

CLAS12 Software

G.Gavalian (Jefferson Lab)

- ▶ Software organization
- ▶ Data formats
- ▶ Visualization
- ▶ Reconstruction
- ▶ Summary



▶ Source Code:

- ✓ reconstruction software common tools are in JAVA
- ✓ reconstruction engines are JAVA plugins with CLARA
- ✓ code management is done using github (reconstruction plugins)
- ✓ build system MAVEN (nightly builds)

▶ Software Distribution:

- ✓ software packages distributed via MAVEN repository
- ✓ software bundle distributed through git (reconstruction codes and utilities)

▶ Databases:

- ✓ calibration database CCDB (Hall-D development)
- ✓ sqlite version is distributed with software bundle
- ✓ geometry database CCDB, run number and variation dependence
- ✓ nightly database dump into sqlite (included in software package build)

► Input/Output

- ✓ raw data in EVIO format composite bank structure
- ✓ reconstruction output HIPO format (record based and compressed)
- ✓ common interfaces to read EVIO, BOS, HIPO files
- ✓ data processing interface (format agnostic) for calibration and monitoring

► Databases:

- ✓ calibration constants and geometry definitions (CCDB)
- ✓ utilities to display calibration constants, compare for different sets
- ✓ caching algorithm of database constants for reconstruction

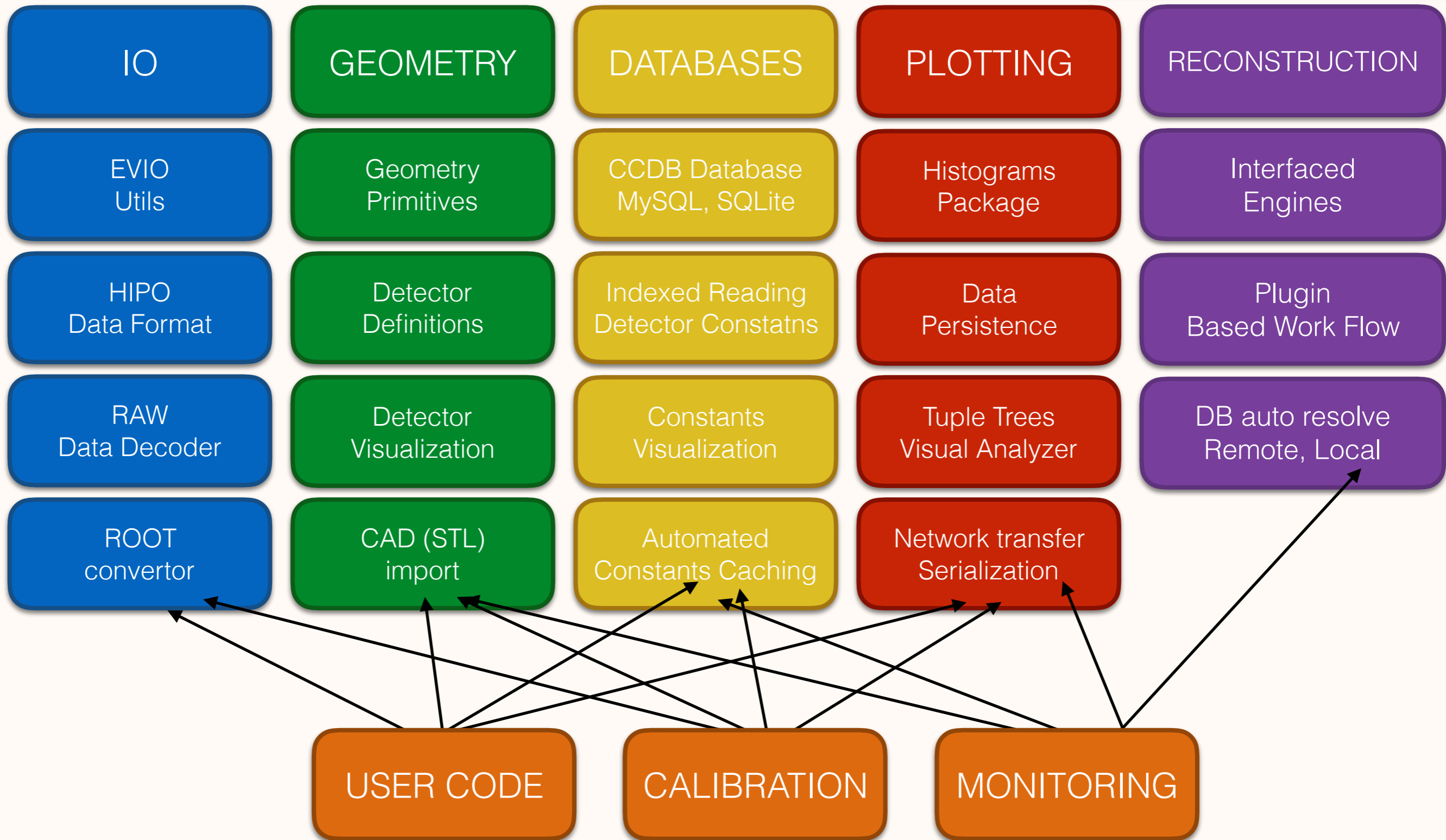
► Geometry:

- ✓ unified geometry package used by simulation, reconstruction and event display
- ✓ ability to import CAD files (GEANT4 tessellated shapes)
- ✓ detector visualization package with callbacks and automated occupancy display

