BigPanDA Work Package 1
The WMS Report

Kaushik De

BigPanDA TIM, Arlington

January 17, 2018

A collaboration of BNL, ORNL, Rutgers, UTA

DE SC0016280 from the Office of Advanced Scientific Computing Research

The University of Texas ARLINGTON™ BROOKHAVEN NATIONAL LABORATORY OAK RIDGE National Laboratory THE STATE UNIVERSITY OF NEW JERSEY RUTGERS

Introduction

- BigPanDA project is now in the middle of Year 2
- Work Package 1 (WP1)
  - Software developments for BigPanDA Workload Management System (WMS)
  - In support of BigPanDA deployment and operations at OLCF Titan
  - Objectives: improve efficiency and add new capabilities
  - In this talk we summarize status and plans

Technical Goals

- Improve backfill efficiency
- Enable utilization of ALCC allocation
- Transition from Pilot 1.0 to Pilot 2.0
- Deploy Harvester
- Deploy Event Service
- Integration with NGE – described in WP3 talk 11:45 and Matteo’s talk 14:30

Improving BackFill Efficiency

- For past ~1.5 years we have been using Pilot 1.0 at Titan
  - With a lot of custom modifications specific to HPC/Titan
  - Runs as edge service on DTN nodes to submit payloads on Titan
  - Used by ATLAS and other experiments/users at Titan
  - Works well – 10-20M Titan core hours per month
  - But needs to be replaced – this was a temporary hacked pilot
  - However, we needed to continuously tweak in 2017 for operations
    - To improve efficiency of backfill – achieved 95% overall utilization in November!
    - To reduce I/O bottlenecks – see next talk on WP2
    - To streamline job startup
  - See Danila’s talk at 16:15 today for results from ATLAS

