Harvester for US HPC

Harvester at US HPCs

- ALCF/Theta
  - Full production with Jumbo jobs + Multi Workers + No dedicated tasks
- NERSC/Cori
  - Full production with Jumbo jobs + Multi Workers + No dedicated tasks
- OLCF/Titan
  - ALCC
    - Full production with traditional (non event service) jobs + Many-to-One (to combine many jobs to a single batch submission) + dedicated tasks
  - Backfill on a large scale
    - Full production with traditional jobs + Many-to-One + dedicated tasks
  - Backfill with event service
    - Running production on a limited scale with Jumbo jobs + Multi Workers + No dedicated tasks
    - Limited due to scalability issues of Lustre FS at OLCF until Yoda is improved to allow usage of RAM disk

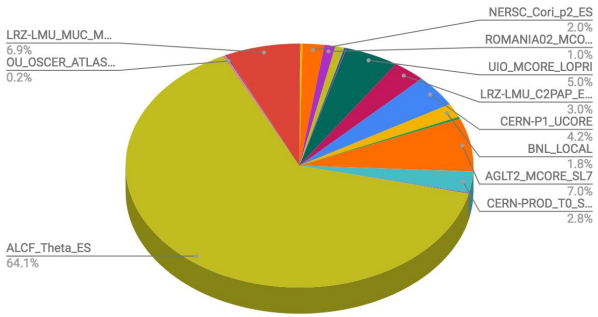
Dedicated Tasks

- The number of events per job in the task is so different from that of other tasks and/or all jobs in the task are assigned to a site
  - To have proper walltime for each batch submission with traditional jobs
  - Required for special resources
- E.g. for Titan
  - Backfill with traditional jobs needs short jobs to minimize a waste of CPU due to preemption
  - ALCC needs to combine many jobs with similar execution time to reduce idle nodes in a single batch slot
- Why not good?
  - Need human intervention
    - Prod managers have to look for proper tasks and tweak parameters
  - Preassignment of workload
    - Suboptimal resource usage from global point of view

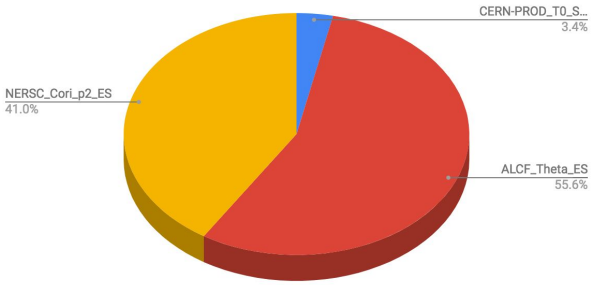
Jumbo Jobs 1/2

- Jumbo jobs in nutshell
  - Jumbo jobs ~= big event service jobs
  - Workload sharing among pledged and opportunistic resources
  - Flexible and dynamic partitioning of workload to be optimal for each resource
  - On-demand delivery of workload to resources
- Multiple jumbo jobs from a single task
  - Distributed to many PQs with useJumboJobs in catchall
  - Multiple jumbo jobs per PQ from a single task
- Events in a single task are shared by jumbo jobs and normal event service jobs (co-jumbo jobs)
  - Each event is exclusively processed by a one job

Processed events in the task per PQ



Processed events in a job per PQ



Jumbo Jobs 2/2

- Each jumbo job can process any events in the task
  - 0 <= jumboJob.nEvents <= task.nEvents (e.g. 10M)
- Each co-jumbo job can process a subset of events in the task
  - 0 <= coJumboJob.nEvents <= traditionalJob.nEvents (e.g. 1k)
  - The number of events in the subset is similar to that of traditional jobs → Good to run on pledged resources
- (co-) jumbo jobs are empty in the beginning and then are filled with events dynamically
- One jumbo job can be shared by multiple workers (= pilots)
  - All workers are running at the same resource
  - Workers don't have to be submitted at the same time
  - Holes in the batch system can be filled quickly
    - Workers can be submitted as soon as holes pop up
  - Stage-out per worker during the job is still running

PandaID	Owner WG	TaskID	Mode	Cores	Status	Substate	Created
4146376901	dsouth / AP_TOPQ	15975812		128	running	running	2018-11-15 12:47:12

Job type: simul **Job transform:** Sim_tf.py **Dispatched event states:** finished(255748) done(12198) merged(8059) sent(1133019)

Events were being staged out to be merged in the subsequent merge step while the job was still running

Jumbo Jobs To Production

- Issues to be addressed
  - Inefficiency in the current workflow
    - No real showstopper
    - Need continuous and incremental improvements anyway
  - Monitoring
    - Very good shape thanks to Tatiana and Sergey P
    - Hiding bloody details and making the workflow understandable
  - Slow task completion
    - New default task parameters for event service tasks simplified the workflow a lot and removed a hopelessly long tail of task completion
    - To be understand why still slower than normal tasks
    - New monitoring is helping us to identify and improve slowness
  - Automation
    - To automatically enable/disable jumbo jobs in event service tasks without manual intervention
      https://twiki.cern.ch/twiki/bin/view/PanDA/PandaJEDI#Jumbo_job_switcher
    - Running in the dry-run mode
  - Technical validation
    - Details in Doug's talk

Plans

- The original goal was to bring HPC+Jumbo into production in Dec even with a limited scale with some sort of automation
  - Already processing real production tasks with jumbo jobs on 3 US HPCs in addition to conventional event service sites
    - Only one task with jumbo in the system at a time
    - With manual task conversion
    - Should wait results of technical validation before full throttle
- Not realistic to add big changes just before Christmas
- Possible scenario to full production (TBD next week)
  1. Enable the automation to always have one or two tasks with jumbo in the system, for example, in Jan 2019
     a. Total workload for event service is unchanged
     b. Still only low-prio tasks for event service tasks to avoid interference with main physics activities
     c. Jumbo jobs don't block tasks at HPCs as long as co-jumbo jobs are consumed somewhere
  2. Increase the priority limit on event service tasks once causes of slow task completion are solved and technical validation is successfully done
  3. Improve the workflow and the automation algorithm

Long term

- Current ATLAS production workloads like Geant4 simulation will not be fit very well to the next generation of LCF machines
  - Summit, Aurora
  - Titan end of life in next 9 months
- Ongoing discussion about GPU usage at HPCs through Harvester
  - Hybrid of CPU and GPU workloads in a single batch submission
  - Event service and/or job late-binding to adjust different execution time of two workloads
  - Serious studies on actual GPU payloads are required
  - More discussion in next week