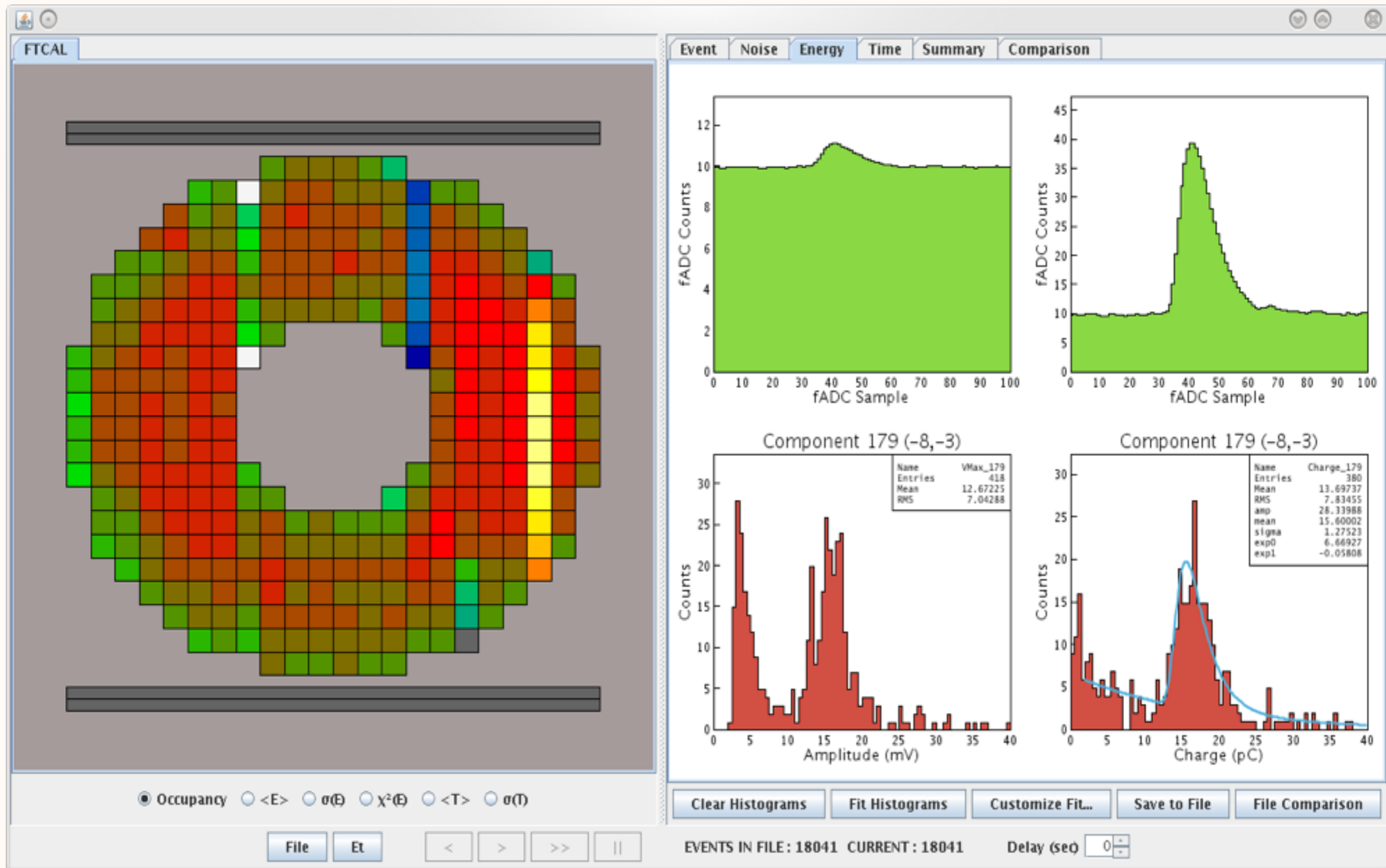
► Reconstruction:

- ✓ reconstruction engine hides CLARA complexity from users
- ✓ automatic detector initialization from database (run dependent)
- ✓ run dependent calibration constants caching for reconstruction engines

