Trigger Integration Tools

Kurt Biery
ADSS Department
Fermilab Scientific Computing Division
04-Feb-2013

Online Data Streaming, Filtering

Large collider experiments traditionally have one or more levels of hardware trigger and a computer farm to perform event reconstruction, analysis, and filtering in software.

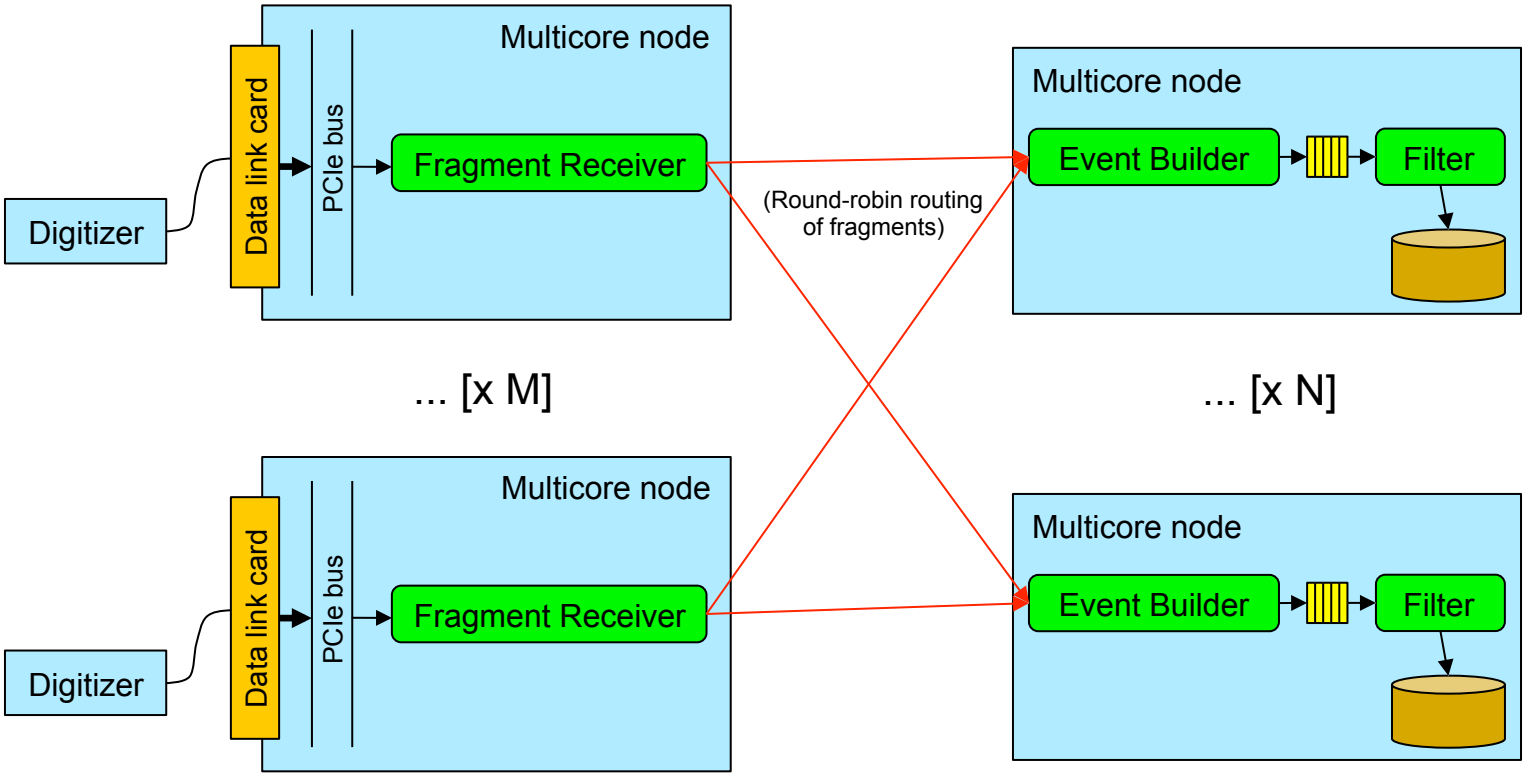
- Triggering – a fast analysis on a subset of the full event data that selects events of interest and triggers the readout of the full event data.
- Filtering – analysis of complete events to select the ones of interest and discard the rest.

