The ATLAS Event Service: A New Approach to Event Processing

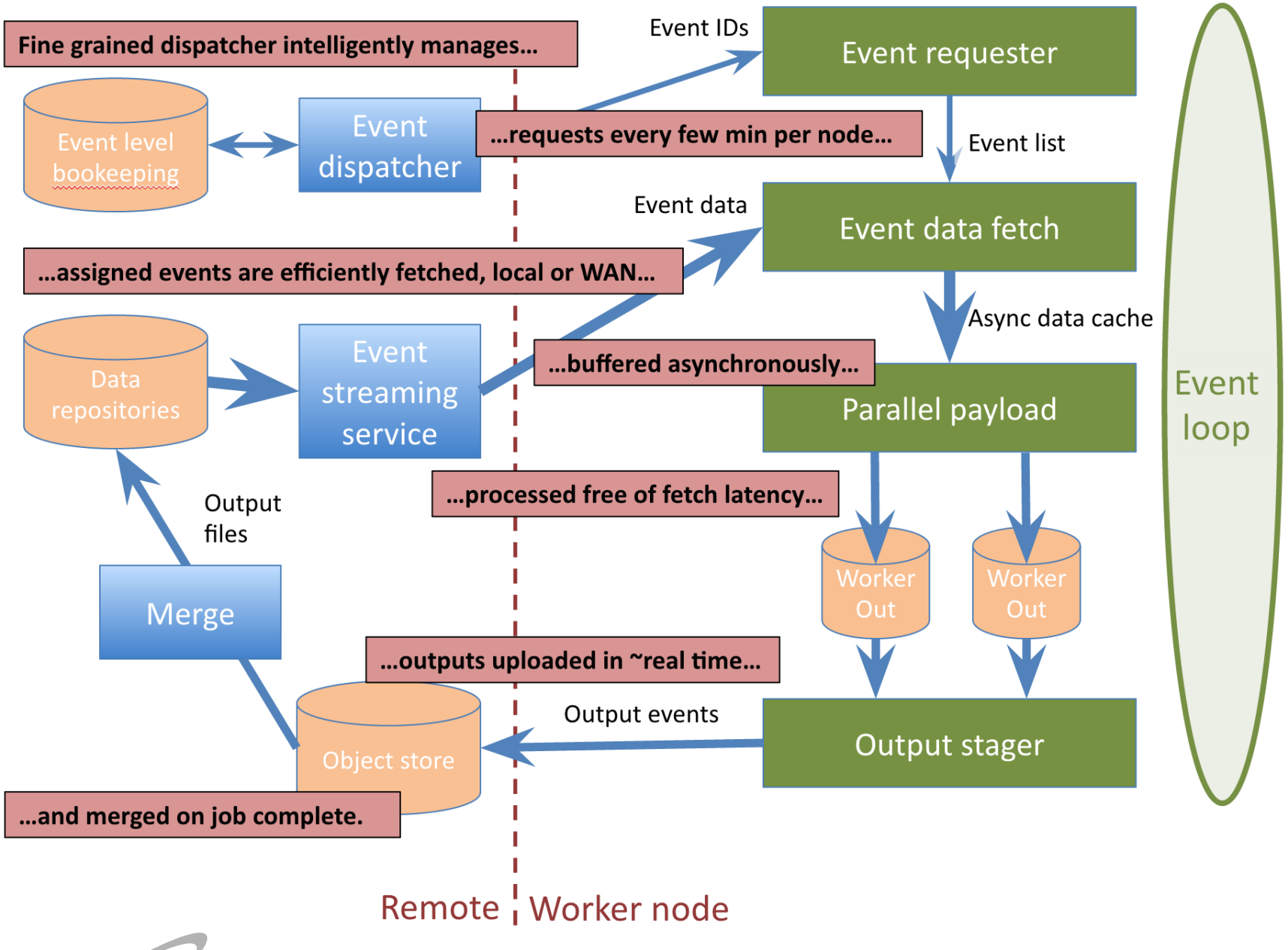
P Calafiura (LBNL), K De (UT Arlington), W Guan (U Wisconsin Madison), T Maeno (BNL), P Nilsson (BNL), D Oleynik (UTA), S Panitkin (BNL), V Tsulaia (LBNL), P Van Gemmeren (ANL), T Wenaus (BNL)
for the ATLAS Collaboration