Software Structure



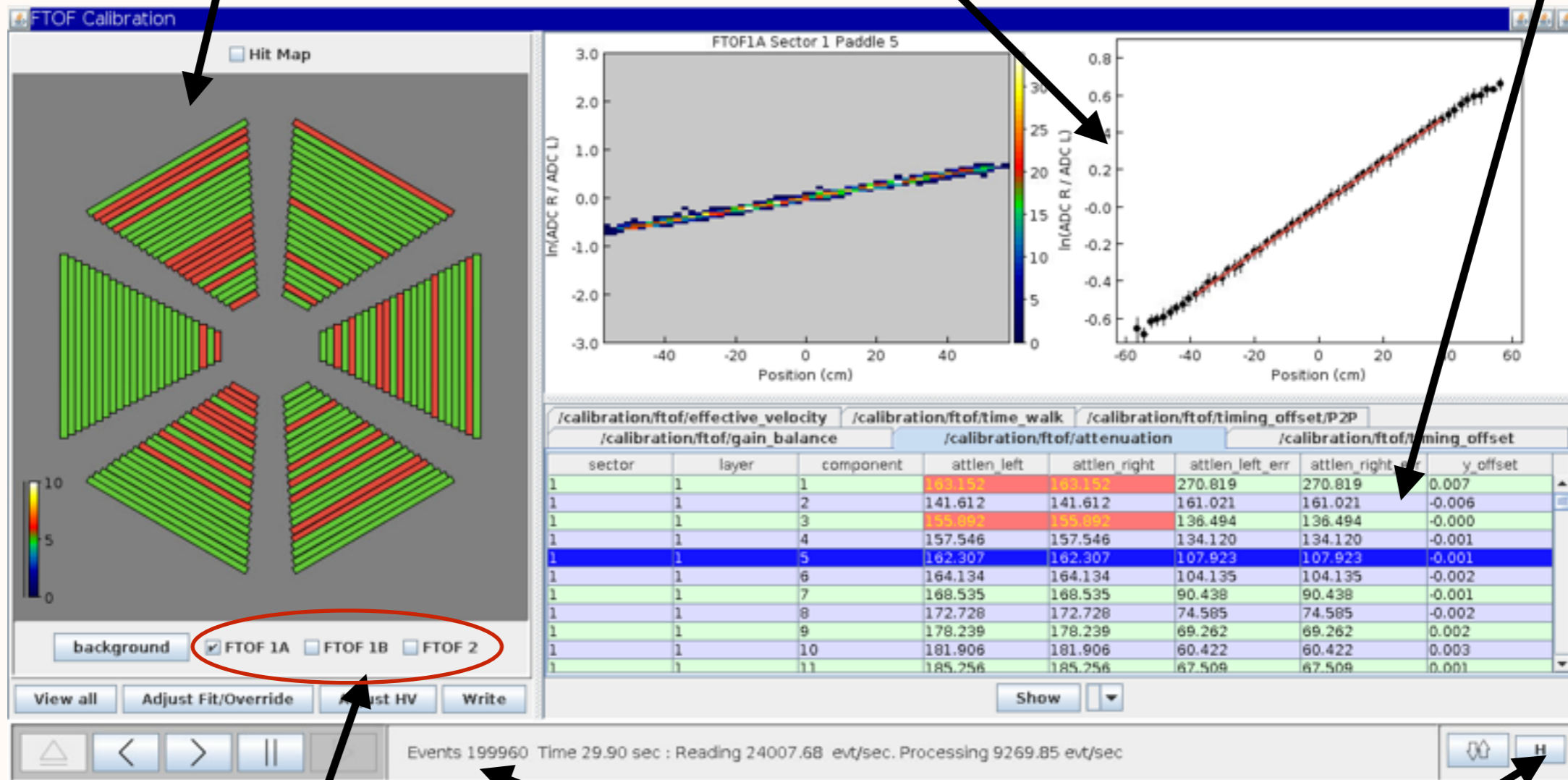
User Code



Detector Visualization
 callback on clicks
 coloring by occupancy

Data Canvas
 callback for detector
 callback for table

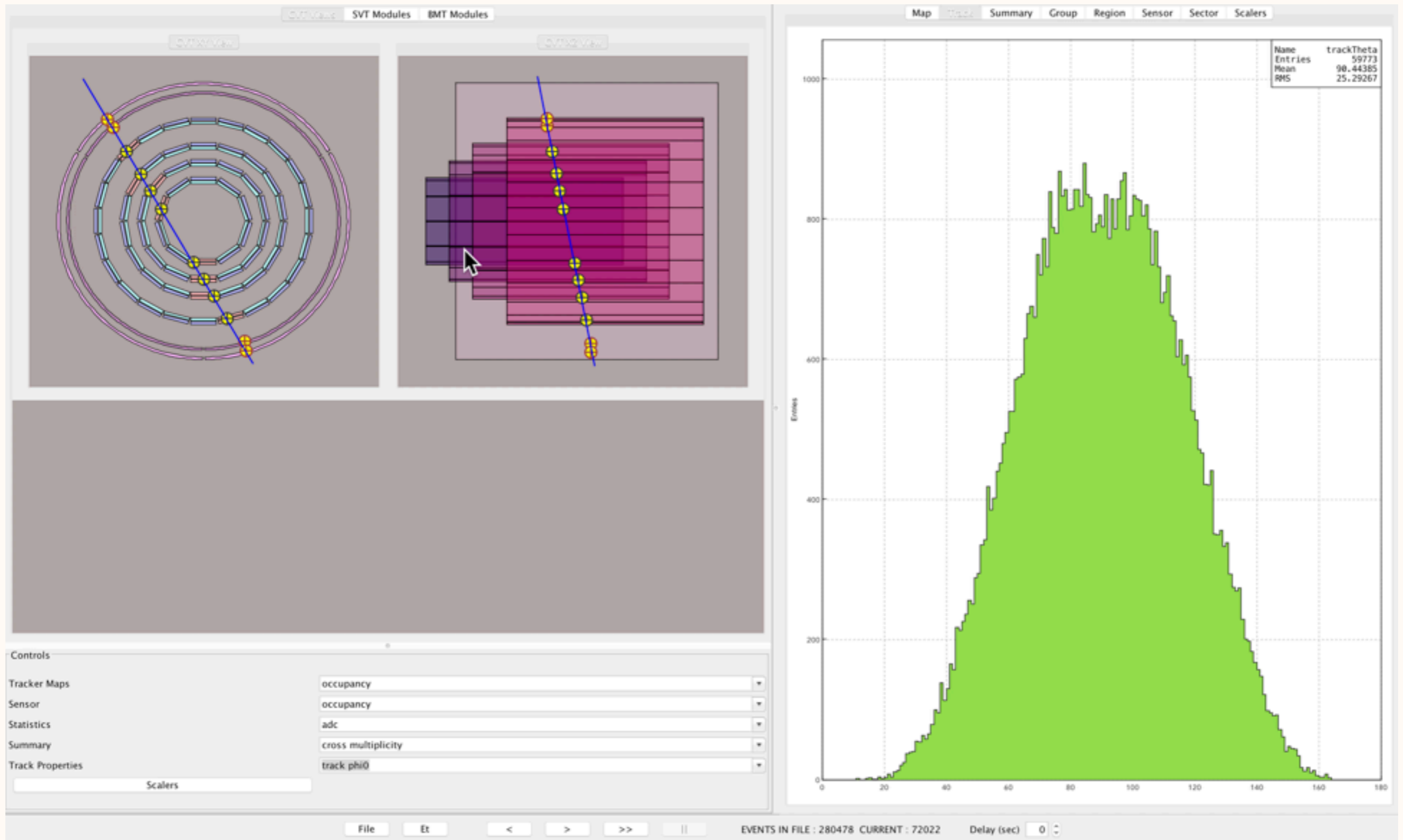
Database Table
 component callback
 constraint coloring
 comparison utils