With current commercial components (ADCs, PCIe cards, PCs, networks), the need for a hardware trigger is no longer a necessity for some experiments.

In these cases, it is possible, and desirable, to stream all of the data into a farm of commodity computers and run software algorithms to filter the data.

Examples include NOvA, MicroBooNE, DarkSide-50, Mu2e.

Generic DAQ



Lots of variations:

- multiple fragment receivers per front-end node
- multiple event builder/filter process pairs per reconstruction node
- multiple filter processes per event builder
- parallel processing inside the reconstruction processing
- everything run on a single node

artdaq

artdaq is a toolkit for creating DAQ systems to be run on commodity servers, including the core functionality of high performance event building and event filtering.

- A set of ready-to-use components along with hooks to support experiment-specific customization.
- Integrated with the *art* event reconstruction and analysis framework for event filtering and data compression.

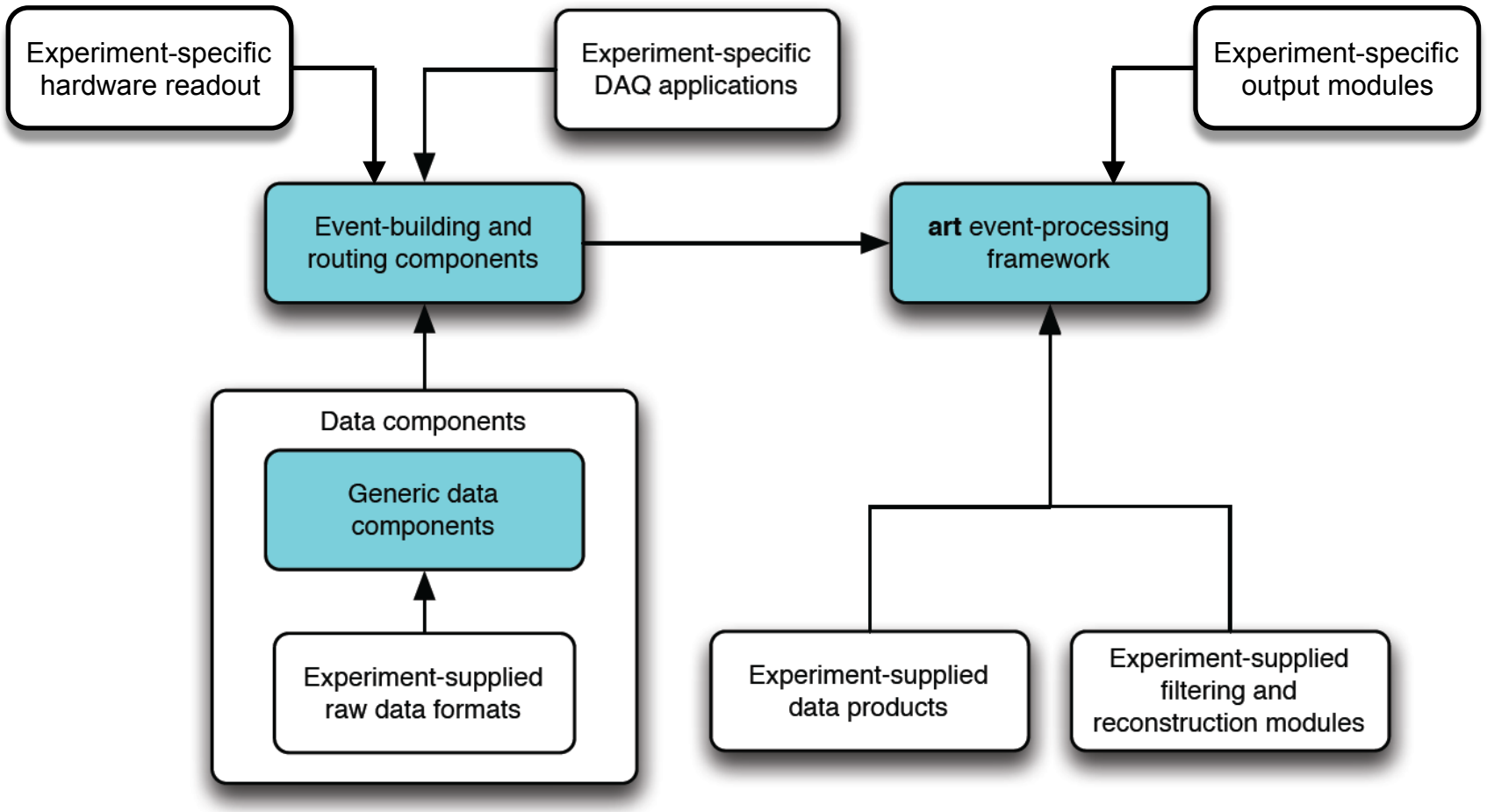
The core functionality was designed and developed by Jim Kowalkowski, Marc Paterno, and Chris Green. Steve Foulkes, myself, Paul Russo, Chris, Marc, Jim, and others are extending the toolkit for use on current and future experiments.