CHEP 2015
Okinawa, Japan
April 13-17 2015

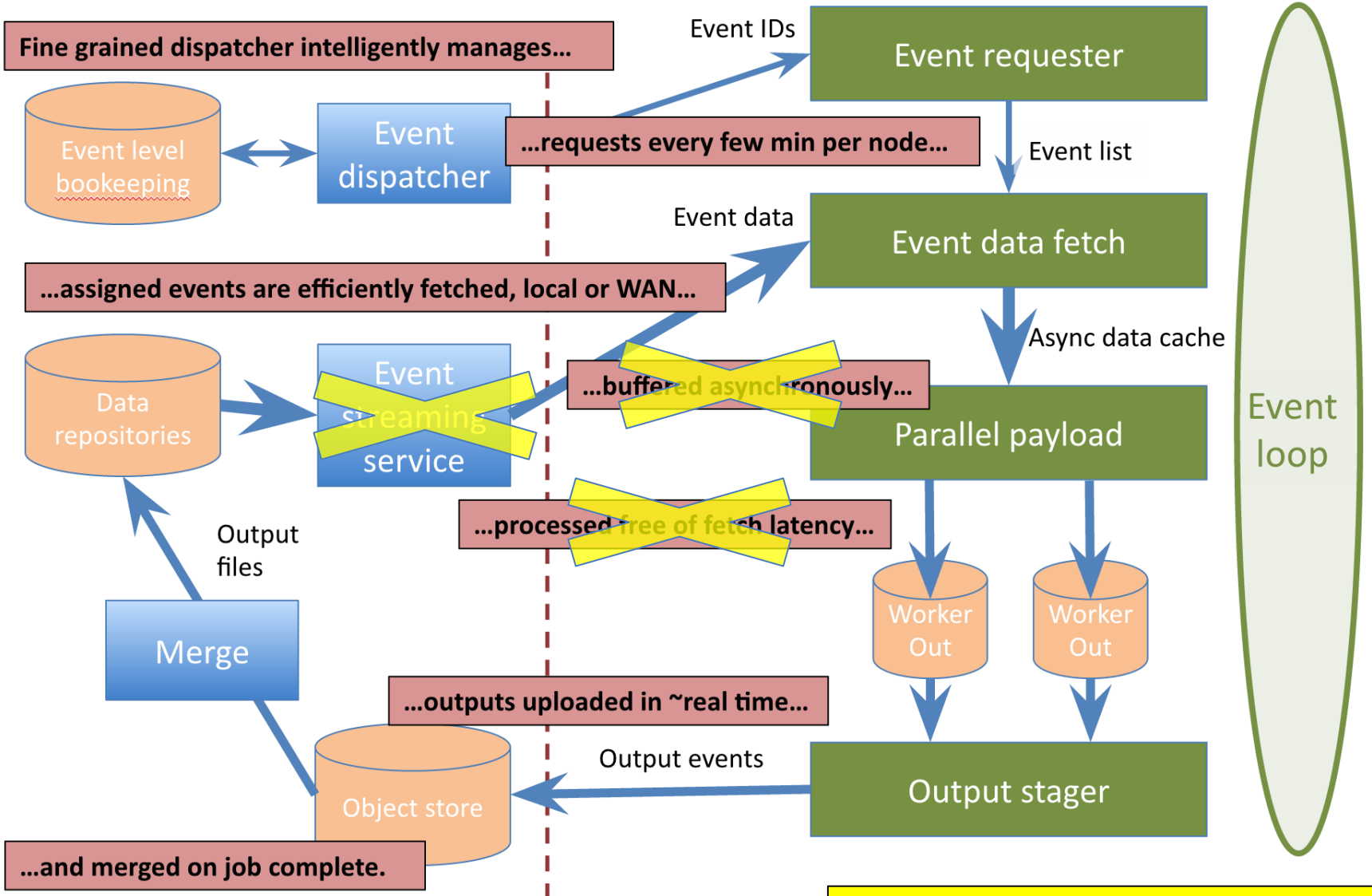
The ATLAS Event Service

- A new **fine grained approach to event processing**: quasi-continuous event streaming through worker nodes
- **Exploit workers fully and efficiently through their lifetime**, whether that is 30 minutes, or 30 hours, or 10ms from now with no notice
- Decouple processing from the chunkiness of files, from data locality considerations, from WAN access latencies
- **Stream outputs away quickly**, for negligible losses if the worker vanishes, minimal local storage demands, and promptly accessible outputs
- Great for **exploiting diverse, distributed, potentially short-lived resources**
  - **HPCs** when 'full' are full of big hulking rocks; they still have plenty of room for sand, for those able to efficiently pour fine grained work into the cracks
  - **Amazon spot market** rewards short-lived, transient workers
  - **Volunteer computing** (BOINC) rewards robustness against unreliable, unpredictable transient workers
  - **Conventional clusters** can be more easily managed when workloads can be instantaneously jettisoned with negligible losses: no long drain times

