

Event Service in ATLAS

Vakho Tsulaia (LBNL)

For the ATLAS Event Service Team

US ATLAS Distributed Facilities Meeting
Clemson University
March 14, 2016

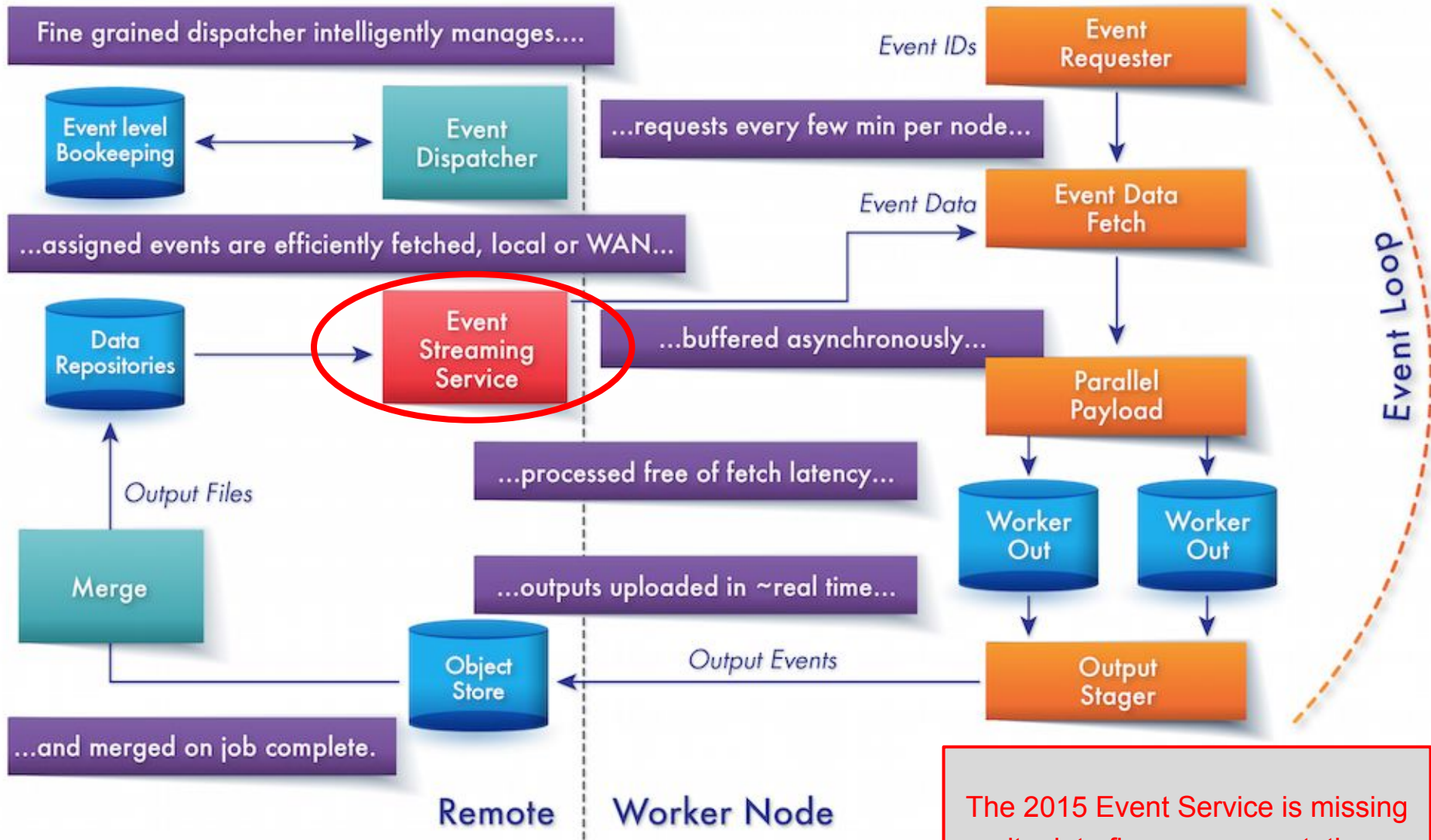
Contents

- Introduction to the Event Service (**ES**)
- Commissioning of the Event Service with ATLAS Geant4 Simulation
 - Amazon EC2
 - HPCs at NERSC, LBNL
- Backfilling Grid sites with the Event Service
- Event Service at Tier 3's
- Extending Event Service functionality beyond Simulation

