# Distributed Computing Challenges

Tadashi Maeno (BNL)
on behalf of
Harvester team and HPC devops

ATLAS Sites Jamboree and HPC strategy,
6 March 2019, CERN, Switzerland

# Acknowledgements

Doug Benjamin, Danila Oleynik, Wen Guan, Tatiana Korchuganova, Sergey Padolski, Fernando Harald Barreiro Megino, Aleksandr Alekseev, Taylor Childers, Nicolo Magini, and Harvester team and HPC devops

# Special Characteristics of HPCs

- Preemptible or very short walltime limit
  - Backfill, HPC site policies, low priorities to save allocation, ...

  → Event service or very short jobs
- Large number of cores per job and/or submission
  - Lower limit on the number of nodes per submission, preference to MPI, monolithic whole node scheduling, …

  → Packing of multiple normal jobs to a single submission payload (ManyToOne) or Event service with large workload (Jumbo jobs)
- Queue limit
  - The number of submission is limited in the batch queue

  → ManyToOne or Jumbo jobs
- Network-less compute nodes

  → Communication with PanDA through a delegation service like Harvester and aCT
- No local CE
  - Not integrated with the grid

  → ssh through ARC CE, via CONNECT, direct ssh access, or harvester instance inside HPC's local network
- No local Rucio Storage Element
  - Not fully integrated with ATLAS DDM
  - Direct access to remote RSE from compute nodes doesn't work in many cases due to limited network access

  → 3rd party service (e.g, Globus Online, FTS, ...), SSH FS, or custom data transfer mechanism in local harvester instances

# Active HPCs

| PQ name | Normalized total running cores | Total running cores | Cores per job | nJobs | HS06 | Maxtime | Event service | Batch submission |
|---|---|---|---|---|---|---|---|---|
| ALCF_Theta_ES | 24576 | 81920 | 81920 | 1 | 3 | 0 | Y | 6 |
| ORNL_Titan_MCORE | 6441 | 10224 | 16 | 639 | 6 | 7200 | | 8 |
| praguelcg2_IT4I_MCORE | 6404 | 3432 | 24 | 143 | 19 | 43200 | Y | = nJobs |
| HPC2N_MCORE | 5282 | 3830 | 12 | 319 | 14 | 0 | | = nJobs |
| NERSC_Cori_p2_ES | 3250 | 13600 | 13600 | 1 | 2 | 0 | Y | 6 |
| OU_OSCER_ATLAS_UCORE | 2710 | 1906 | 16 | 119 | 14 | 259200 | | = nJobs |
| CSCS-LCG2-HPC_MCORE | 1522 | 1322 | 8 | 165 | 12 | 172800 | | = nJobs |
| UIO_MCORE_LOPRI | 872 | 632 | 8 | 79 | 14 | 0 | Y | = nJobs |
| UIO_MCORE | 615 | 446 | 7 | 63 | 14 | 0 | | = nJobs |
| LRZ-LMU_MUC_MCORE1 | 24 | 40 | 40 | 1 | 6 | 172800 | Y | = nJobs |
| NSC_MCORE | 17 | 12 | 4 | 3 | 14 | 0 | | = nJobs |
| RRC-KI-HPC2 | 5 | 3 | 1 | 3 | 16 | 259200 | | = nJobs |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| FZK-LCG2 (Grid example) | 13911 | 11078 | 8 | 1390 | 13 | 345600 | | = nJobs |

