Orchestration of Streaming Data Processing for ePIC

Torre Wenaus

Nuclear and Particle Physics Software (NPPS) Group Leader, BNL Physics Department

Streaming Readout Workshop SRO-XII, University of Tokyo

December 3 2024

Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

@BrookhavenLab

Thank you for the invitation to present on this topic!

- About me: head of a group of 25 working on scientific software across 9 NP and HEP experiments and projects in the BNL Physics Department's program
- Personally mostly ATLAS until recently, distributed computing and HL-LHC computing R&D
- EIC/ePIC more recently, deputy to Markus for infrastructure
    - Encompassing distributed computing and the streaming computing model
- This talk:
    - Experimental context is ePIC
    - First an overview, hopefully complementing Markus, of the ePIC computing model & requirements aspects that motivate and set the context for distributed computing and streaming data orchestration
        - What I present is grounded in ePIC plans developed thus far, but don't take the talk as ePIC plan!
        - ePIC discussions on (part of) this content are only months old
    - Then thoughts from me on streaming data orchestration proper
    - The context I am coming from is
        - the workload/workflow systems developed in my group, for ATLAS and others
        - the event streaming capabilities we developed for fine grained processing, for ATLAS and recently for others
    - I will use our system, PanDA, as example context for implementing ePIC streaming orchestration
- ePIC is expected to evaluate workload management systems in 2025
    - PanDA is preparing for that, and I will touch on the preparations
- Distributed computing is also data management, with ePIC having chosen Rucio (also from ATLAS)
    - PanDA + Rucio were developed to orchestrate workload/data management together
    - In this talk I'll focus on workflow management, Rucio entering where the orchestration happens

The ePIC Streaming Computing Model
Version 2, Fall 2024

Marco Battaglieri[1], Wouter Deconinck[2], Markus Diefenthaler[3], Jin Huang[4], Sylvester Joosten[5], Dmitry Kalinkin[6], Jeffery Landgraf[4], David Lawrence[3] and Torre Wenaus[4]
for the ePIC Collaboration

[1]Istituto Nazionale di Fisica Nucleare - Sezione di Genova, Genova, Liguria, Italy.
[2]University of Manitoba, Winnipeg, Manitoba, Canada.
[3]Jefferson Lab, Newport News, VA, USA.
[4]Brookhaven National Laboratory, Upton, NY, USA.
[5]Argonne National Laboratory, Lemont, IL, USA.
[6]University of Kentucky, Lexington, KY, USA.

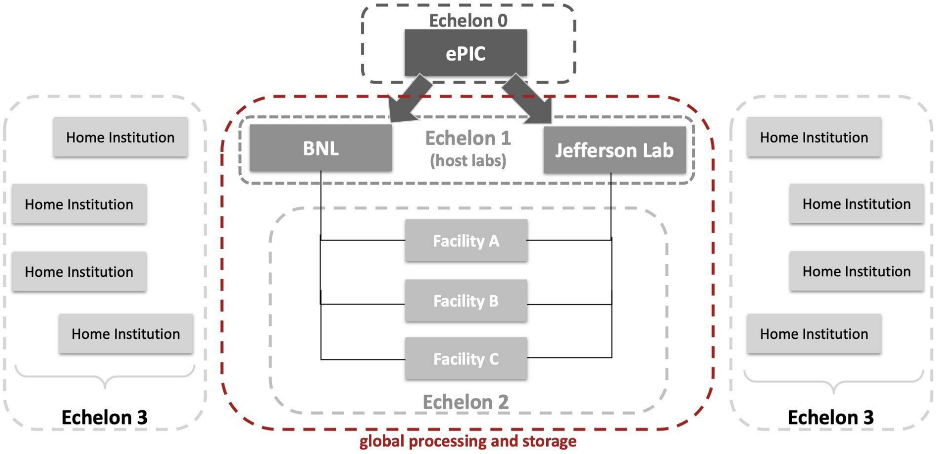
Abstract

This second version of the ePIC Streaming Computing Model Report provides a 2024 view of the computing model, updating the October 2023 report with new material including an early estimate of computing resource requirements; software developments supporting detector and physics studies, the integration of ML, and a robust production activity; the evolving plan for infrastructure, dataflows, and workflows from Echelon 0 to Echelon 1; and a more developed timeline of high-level milestones. This regularly updated report provides a common understanding within the ePIC Collaboration on the streaming computing model, and serves as input to ePIC Software & Computing reviews and to the EIC Resource Review Board. A later version will be submitted for publication to share our work and plans with the community. **New and substantially rewritten material in Version 2 is dark green. The present draft is preliminary and incomplete and is yet to be circulated in ePIC for review.**