Event Service. Concept

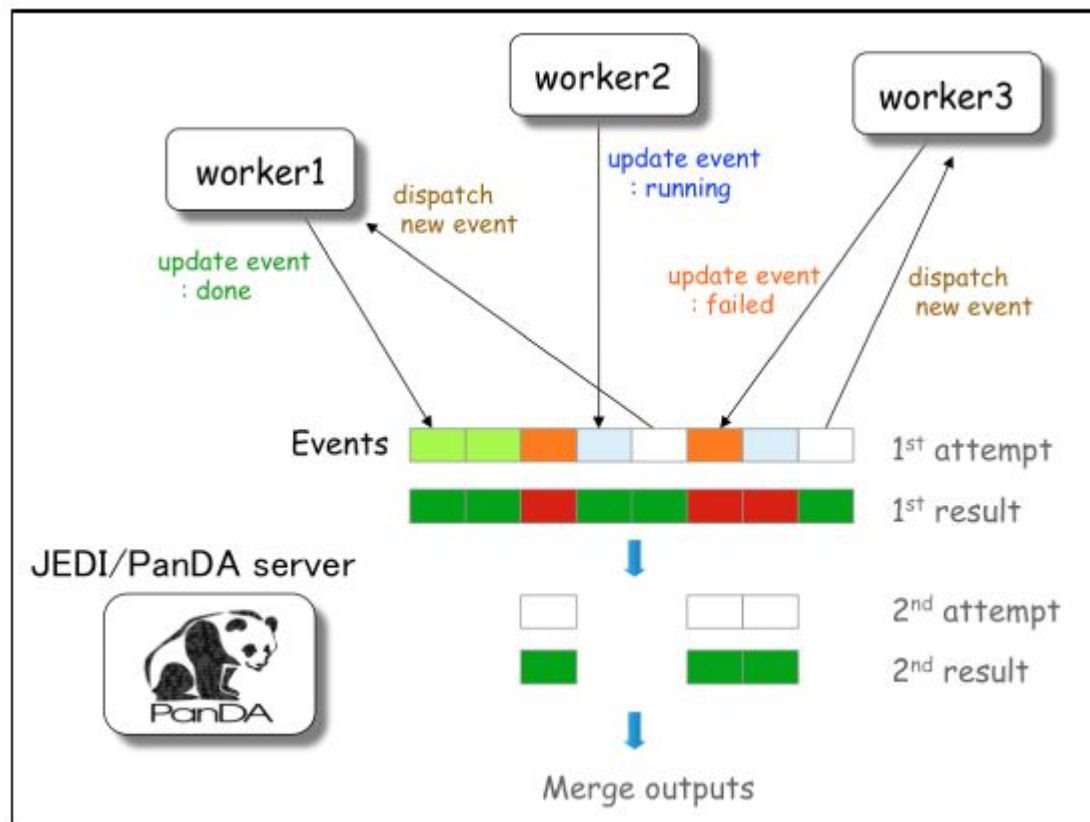
- A fine-grained approach to event processing
 - Quasi-continuous event streaming through worker nodes
- Exploit event processors fully and efficiently through their lifetime
 - **Real-time delivery of fine-grained workloads to running application**
 - Be robust against disappearance of the compute resource on short notice
- Decouple processing from chunkiness of files, from data locality considerations and from WAN latency
- **Stream outputs away quickly**
 - Negligible losses if the worker node vanishes
 - **Minimal local storage demands**
 - Promptly accessing outputs
- **Great for exploiting diverse, distributed, potentially short-lived resources**

