# Effective End-to-End Management of Data Acquisition and Analysis for X-Ray Photon Correlation Spectroscopy

F. Khan*, J. P. Hammonds,
S. Narayanan, A. Sandy, N. Schwarz
Advanced Photon Source
Argonne National Laboratory, Argonne IL 60439 U.S

*Presenter

# End-to-End Management

Managing scientific operations from acquisition to analysis through automated software.

Faster turn-around time between acquisition and analysis.

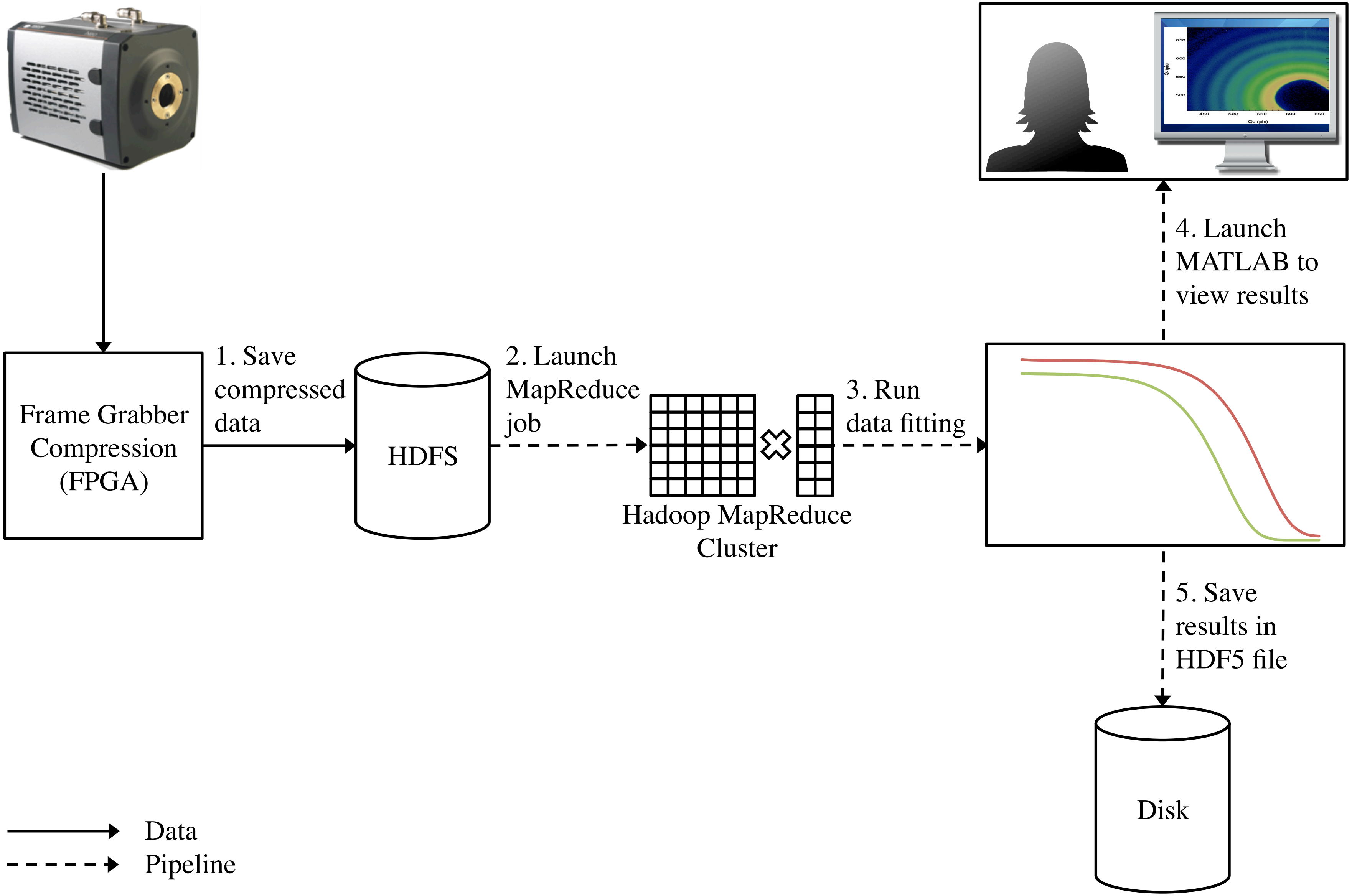
Operational efficiency of the scientific apparatus in a multi-user environment.