Marc presented *artdaq* at the Real-Time 2012 conference last June, and his paper has been accepted for publication in the IEEE Transactions on Nuclear Science.

Initial *artdaq* Goals

- Support the use of commodity computers as close to the data collection as possible.
- Make efficient use of multi-core computers.
- Take advantage of high-speed networking and hardware buses.
- Support modular algorithms, enable the use of GPGPUs.
- Enable collaborators to contribute to online code development.
- Allow for concurrent processing of events to best utilize all of the cores on a node.
- Support easy system reconfiguration.
- Allow for similar or identical algorithms to be used offline and online.
- Provide an environment for R&D tasks.
- Provide a springboard for DAQ development in future experiments.

Initial Architecture

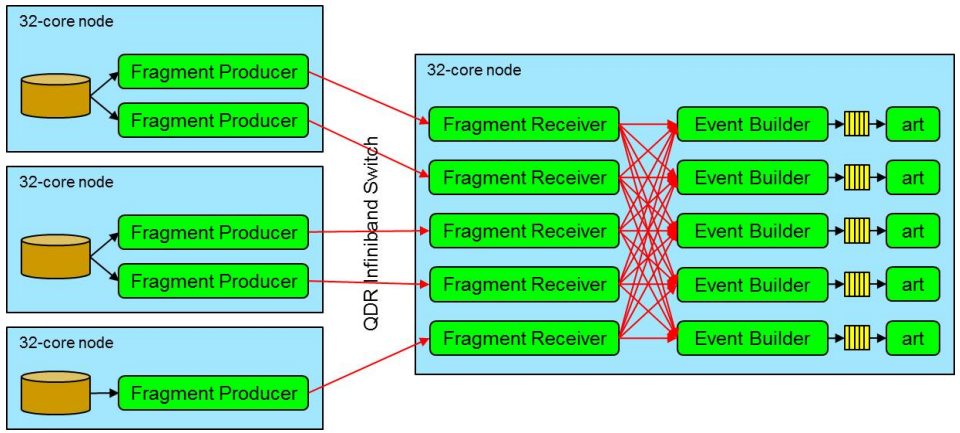


Some Details

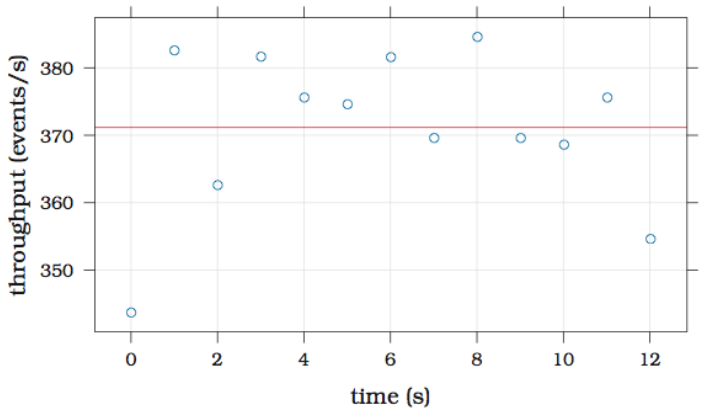
From Marc's paper: "*artdaq* makes use of the Message Passing Interface (MPI) to create a multi-process, potentially distributed, event-building program. The use of MPI allows us to take advantage of high-performance network drivers written for the supercomputing community. We also obtain the flexibility of being able to move different computational tasks to different nodes with just a change in our configuration scripts, and with no need to recompile the application."

Classes were developed to handle the sending and receiving of data fragments over MPI, the collection and assembly of fragments into full events, for queuing events for *art*, and for running *art* itself inside the EventBuilder process.