Event Service in 2015



ES Components

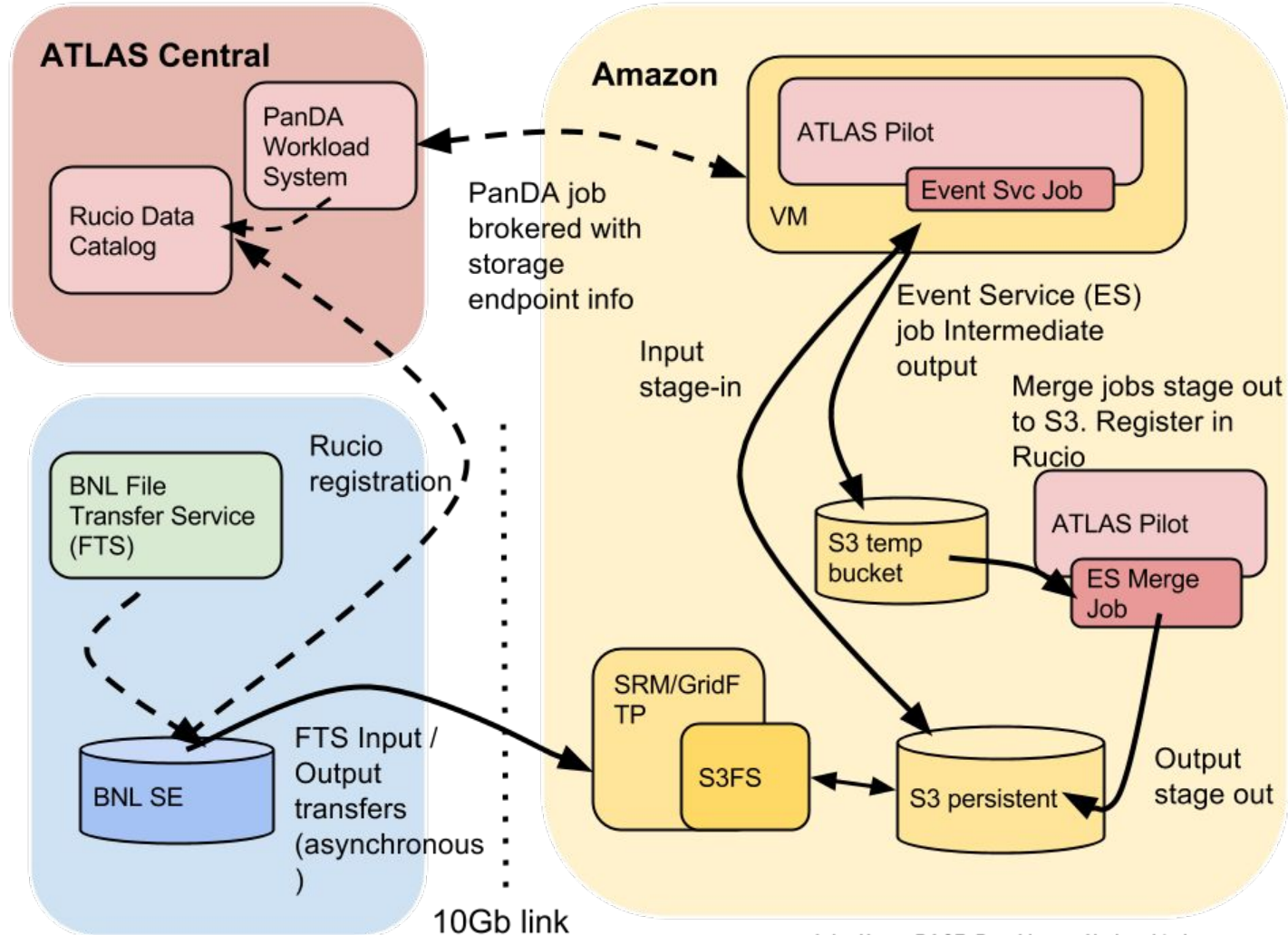
- **The ES Engine:** PanDA Distributed Workload Manager
 - JEDI extension to adds flexible task management and fine-grained dynamic job management
- **AthenaMP payload**
 - Efficient usage of CPU and memory resources on the compute node
- **Remote I/O**
 - Efficient delivery of event data to compute nodes
- **Object Stores**
 - Efficient management of outputs produced by the ES