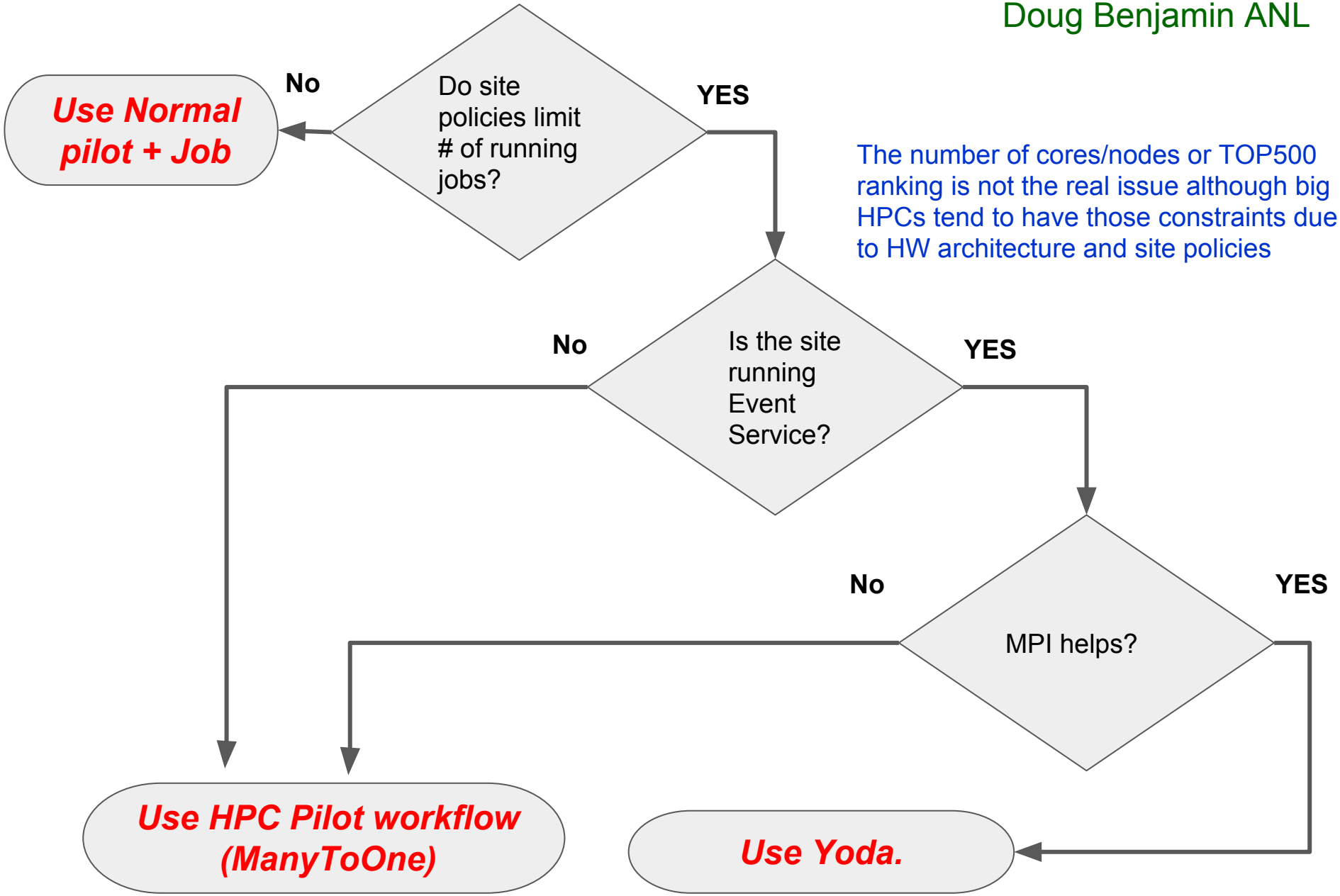
- A snapshot in Feb, not historical average
- Normalized = sum(actualCoreCount)*HS06/10

# Categories and Configurations of HPCs

- Grid-friendly HPCs
  - CE (equivalent) + RSE + network access + large or indefinite queue limit
    - E.g. OU_OSCER_ATLAS_UCORE, CSCS-LCG2-HPC_MCORE, CONNECT_*, …
    - Normal grid jobs
- HPCs working like CPU farms behind SSH
  - SSH gateway + network-less + large or indefinite queue limit
    - E.g. praguelcg2_IT4I_MCORE, HPC2N_MCORE, …
    - aCT + SSH FS, or Harvester + aCT/ARC/SSH plugins
    - Normal grid jobs, or event service jobs for preemptible or short walltime
    - PUSH + delegated heartbeats
- "Special" HPCs (US/DOE : Theta, Cori, and Titan)
  - Multi-factor authorization + network-less + tight queue limit + MPI for efficiency
    - Local harvester instances
    - Special payload
      - Yoda + Jumbo jobs for Theta and Cori
      - ManyToOne for Titan
        - File system at OLCF doesn't scale with many files produced by event service
    - Custom data motion
      - Globus Online + Rucio/Globus dual endpoint
      - Rucio download/upload from/to remote RSE