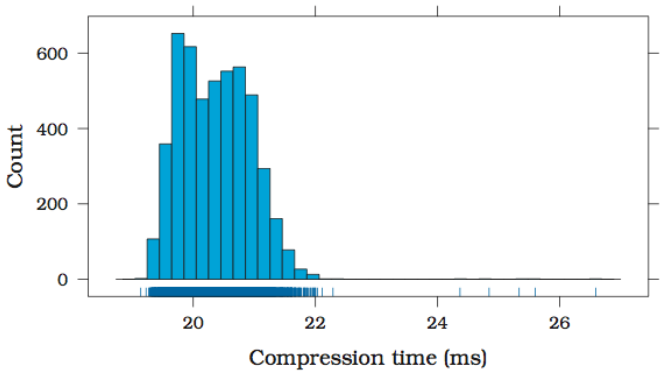
Performance Studies

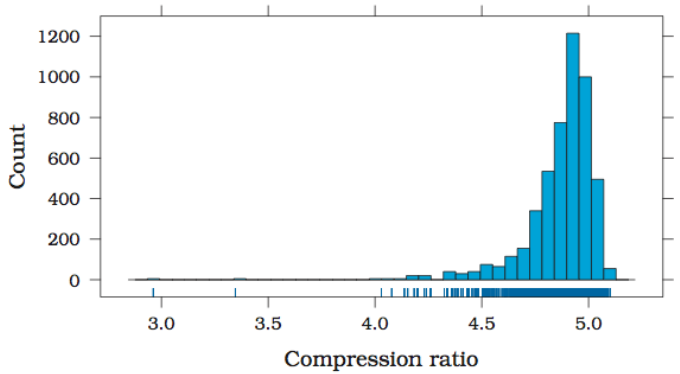


To test the performance of a candidate compression algorithm for DarkSide-50, a cluster of four 32-core nodes connected by a QDR Infiniband switch was used. In this test, the detector electronics was simulated by a Linux process that read DS50 fragment data from disk. In this test, all five Fragment Receivers, all five Event Builders, and all five *art* reconstruction applications (with 5 threads – one per fragment) were run on the same host to make use of all of the 32 cores on the machine.



Throughput results, with no compression, as a function of time during the test. In the test, the event size was 6 MB, so the average throughput data rate was 2.2 GB/s.





Performance results from a test of a candidate Huffman compression algorithm. The compression time results are per event per *art* process, so the overall measured rate was ~250 events per second.

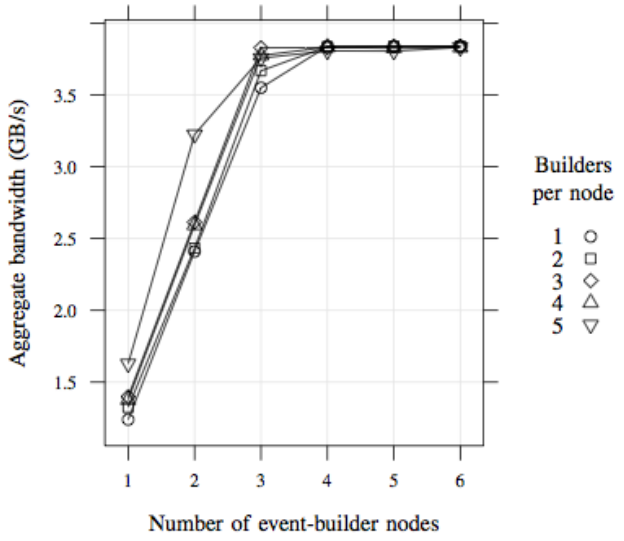
Additional Performance Results

Mu2e-like system

- In Mu2e, data will be received directly into the processor farm PCs, and the event building will be done between those PCs.
- Using a test cluster of five 32-core nodes connected by an InfiniBand QDR network, we measured the throughput of configurations where a FragmentReceiver process and an EventBuilder process ran on each node. The result was ~730 MB/s per node.
- To handle the 30 GB/s needed for Mu2e, this would translate into ~42 nodes (a reasonable number).

DarkSide-50 system

- When the DarkSide-50 DAQ machines were running here at Fermilab (before being shipped to LNGS), tests were run in which a FragmentReceiver process produced simulated data on each of 4 front-end computers. Studies were done to see how many EventBuilder processes were needed to handle the rate from the 4 FragmentReceivers. The results are shown in the graph on the right.



