node/DPE

NAIADS Parallel Event Processing Application



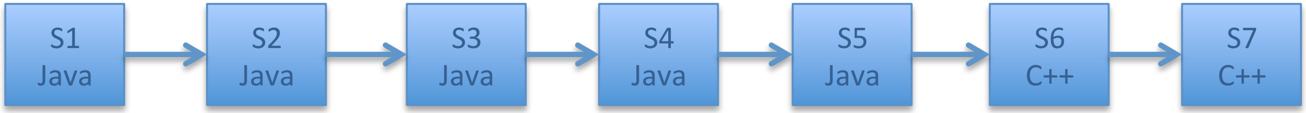
Thank You

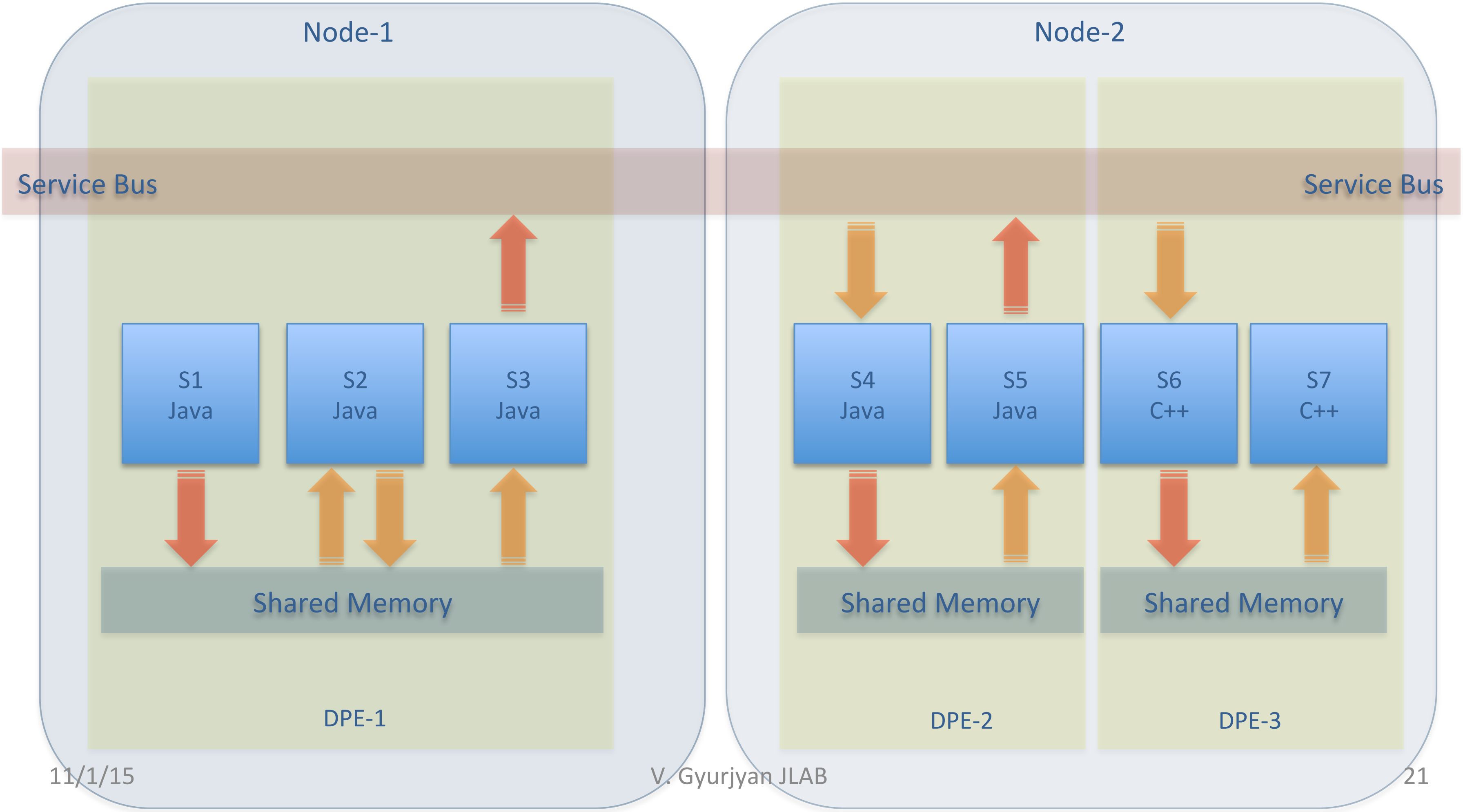
e-mail: gurjyan@jlab.org

web site: claraweb.jlab.org (under construction)