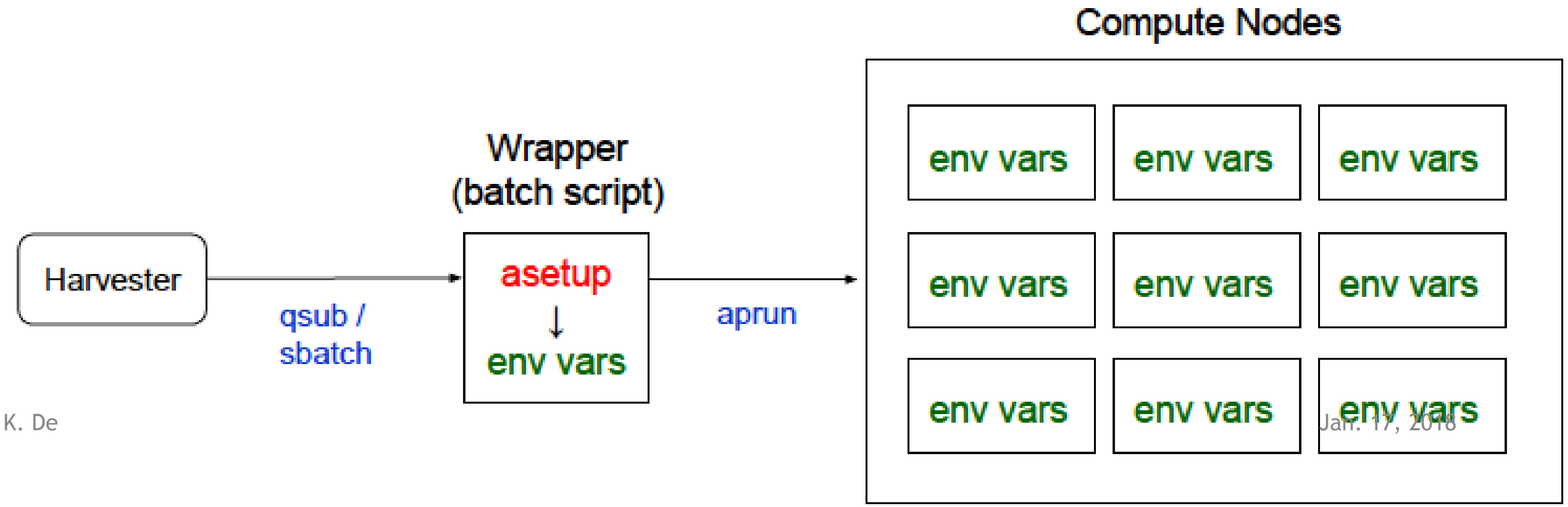
Enabling Utilization of ALCC Allocation

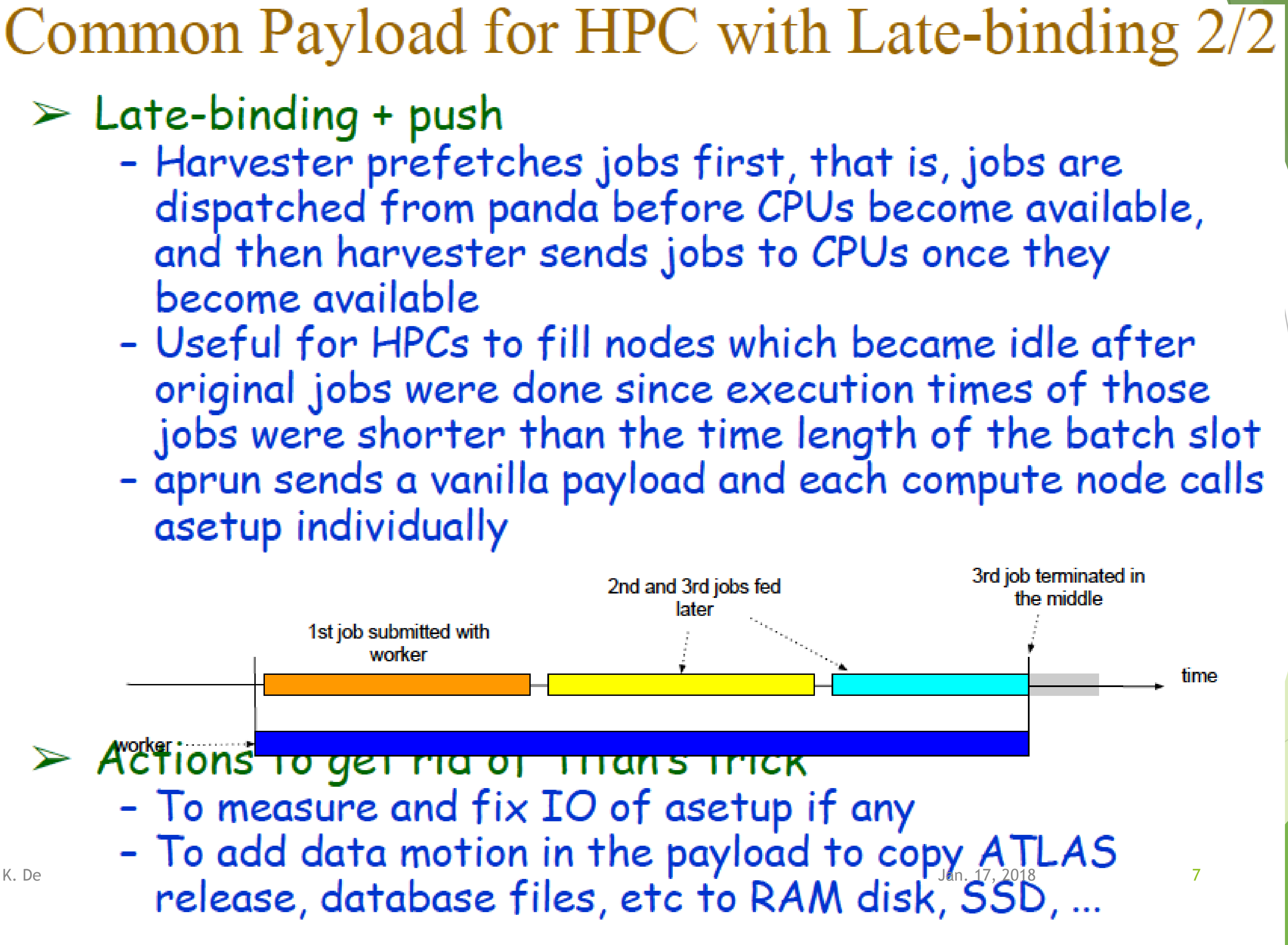
- 90M core hours of ALCC allocation for ATLAS on Titan
  - Required changes to pilot – scheduling policy is different than backfill
  - Started with custom tasks – queue wait time is not a problem
  - Working to reduce queue wait time through intelligent scheduling (late binding)
    - Use knowledge of queued jobs to schedule work as late as possible
    - Allows use of allocation for the highest priority task (fastest execution)
    - See talk by Titov Thursday morning
  - Work in progress – to be implemented in Harvester