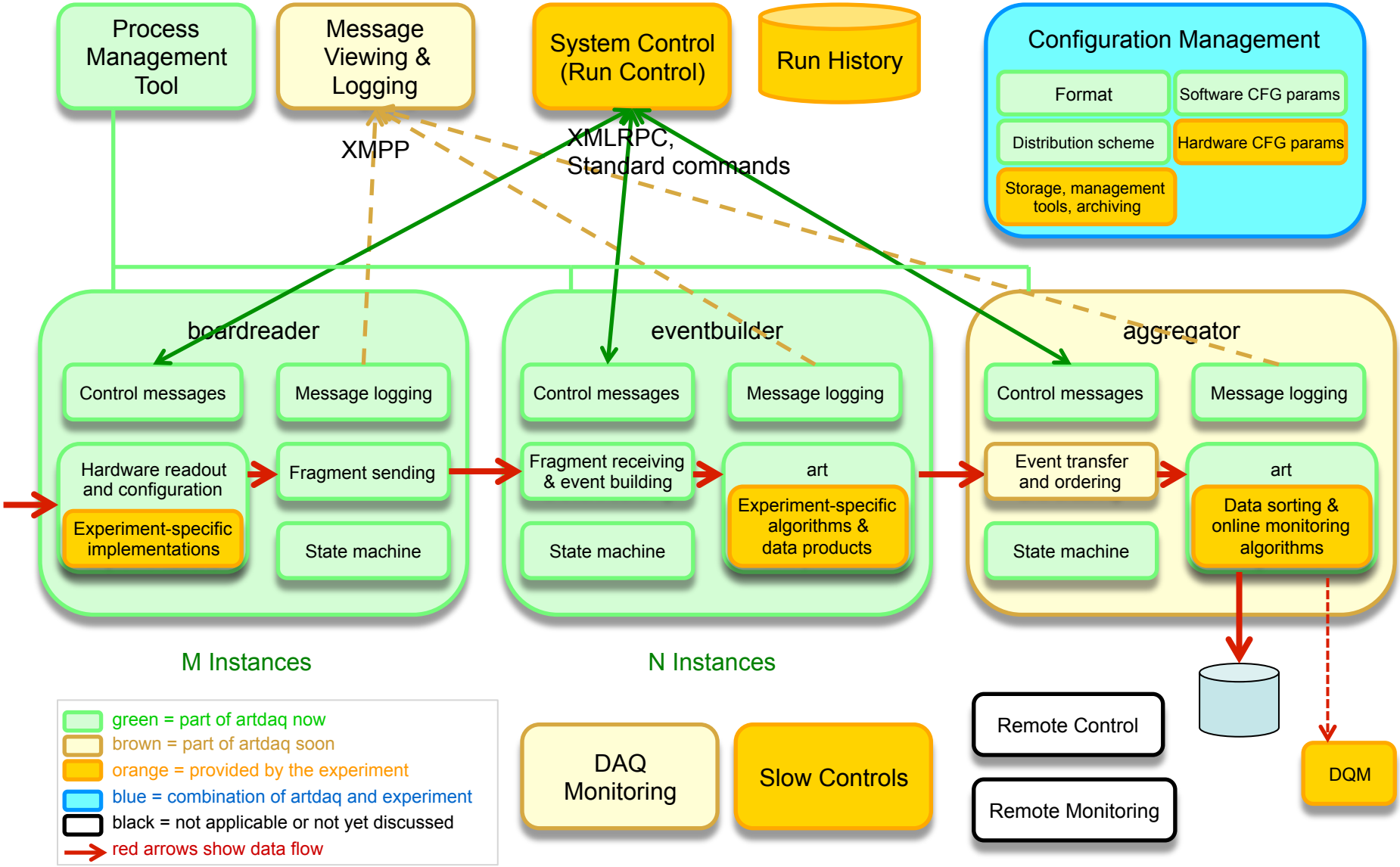
Current Work

We're developing additional DAQ functions for *artdaq* in preparation for it being used as part of the data acquisition of the DarkSide-50 experiment at Gran Sasso Lab.

