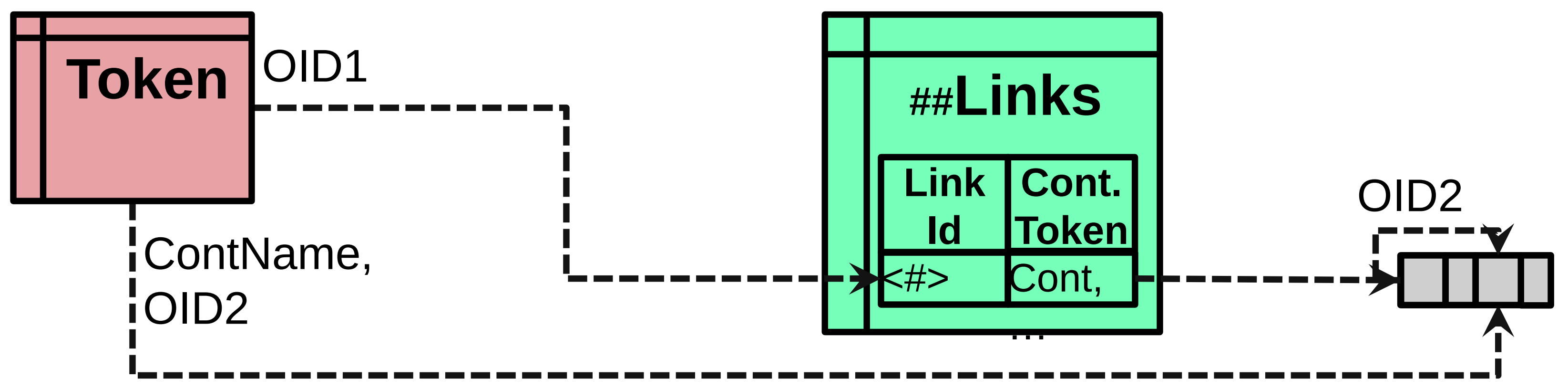
ATLAS Event Data Organization and I/O Framework Capabilities in Support of Heterogeneous Data Access and Processing Models

Jack Cranshaw, Peter van Gemmeren, David Malon, and Marcin Nowak on behalf of the ATLAS Collaboration

Abstract: Choices in persistent data models and data organization have significant performance ramifications for data-intensive scientific computing. In experimental high energy physics, organizing file-based event data for efficient per-attribute retrieval may improve the I/O performance of some physics analyses but hamper the performance of processing that requires full-event access. In-file data organization tuned for serial access by a single process may be less suitable for opportunistic sub-file-based processing on distributed computing resources. Unique I/O characteristics of high-performance computing platforms pose additional challenges. The ATLAS experiment at the Large Hadron Collider employs a flexible I/O framework and a suite of tools and techniques for persistent data organization to support an increasingly heterogeneous array of data access and processing models.

ATLAS Run 2 Event Data Model

- Optimized for attribute-level retrieval and histogramming
- Column-oriented, preferring structs of vectors to vectors of structs
- Designed with direct mapping of transient data model to persistent data model in mind
- Addition of new attributes and decorations is easy
- Serves end-user analysis use cases well
- Event-by-event variation in content is not supported (missing content must be back-filled because of persistence technology (ROOT) constraints), and schema evolution support is deliberately limited in favor of simplicity and efficiency in late-stage analysis