Common Payload for HPC with Late-binding 1/2

- Trick in Titan's payload
  - asetup is called in the wrapper and all compute nodes inherit env variables
    - It was reported that calling asetup from each node was problematic for the Lustre file system at Titan/OLCF in terms of IO (GPFS for Summit)
  - Cannot change ATLAS release after the payload is submitted → Lack of late-binding

From
T. Maeno





Common Payload for HPC with Late-binding 2/2

- Late-binding + push
  - Harvester prefetches jobs first, that is, jobs are dispatched from panda before CPUs become available, and then harvester sends jobs to CPUs once they become available
  - Useful for HPCs to fill nodes which became idle after original jobs were done since execution times of those jobs were shorter than the time length of the batch slot
  - aprun sends a vanilla payload and each compute node calls asetup individually

2nd and 3rd jobs fed later
3rd job terminated in the middle
1st job submitted with worker
time
worker

- Actions To get rid of "Titan's Trick"
  - To measure and fix IO of asetup if any
  - To add data motion in the payload to copy ATLAS release, database files, etc to RAM disk, SSD, …

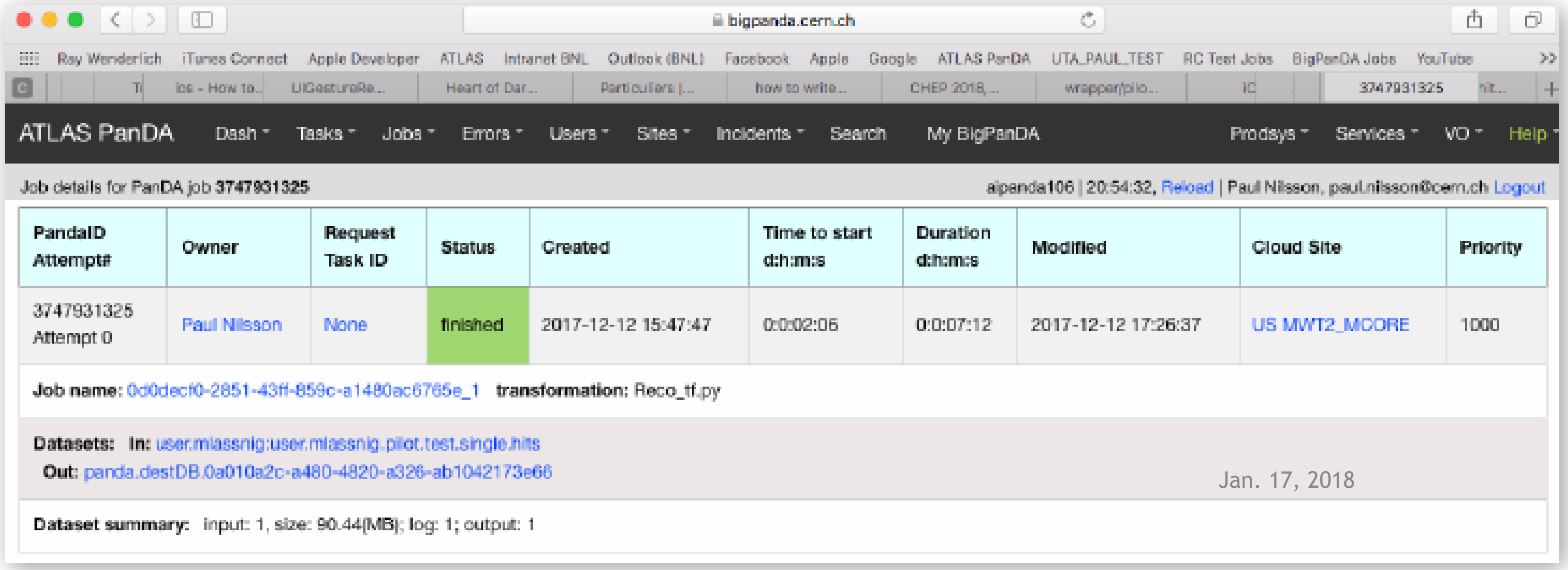
From
T. Maeno

Transition from Pilot 1.0 to 2.0

- Pilot 1.0 has been used by PanDA for >10 years
- Pilot 2.0 development is ongoing – not used in production yet
  - Some components are available and being tested
- In the meantime, HPC’s have been using miniPilot
  - MiniPilot - a lightweight Pilot 2.0 component available for development testing
  - Will start migration to full Pilot 2.0 in the Spring
- Pilot 2.0 was developed from scratch with HPC capabilities

