

HSF Event Delivery WG – Introduction



MORGRIDGE
INSTITUTE FOR RESEARCH
CORE COMPUTATION

FEARLESS SCIENCE

A few thoughts before we begin...

First, thanks to ATLAS for hosting this WG meeting during their S&C Week!

- And apologies I can't be there in person.

I wanted to remind everyone of questions I asked last time:

- What is “event delivery”, anyway?
 - Writ large, an experiment's computing program is “event delivery” – all we do is deliver events (via files written to a storage element)!
 - What distinguishes what we want to do from a data processing system?
 - What does a successful event delivery system look like?
 - How does “event delivery” make HL-LHC more tractable?



Toward those ends...

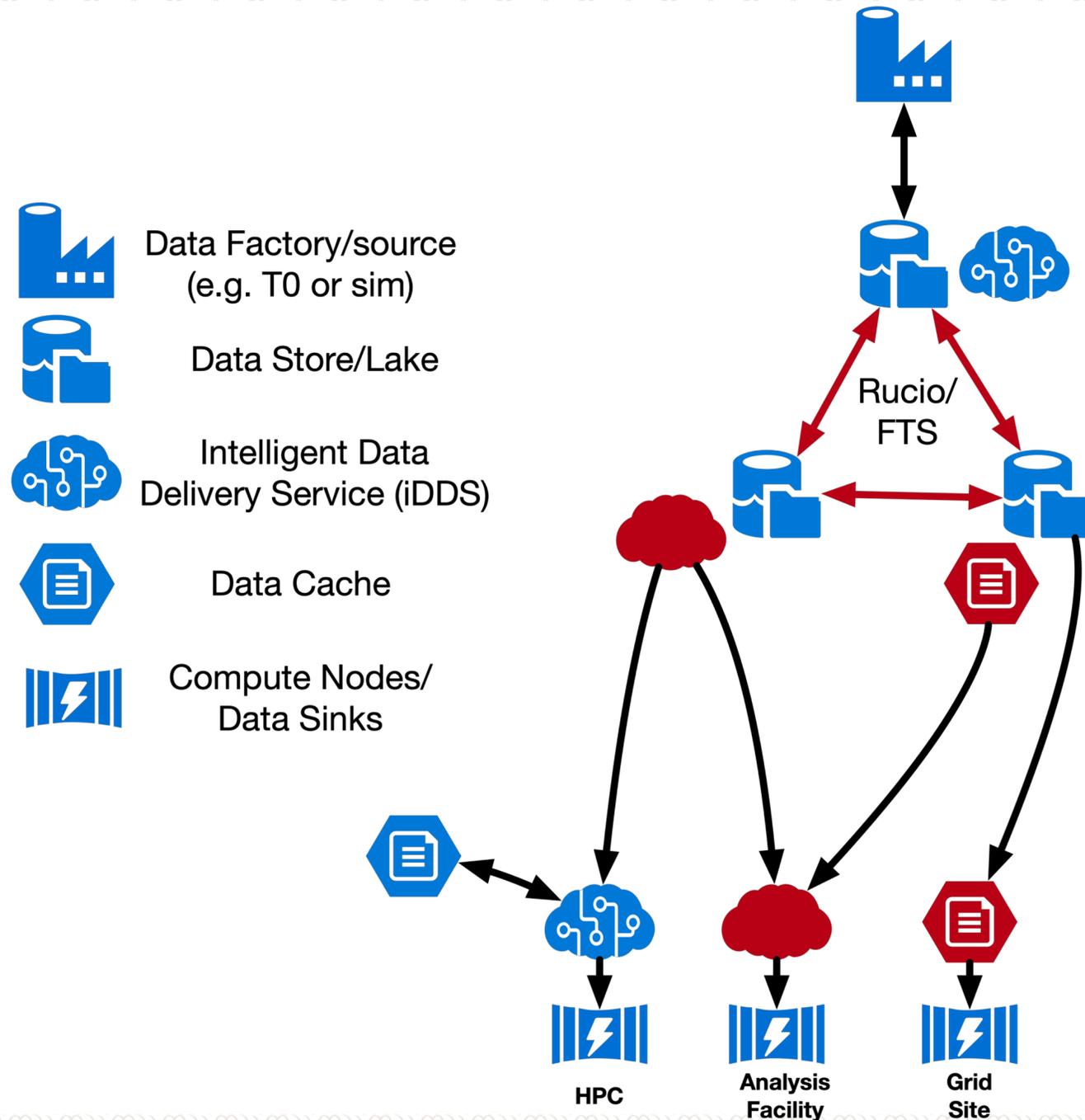
I think of a **cache** as a service that can deliver specific bytes to the user. Responsible for finding the right data (either locally or remotely) and sending it.

I see a **data delivery service** as something that can deliver the requested data - possibly transforming or pre-conditioning the data as needed.

- A cache responds to “deliver **bytes 1024 - 2048 of file** /foo/bar/baz.root”.
- A data delivery service responds to “**deliver the electrons** and muons from events 1 - 10,000 of dataset foo_baz_AOD”.

The event delivery service be responsible for handling user queries, determining the appropriate data source, transforming the data (potentially working with another IDDS at the source side), deciding whether to store it locally, and finally delivering to the user.