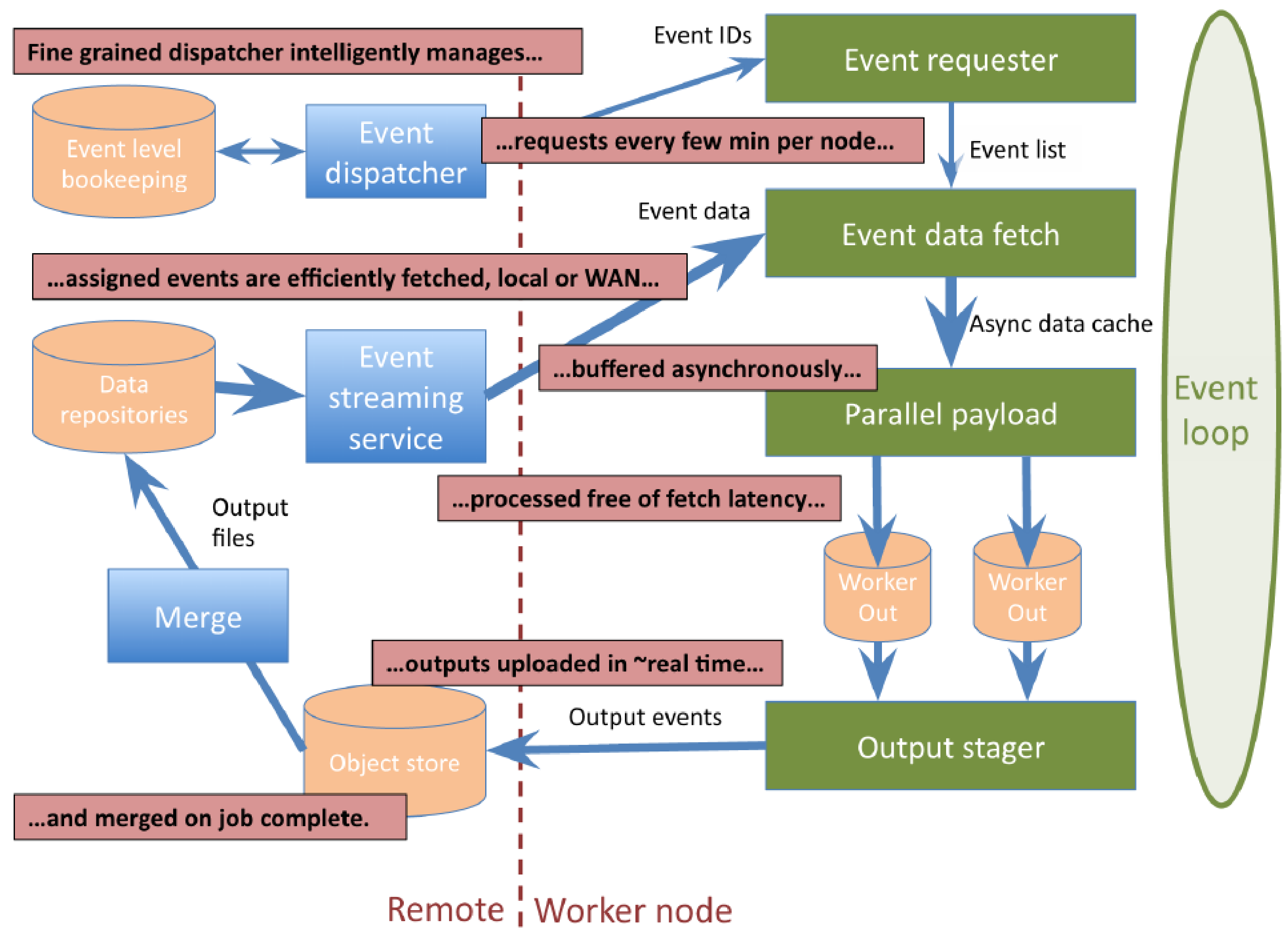


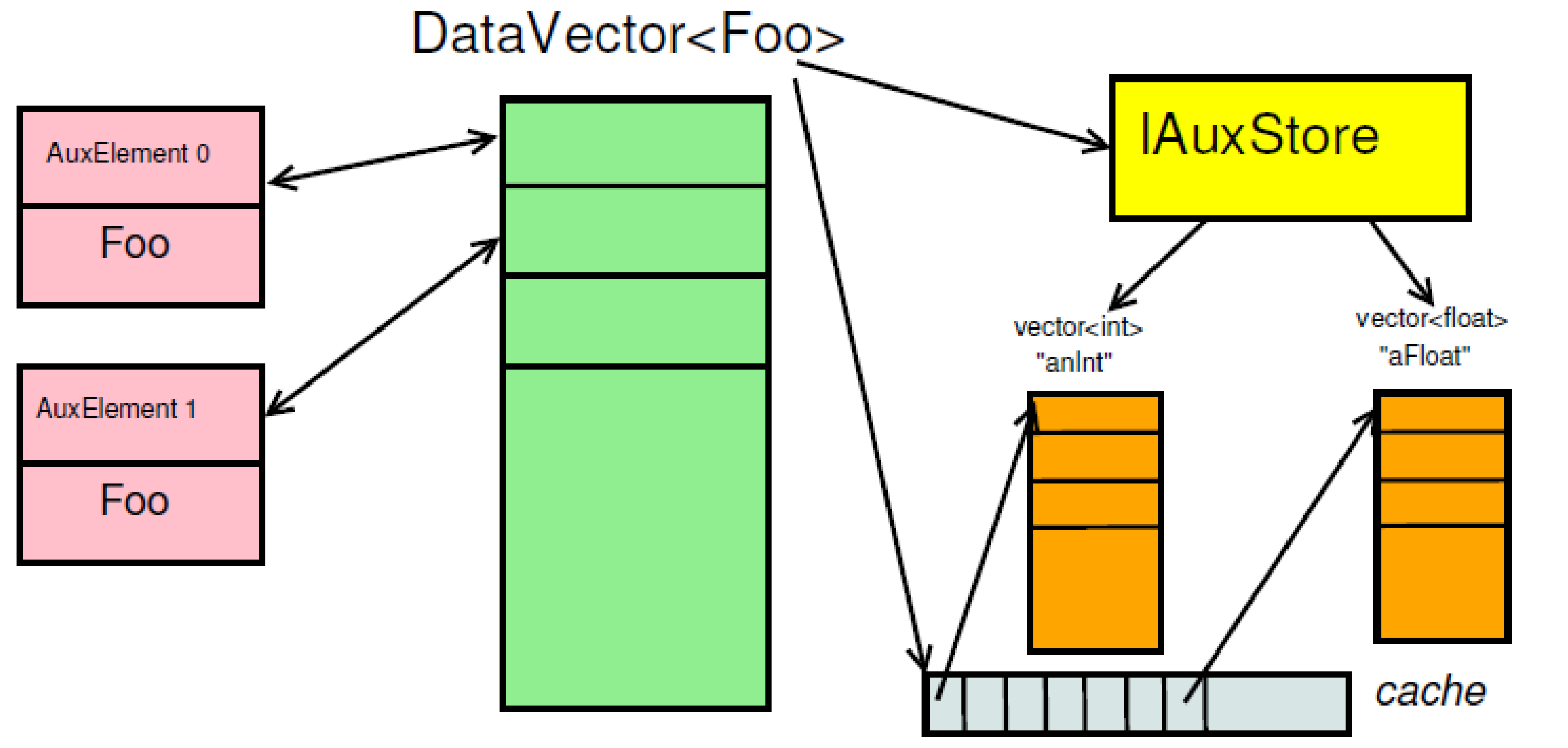
Performance tradeoffs

- ATLAS Run 2 event data model is less tailored to full-event processing (reconstruction, input to derivation framework that produces ATLAS analysis data products, etc.) than to selective content retrieval
- Substantial memory consumption for writing and reading when individual attributes have first-class status in persistent data model