Event Service Schematic



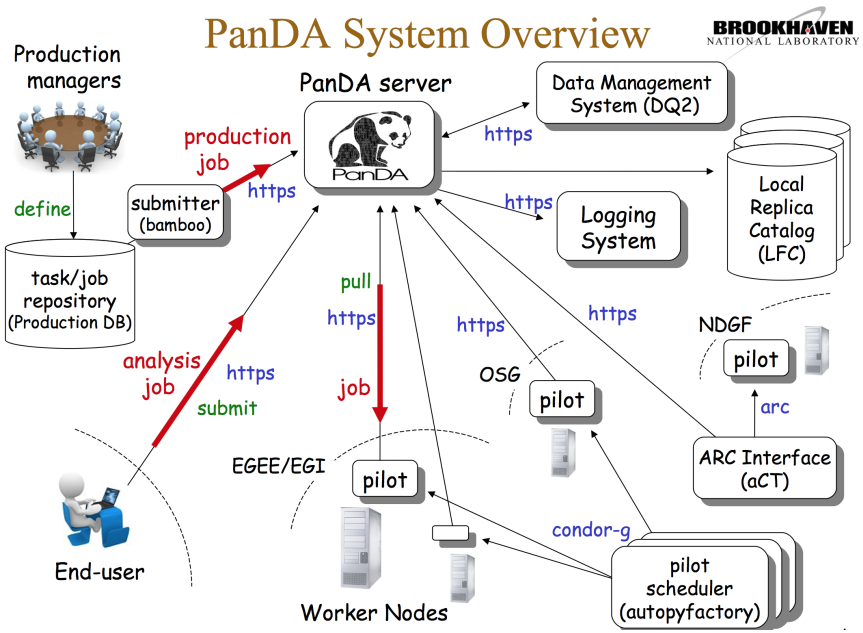
Event Service as realized today



The missing piece: intelligent asynchronous data fetch (more later)

The ES Engine: PanDA Distributed Workload Manager

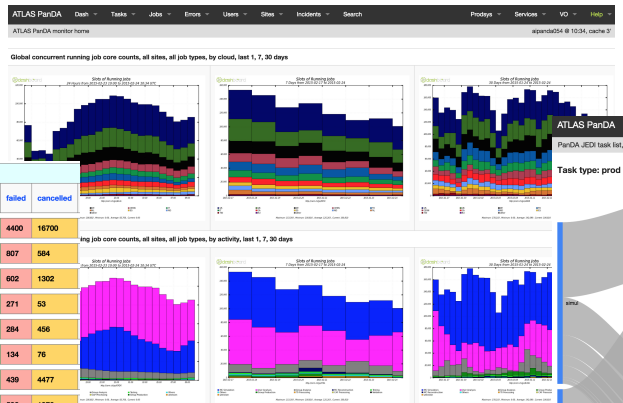
https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA



PanDA manages processing and data workflows for large scale data intensive computing

- 2005: Initiated for US ATLAS production
- 2008: ATLAS-wide production & analysis
- 2011: Dynamic usage-driven data caching
- 2012-15: BigPanDA project extending to HPCs, other experiments
- 2014: Network-aware brokerage
- **2014: JEDI extension adds flexible task management and fine grained dynamic job management**
- **2014: JEDI enables the Event Service**

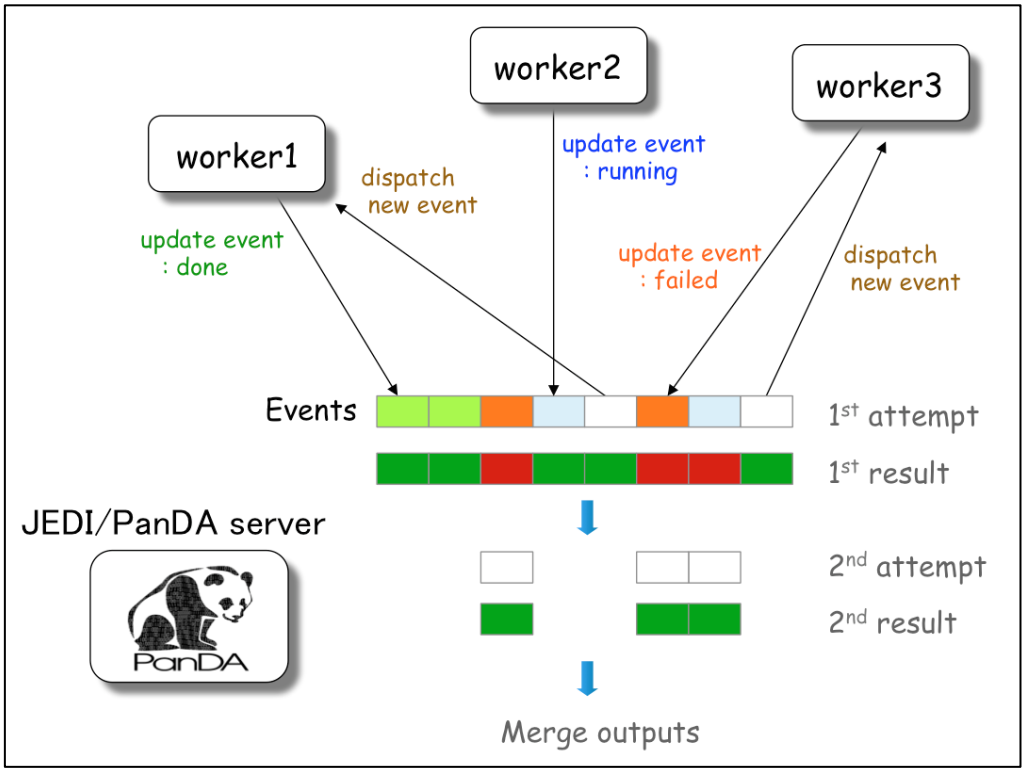
PanDA (and Event Service) Monitor
http://bigpanda.cern.ch/



Cloud / Site summary of production jobs - Cloud view For a description of cloud view see below