Version 2 of the ePIC streaming computing model report is in progress

Context: The ePIC streaming computing model

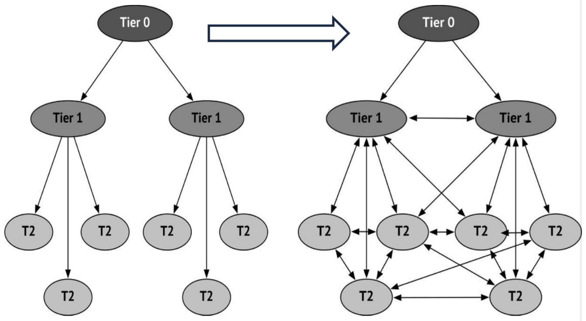
- The EIC's unique two-host-lab organization motivates the 'butterfly' computing model in which BNL and JLab are symmetric peers in their capability for post-Echelon 0 processing
- Globally distributed community and compute
- The streaming computing model report highlights the importance of distributed computing for ePIC
  - Quantitative estimates of computing needs
    - ePIC is compute-intensive
  - Echelon 0 - Echelon 1 data streaming & workflows
    - Complex workflow orchestration as soon as data arrives at Echelon 1s
- ePIC can draw on existing experience and tools from LHC & elsewhere, while addressing the unique aspects of its streaming computing model



Lesson from LHC: Mesh outperforms hierarchy

Fully interconnected facilities flexibly and efficiently serve many roles

A hierarchy is *more complex* and less efficient & flexible



Echelon 0: ePIC experiment, DAQ system

Echelon 1: Two host labs, two primary ePIC computing facilities

Echelon 2: Global contributions leveraging commitments to ePIC computing from universities and labs domestically and internationally

Echelon 3: Supporting the analysis community 'where they are' at their home institutes, primarily via services hosted at E1s/E2s

Computing use cases and their Echelon distribution

Use Case	Echelon 0	Echelon 1	Echelon 2	Echelon 3
Streaming Data Storage and Monitoring	✓	✓		
Alignment and Calibration		✓	✓	
Prompt Reconstruction		✓		
First Full Reconstruction		✓	✓	
Reprocessing		✓	✓	
Simulation		✓	✓	
Physics Analysis		✓	✓	✓
AI Modeling and Digital Twin		✓	✓	

Prompt = rapid low-latency processing

Prompt processing of newly acquired data typically begins in seconds, not tens of minutes or longer

Assumed Fraction of Use Case Done Outside Echelon 1	
Alignment and Calibration	50%
First Full Reconstruction	40%
Reprocessing	60%
Simulation	75%

- **Echelon 1s uniquely perform the low-latency streaming workflows consuming the data stream from Echelon 0**
  - Archiving, monitoring, prompt reconstruction, rapid diagnostics
- There's been some discussion over whether Echelon 2s have a role in low-latency streaming processing
  - We should not exclude it in the design, but doing it in year 1 is very unlikely, should ensure it's a supported option later
  - LHC experiences suggest caution, and strong motivation before undertaking it
  - e.g. ATLAS supports it when important to mitigate tight Tier 0 resources, but it is highly complex and operationally demanding
- **The conservative approach is to ensure the E1s have sufficient processing power for the low-latency workflows**
- Apart from low-latency streaming, Echelon 2s are full participants in the use cases
  - Our (new, tentative, approximate) resource requirements model assumes a substantial role for Echelon 2