Shared Memory Support

Event processing application composition (task service)



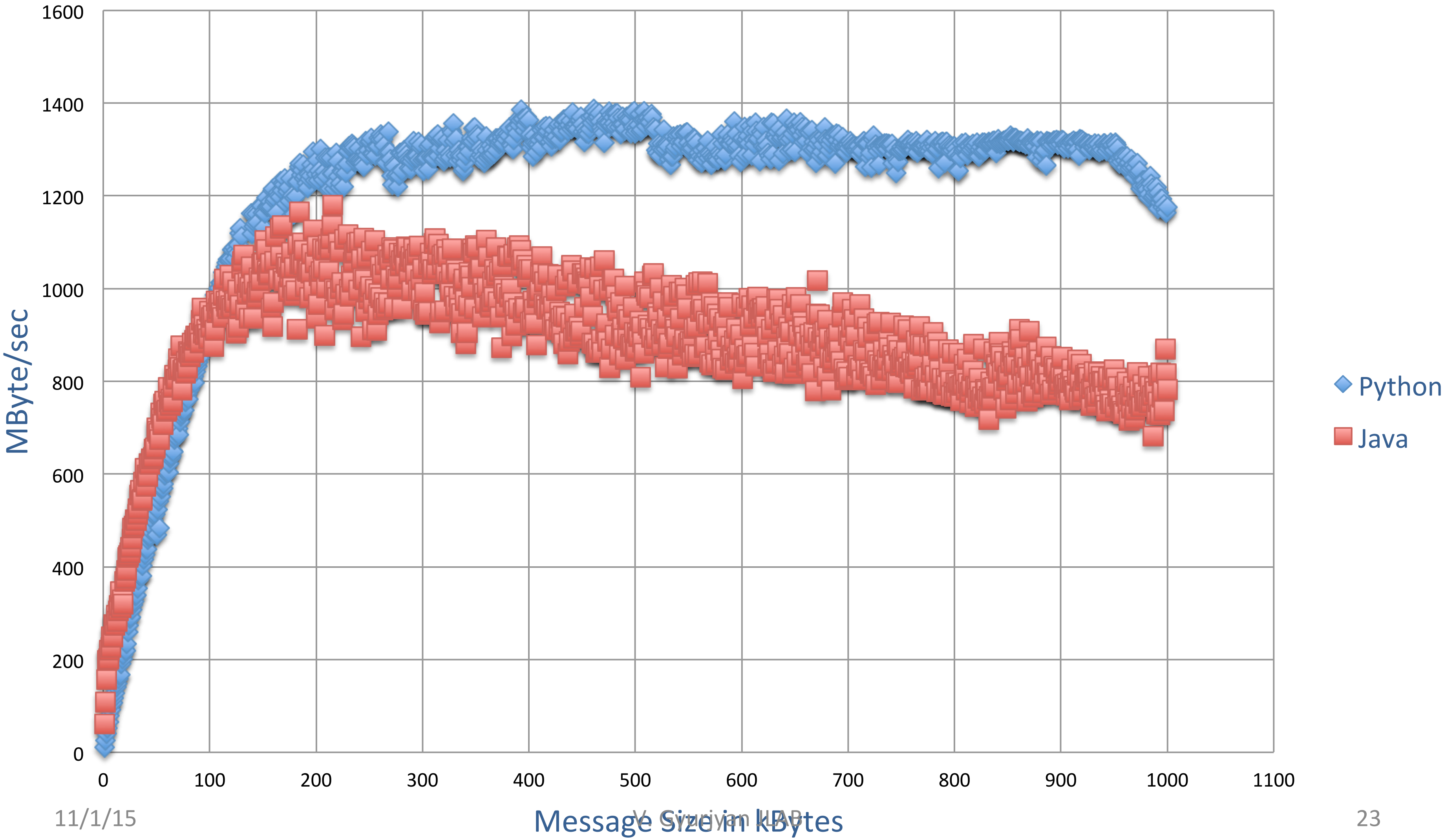


Service Bus

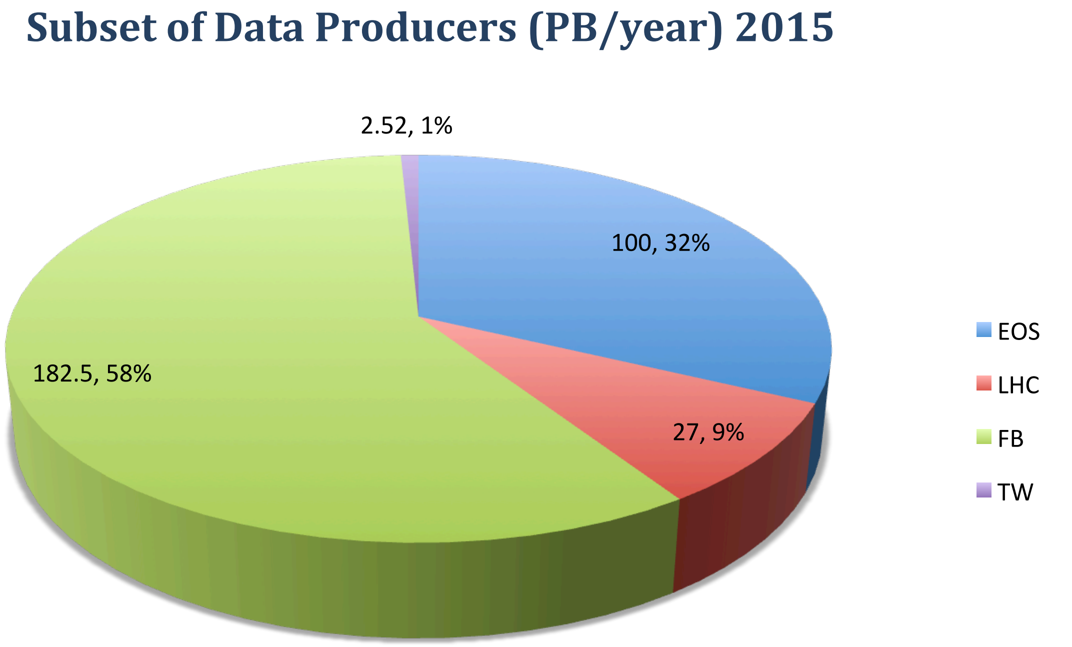
- Fast, multilingual publish subscribe messaging system: xMsg
  - Based on ZeroMQ socket libraries.