First use-case: ATLAS Geant4 Simulation

- Current implementation of the Event Service **can run only ATLAS Geant4 Simulation**
- Simulation jobs use large fraction of ATLAS CPU-budget on the Grid
 - **By offloading Simulation jobs to other platforms (e.g. Clouds, HPCs) we can free a substantial amount of Grid resources**
- G4 Simulation is a **CPU-intensive job** with minimal I/O requirements
- **Meta-data handling** in Simulation jobs is **relatively simple** (wrt other types of workloads, e.g. reconstruction)
- ATLAS Geant4 **Simulation transforms run in single step**

Event Service at Amazon EC2



John Hover RACF, Brookhaven National Lab

Scaling Test Properties

- Follow up to the 40k-core, one region, 1-week run on Amazon EC2 in September 2015
- **3 EC2 regions**, each configured as separate PanDA site with dedicated SE
- Mix of EC2 instance types: **8-, 16- and 32-core types**
- **MCORE only**, so ~13k 8-core jobs
 - **BNL_EC2*_MCORE_PanDA** Queue
- Plan to run 2-3 days at **~100k core** level
- Demonstrate that we can **ramp to this scale in ~½ day**

100k run status

- Very busy ~2 weeks of preparation for the run in the second half of February
- ES jobs succeeded on all three EC2 regions in early March
- The last attempt to scale out to 100k cores was made on Friday last week
 - ES seemed to be working fine, but upon ~6k VMs (~11k slots, ~80k cores) Condor hit Amazon API query throttling
 - This resulted in Condor losing track of VM job submissions, causing follow on problems
 - The ES activity during the ramp-up period is yet to be investigated

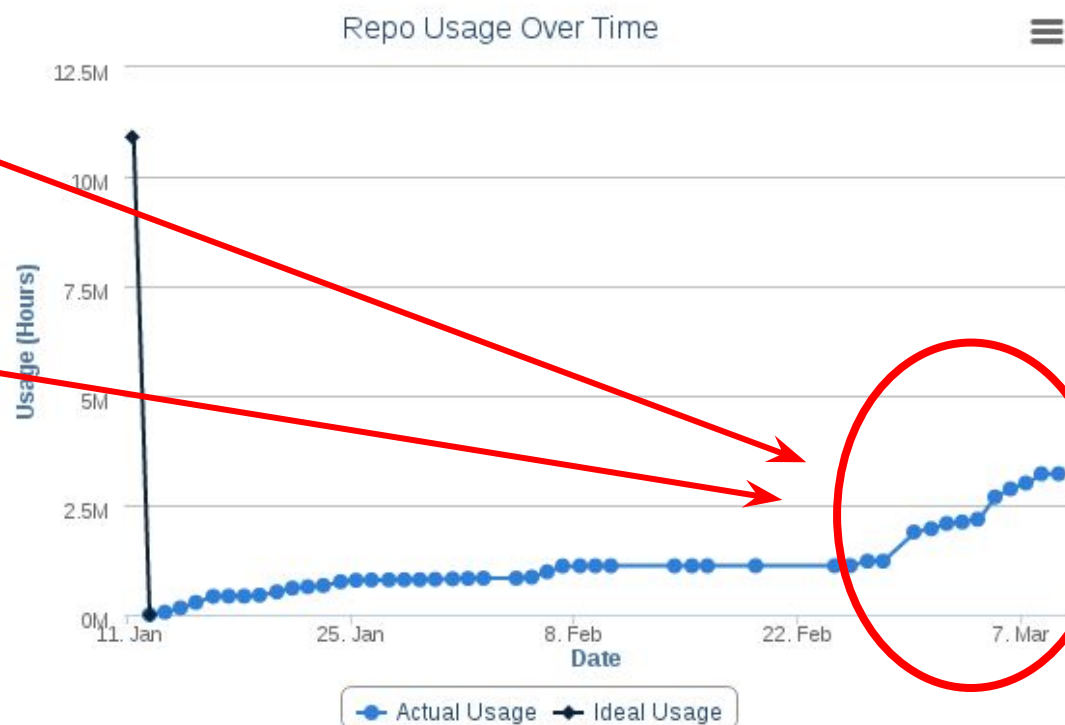
Yoda: Event Service on Supercomputers

- Specifically for running on supercomputers we have developed an MPI-based implementation of the ES called **Yoda**
 - We **reused the code** of the conventional ES wherever possible (e.g. AthenaMP payload)
 - Otherwise we **implemented lightweight versions of existing components** (e.g. JEDI and Pilot)
 - Replaced HTTP communication between ES components with **MPI**
- **HPC Pilot** runs on the edge node
 - Talks to PanDA, gets the jobs, stages in the input files (to the Shared FS)
 - Submits MPI jobs (Yoda) to the HPC batch system
 - Stages out the outputs to the Object Stores