# When to use Yoda?

Doug Benjamin ANL




The number of cores/nodes or TOP500 ranking is not the real issue although big HPCs tend to have those constraints due to HW architecture and site policies

# Integration of HPCs with Production

- Grid-friendly or SSH HPCs
  - No special treatment
  - Not so "special" in terms of workload management since
    - the number of jobs is equal to the number of batch submissions, and
    - job sizes are the same as normal grid jobs
- Special HPCs
  - Custom tasks with short jobs for Titan
    - Titan cannot scale with event service due to a scalability issue of Lustre file system at OLCF
    - No major development for Titan since it's going to retire soon
  - Conversion of normal simulation tasks to generate jumbo jobs at Theta and Cori while co-jumbo jobs (normal event service jobs) at grid sites
    - No custom tasks
    - Conversion after prod manages submit tasks
    - To be fully automated to get rid of any manual interventions
    - Events can be processed at grid sites even if HPCs and/or local harvester instances are down
    - Mixture of ES and non-ES jobs in a single task

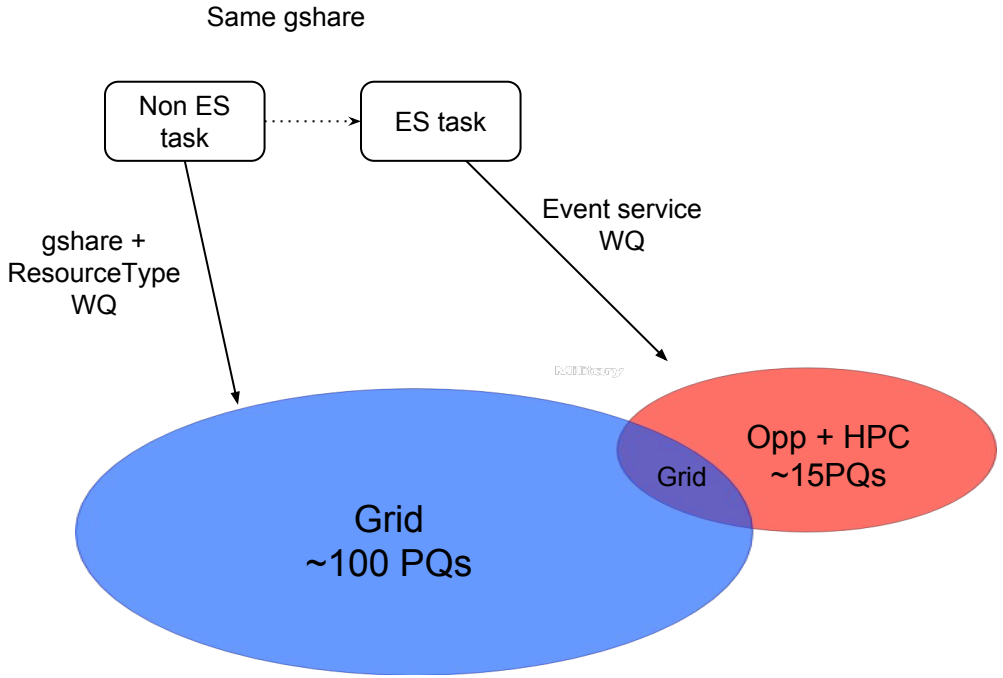
# Checklist before Yoda+Jumbo in Production

- Better monitoring for Event service
  - Done
    - Performance improvements
    - Input-based views to show what's actually going on
    - Lightweight job status summary with a capability of switching between total and split statistics
- Getting rid of slow tails of Event service task completion
  - Done
    - Next slide
- Event duplication check
  - Done
    - Next slide
- Automation
  - Half done
    - Implemented task manipulator
    - Partially running
    - Details later