Pilot 2 Project Overview

- Lots of progress since especially last summer
  - Containers, proper payload setup, payload + thread monitoring, information service, copy tools, event service, minipilot migration, server update, poster + publication, etc
  - Fewer Pilot 1 requests means more time for Pilot 2 development (!)
  - Active development / contributions from D. Benjamin, D. Oleynik, P. Svirin, W. Guan, A. Anisenkov, D. Drizhuk, T. Wegner, P. Nilsson
- A major goal was to have a test job run to completion on the grid before the end of 2017
  - https://bigpanda.cern.ch/job?pandaid=3747931325



Job details for PanDA job **3747931325**

aipanda106 | 20:54:32, Reload | Paul Nilsson, paul.nilsson@cern.ch Logout

PandaID Attempt#	Owner	Request Task ID	Status	Created	Time to start d:h:m:s	Duration d:h:m:s	Modified	Cloud Site	Priority
3747931325 Attempt 0	Paul Nilsson	None	finished	2017-12-12 15:47:47	0:0:02:06	0:0:07:12	2017-12-12 17:26:37	US MWT2_MCORE	1000

Job name: 0d0decf0-2851-43ff-859c-a1480ac6765e_1 **transformation:** Reco_tf.py

Datasets: **In:** user.mlassnig:user.mlassnig.pilot.test.single.hits

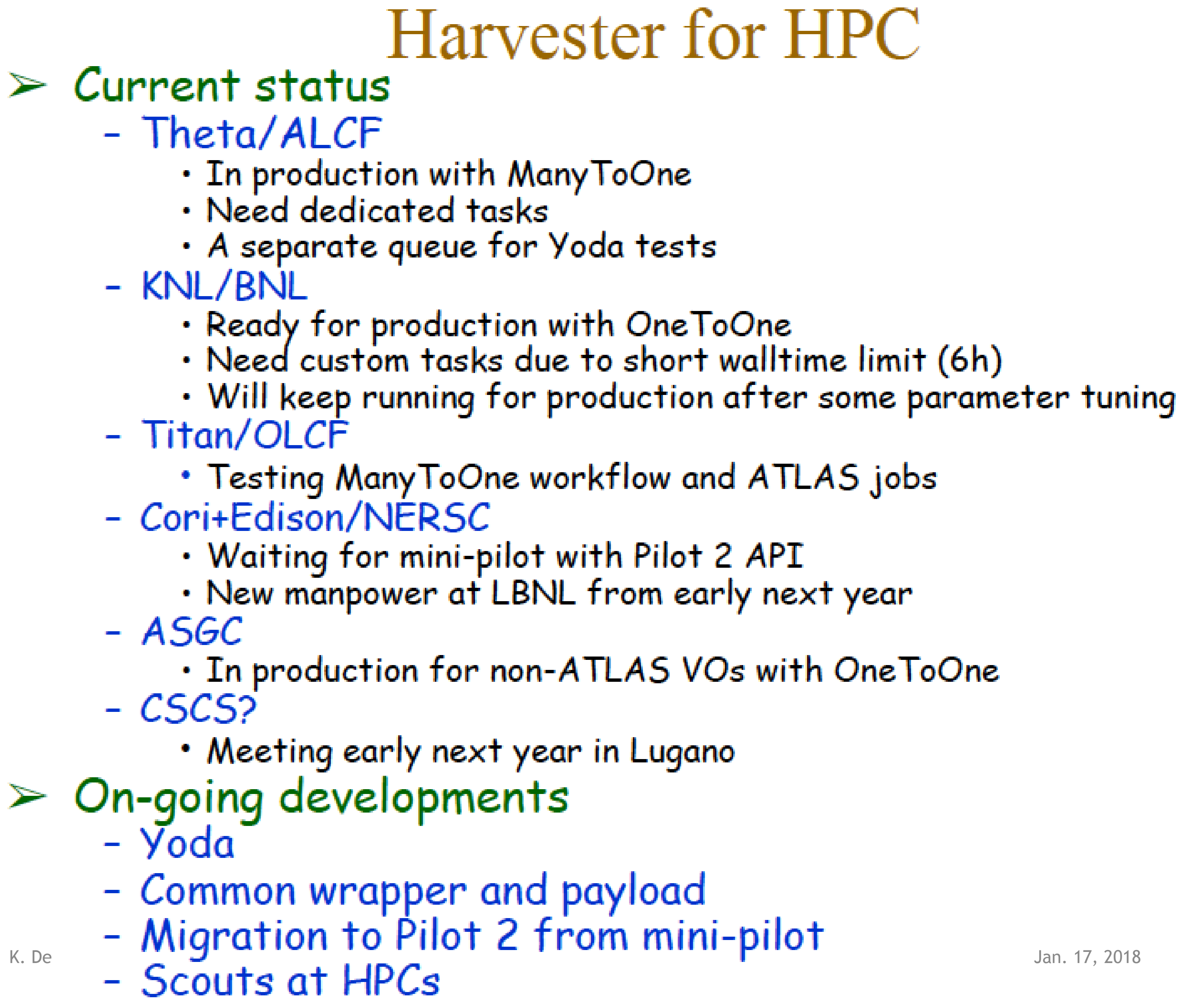
Out: panda.destDB.0a010a2c-a480-4820-a326-ab1042173e66

