

# Update on the *art* framework

---



**Fermi National Accelerator Laboratory**

 Office of Science / U.S. Department of Energy

Managed by Fermi Research Alliance, LLC

**Marc Paterno**

Scientific Computing Division

*for the art team*

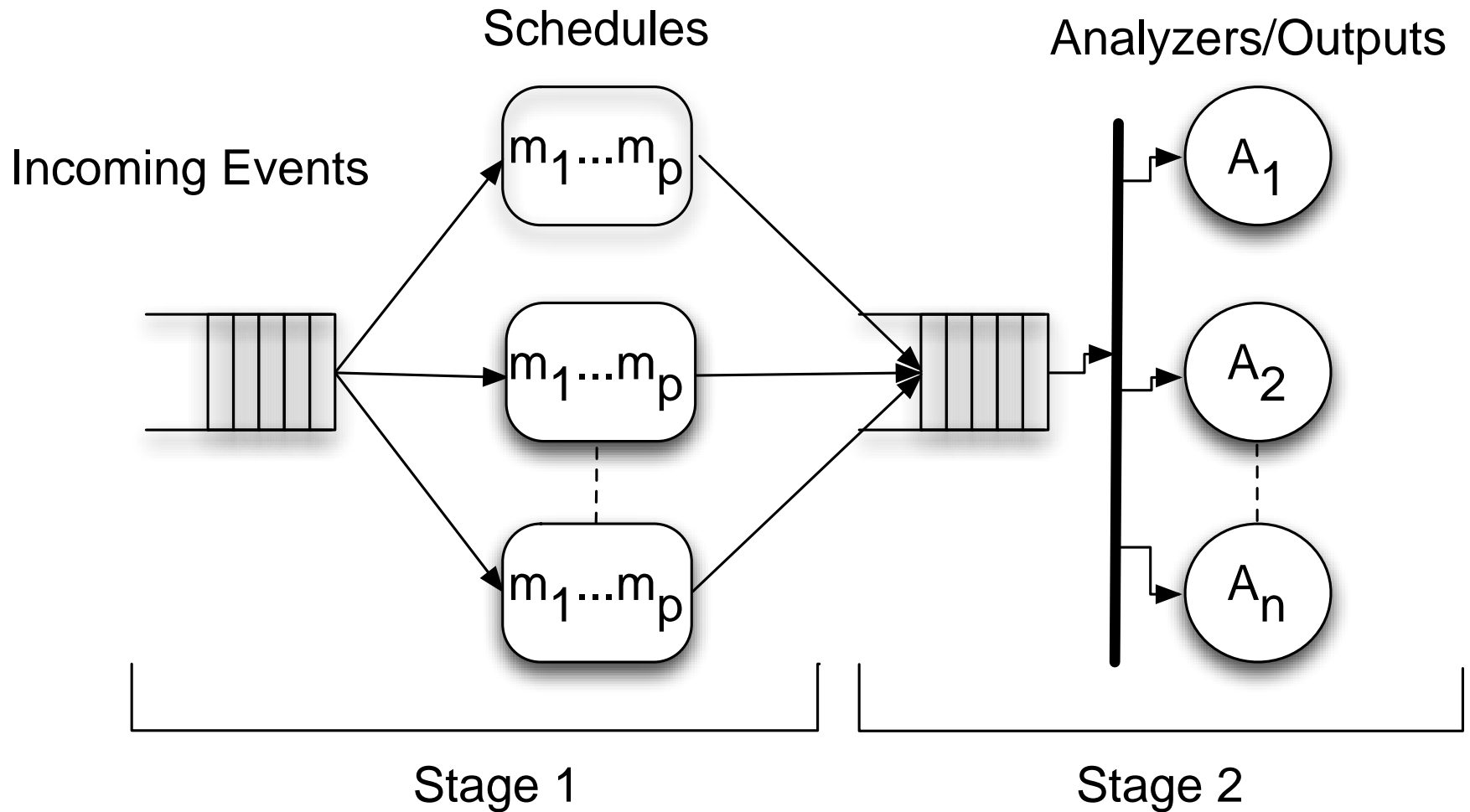
## Our context

- *art* is the framework used by several experiments and efforts at Fermilab and elsewhere.
  - NOvA, muon g-2, Mu2e, microBooNE, LBNE, DarkSide-50
  - Various other Liquid Argon activities, through LArSoft
- It is an offshoot of the CMSSW framework, packaged as an “external product”, targeted for DAQ and offline use by many small collaborations.
- Both *art* and the CMSSW framework are now developed by members of the same department at Fermilab.

# Multi-event parallelism support

- We implemented high-level coordination appropriate for the event processing tasks in the experiments we support.
- Hybrid parallelism scheme: 2-stage pipeline
  1. serial trigger-path module operation for multiple events simultaneously
  2. completed events queued for processing one at a time; each event processed by multiple analyzers and output modules in parallel.

# The 2-stage pipeline



## Multi-event parallelism support

- We implemented high-level coordination appropriate for the event processing tasks in the experiments we support.
- Hybrid parallelism scheme: 2-stage pipeline
  1. serial trigger-path module operation for multiple events simultaneously
  2. completed events queued for processing one at a time; each event processed by multiple analyzers and output modules in parallel
- In addition, any module may invoke task-based parallelism at any time using TBB coordinated across the framework execution instance.

