SHARED I/O COMPONENTS FOR THE ATLAS MULTI-PROCESSING FRAMEWORK

Peter Van Gemmeren (ANL), David Malon (ANL), Marcin Nowak (BNL), Vakho Tsulaia (LBNL) on behalf of the ATLAS Collaboration

ABSTRACT

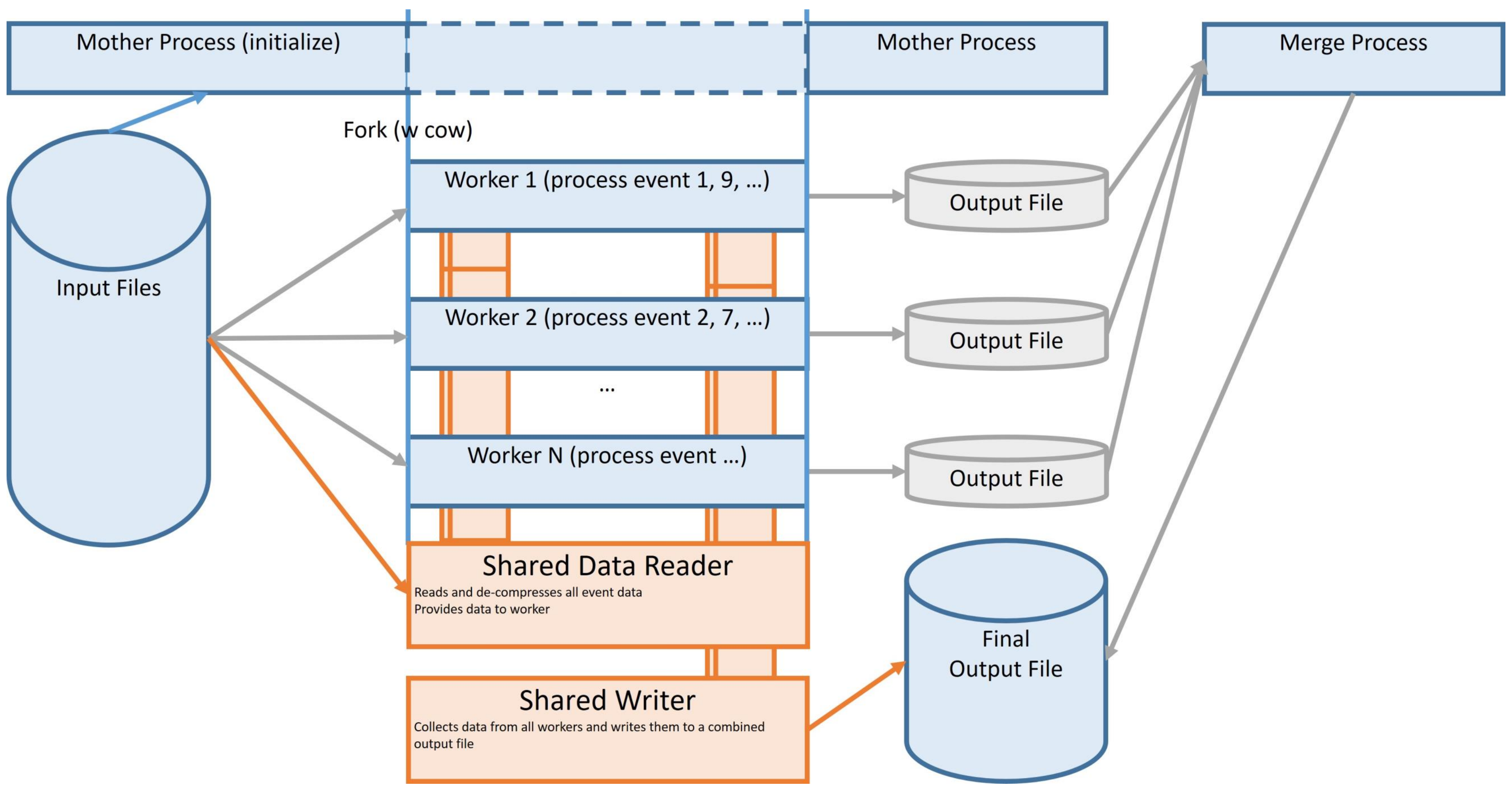
- ATLAS uses its multi-processing framework AthenaMP for an increasing number of workflows, including simulation, reconstruction and event data filtering (derivation). After serial initialization, AthenaMP forks worker processes that then process events in parallel, with each worker reading data individually and producing its own output. This mode, however, has inefficiencies:
  1) The worker no longer reads events sequentially, which negatively affects data caching strategies at the storage backend.
  2) For its non-RAW data ATLAS uses ROOT and compresses across 10-100 events. Workers will only need a subsample of these events, but have to read and decompress the complete buffers.
  3) Output files from the individual workers need to be merged in a separate, serial process.
  4) Propagating metadata describing the complete event sample through several workers is nontrivial.
- To address these shortcomings, ATLAS has developed shared reader and writer components presented in this paper. With the shared reader, a single process reads the data and provides objects to the workers on demand via shared memory. The shared writer uses the same mechanism to collect output objects from the workers and write them to disk. Disk I/O and compression / decompression of data are therefore localized only in these components, event access (by the shared reader) remains sequential and a single output file is produced without merging. Still for object data, which can only be passed between processes as serialized buffers, the efficiency gains depend upon the storage backend functionality.