Computing resource needs and the implications

- cf. Markus' talk for how we got the numbers
- Focus here is on the messages
  - O(1M) core-years to process a year of data is above ATLAS scale today
  - Optimistic scaling of constant-dollar performance gains would reduce the numbers about 5x
    - Based on current LHC measure of 15% per year
    - But the trend is towards lower gains per year
  - Whatever the gains over time, the processing scale is substantial
    - Motivates attention to leveraging distributed and opportunistic resources from the beginning
  - ~400PB/yr storage scale is also above ATLAS scale today
    - Archival storage (probably tape) will play a role
    - Motivates attention to efficiently orchestrating workflows consuming tape-resident data

Estimated needs in 2034 for ePIC Phase I nominal year

Processing by Use Case [cores]	Echelon 1	Echelon 2
Streaming Data Storage and Monitoring	-	-
Alignment and Calibration	6,004	6,004
Prompt Reconstruction	60,037	-
First Full Reconstruction	72,045	48,030
Reprocessing	144,089	216,134
Simulation	123,326	369,979
Total estimate processing	**405,501**	**640,147**

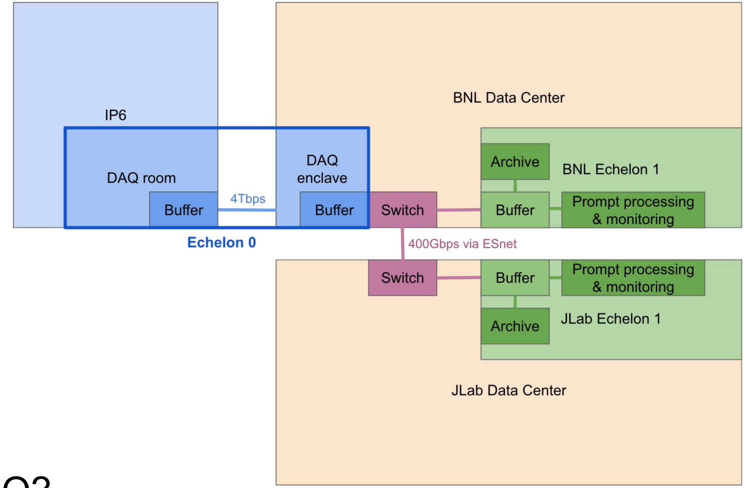
Storage Estimates by Use Case [PB]	Echelon 1	Echelon 2
Streaming Data Storage and Monitoring	71	35
Alignment and Calibration	1.8	1.8
Prompt Reconstruction	4.4	-
First Full Reconstruction	8.9	3.0
Reprocessing	9	9
Simulation	107	107
Total estimate storage	**201**	**156**

ePIC in the “Big Science” distributed computing lineage

- While ePIC is compute intensive, ePIC S&C must do its best to ensure ePIC is not compute-limited in its science
- ePIC joins a lineage of compute intensive experiments looking to distributed computing to meet their needs
- Distributed computing
  - knits the collaboration and its resources together in a global computing fabric
  - makes it possible to fully leverage both ‘owned’ and opportunistic resources
- As Frank Wuerthwein (OSG lead) commented in our recent ePIC S&C review:
  - The power of distributed computing is flexibility in moving processing between opportunistic resources and owned resources as they become available
- Distributed computing capabilities are also key to workflow orchestration, beginning at Echelon 1
  - Orchestrating streaming processing of the raw data streams
  - Managing prompt workflows consuming the stream
    - Monitoring, early calibration, analysis workflows validating data integrity, …
- Batch style distributed computing is well understood
- Where distributed computing gets interesting and relatively unique to ePIC is in the streaming workflows at Echelon 1…

Where ePIC workflow orchestration begins: Echelon 1

- We've (just) begun to work out how data will flow from Echelon 0 to Echelon 1 for prompt consumption by near real time workflows
- Plan in development includes DAQ leveraging data center resources through a 'DAQ enclave'
- Symmetric BNL, JLab Echelon 1 facilities downstream of DAQ are equally capable of performing Echelon 1 workflows
  - Ensuring the capability exists is our job; the E1 roles will be decided by ePIC and the facilities