Yoda running in production

- Yoda has been commissioned on the Edison HPC at NERSC
 - Integration with ARC underway
- Tested for running at large scale last year (Oct-Nov)
 - Geant4 10 “battle-testing” campaign
- Currently running two production tasks
 - **Task 1.** Part of the “500M simulation events” campaign
 - **Task 2.** Testing **AthSimulationBase-1.0.4** with 2M ttbar events



Backfilling Grid Sites

- **Objective:** Never shall a pilot get error code 20 “*I have nothing to dispatch to you*”
 - Turn the slot into an ES consumer until it disappears or higher priority work comes along
 - Don’t empty a resource as it drains for maintenance, switch it to ES backfill
- **Requirements:** 1. Direct WAN data access
 - Normally used with user jobs
 - Implemented and tested for regular (non-ES) workflows. Activated using **transferType=direct** in job definition
 - Now also extended for ES job, but ...
 - Not possible to test right now given the current AthenaMP inability to work with direct reading over XRootD
 - The fix (in Frontier) is on its way to the next LCGCMT release

Backfilling Grid Sites (2)

- **Requirements:** 2. Release CPU as soon as high priority jobs appear in the queue (Preemption)
- New cron in production to address this requirement
 - **Running ES jobs are killed** based on the priority of ES jobs (prio<200) and the wait time (15min) of high priority jobs
 - The **soft-kill command** is propagated to the pilot via regular heartbeats
 - **The pilot** sends a message to running AthenaMP payload saying that “*No more events*” are expected, waits until all running ranges are completed and **exits gracefully**
 - **High priority jobs are dispatched** to next pilots
 - **ES jobs are automatically regenerated** to resume remaining work after the high priority jobs are done

Filler Jobs in Backfill Mode

- JEDI gathers tasks from a task pool based on their priorities in descending order and stops generating jobs once enough jobs are queued
- Backfill jobs tend to have lower priorities
- **Solution:** introduce a new workqueue for tasks in the backfill mode
 - E.g. `processingType=(evgen or simul)` and `priority < XYZ` and `eventService=True`
 - JEDI gathers tasks per workqueue. Backfill jobs can go even if high priority jobs from other workqueues are there
 - Jobs could go to large sites as well, but they would be eventually reassigned

Tadashi Maeno, Paul Nilsson

Event Service at Tier 3's

- ES allows local site to quickly evict Panda pilot to allow local users to use their resources
- ES for T3's first prototyped at Univ. of Arizona
 - Local pilot submission with service proxy (needs to be updated). Need to reestablish running jobs there
- Oklahoma, Northern Illinois Univ expressed interest in ES there - BNL also mention using ES in shared T3 resource as backfill
- Next steps are to compare ES in Tier 3 vs traditional Panda job for event simulation efficiency and effects on local users

Doug Benjamin

Outlook: ES beyond Simulation

- Extension from full simulation to **FastSim** seems to be an easy step
 - Just need to **increase granularity of Event Ranges** (from 1 to 5-10 events)
 - *Already supported!*
- **Derivation**
 - **Low hanging fruit:** single step transform, confirmed that can technically run in the ES
 - **Issue:** correct handling of the meta-data (currently being looked at ...)
 - *The development of a new infrastructure which would support correct handling of meta-data for event-based workflows goes well beyond the utilization of ES for derivation*
- **Reconstruction**
 - The **next major target**
 - **Need to come up with a strategy for running multiple-stage transforms within the Event Service**

Summary

- The Event Service has been successfully commissioned by running full ATLAS G4 Simulation workloads on various platforms, e.g. Amazon EC2, HPC @ NERSC
- New functionality is available for starting to backfill Grid sites with the Event Service. Testing blocked by the Frontier bug which prevents AthenaMP from direct reading over XRootD
- We are taking steps to extend ES functionality beyond Simulation. Although running Reconstruction in the ES is not yet around the corner