Combining components developed over time by different teams.

# Outline

- System overview
- Analysis pipeline
  - Acquisition
  - Analysis - MapReduce
  - Analysis - Data fitting
  - Data format (HDF5)
  - Automation
- Performance
- Conclusion

# System Overview



Frame Grabber Compression (FPGA)

1. Save compressed data

HDFS

2. Launch MapReduce job

Hadoop MapReduce Cluster

3. Run data fitting

4. Launch MATLAB to view results

5. Save results in HDF5 file

Disk

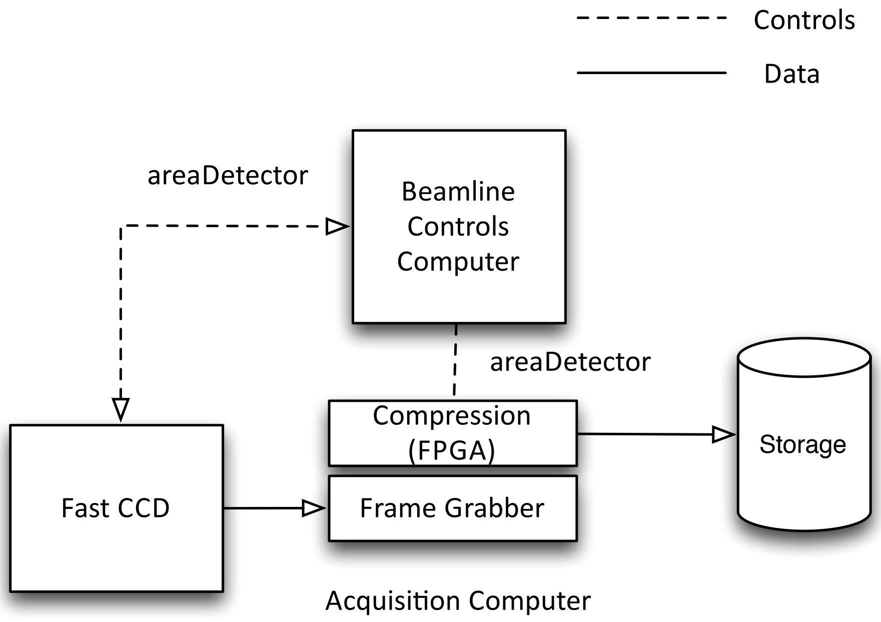
Data

Pipeline

# Acquisition

- CCD-based data acquisition system
  - Operating at 60 frames/s producing 128 MB/s raw data
- Hardware compression to reduce upstream bandwidth
  - Done using estimation of average dark signal and noise

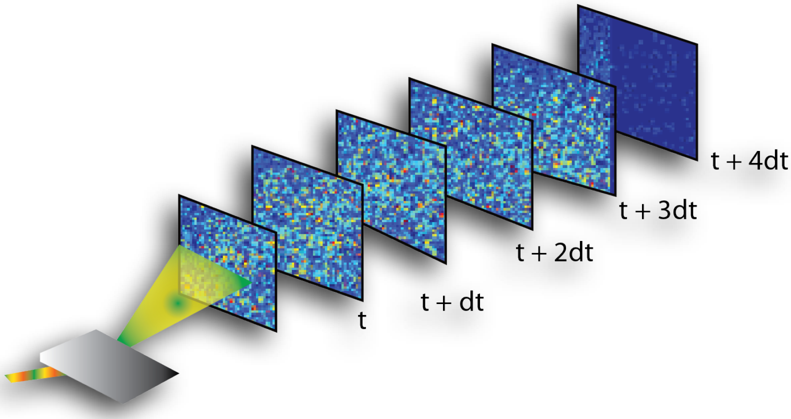
## Acquisition system overview[1]



[1] T. Madden et. al. "Firmware Lower-Level Discrimination and Compression Applied to Streaming X-ray Photon Correlation Spectroscopy Area-Detector Data," Rev. Sci. Instrum. 82. 075109 (2011).