Detector Visualization
 Layers

Event Processing Pane
 event by event, or whole
 opens: File, ET ring, CLOUD

Online Monitoring

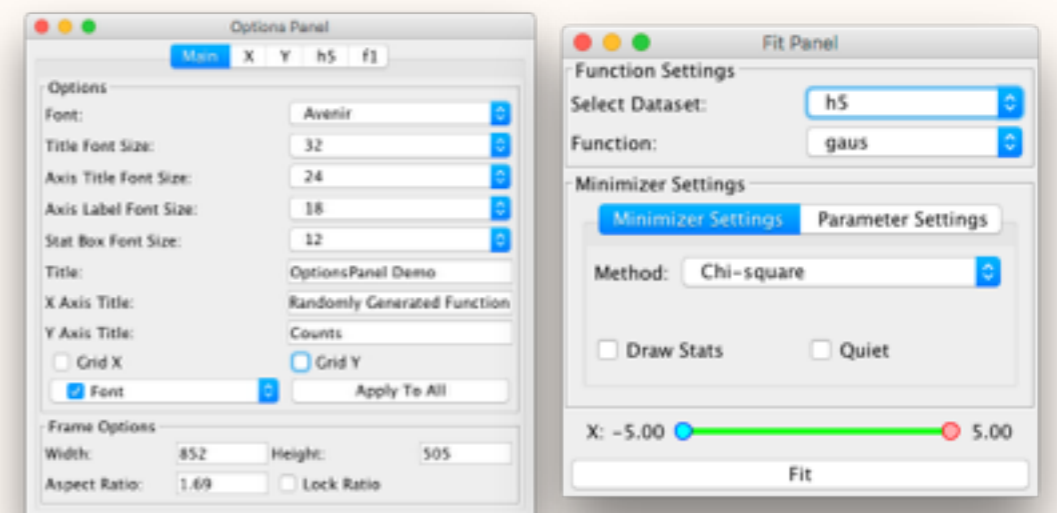
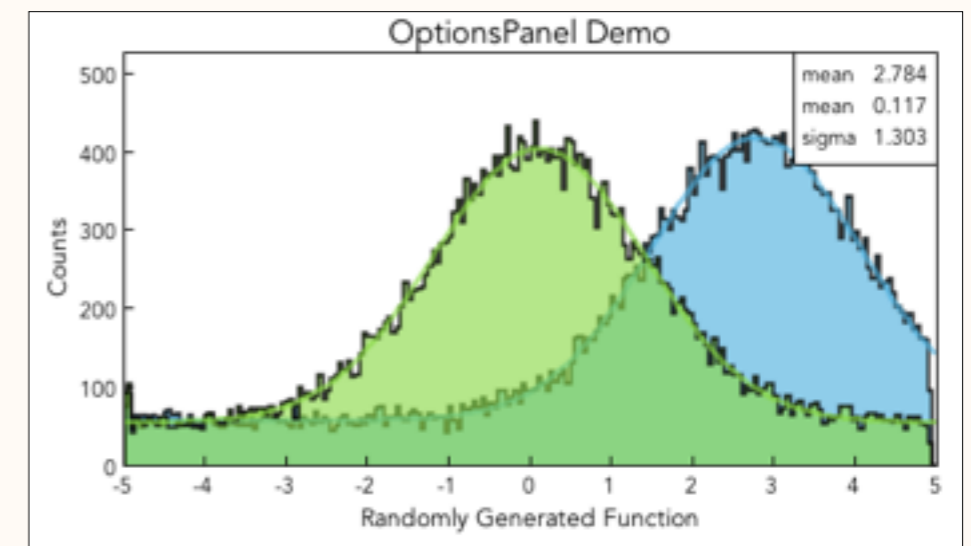
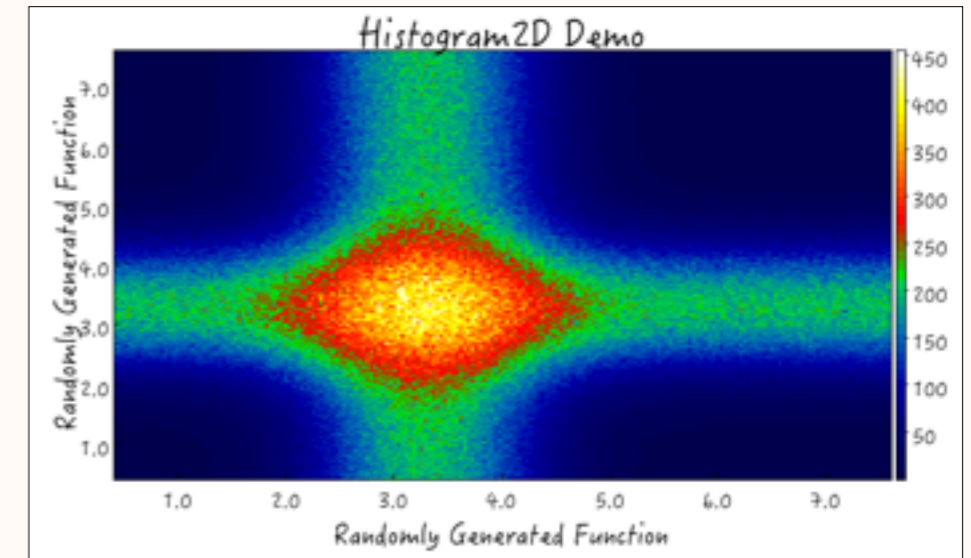


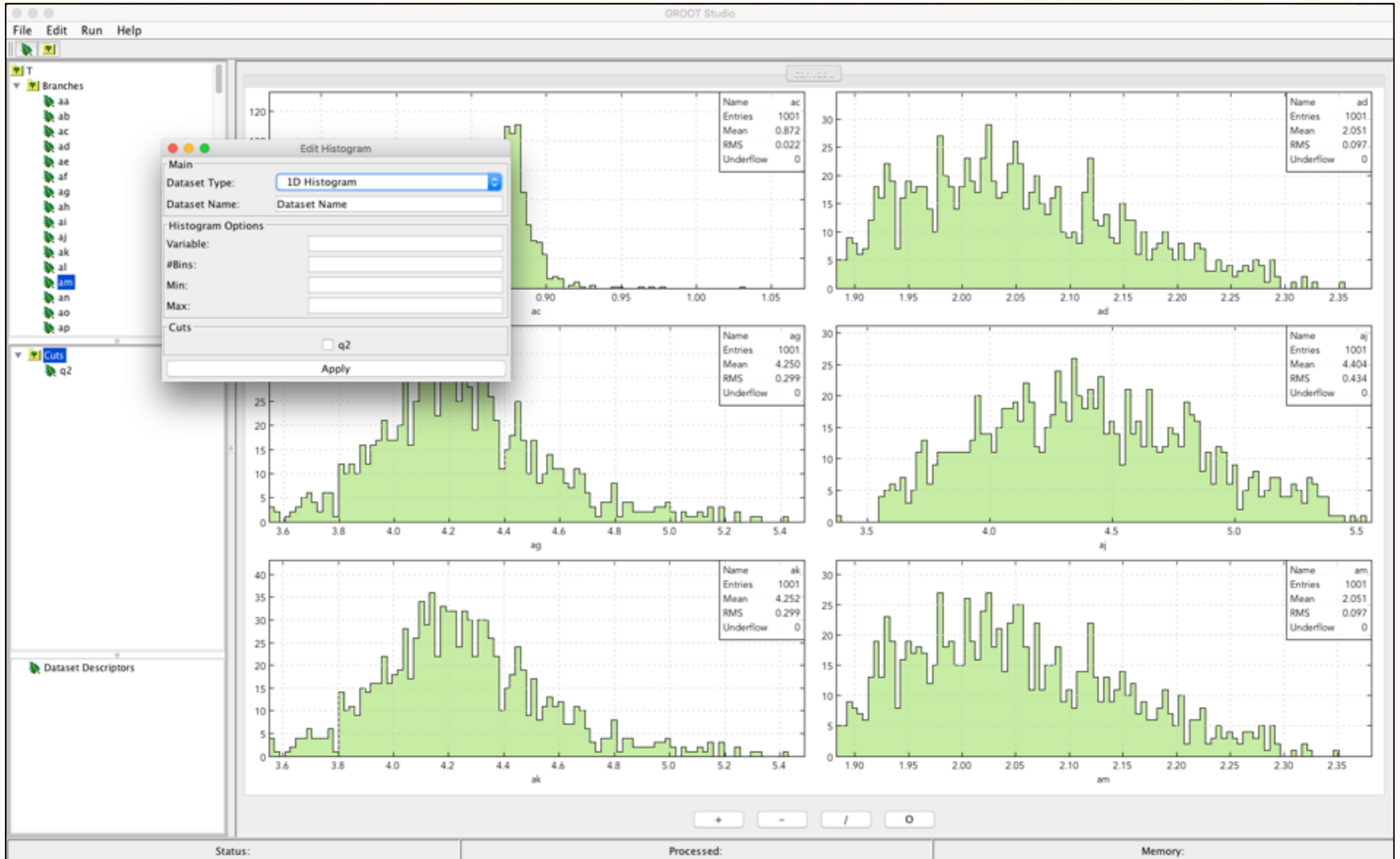
► Data Visualization Package:

- pure Java implementation of plotting
- histograms 1D, 2D and GraphErrors
- functions and MINUIT fitting
- interactive styles and property editors
- tuple tree implementation
- saves data to HIPO files (compressed)
- data serialization for network transfer

► Studio UI

- analysis studio for visual data analysis
- interactive fitting, custom function builder
- interactive data set comparison algorithms
- ASCII tuple import/export
- serialized data export with analysis procedure







► xMsg CLARA service bus

- ✓ general purpose public subscribe MPI
- ✓ utilizes zeroMQ socket libraries

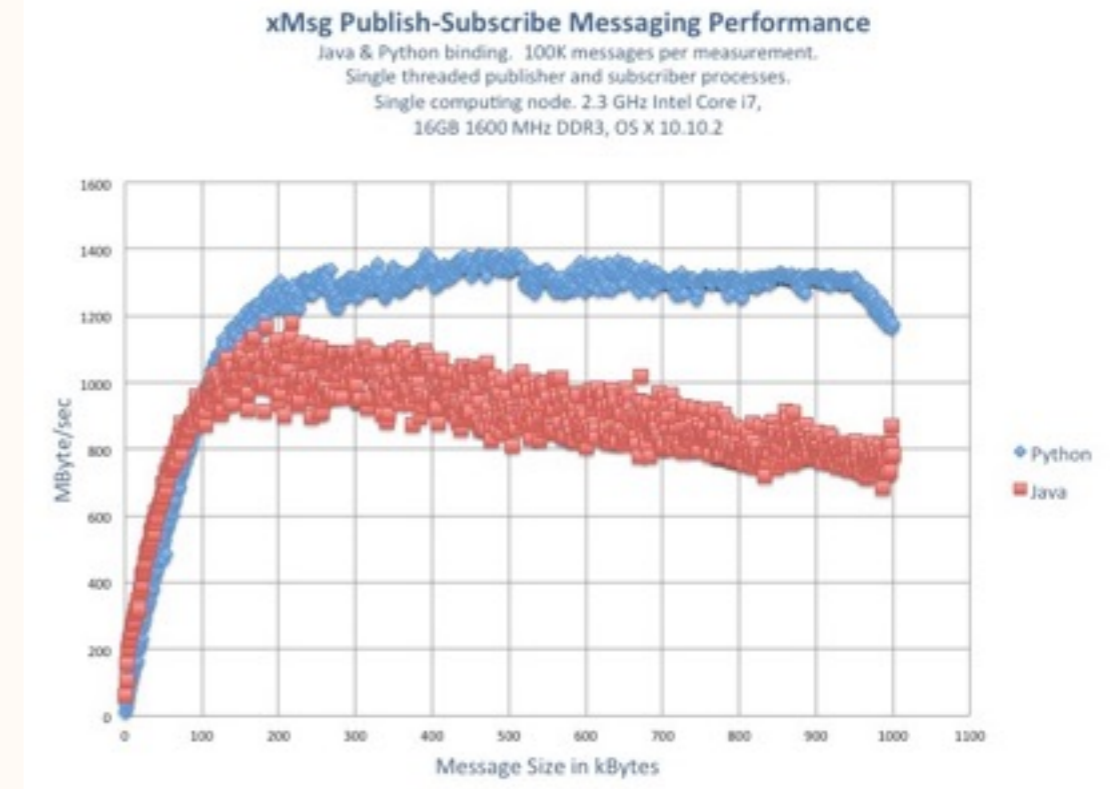
► Sockets that carry messages across various transports

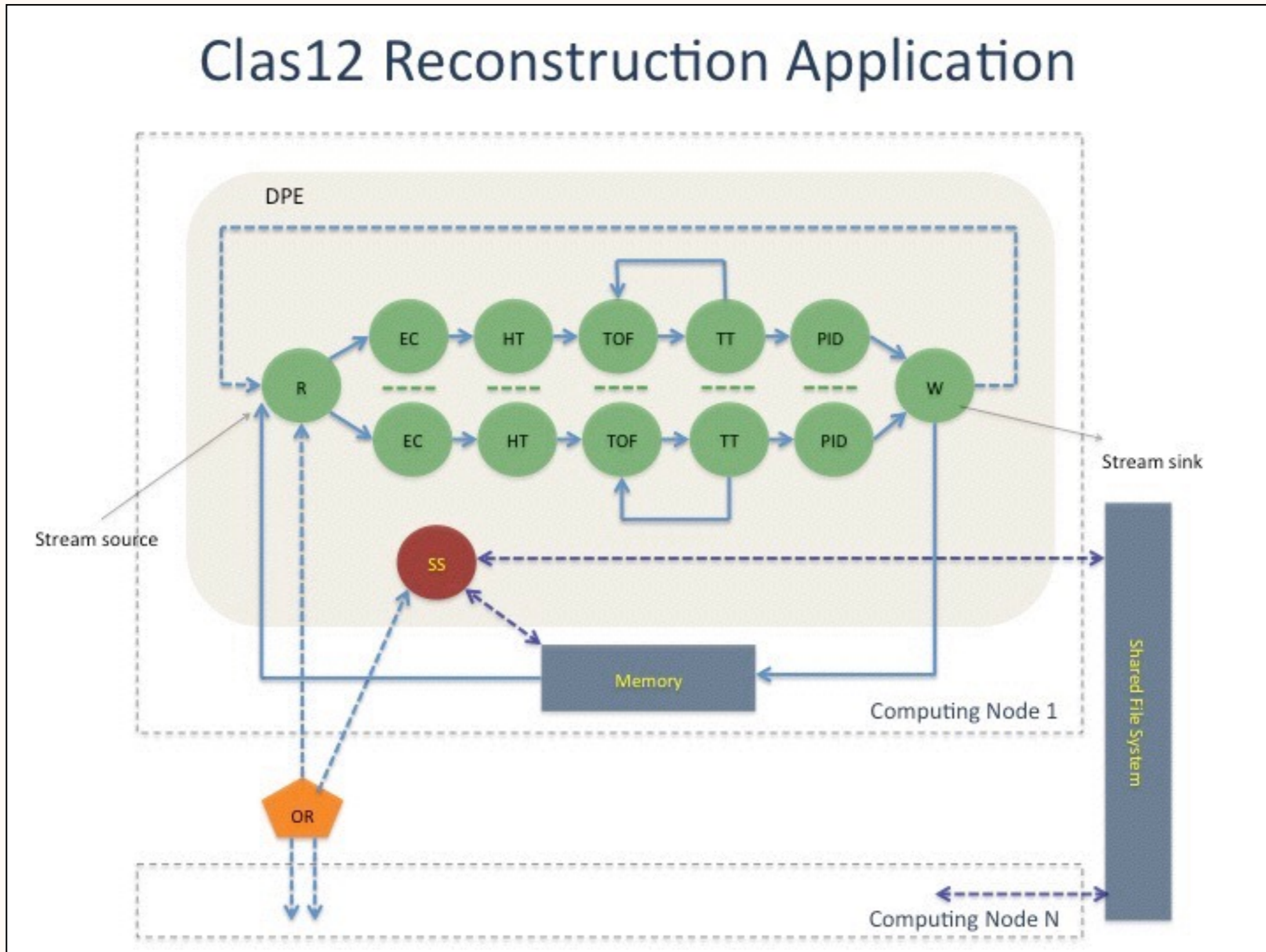
- ✓ In-process
- ✓ Inter-process
- ✓ TCP
- ✓ Multicast