ATHENA MP AND I/O

AthenaMP starts as a single process which after initialization forks off multiple worker processes, sharing memory via the ‘copy on write’ mechanism. Each worker processes events independently, reading the input data directly from file and producing its own output. In its default mode. AthenaMP dispatches individual events to the workers. Because data is compressed among events, different worker will spent CPU time and memory to de-compress the same data.

Because worker access data directly and independently, they are no longer reading events sequentially and caching by the storage backend (e.g.: ROOT TTreeCache) is less efficient.

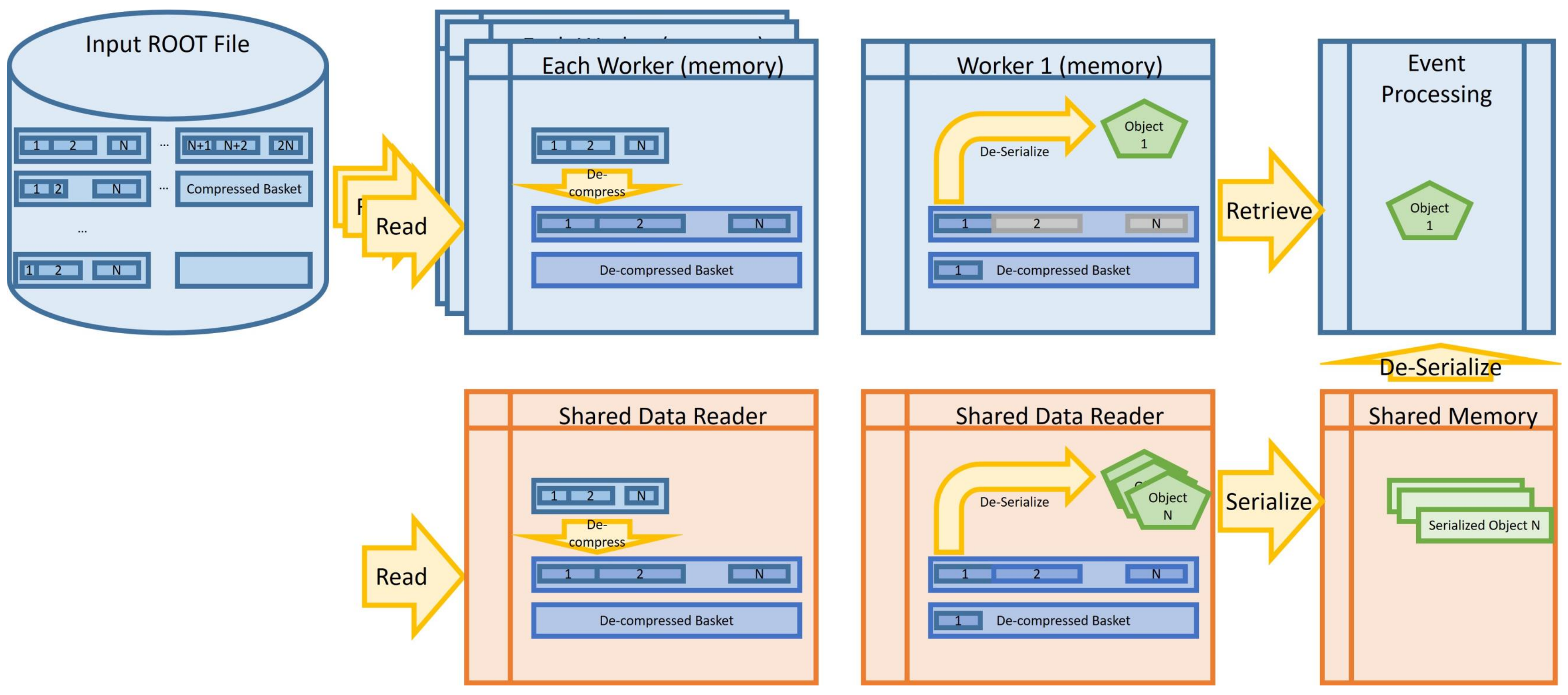
A separate, serial process is used to merge the temporary output files of each worker, essentially duplicating the I/O of the original workflow (including compression and serialization).