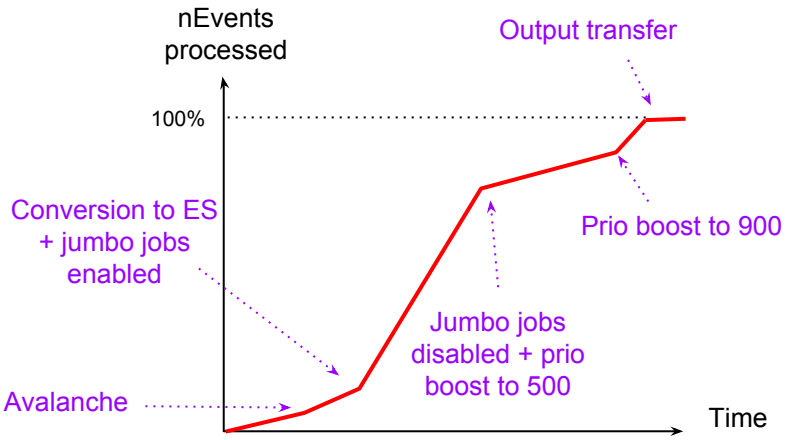
# Issues in Event Service Tasks

- Slow task completion
  - The most problematic when using ES in production
  - Actions
    - Removal of redundant and too many job attempts
    - Reduction of attempts in merge step
    - Automatic recovery of corrupted zip files
    - Protection against missing co-jumbo jobs due to a race condition
    - Gradual priority boost
    - Using only MCORE PQs for high prio ES jobs
    - Using a subset of grid resources to speed up task completion
- Event duplication
  - Some EVNT files already have duplicated event numbers when they were generated with filters (fixed in rel 21.6)
    - "Unique" events in terms of physics
  - Two tasks had real event duplication due to an AthenaMP bug when jumbo jobs set wrong values to --skipEvents
    - Normal event service jobs are not affected
  - Actions
    - Patch for AthenaMP
    - Protection in JEDI when generating jumbo jobs

# Automation with Task Manipulator 1/2




- ES and non-ES tasks in the same global share
  - No special gshare for ES
- Separate WQs to submit jobs independently
  - Not so many resources shared by non-ES and ES
- Resources for non-ES jobs
  - Many grid PQs shared with other gshares
- Resources for ES jobs
  - Only opportunistic PQs and HPCs when task priority < 900
    - Rather small but dedicated to ES
  - Additional grid PQs (~20) when task priority ≥ 900
    - Generally very good PQs with jobseed=eshigh
- ES tasks don't change gshare to Express even if priorities are boosted




Typical event processing with Jumbo jobs

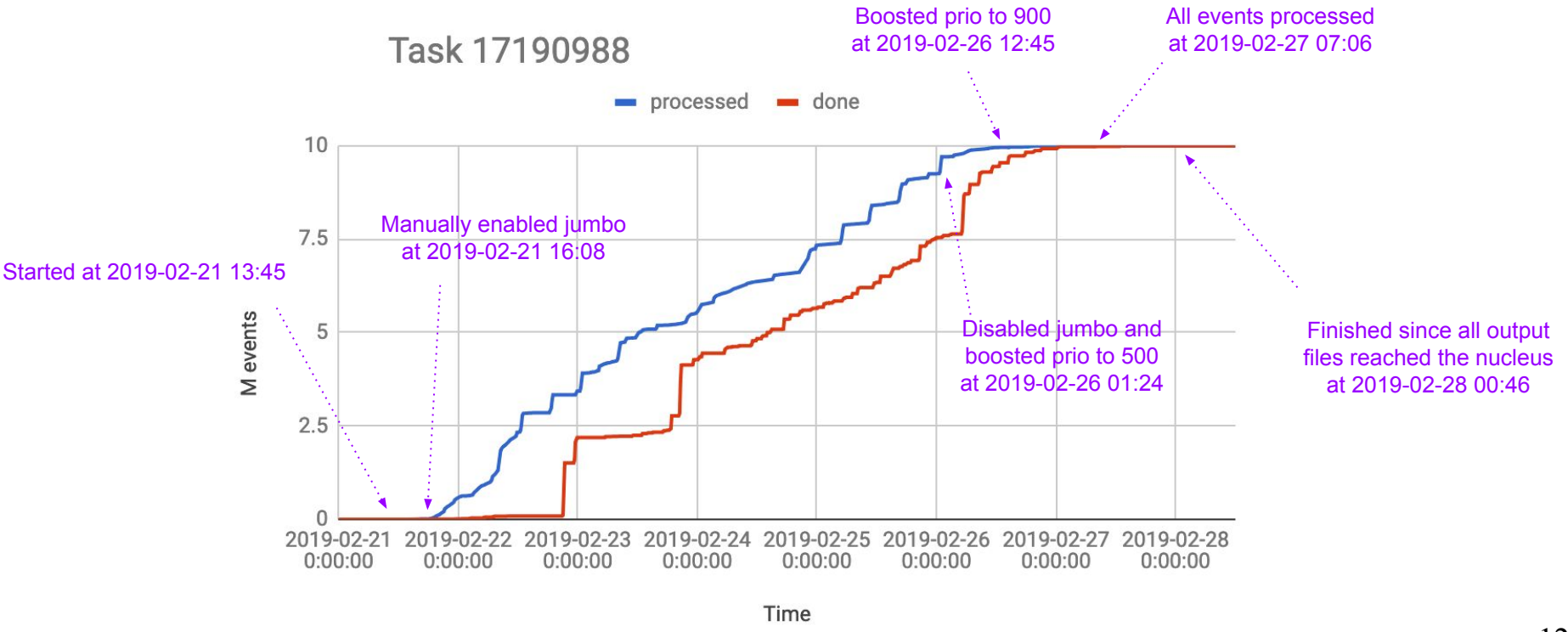
Only grid in scouting and just after avalanche
→ Opp+HPC
→ Only Opp not to send small jobs to HPC
(prio=500 to get all opp resources)
→ Opp + a subset of grid to shorten the tale
(prio=900 to use the additional grid resources)