► Sockets can be connected N-to-N with patterns

- ✓ Fan-out
- ✓ Pub-sub
- ✓ Task distribution
- ✓ Request-reply

► Java, C++, Python bindings





► Reconstruction Code:

- ✓ written in JAVA (calibration and reconstruction engines)
- ✓ runs in CLARA environment (multi-threaded) with dynamic configuration

► Reconstruction package:

- ✓ software comes in one package which includes:
 - ✓ descriptors for data banks
 - ✓ local copy of calibration database (sqlite)
 - ✓ magnetic map definitions and swimmers
 - ✓ scripts specifying different run configurations (YAML files describing Engines)

► Software Structure:

- ✓ uber JAR containing compiled common tools
- ✓ plugin directory for reconstruction engine codes
- ✓ clara environment running tools (multi-threaded)

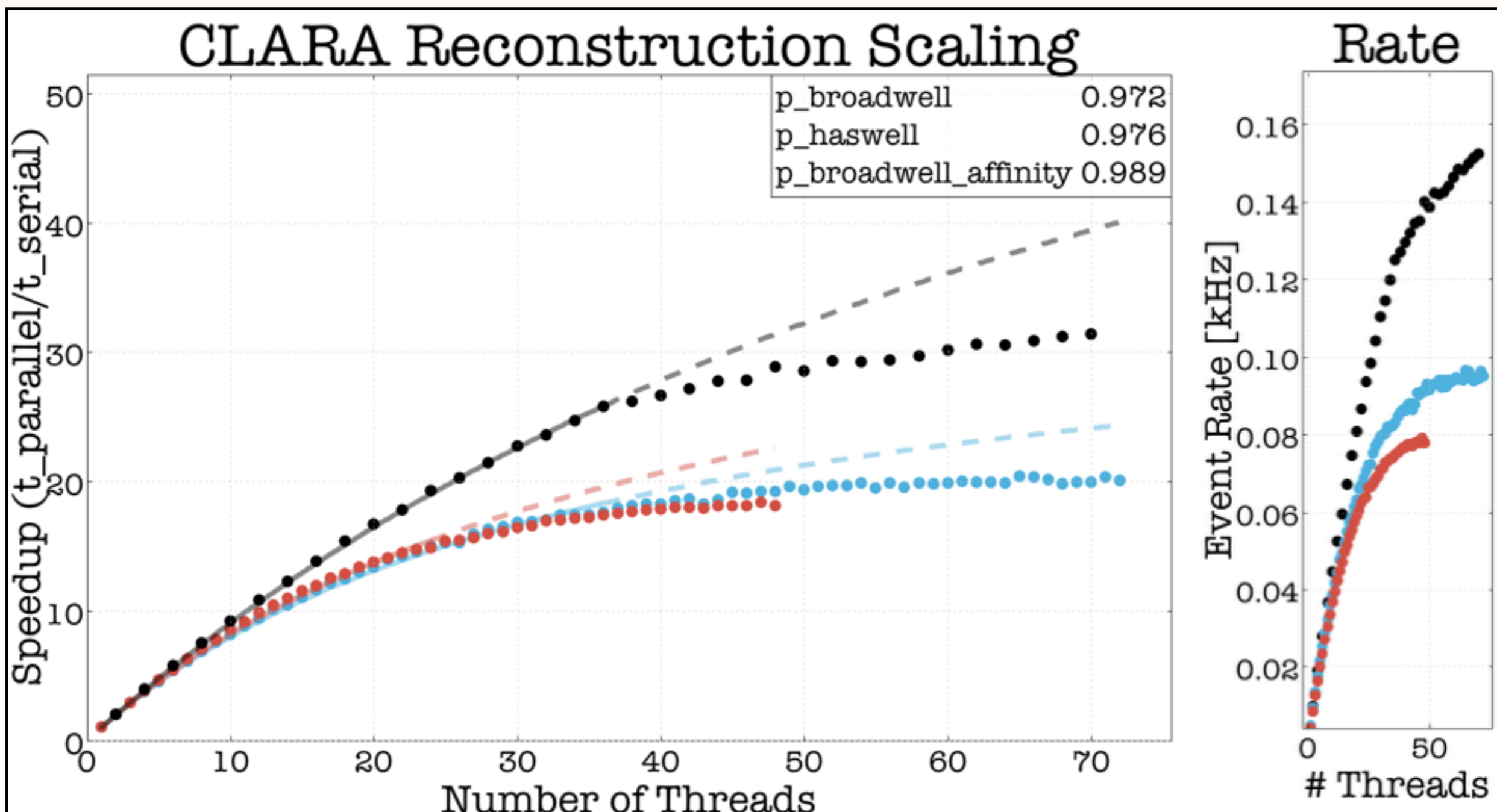
► Tools:

- ✓ interactive data format conversion tools (EVIO to HIPO)
- ✓ interactive analysis studio UI, allows low level tuple analysis
- ✓ data event viewer with bank filtering tools
- ✓ Raw data viewer with event decoder and ADC pulse visualization

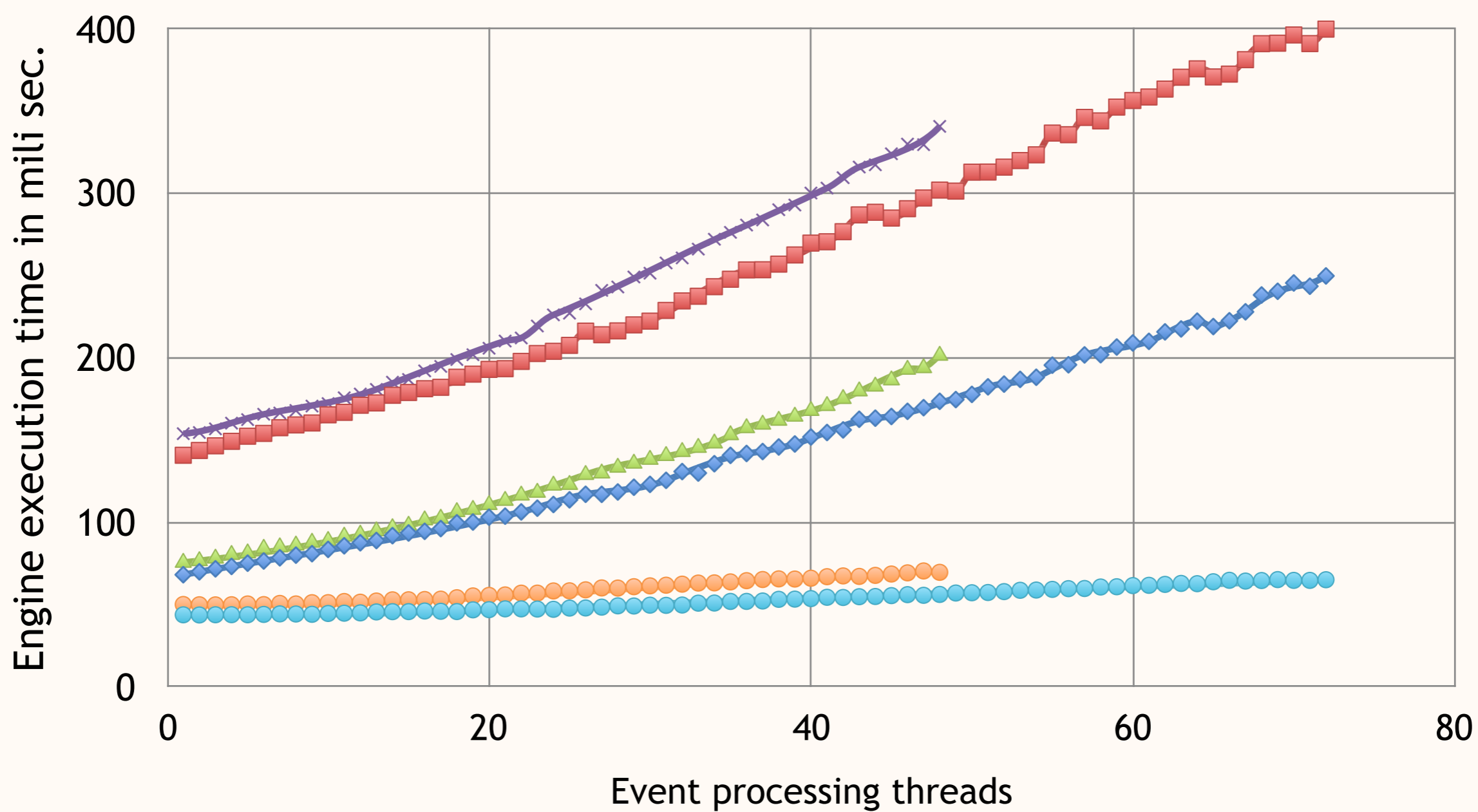
Amdahl's law

gives the theoretical speedup in latency of the execution of a task at fixed workload that can be expected of a system whose resources are improved.