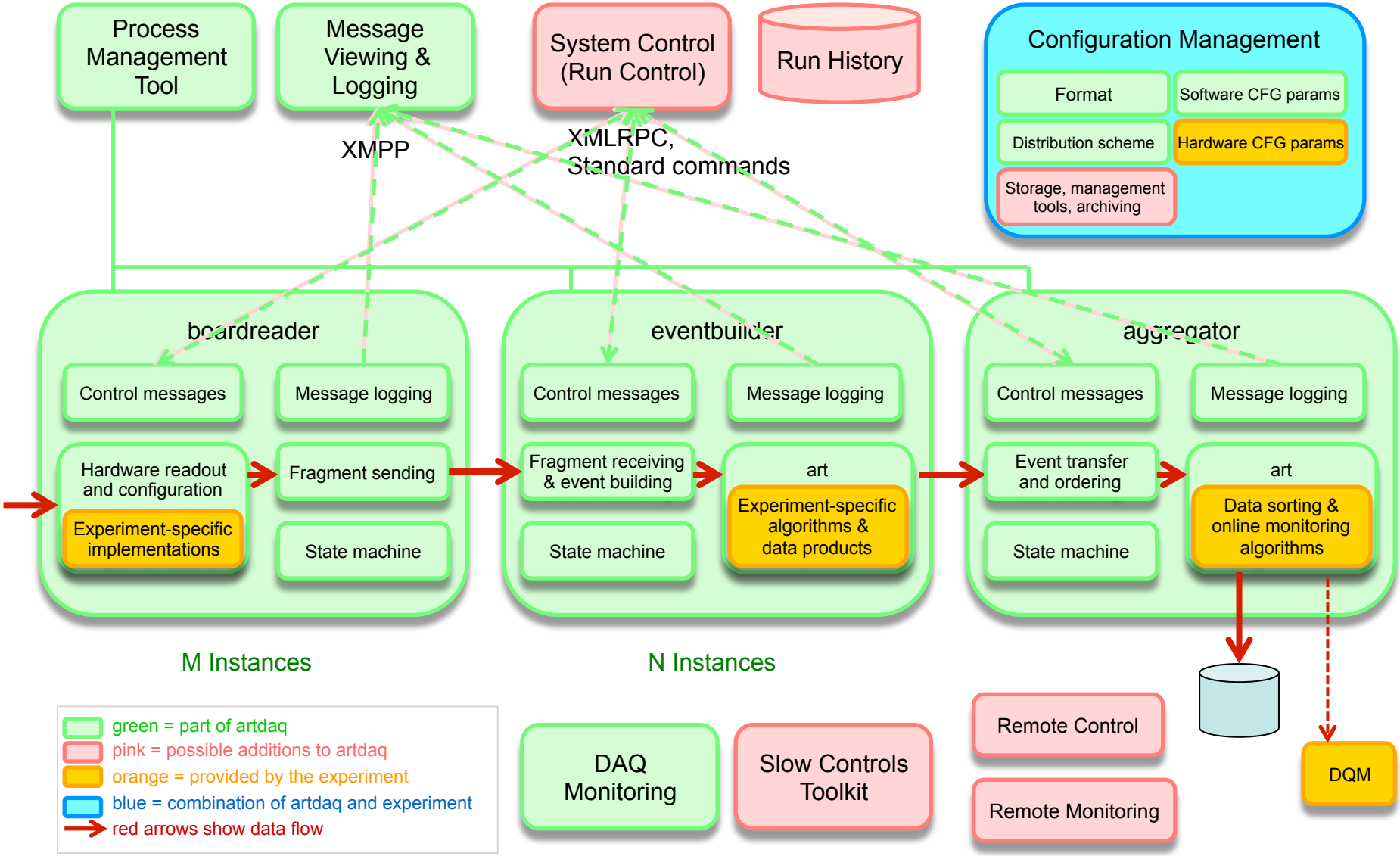
DarkSide-50 is a dark matter search located at the Laboratori Nazionali del Gran Sasso (LNGS) in Italy.

- Peak DAQ data rates in the range of 600 MB/s.
- Desired compression factor of 5+.
- The installation and commissioning of the detector will begin this spring.