# Automation with Task Manipulator 2/2

- All parameters, JUMBO_*, are defined in gdp config https://prodtask-dev.cern.ch/gdpconfig/
- A normal simulation task can be converted to ES + Jumbo when
  - the task has more than JUMBO_MIN_EVENTS_ENABLE unprocessed events
  - the total number of events to be processed by jumbo job tasks is less than JUMBO_MAX_EVENTS
  - the total number of jumbo job tasks is less than JUMBO_MAX_TASKS, and
  - one or more PQs with useJumboJob in catchall deploy ATLAS release/cache
- A jumbo job task disables jumbo jobs and increase priority to 500 when the number of unprocessed events is less than JUMBO_MIN_EVENTS_DISABLE
- A jumbo job task increases priority to 900 when
  - JUMBO_PROG_TO_BOOST % of events are processed, and
  - the number of files to be processed is less than JUMBO_MAX_FILES_TO_BOOST
- Task manipulator runs every 10 min
  - Log https://es-atlas.cern.ch/kibana/goto/b3fa05b7b62fda2b8a2fb5d885d934ba
- Some intelligent mechanism to automatically adjust JUMBO_* parameters in the future

# Current Status and Near-term Plans 1/2

- No event duplication in the latest task after applying the patch to AthenaMP
- Tails of ES task completion have been shortened
- Automatically boosting priorities and disabling jumbo jobs
- The mechanism to enable jumbo jobs is running in the dry-run mode and waiting for the final sign off
  - Prod manages still have to submit dedicated tasks
- Doug on a 24x7 shift