- What is it we're processing in the data stream arriving from DAQ?
  - Time frames, demarcated by file markers and run markers
  - Together with a small non-event data component (e.g. slow controls)
  - Data arrives at Echelon 1s as files in a 1 week deep disk buffer
  - Workflows consume data from that buffer, first and foremost:
    - Archiving the full stream to tape (at least in early years)
    - Prompt processing, monitoring: providing a rapid view of data/detector integrity and quality
- **Acceptable (fast) latencies are incompatible with coarse grained files processed as batch jobs**
- **Prompt streaming workflows require a different approach**

Time Frames and Super Time Frames

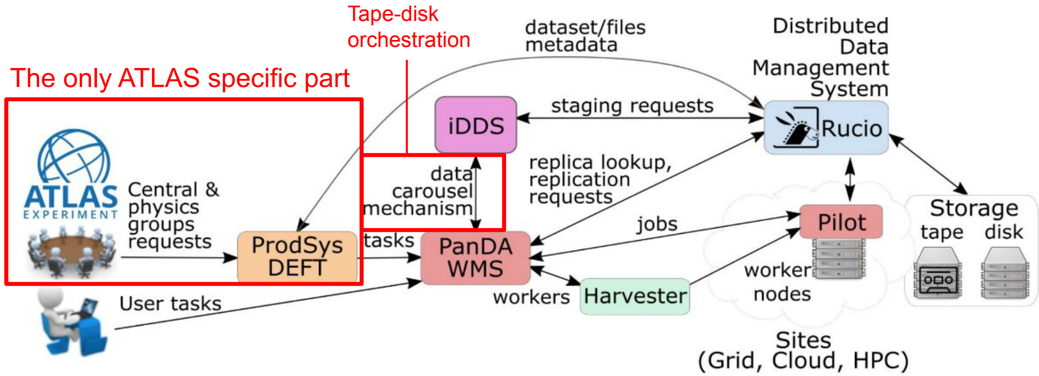
- Time frames aggregate all detector data within a time window of ~0.6ms
- DAQ-S&C discussions are converging on ‘Super Time Frames’ (STFs) as a contiguous set of O(1k) time frames (large enough that losses from the edges are negligible)
- This O(1s), 2GB data sample is an appropriate granularity for Echelon 1 processing -- short enough for prompt processing, long enough for tractable bookkeeping and file size
- Within a STF the TFs are time-ordered
- The STFs are not (required to be) time-ordered
  - Friendly to distributed computing and parallel orchestration

Batch systems, files, and streaming workflows

- Facility systems experts and admins know and love batch systems
- Our processing tasks for the most part map easily and well to batch
- Not, however, for low-latency fine-grained workflows like ePIC streaming processing
  - Quasi-continuous processing of newly arrived data that should not be impacted by the partitioning into files -- file granularity should not drive processing granularity/latency
- Do we address this by abandoning batch and putting something in its place?
  - Begs the question, what replaces it, and would increase complexity, heterogeneity
  - While decreasing flexibility of assigning resources dynamically across workflows
- In the context of ATLAS, and more recently the Rubin Observatory, in my group we've been facing this for many years
  - ATLAS: on HPCs, where full use of resources demands continuous streaming workloads that do not terminate on file boundaries
  - Rubin: in the fast O(1min) notification workflow when a transient is observed
    - and in efficiently processing their extremely fine grained processing units (O(1min))
- Our answer has been: don't abandon batch (which generally isn't an option anyway), implement streaming workflows within it

The PanDA workload manager (developed by BNL and UT Arlington)

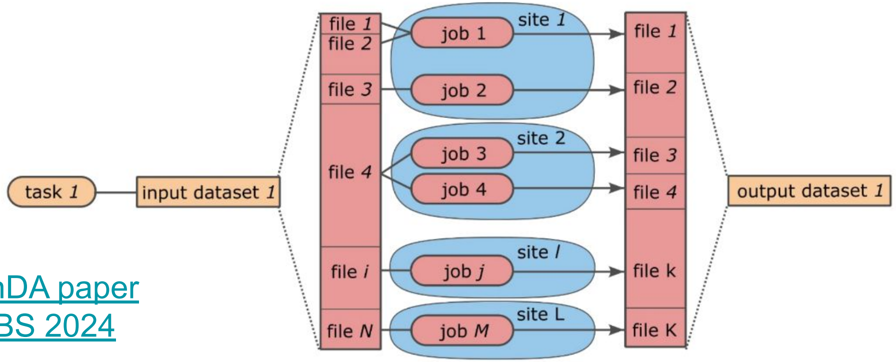
Workflow management in ATLAS: PanDA and its ecosystem