Cloud	Status	nJobs	defined	waiting	assigned	throttled	activated	sent	starting	running	holding	transferring	finished	failed	cancelled
All clouds		213752	1	134	21884	0	39281	11	2647	26290	797	30668	70949	4400	16700
CA	online	15548	0	0	2844	0	656	0	163	1843	93	2505	6053	807	584
CERN	online	24466	0	0	3000	0	6183	0	253	1690	129	4954	6353	602	1302
DE	online	7115	0	0	1367	0	168	0	59	1508	22	611	3056	271	53
ES	online	14699	0	0	3966	0	264	0	5	2264	26	1923	5511	284	456
FR	online	4283	0	0	34	0	1687	0	20	742	9	444	1137	134	76
IT	online	14242	0	134	1163	0	546	0	106	2105	28	1452	3792	439	4477
ND	online	25119	0	0	1235	0	8309	0	1861	5135	65	1603	5248	593	1070
NL	online	56242	0	0	3600	0	12088	9	127	6423	267	11044	17906	305	4473
RU	brokeroff	57	0	0	0	0	2	0	0	2	0	0	52	0	1
TW	online	18192	0	0	2711	0	4391	0	15	2050	71	4438	3913	139	464
UK	online	8115	1	0	526	0	1133	0	33	694	34	1043	3829	252	570
US	online	25674	0	0	1438	0	3854	2	5	1824	53	651	14099	574	3174

The ES Enabler: JEDI extension of PanDA



2013-2014: PanDA server becomes the JEDI/PanDA server. JEDI adds the capability to

- accept work defined in terms of high level tasks
- optimally partition tasks into jobs based on the dynamic state of available resources
- dispatch work down to very fine (event level) granularities
- handle the workflow and bookkeeping requirements of operating at this fine grained, highly dynamic level
- perform fine grained retry
- and automatic final merge

The Event Service makes full use of these capabilities

⟹ PanDA future talk, Track 4

Today's ES Payload: the ATLAS Parallel Framework

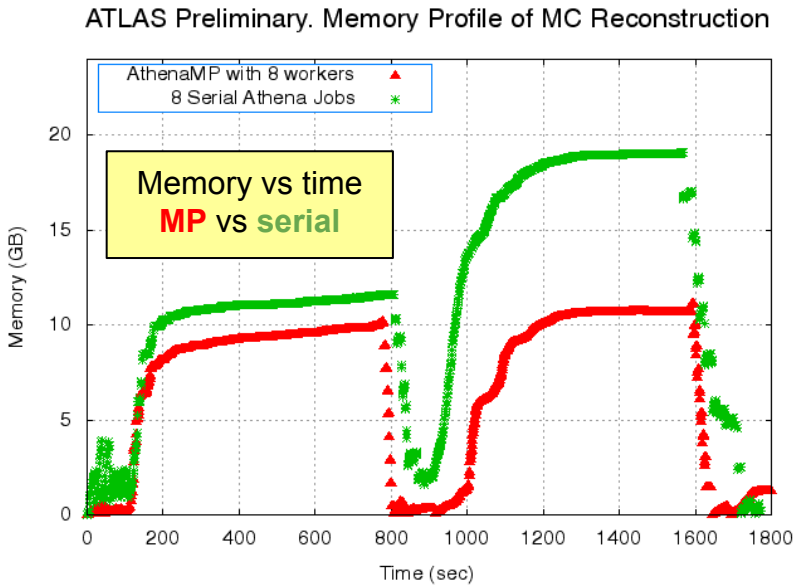
AthenaMP (Multi-Process) – event-parallel framework

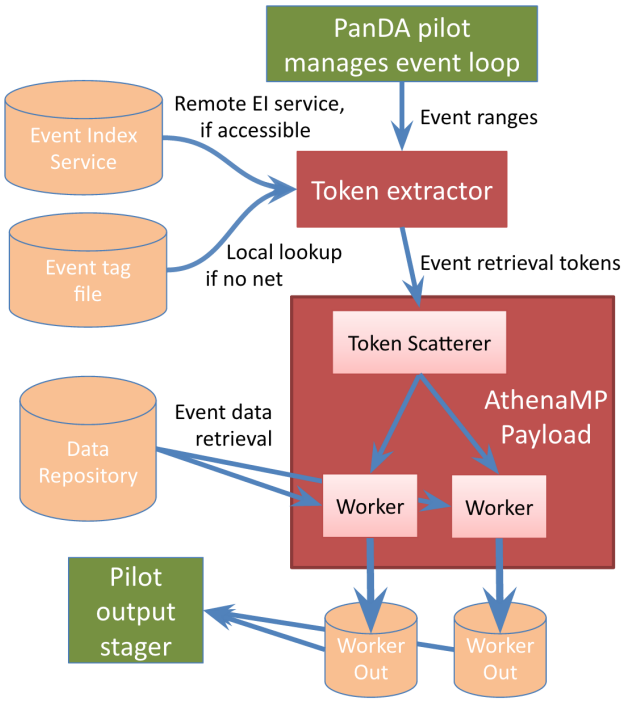
Memory sharing via Copy-On-Write

- Efficient use of all cores

Manages **independent parallel streams of events** and **efficiently supports remote I/O**

- Just what's needed for the Event Service





For the Event Service, AthenaMP manages **distribution of events** (as retrieval tokens) to parallel workers