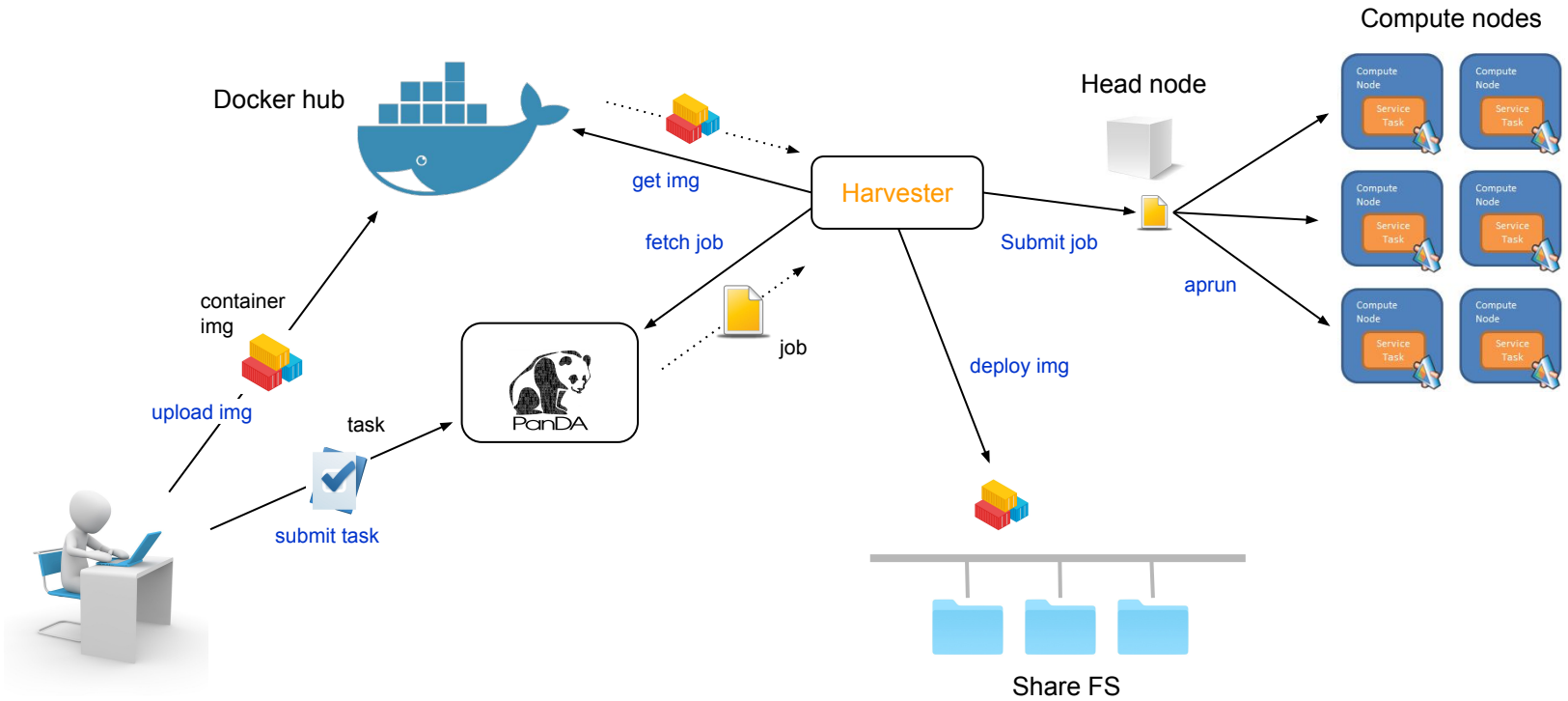


# Current Status and Near-term Plans 2/2

- Ready to go to normal operation mode with full automation
  - DPA, US HPC ops, ADCOS as L1 while experts to the rescue if necessary
  - Procedures
    1. Production managers submit many large tasks in one go
       - With very low task priorities
       - Without any special task parameters
       - Not one by one on request base
    2. Those tasks will sciently sit in the system after scouting due to low priorities
       - Can work as fillers when the normal grid sites become empty
    3. Task manipulator automatically converts some tasks to ES+Jumbo
    4. The task manipulator logging should be periodically checked (for now) to see if there is an enough pool of tasks
       - Could make an overview page and alarms based on logging messages to show # of jumbo job tasks, # of potential tasks, # of events in jumbo tasks, …
    5. Ask production managers to submit more backlog, ask HPCs to deploy proper releases, or ask DPA to adjust JUMBO_* parameters