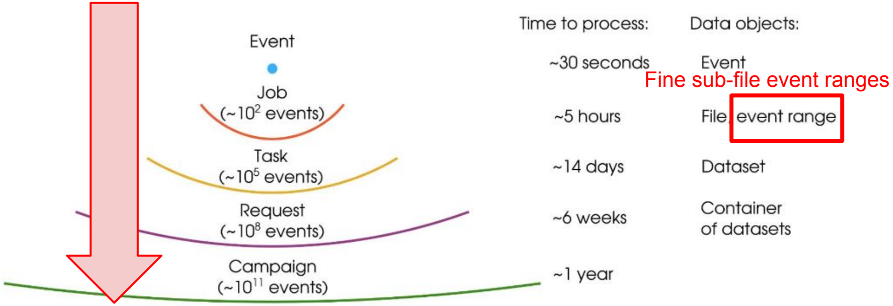
- **DEFT**: Database Engine For Tasks
- **PanDA**: Production ANd Distributed Analysis System
- **Harvester**: resource-facing service between the PanDA and collection of pilots
- **Pilot**: the execution environment on a worker node

Fine grained workflow orchestration

- **iDDS**: Intelligent Data Delivery System
- **Rucio**: Distributed Data Management System



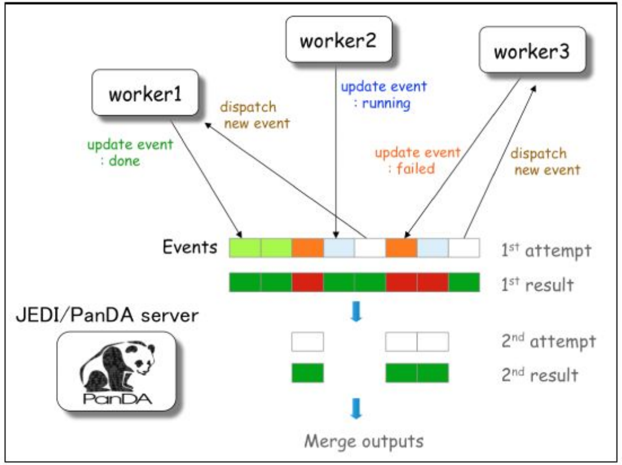
Granularity suited to the use case



PanDA paper
CSBS 2024

Low latencies and streaming in a batch/file context

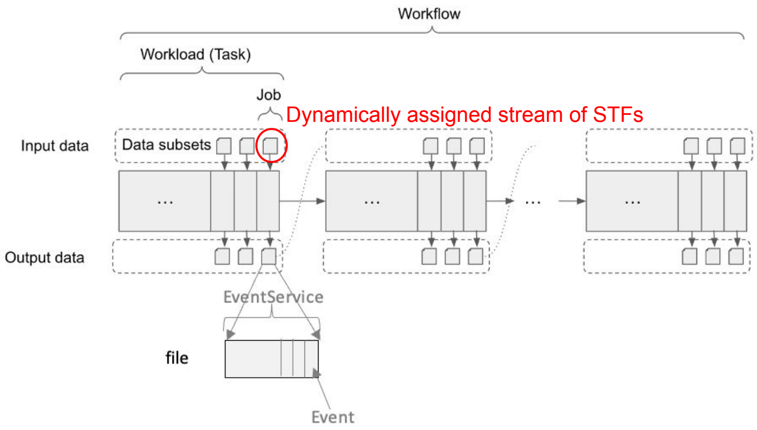
- The way we implement streaming workflows
  - with granularity independent of files and selectable down to event level (event ranges is typical)
  - and operating in batch slots
- is the Event Service, and its more recent generalization as the ‘intelligent Data Distribution’ (iDDS) service
- *(Here I begin referencing PanDA as an implementation context)*
- The key capabilities:
  - the batch job is a long-lived agent dynamically receiving fine grained processing assignments
  - the central orchestrator issues these granular assignments and (the hardest part) does proper bookkeeping, including re-assigning them when they fail
  - the orchestrator feeds assignments to workers such that processing is a continuous stream irrespective of job boundaries
  - Output granularity can be optimized independently of input
    - Major ATLAS lesson: avoid merging of outputs, it is fragile, especially when distributed
  - This is all highly parallelized and scalable