- Event ranges translated to locations (tokens) via Event Index service (or if no net access, local files)

Workers retrieve event data using the token

- Data may be local or remote

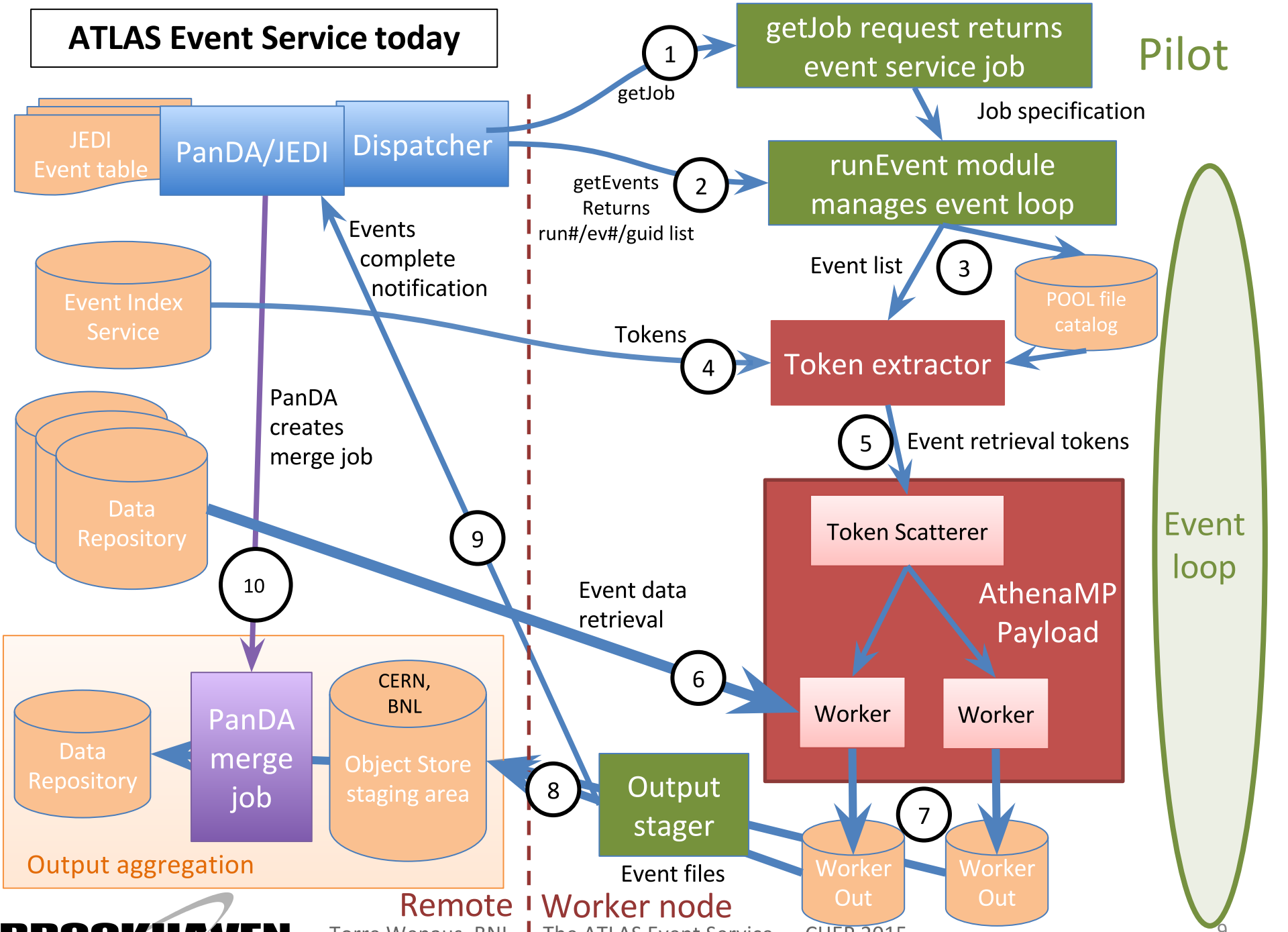
PanDA pilot manages allocation of event ranges, staging of outputs to object store, and success/fail reporting

⇨ athenaMP talk, Track 2

The ES Data: Event Service Data Handling

- The Event Service (and the Event Streaming Service (ESS) extension to come) is designed for **efficient exploitation of storage resources** as well as processing resources
  - Storage is the biggest cost element in ATLAS computing
- Leverages **powerful networks** -- the backbone of LHC computing success -- to minimize use of costly storage
- Leverages **efficient event I/O** making WAN data access practical
  - While insulating payload processing from WAN latencies
- Uses **remote data repositories** such that there are no data locality or pre-staging requirements: flexible, dynamic use of processing resources
  - While integrating well with local cache mechanisms
- Uses **object stores** for fine grained output management to provide highly scalable, easily accessed WAN storage for many small outputs
- This is **data intensive, network centric, platform agnostic computing**
  - An increasingly important paradigm in scientific computing