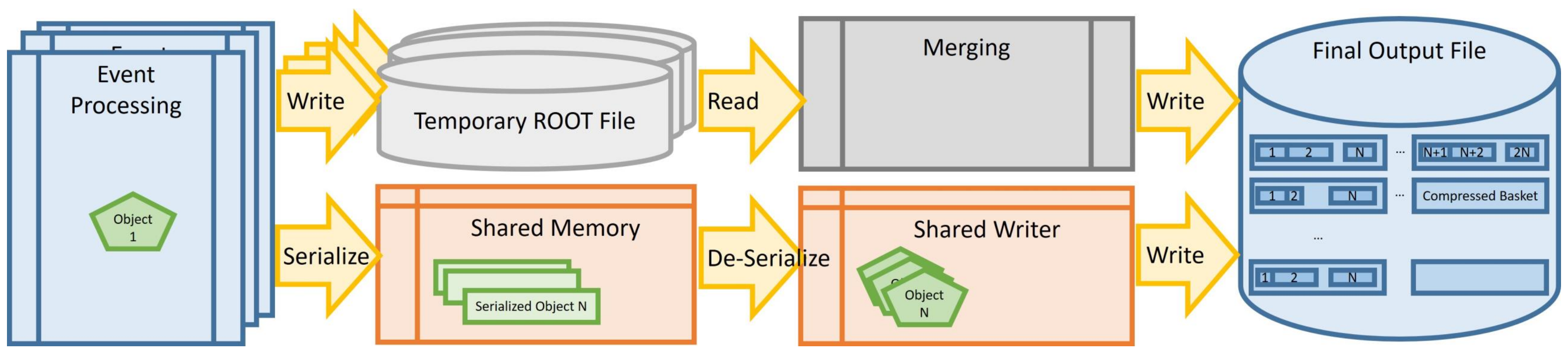
Mother Process (initialize)
Mother Process
Merge Process
Fork (w cow)
Input Files
Worker 1 (process event 1, 9, …)
Worker 2 (process event 2, 7, …)
…
Worker N (process event …)
Output File
Output File
Output File
Shared Data Reader
Reads and de-compresses all event data
Provides data to worker
Shared Writer
Collects data from all workers and writes them to a combined output file
Final Output File

SHARED I/O COMPONENTS

The Shared Data Reader reads, de-compresses and de-serializes the data for all workers and therefore provides a single location to store the decompressed data and serve as caching layer.



The Shared Writer collects output data objects from all AthenaMP workers via shared memory and writes them to a single output file. This helps to avoid a separate merge step needed in regular AthenaMP processing.



SHARED I/O FOR DERIVATIONS

For ATLAS derivation, the worker spend most of their time producing multiple output streams, according to different physics selection criteria.

Merging the different outputs across worker nodes requires significant time and computing resources.

But, a single Shared Writer can become a bottleneck trying to write data for multiple worker.

A new feature is being developed to allow a separate Shared Writer for each output stream writing to a separate file.

- These files do not need to be merged.

CONCLUSIONS

The Shared Data Reader can be deployed in AthenaMP to avoid multiple de-compression of the same data and make reading more sequential, which is beneficial to caching strategies.

- To share objects via memory, they need to be serialized and de-serialized.

The Shared Writer will help to avoid serial merge steps in AthenaMP, potentially saving CPU time and simplifying the workflow.

REFERENCES

- Van Gemmeren P, Binet S, Calafiura P, Lavrijsen W, Malon D and Tsulaia V (for the ATLAS Collaboration) (2012) “I/O Strategies for Multicore Processing in ATLAS”, J. Phys.: Conf. Ser. 396 022054
- Calafiura P et al. on behalf of the ATLAS Collaboration (2015) "Running ATLAS workloads within massively parallel distributed applications using Athena Multi-Process framework (AthenaMP)", J. Phys.: Conf. Ser. 664 072050

THE UNIVERSITY OF CHICAGO
U.S. DEPARTMENT OF ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.
Argonne NATIONAL LABORATORY