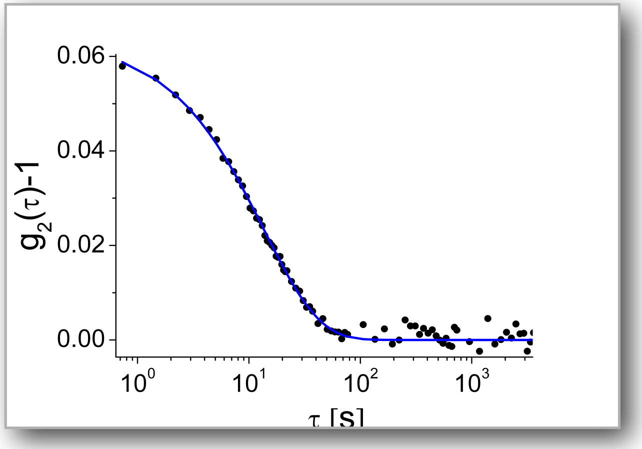
# Analysis

XPCS[1] Technique : Auto-correlation[2] of pixel intensities over different time (Tau) and length (q) scales



Computing **g**$_2$ : It includes computing correlations for every pixel and then normalizing them for different **Tau** and **q** values

$$g_2(\tau, q) = \frac{\langle I(t,q) I(t+\tau,q)\rangle}{\langle I(t,q)\rangle^2}$$



Auto-correlation is computation intensive but easy to parallelize problem

[1] XPCS is done at Sector 8-ID and have wider usage, e.g., I. M. Sikorski, A.R. Sandy, and S. Narayanan, ``Depletion-Induced Structure and Dynamics in Bimodal Colloidal Suspensions,'' Phys. Rev. Lett. 106. 188301-1 (2011).
[2] B. Tieman, S. Narayanan, A. Sandy and M. Sikorski, "MPICorrelator: A parallel code for performing time correlations," Nucl. Instrum. Method A 649(1), (2011) 240.

# MapReduce

Parallel processing of large datasets using larger number of computer nodes (cluster).

Computations based on “map” and “reduce” phase.

**map** - Data is split into smaller sub-problems usually with a unique identifier per problem.

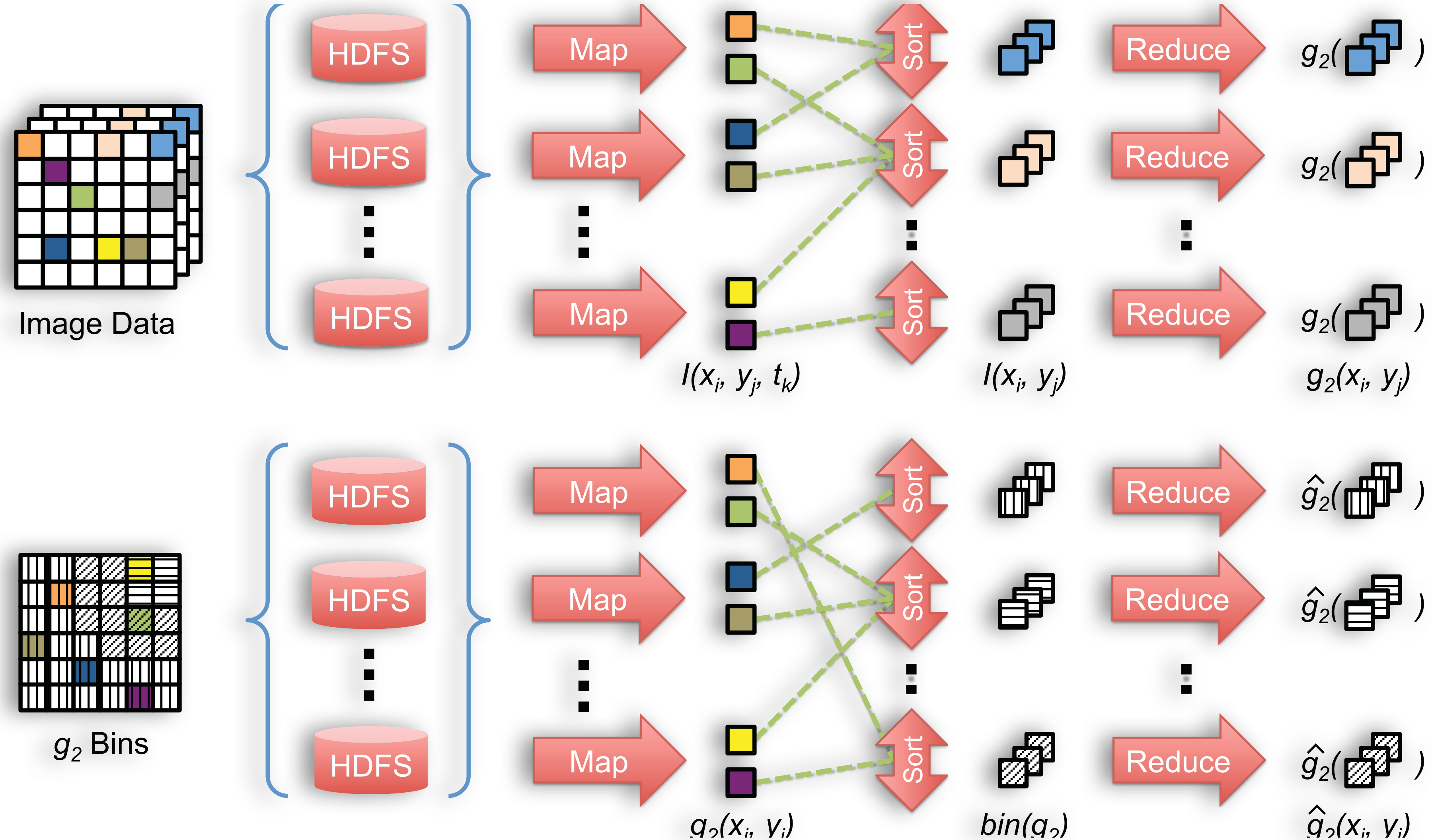
**reduce** - Output from individual sub-problems is combined to form the final result.