ATLAS Event Service today



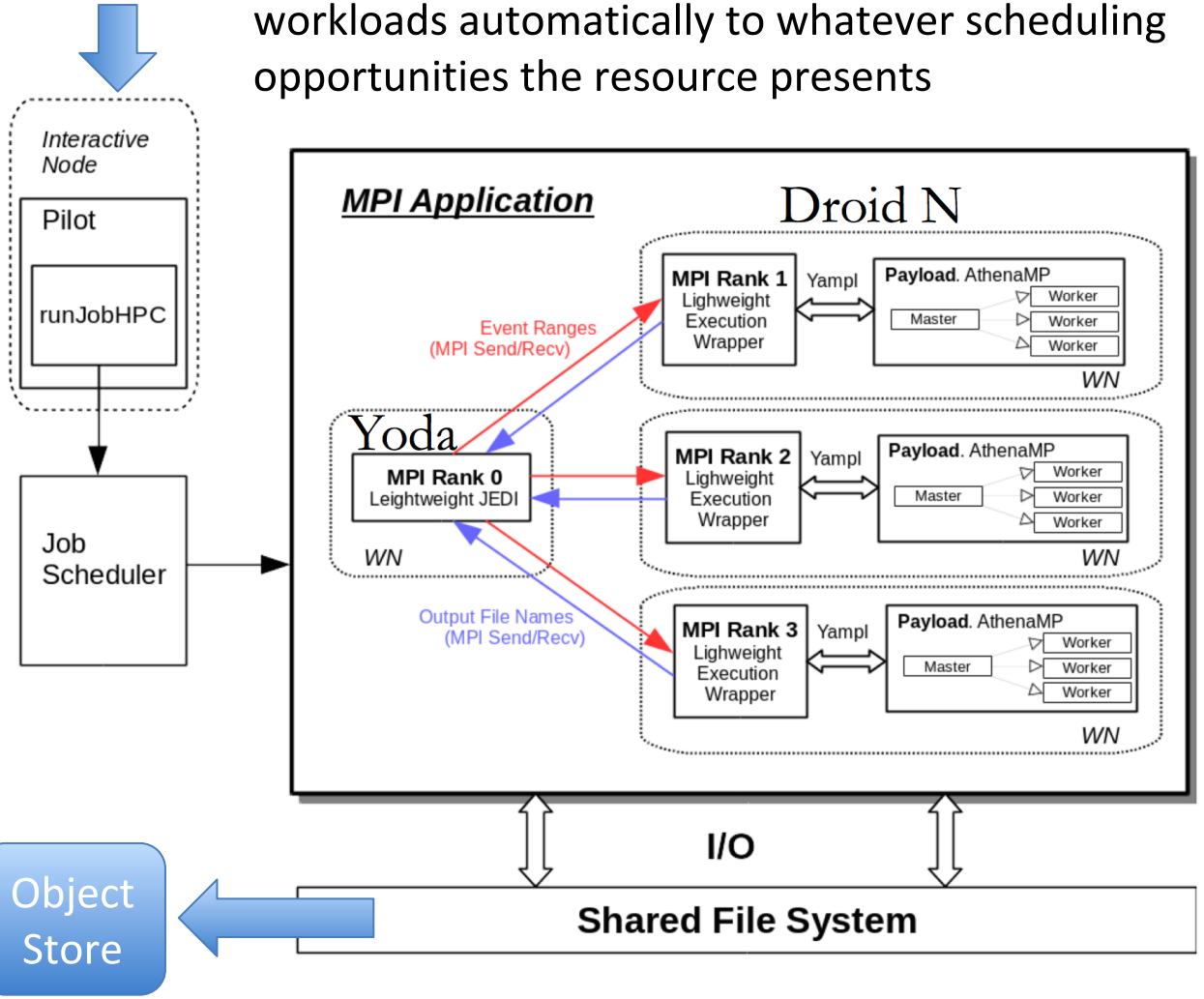
Using the ES: Yoda

JEDI based event service miniaturized for HPCs

On HPCs, the MPI-based master/client adaptation 'Yoda' of the Event Service allows tailoring workloads automatically to whatever scheduling opportunities the resource presents

- Work assignments stream in with fine granularity
- Outputs streamed promptly to secure location
- Processing proceeds until slots die, with full utilization

Offers the efficiency and scheduling flexibility of preemption without the application needing to support or utilize checkpointing