Important aspects of Event Delivery

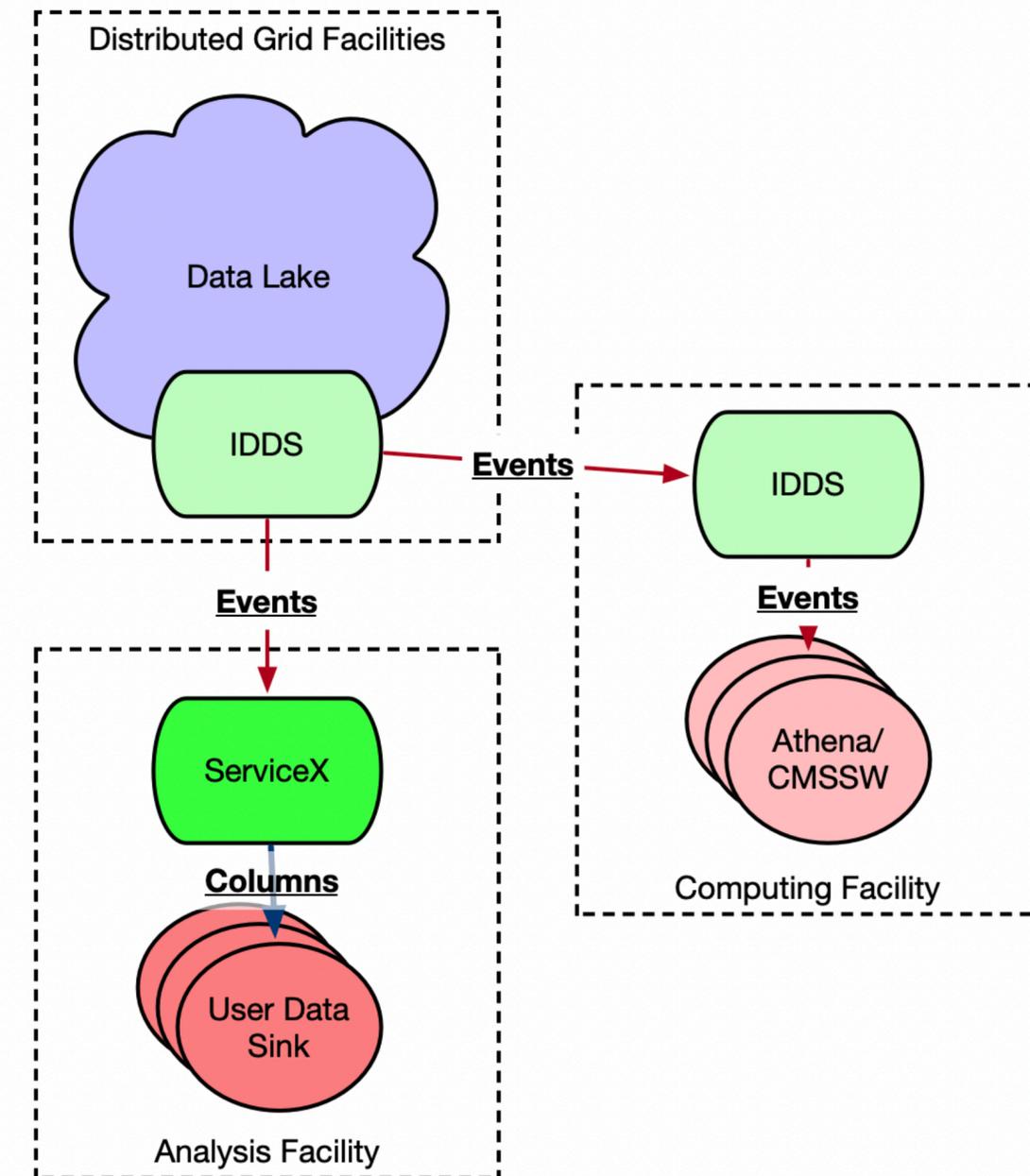


- Event delivery systems tend to feed computing sinks – analysis facility, HPC, or grid site. Grid sites are familiar.
- **HPC:** Significant computing resources for processing – but often only available in bursts. Important to have high utilization. Data present may be “transient”.
- **Analysis Facilities:** Multi-user facilities for high-speed data processing (target MHz event rates). Aggregate many events, in reduced format.
 - Ideally want data reduced *before* going to the analysis facility.
 - Likely want to have something intelligent at the data lake as well.

Events versus Columns

We should differentiate between use cases where we deliver “events” versus “columns”.

- Event delivery delivers most of the event content for a data tier - typically no data reduction.
- “**Event**” delivery provides events few-at-a-time: appropriate for fine-grained processing where we benefit from the extra costs of tracking at finer granularity.
 - Think: **processing measured in Hz** per core. Backfill at HPC.
- “**Column**” delivery focuses on delivering specific objects for a large number of events.
 - Think: **processing measured in MHz**; feeding analyses.



Why Event Delivery?

We should think about our impact metrics:

- For delivering events for data processing, how much **do we increase “goodput”** per unit of computing resources? Can event delivery help reduce time spent in processing tails?
- **Can event delivery save space on tape?** Can we separate the “archival format” from the “processing format”? How much do we “pay” on tape for the ability to read random events when processing?
- Can analysis facilities replace the need for *most* user ntuples?
 - Can event delivery enable models where graduate students spend less time writing code and more time doing physics?
 - Can we significantly speedup “time to first event”?

Today's Schedule

We have a few presentations talking about projects working toward an event delivery system:

- Run3/4 Perspectives – Gordon Watts
- iDDS – Wen Guan
- ServiceX – Marc Weinberg
- XCache for Event Delivery – Andy Hanushevsky

Thanks to everyone who agreed to present!



MORGRIDGE
INSTITUTE FOR RESEARCH
CORE COMPUTATION

morgridge.org

FEARLESS SCIENCE