Dataset summary: input: 1, size: 90.44(MB); log: 1; output: 1

From
P. Nilsson

Harvester

- Harvester is a new component of PanDA
  - Autonomous light-weight stateless service between PanDA server and Pilot
  - Provides intelligent pilot scheduling capabilities
  - Works with (and shares components with) Pilot 2.0
  - Crucially important for HPC's – Titan
  - In active development, testing and prototyping at Titan
- See talk by Danila and Pavlo 14:00



Harvester for HPC

- Current status
  - Theta/ALCF
    - In production with ManyToOne
    - Need dedicated tasks
    - A separate queue for Yoda tests
  - KNL/BNL
    - Ready for production with OneToOne
    - Need custom tasks due to short walltime limit (6h)
    - Will keep running for production after some parameter tuning
  - Titan/OLCF
    - Testing ManyToOne workflow and ATLAS jobs
  - Cori+Edison/NERSC
    - Waiting for mini-pilot with Pilot 2 API
    - New manpower at LBNL from early next year
  - ASGC
    - In production for non-ATLAS VOs with OneToOne
  - CSCS?
    - Meeting early next year in Lugano
- On-going developments
  - Yoda
  - Common wrapper and payload
  - Migration to Pilot 2 from mini-pilot
  - Scouts at HPCs