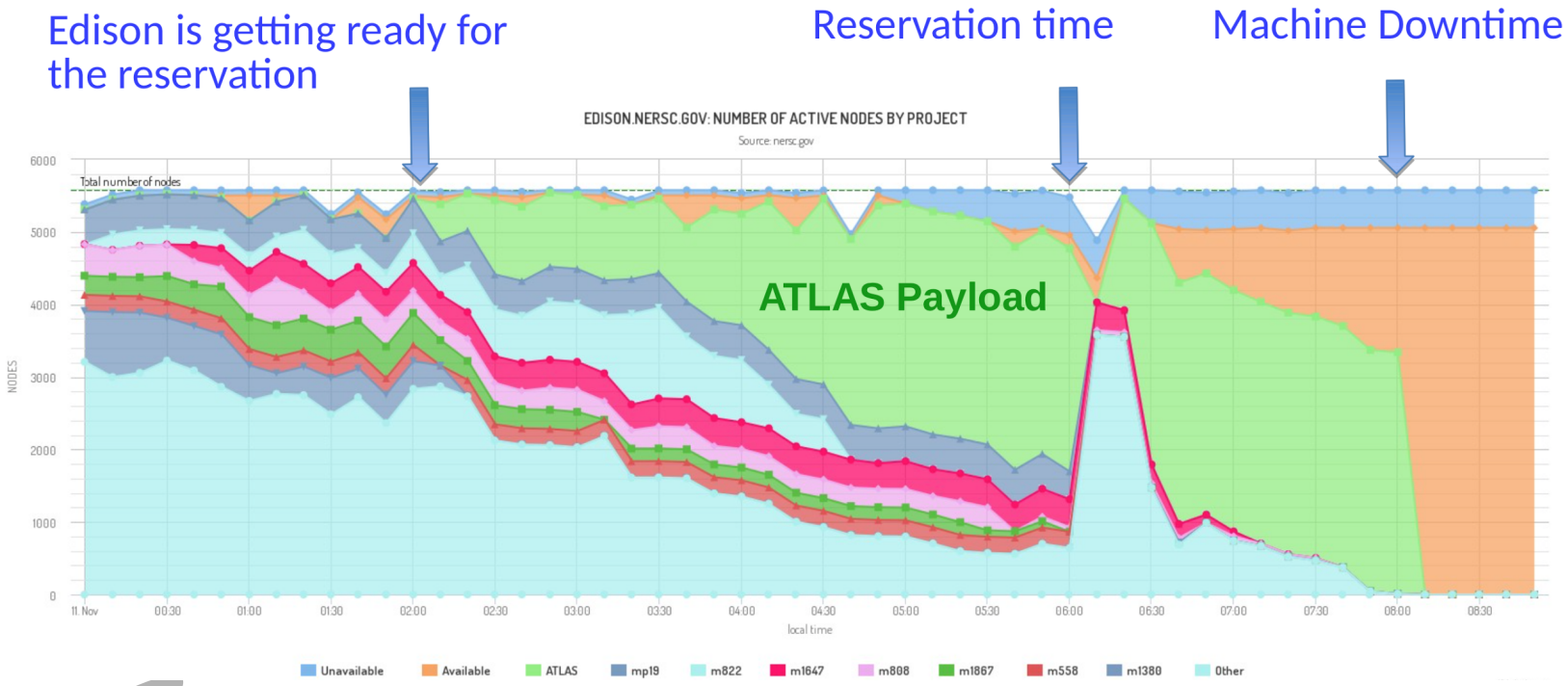


Yoda talk, Track 8

Demoed at Supercomputing 2014, Nov 2014
http://goo.gl/WSdU4a

Yoda scavenging resources @ NERSC Edison

- As the machine is emptied for downtime or large usage blocks, a killable queue makes transient cycles available
- Yoda sucks them up efficiently and processes events until the moment they vanish, with negligible losses to the processing
- And refills when they appear again



Using the ES: Amazon EC2 Spot Market

- A PanDA site at BNL sends jobs to EC2 spot market VMs
  - ~7x cheaper than on-demand
  - Free if Amazon reclaims the node in less than one hour
- ES maximizes the return on these transient short lived slots
- Work leverages BNL's R&D collaboration with Amazon
  - Up to ~50k concurrent job slots available
- Uses BNL Tier 1-developed capability to elastically and transparently expand PanDA workloads into cloud resources
  - **Send peak usage to the cloud**
- Now in physics validation to begin production

Using the ES: Volunteer Computing

http://atlasathome.cern.ch/

ATLAS@Home

ATLAS@Home is a research project that uses volunteer computing to run simulations of the ATLAS experiment at CERN. You can participate by downloading and running a free program on your computer.

ATLAS is a particle physics experiment taking place at the Large Hadron Collider at CERN, that searches for new particles and processes using head-on collisions of protons of extraordinary high energy. Petabytes of data were recorded, processed and analyzed during the first three years of data taking, leading to up to 300 publications covering all the aspects of the Standard Model of particle physics, including the discovery of the Higgs boson in 2012.

Large scale simulation campaigns are a key ingredient for physicists, who permanently compare their data with both "known" physics and "new" phenomena predicted by alternative models of the universe, particles and interactions. This simulation runs on the WLCG Computing Grid and at any one point there are around 150,000 tasks running. You can help us run even more simulation by using your computer's idle time to run these same tasks.

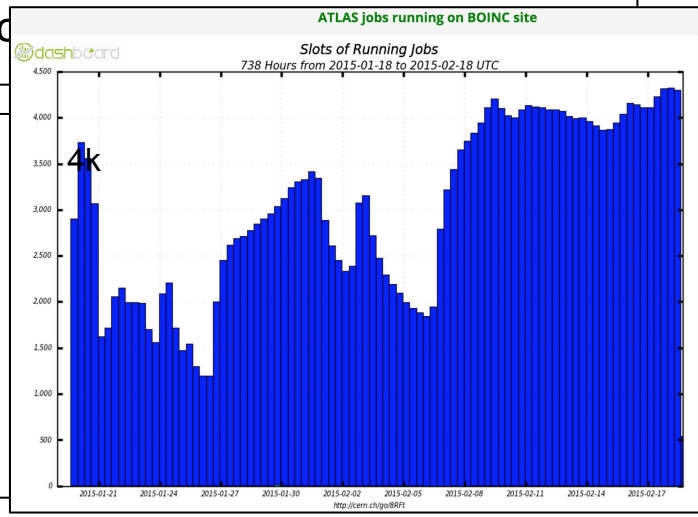
User of the day

yank

Born in Brooklyn, NY a few years ago (town in Arkansas. Aircraft mechanic a retired...