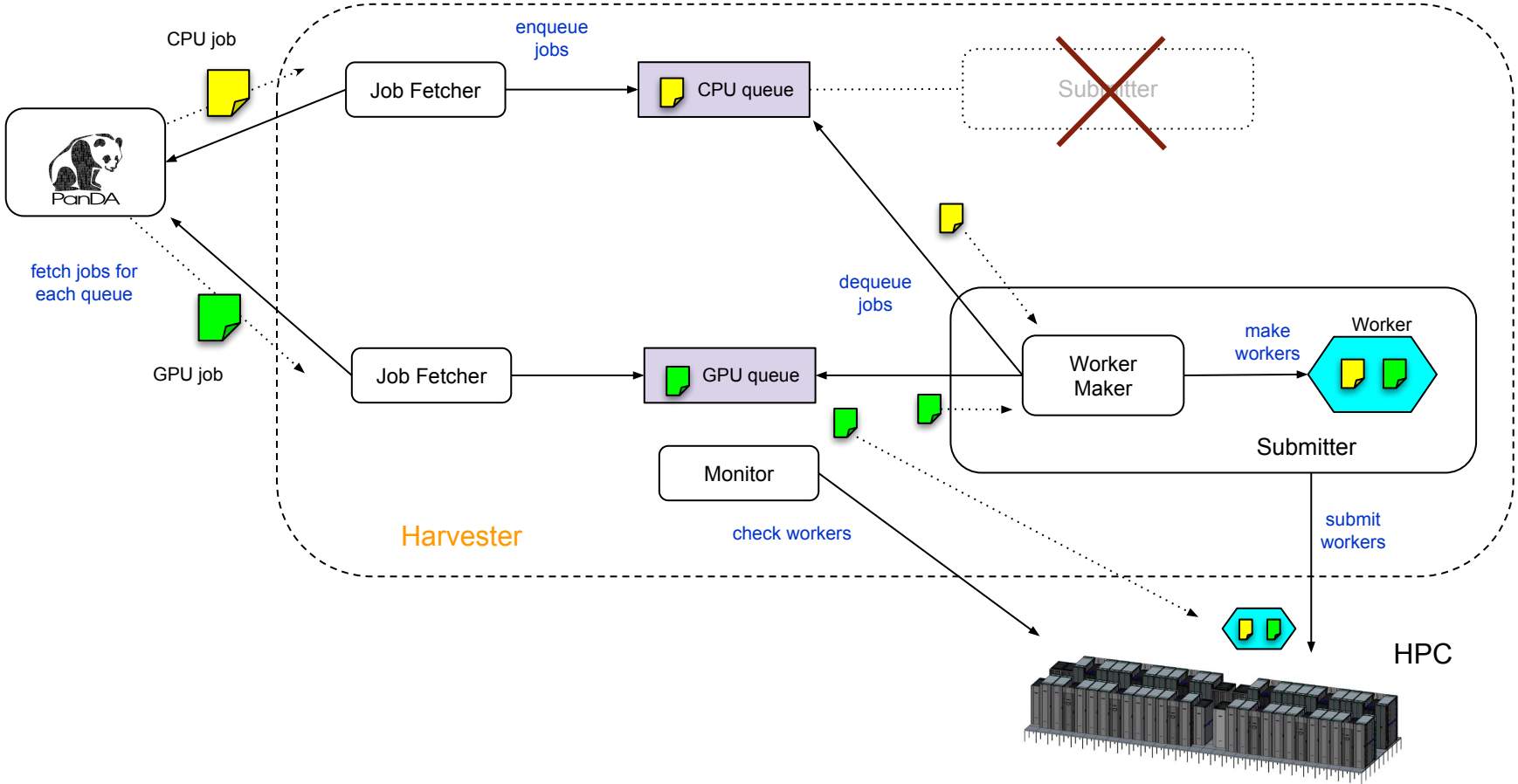
# Future Plans 1/2

- HPC + ML + MPI
  - Distributed training on HPC through PanDA and Harvester
  - Multi-node payload with MPI to be prepared by users for now
    - Might provide a common MPI framework in the future
  - On-demand deployment for user container images
  - New job attribute to specify the number of nodes in addition to the number of cores
  - Container names from job attributes
  - Hints from users to shape jobs with proper execution time
    - E.g, Seconds per training data for each model



# Future Plans 2/2

- Workload co-location
  - CPU-only payload is disallowed to use the next generation of large US/DOE HPCs
  - Hybrid payload, combination of CPU (ATLAS MC simulation) jobs and GPU (e.g. ML) jobs, due to monolithic node-level scheduling
    - CPU and GPU jobs cannot be submitted independently due to whole node scheduling
    - Single payload from HPC's point of view, but CPU are shared by CPU and GPU jobs
    - Assuming that GUP jobs use a part of CPU but not all
  - Job-late binding and/or event service to align execution time



# Conclusions

- Rather straightforward to integrate grid-friendly or SSH HPCs with production
- Large US/DOE HPCs are tricky due to site policies, HW architecture and configuration
- Yoda + Jumbo jobs + Task manipulator are ready for full production to get rid of any manual intervention
  - ADC sign-off?
  - Theta and Cori will be fully integrated
  - Normal operation mode with DPA, US HPC ops, and ADCOS
  - Further optimization and cosmetics to be done as a next step
- Most of Titan resources will run with custom tasks as they are since it retires soon
  - Limited manpower to focus on R&D activities at OLCF/Summit
- New challenges for the next generation of US/DOE HPCs
  - Usage of accelerator is a must
    - New GPU-oriented applications via PanDA/Harvester
    - Co-location of GPU-oriented and CPU-oriented applications