- Java, Python and C++ bindings.

xMsg Publish-Subscribe Messaging Performance

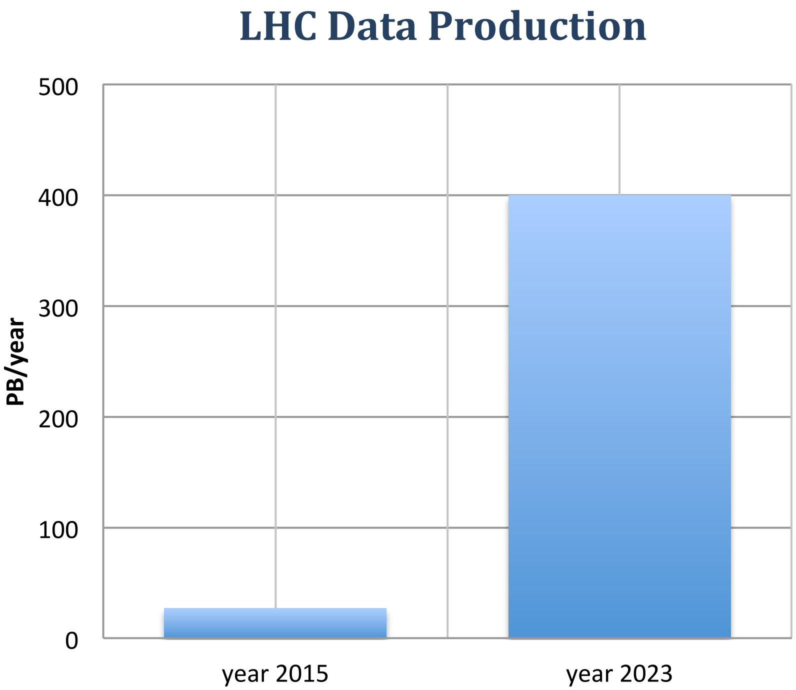
Java & Python binding. 100K messages per measurement.
Single threaded publisher and subscriber processes.
Single computing node. 2.3 GHz Intel Core i7,
16GB 1600 MHz DDR3, OS X 10.10.2