Apache Hadoop[1] provides an open source implementation of MapReduce[2] along with a distributed storage system.

[1] http://wiki.apache.org/hadoop

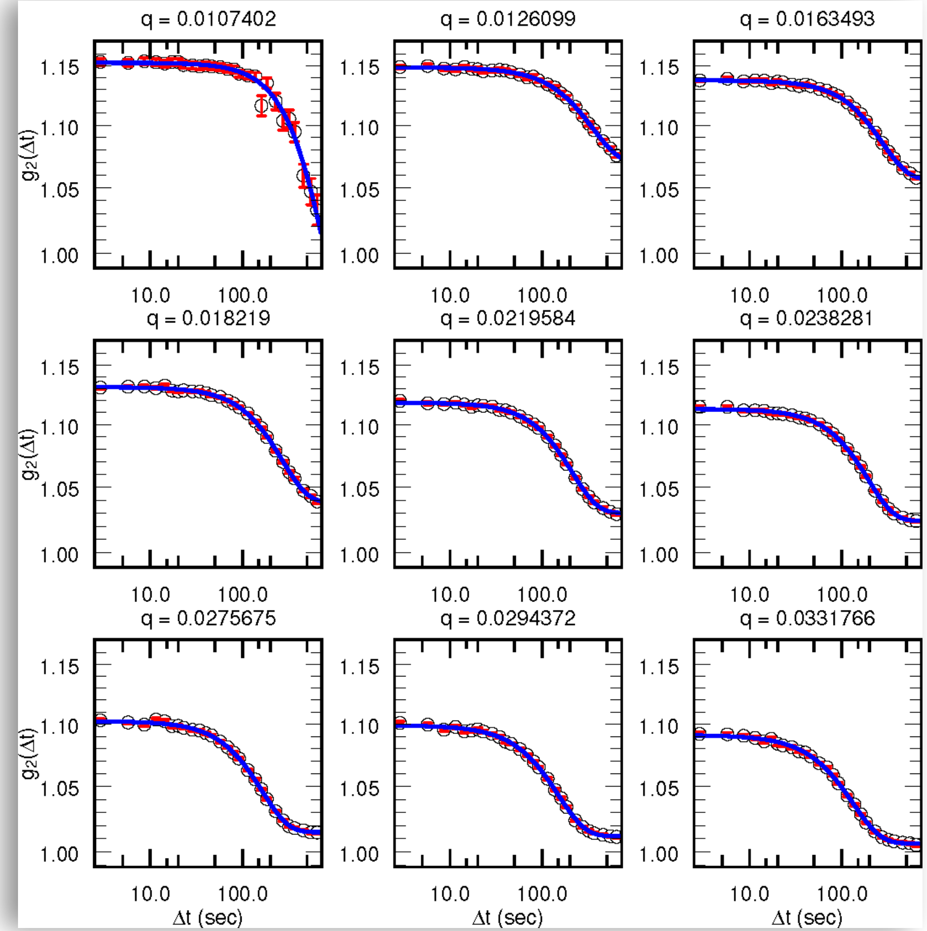
[2] Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” Communications of the ACM 51(1), (2008) 107.21.