Performance tuning

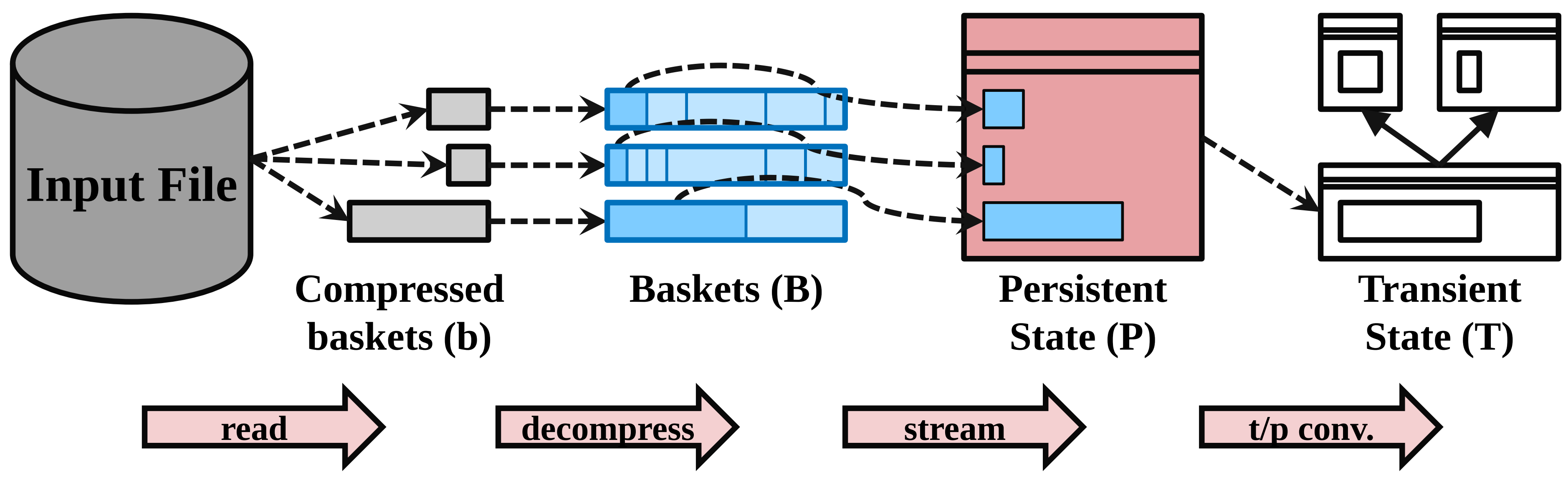
- Must balance such trade-offs
- Examples of tunable parameters include buffer and (ROOT) basket sizes, commit intervals, buffer flush settings
- Careful measurement for a variety of use cases is required
- Reordering of data within files can also help, and can be optimized for specific use cases
- Substantial potential savings in efficient aggregation and de-aggregation of attributes in transient←→persistent conversions (an area of ongoing work)





ATLAS Persistent Data Model Infrastructure

- Supports direct navigation to and retrieval of arbitrary data objects
- Uniform reference model for event, sub-event and non-event data, in-file and cross-file references, and back navigation to upstream data
- Supports event entry points that also record provenance and allow restoration of the state of the transient event store
- Independent of persistence technology
- Capabilities are more extensive and more general than what ATLAS tends to exploit in practice
- Well suited to a world of distributed object stores and to an environment in which any data should be readable from anywhere via wide-area access protocols



Emerging workflows

- Opportunistically-available resources are playing an increasingly important role in ATLAS computing
- Efficient use requires a scatter-gather architecture capable of delivering one or a few events rather than full files of events to ephemerally-available resources—finer granularity
- ATLAS event service implements this model
- Feeding the very large numbers of processors on HPC platforms is another increasingly important use case

Support for emerging workflows

- I/O components have been successfully adapted to support the ATLAS event service model, including support for multi-process worker jobs
- Simplest persistent data model to support wide-area event-by-event data distribution would involve storing all data for a single event in a single contiguous block of bytes
- A significant disadvantage to such an approach, though, is a substantially larger storage footprint because of poor compression (no compression across events)
- At LHC data volumes and given collaboration storage resource constraints, such a disadvantage may be decisive
- For processing that requires access to only a handful of event data attributes, there could be further disadvantages (reading unneeded data or multiple roundtrips, …)
- Current ATLAS implementation does not adjust or tune the persistent data model differently to support event service (versus "standard") workflows

Conclusions and next steps

- Performance tuning in ATLAS is an ongoing activity
- Approach to I/O and persistence tuning must balance analysis and production use cases, wide area and local access, and standard and emerging workflows
- Tuning choices may change as use case and workflow balances shift
- **Under investigation**: a storage-chunk-aware event streaming service, that could distribute parcels of events matched to how events are bundled in the input file (a result of flush settings and commit intervals), in a way that keeps framework components as technology-independent as possible

Argonne NATIONAL LABORATORY

BROOKHAVEN NATIONAL LABORATORY