ATLAS@Home is up and running ATLAS simulation

You can join and c



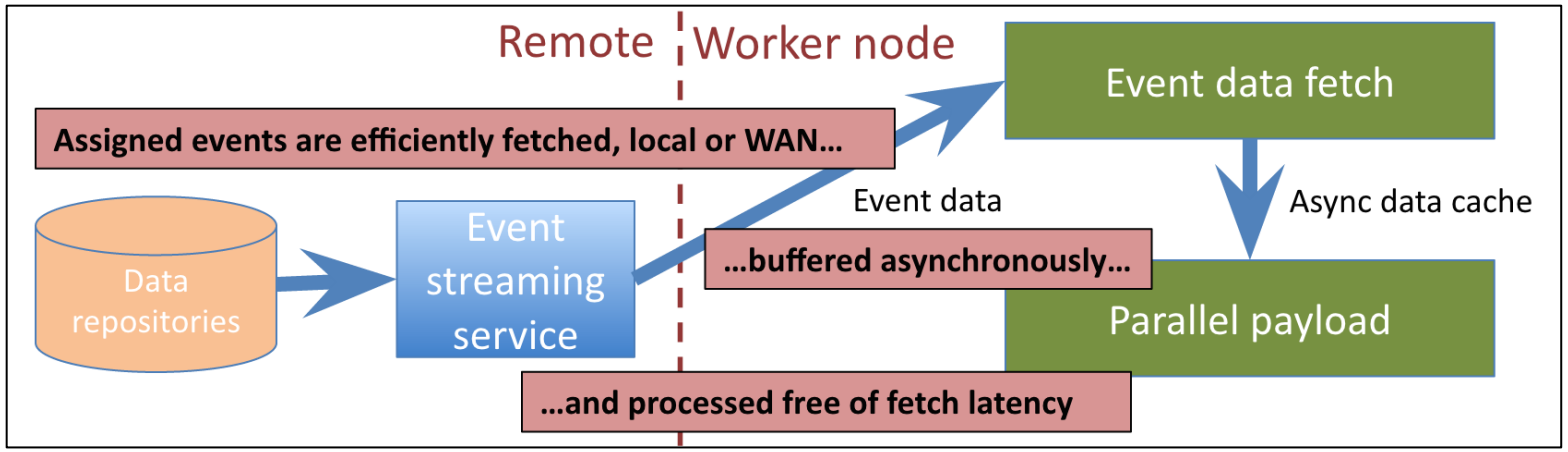
Event service port to ATLAS@Home well underway

ES is well suited to exploiting transiently available PCs

- Accommodating of machines disappearing suddenly
- Results streamed off incrementally, not trapped locally
- No need to shape job duration to the resource

⟹ ATLAS@Home talk, Track 3

The Next ES Development Phase: The Event Streaming Service (ESS)



- Event data delivery service on the content delivery network model
  - Delivering event data, intelligently, efficiently, transparently
  - With locality knowledge, leveraging available caching
- When data is delivered over the WAN, marshal it on the source side so only the bytes actually needed by the client traverse the net
  - Amenable to analysis of sparsely sampled data
- ES architecture allows ESS data delivery to be fully asynchronous with the processing, avoiding impact from WAN latencies
- **Work has started to deliver a first version in 2015**

Finally: What's Next for the Event Service

- **Bring it into simulation production**
  - Physics validation in progress (Yoda is already GO)
  - Early production targets are Amazon, US grid resources, HPCs (NERSC), ATLAS@Home
- Simulation is the 'low hanging fruit' – longer term, we would like to use the approach **for analysis as well**
  - Analysis is I/O intensive – requires more sophisticated event data transit over the net: the **Event Streaming Service (ESS)**
- SC14 demo was a great success, and a strong driver for progress (immovable milestone) -- planning aggressive goals again for SC15
  - Run with ESS, on more HPCs, possibly with multithreading
- ES effort is open to other experiments and a wider collaboration
  - Amenable to applications with finely partitionable workloads
  - Talk to us!

Postscript: Credits

A broad collaboration within ATLAS has applied their expertise in workload management, parallel payloads, parallel I/O, and HPC porting to build the Event Service and Yoda

- **BNL**: (Big)PanDA and its JEDI fine grained extension; HPC porting (ORNL Titan); object store
- **LBNL**: athenaMP parallel framework; HPC porting (NERSC Edison); Hive MT framework
- **ANL**: parallel I/O; WAN data access; HPC porting (ANL MIRA)
- **UTA**: (Big)PanDA; HPC pilot; HPC porting (ORNL Titan)
- **U Wisconsin - Madison:** Pilot extensions for ES and object stores; ES development & operation at NERSC