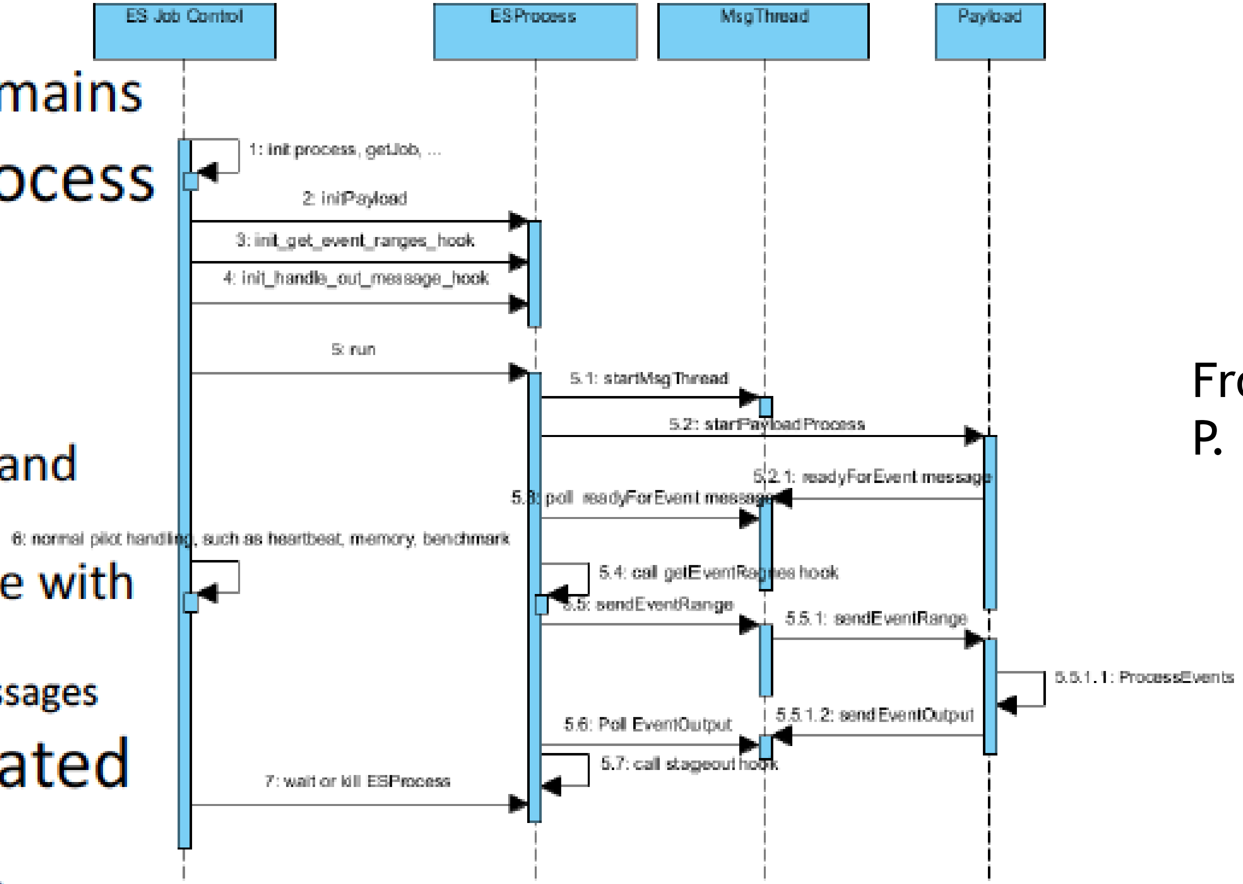
From
T. Maeno

Event Service

- New service for PanDA
  - Allows processing of small chunks of data – file size becomes irrelevant
  - Very powerful in conjunction with event streaming service – being developed
  - Allows preemption of jobs without loss of the work done till preemption
  - In early production stage
  - In use for 25k-50k cores for ATLAS daily
  - High priority for deployment at Titan – but waiting for Pilot 2.0 and Harvester (Yoda in previous slide)

Event Service

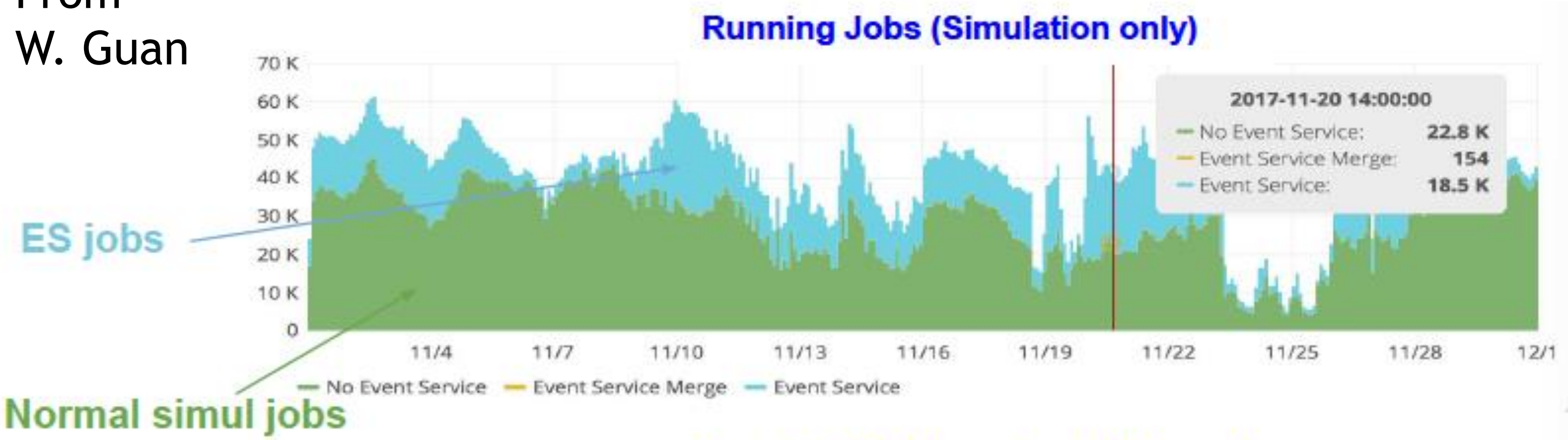
- Most of the event service code currently implemented as unit tests
  - Some development remains
- Main event service process ESProcess is finished
  - Class similar to Python Process
    - Easy to handle yampl and AthenaMP messages
    - Hooks to communicate with other processes
      - Event ranges, out messages
- Currently being integrated with Pilot 2 in generic workflow with ES hooks
  - Should be ready in January