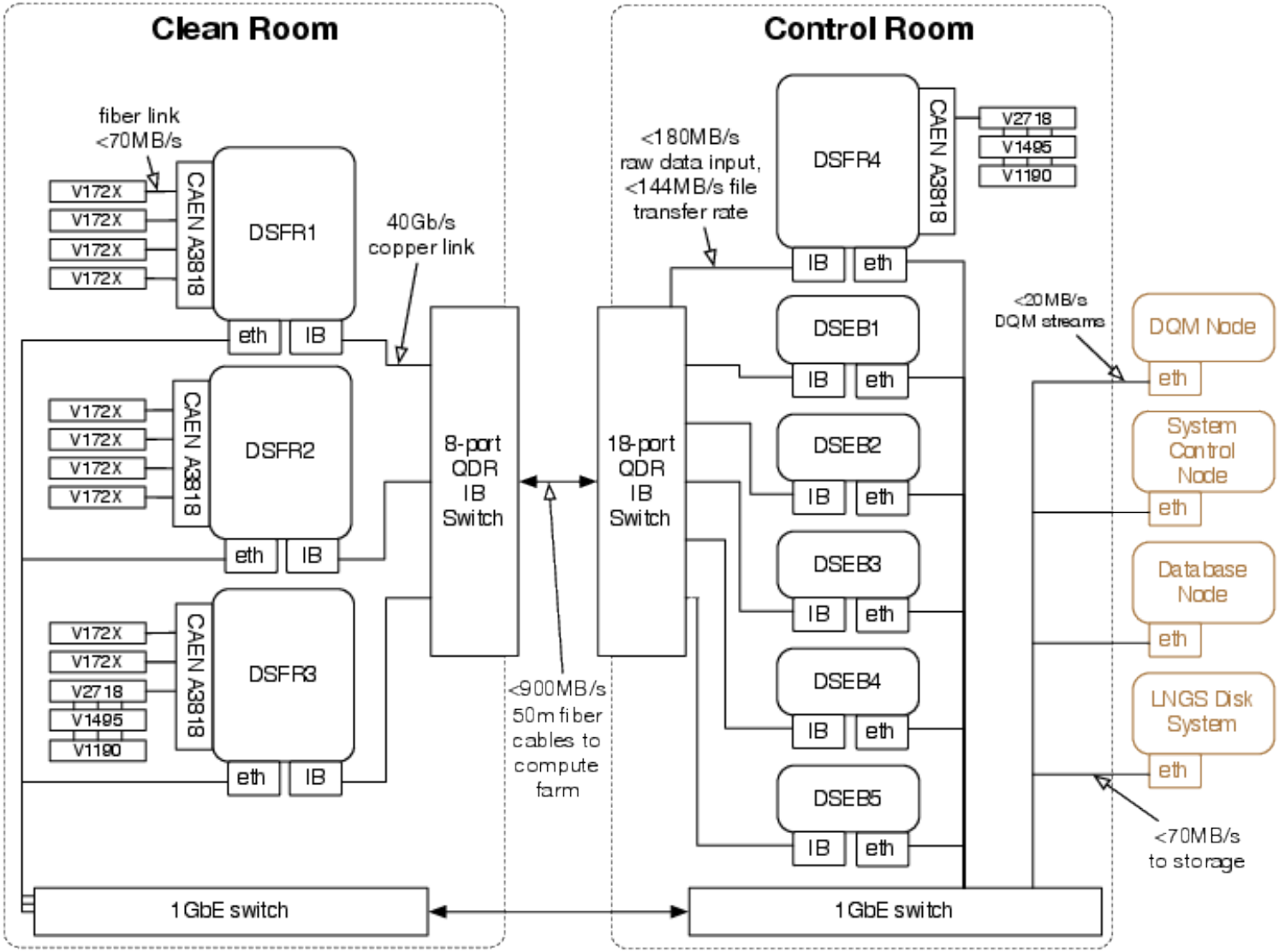
artdaq for DarkSide-50



Future *artdaq* Visions



DarkSide-50 DAQ Hardware



Current and Future Uses

NOvA Data-Driven Trigger:

- Using a couple of components from *artdaq* to help interface the existing BufferNodeEVB process to *art, art* processes have been tested and will soon be run regularly on the buffer nodes to provide data-driven triggering. The overhead from the framework has been measured to be less than 5% of the time budget.

DarkSide-50:

- Described earlier.

Mu2e:

- *artdaq* will be used for initial DAQ development and testing with an eye toward using it in the full DAQ. (The current baseline design uses FPGAs to do the event building, but *artdaq* should provide a more cost-effective solution.)

We are talking with other future experiments to see if *art* and *artdaq* can meet their data acquisition and filtering needs.

DarkSide-50 Software Architecture