Most elements of this workflow/workload orchestration and management infrastructure are already serving ePIC/EIC in the AID2E AI-based detector design optimization project

Streaming workflow orchestration example in PanDA

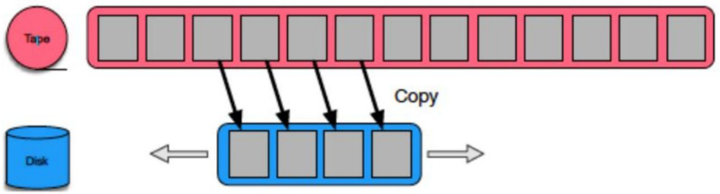
- Prompt data processing at E1-BNL and E1-JLab via a centrally managed **Workflow**
- **Tasks** are workflow instances that aggregate related processing (e.g. a given run configuration) for dispatch to compatible computing resources
  - e.g. a Task releases Jobs to the batch resources of an E1 as they become available
- **Jobs** are the Task execution units corresponding to an allocated batch slot



- Streaming orchestration enters the picture at the Job level via event streaming services
- As STFs arrive from DAQ they are registered as available
  - added to a growing Rucio container, which is input to an active PanDA task
  - as STFs arrive, PanDA dynamically assigns them to Job workers, each for as long as their job slot lives
  - Outputs/results are registered with PanDA, kept in a fine grained bookkeeping database to manage retries in case of failure
- How and where we store STFs and the downstream outputs is (like everything else) a matter of discussion, e.g. would object stores make sense

PanDA - Rucio streaming orchestration

- The capabilities of PanDA and Rucio mesh well to serve near real time streaming workflows
- Rucio datasets/containers can dynamically receive new registered content
- Rucio messages PanDA/iDDS that content is available, triggering prompt action
- iDDS provides the highly granular responsiveness at the STF level
- e.g. in the ATLAS Data Carousel, a sliding window of disk-staged data from a tape-based origin, PanDA+Rucio orchestrate the stage-in at the dataset level and iDDS handles fine-grained quasi-continuous processing initiation as new files become available



Data Carousel concept

Streaming orchestration for ePIC

- Prototyping of ePIC streaming orchestration workflows should begin in 2025, in the context of ePIC evaluating workload management options
- Planning the evaluations hasn't begun yet (I will recuse myself as appropriate :-)
- The PanDA team is ready to participate
  - Using a new PanDA instance running at BNL, for AID2E and other (primarily) NP applications
- Other workflow management system participants in the evaluation TBD
  - DIRAC(x) would be a natural if there are advocates in ePIC
  - My group also works/develops with DIRAC(x), we did the Rucio integration for Belle II, but we're not going to work two evaluations :-)

Conclusion

- ePIC’s streaming computing model presents novel and exciting challenges in the orchestration of fine grained streaming processing at the Echelon 1 centers
- Distributed computing will extend to Echelon 2 and 3; streaming orchestration may one day extend to E2, if well motivated
- The fact that ePIC will be a compute-intensive experiment becomes clearer as we look in more detail into the requirements and implications of streaming readout and an all-inclusive data sample ready for high-precision ePIC physics
- Workflow management must serve a wide range of workflows from streaming orchestration at E1s, to batch production at E1/E2, to opportunistic processing wherever resources can be found to answer ePIC’s compute-intensive needs
- Putting systems to the test will begin next year

Acknowledgements

Thank you to Wouter Deconinck, Markus Diefenthaler, Wen Guan, Dmitry Kalinkin, Alexei Klimentov, John Lajoie, Jeff Landgraf, Tadashi Maeno, my BNL NPPS and SDCC colleagues, and many others whose work and ideas have contributed to what I've presented

References

- ePIC streaming computing model report V1, 2023
- PanDA paper https://link.springer.com/article/10.1007/s41781-024-00114-3
- PanDA event service description (written for Rubin but mostly applicable)