# Analysis - MapReduce

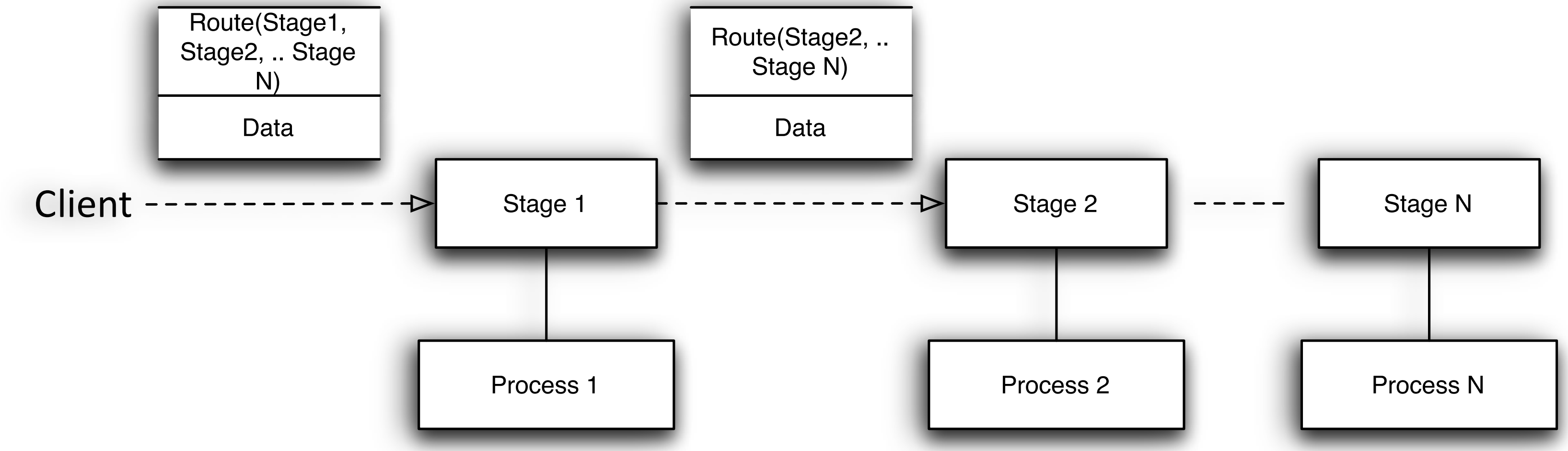


# Analysis - Data Fitting

- Least square fitting
  - Individual g2 (y-axis) values for different time (Tau, x-axis) and length (q) scale.
- Fitting is a done as a last stage in processing.
- Pipeline launches Matlab script on user's desktop for visualizing the results of fitting.



# Analysis Automation



A thin wrapper around user process

A common data format

Client starts the pipeline analysis by constructing a JMS message containing:

**input data**[1, 2] - A HDF5 file with input parameters, e.g., location of Hadoop file to process

**route** - specify individual processing step

Next stage is triggered by passing a JMS[3] message between the current and the next stage.

[1] Hierarchical Data Format version 5 (HDF5), 2000-2010. http://www.hdfgroup.org/HDF5.

[2] The Scientific Data Exchange, http://www.aps.anl.gov/DataExchange.

[3] Apache ActiveMQ, http://activemq.apache.org based implementation of JMS standard is used.

# Performance

Acquisition

- 60 Hz compressed data

Analysis - MapReduce

- 1-3 minutes for most datasets.
- Running multiple analysis can reduce this time even further.

Pipeline - Backend

- Based on ActiveMQ - an industry standard messaging backbone.
- Capable of dispensing a couple thousands of messages per second (depending upon the environment).

## MapReduce analysis results

| Size | Frames | Time |
|---|---|---|
| 45 GB | 120, 000 | 10 minutes |
| 11 GB | 20, 000 | 9 minutes |
| 0.8 GB | 80, 000 | 3 minutes |
| 0.8 GB | 20, 000 | 2.5 minutes |
| 0.4 GB | 4,000 | 1 minutes |

# Conclusion

Near-real-time analysis using

- Parallel analysis algorithms and
- A state-of-the-art messaging backend

Maximizing utilization of the equipment

Lowering turn-around time between acquisition and analysis

Future **challenge**

- Ability of the modern detectors to take images at **faster rate** and **higher resolution.**

# Thanks

Questions?