# Offloading algorithms to Xeon Phi

- The Mu2e DAQ system needs high-performance tracking for their primary event filter. We are investigating the use of Xeon Phis for this task.
- We have targeted the use of Intel TBB in “offload” mode. This requires building the code to be offloaded with the Intel C++ compiler. *art* uses much modern C++, and the Intel compiler doesn’t understand all of it.
- We have enhanced the *art* build system to support *hybrid compilation* with ICC and GCC
  - TBB library and high-level “user” module package compiled with Intel 14.0.2 over GCC 4.8.2
  - *art* infrastructure compiled with GCC 4.8.2
  - allows us to build our code, using as-shipped *art* framework libraries and executables.
- We are able to run *art* modules that offload algorithms to the accelerator, using TBB. “Fallback” operation using TBB on the multi-core in the absence of an accelerator has also been achieved, without alteration of source code or recompilation.
- Four Intel C++ compiler bugs have been reported back to Intel (with full reproduction instructions). One is already in the queue for release with Intel C++ 14.0.3 and the others are all “with engineering”.

## Multi-threading support for *art* services

- Parallel event processing requires controlled access to global resources (services).
- Services may be
  - truly global (provider gives thread safety guarantee)
  - private to a single schedule (state synchronized for the one event the schedule is processing)
- In our 2-stage pipeline system, context management allows the correct service to be provided when a *ServiceHandle* is used.
- Service callback system upgraded to C++11; signals are also classified as either global or per-schedule.

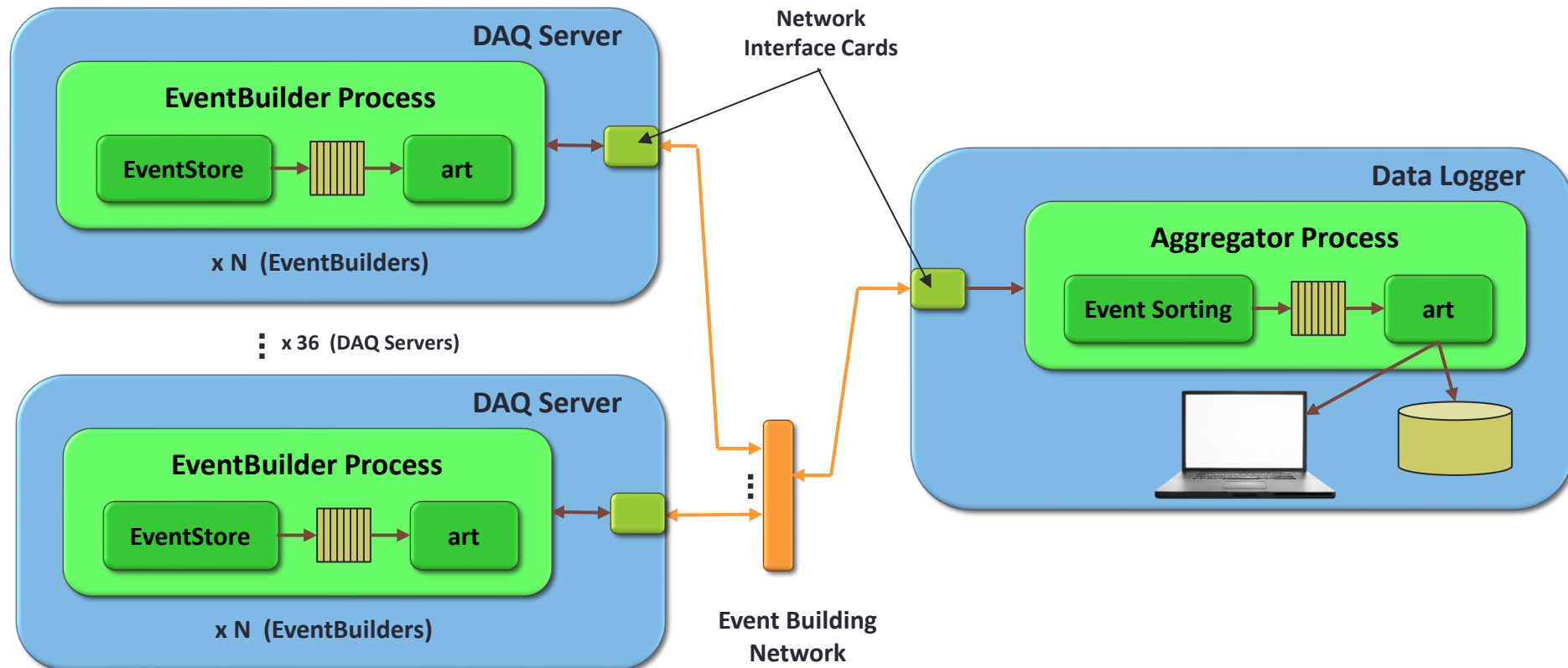
## Delivery status

- All changes (infrastructure, internal reorganization and new features) have been applied to the as-shipped code to lessen conversion shock and risk of conflict with development not related to parallelism.



# art use in a data acquisition system

- *art* is an integral part of multi-process MPI *artdaq* application
- streaming I/O modules move data without writing to disk



## Ongoing directions

- Concentrate on allowing algorithm development to use all available resources, accelerator or multi-core.
- Allow easy integration of the algorithms into *art* framework applications through our build system.
- Allow *art* to be readily incorporated into higher-level systems, using HPC technologies to take advantage of modern hardware.