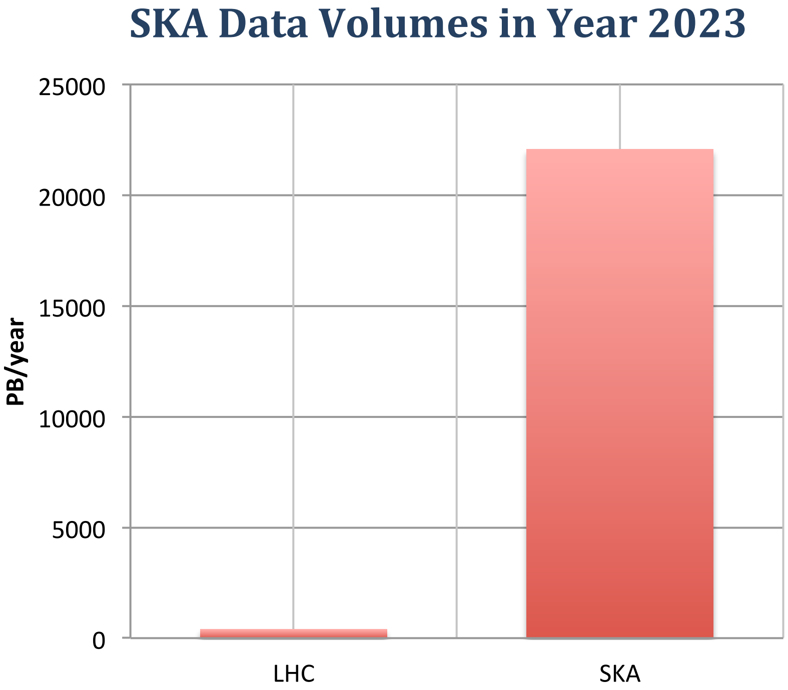


Digital Data Demography

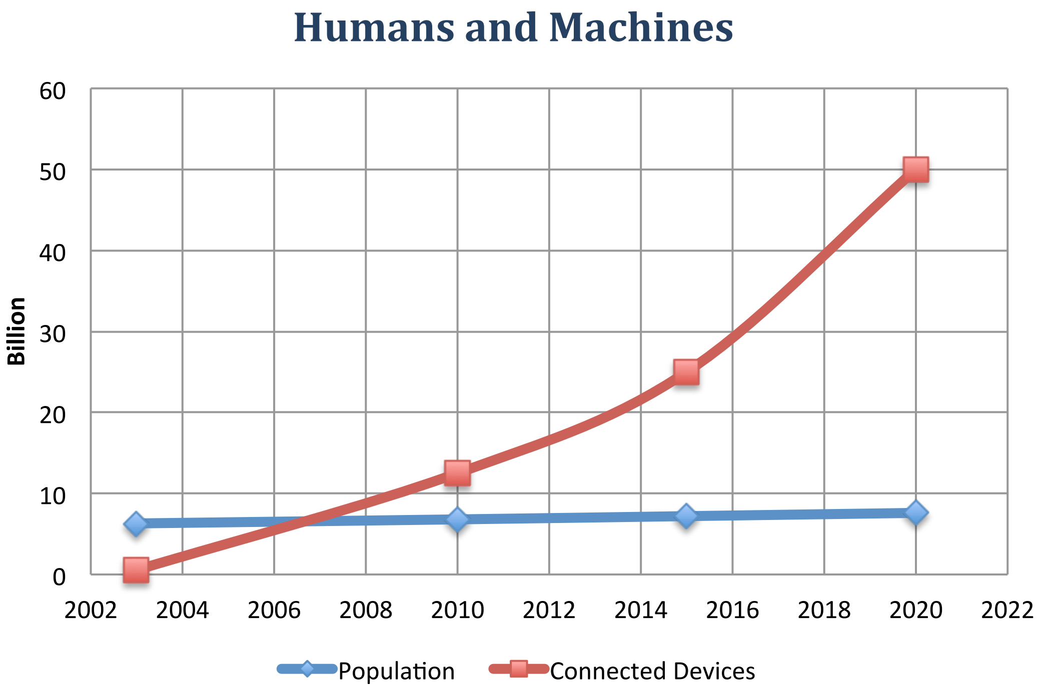


Scientific Data Expansion





The Rise of IoT


Humans and Machines
Billion
Population
Connected Devices