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

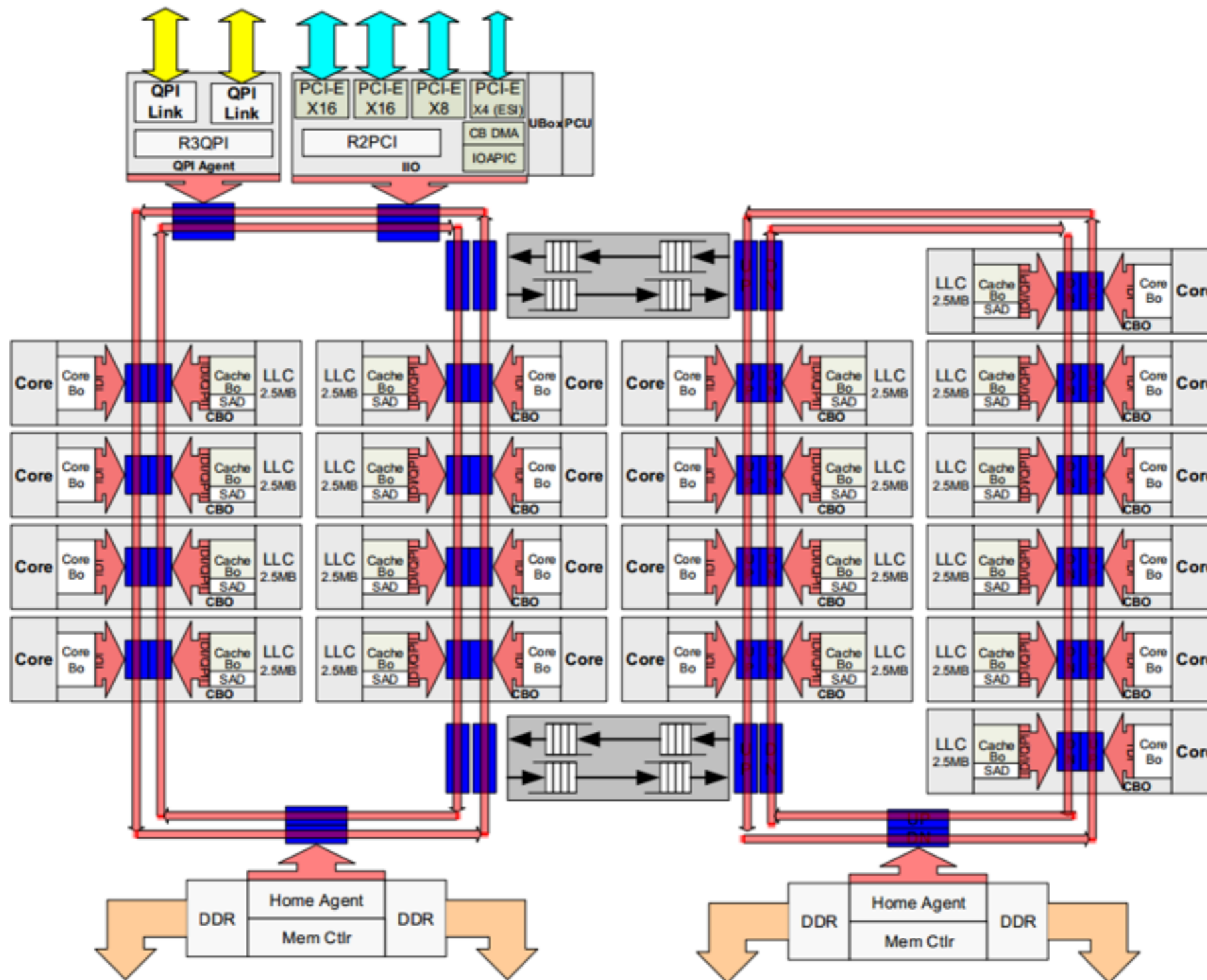


DCHB, DCTB and Factorial Engine Execution Times

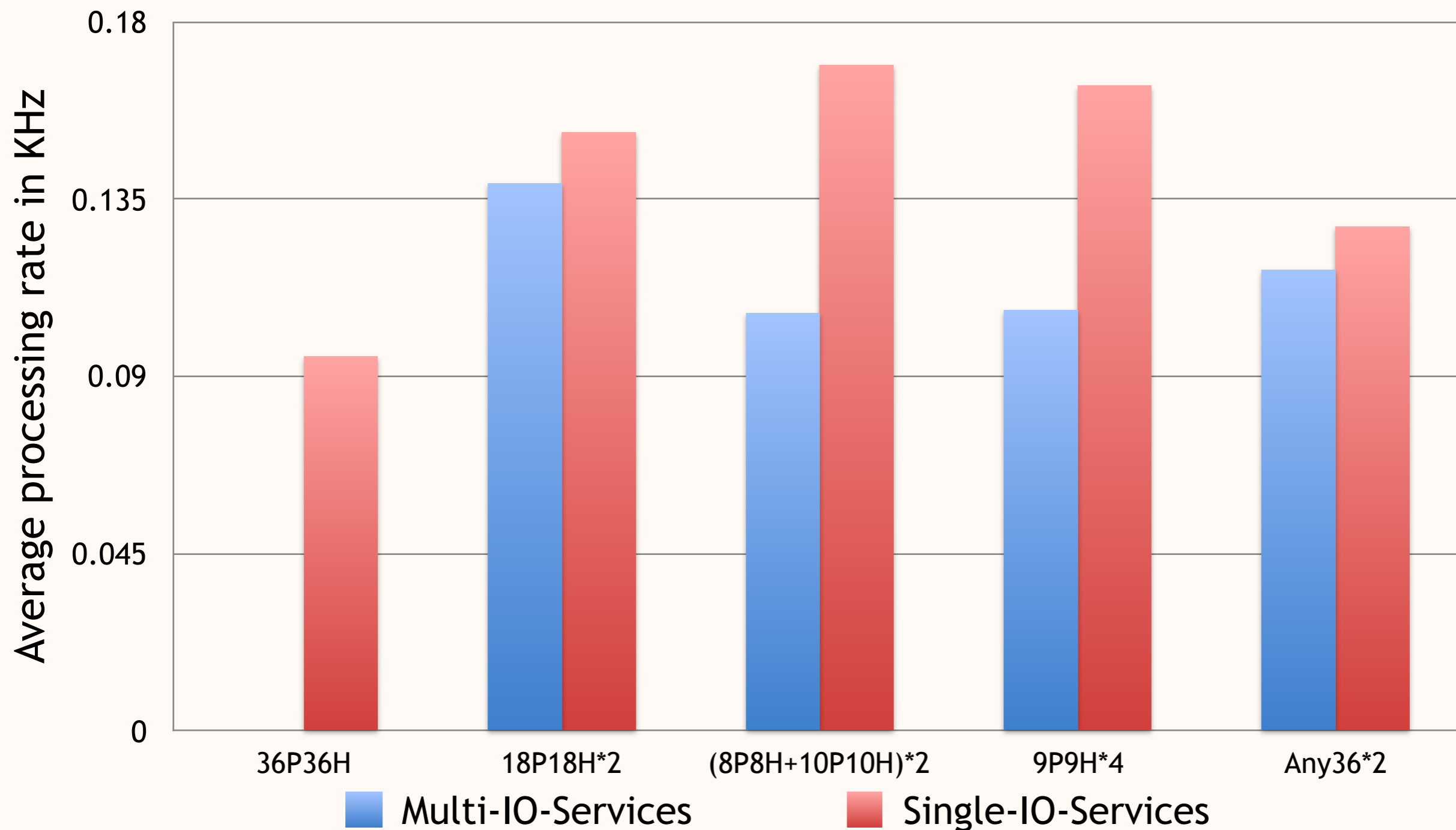


- ◆ DCHB-Broadwell
- DCTB-Broadwell
- ▲ DCHB-Haswell
- ✕ DCTB-Haswell
- Factorial-Broadwell
- Factorial-Haswell

14-18 Core (HCC)



CLAS12 Reconstruction Application Performance for Different CLARA Configurations



► Common Development Environment:

- common tools library aids users for fast application development
- unified interface for database constants access and comparison
- unified detector visualization (common detector enumeration scheme)
- efficient data format for storing DST and intermediate results
- reconstruction framework for multi-threaded engines for detectors

► Reconstruction:

- reconstruction software complete in CLARA 4.3 (with xMsg)
- multithreading tests show smooth vertical scaling (no thread contention)
- plugin driven multithreaded reconstruction package is in place
- all in one package includes magnetic maps, local database copy

• Software Distribution:

- github repository for the common CLAS12 package with build scripts
- detector packages (plugins) have separate github repositories
- versioned reconstruction package distributed via web download