ES Job Control
ESProcess
MsgThread
Payload
1: init process, getJob, ...
2: initPayload
3: init_get_event_ranges_hook
4: init_handle_out_message_hook
5: run
5.1: startMsgThread
5.2: startPayloadProcess
5.2.1: readyForEvent message
5.3: poll readyForEvent message
6: normal pilot handling, such as heartbeat, memory, benchmark
5.4: call getEventRanges hook
5.5: sendEventRange
5.5.1: sendEventRange
5.5.1.1: ProcessEvents
5.5.1.2: sendEventOutput
5.6: Poll EventOutput
5.7: call stageout hook
7: wait or kill ESProcess

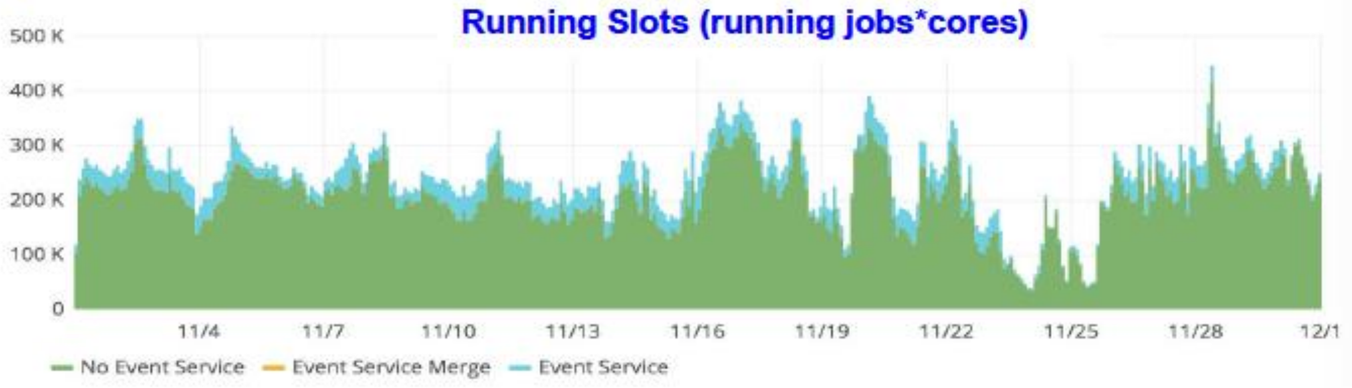
From P. Nilsson

How much Event Service(ES VS normal Simul)

From
W. Guan



Running Jobs (Simulation only)



Running Slots (running jobs*cores)

- EventService vs normal simul:
  a. Number of jobs: up to 50% are ES
  b. Events: ES is ~17% of total Simul(normal simul is using MCORE. ES is using both MCORE and SCORE. So 50% jobs produces 17% of events)
  c. Ramping ES slowly

- ES framework is scaling well (running jobs are comparable to the “standard” one)
- We are grabbing a lot of single core resources which might have been wasted otherwise

Schedule and Plans

- The 5 development tracks described + NGE are the highest priority
  - Additional areas include Jumbo jobs, Task Management, Data movers… in progress
  - The BigPanDA team is fully integrated with ATLAS PanDA development team
  - However, support of continuous operations is higher priority
- Progress is limited by developer effort available
- All 5 areas described are expected to be operational in Spring 2018