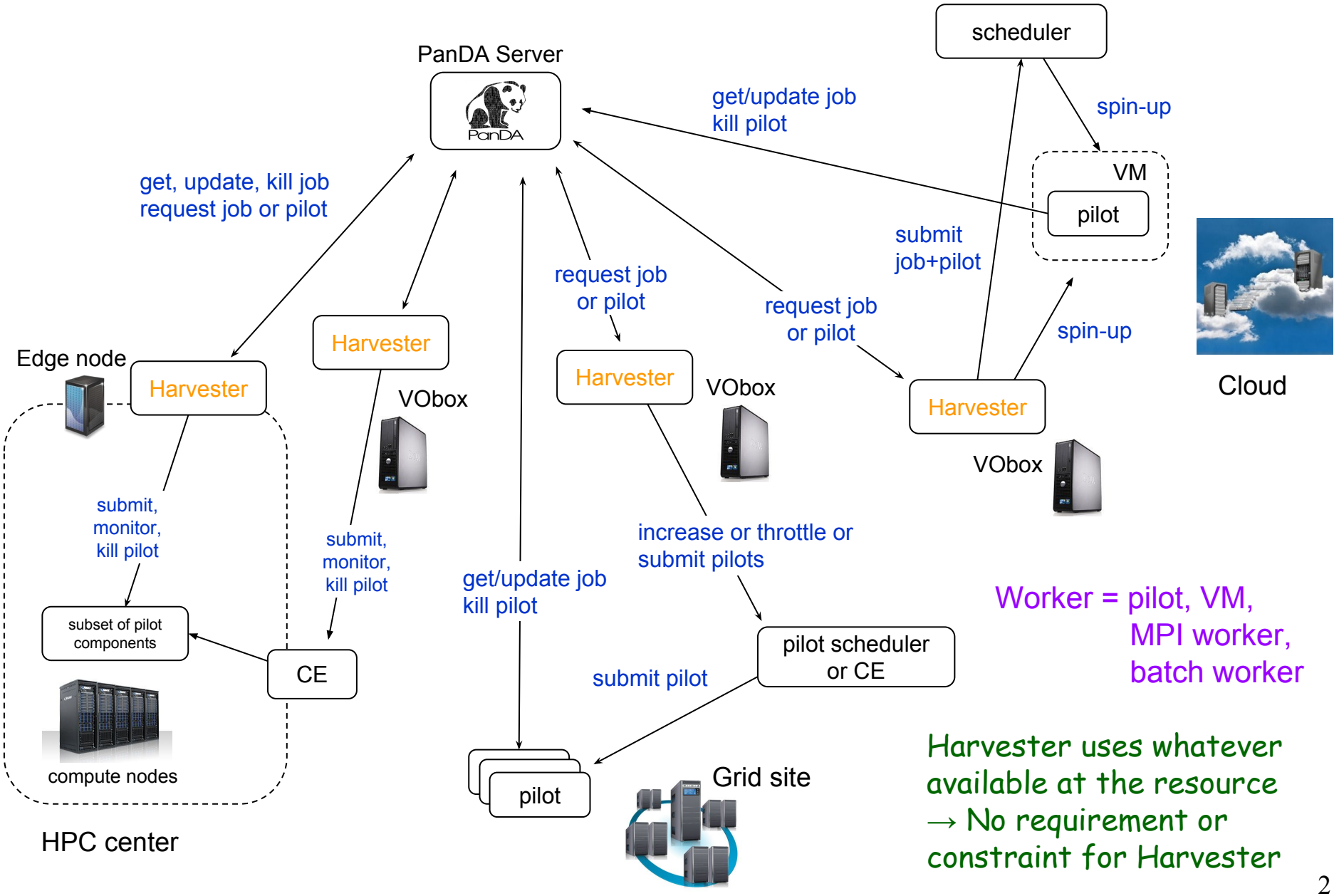
Harvester

Tadashi Maeno (BNL)
on behalf of Harvester team

BigPanDA TIM,
26 April 2018, BNL, USA

Schematic View



Worker = pilot, VM,
MPI worker,
batch worker

Harvester uses whatever available at the resource → No requirement or constraint for Harvester

Current Status with ATLAS Resources

N.B.

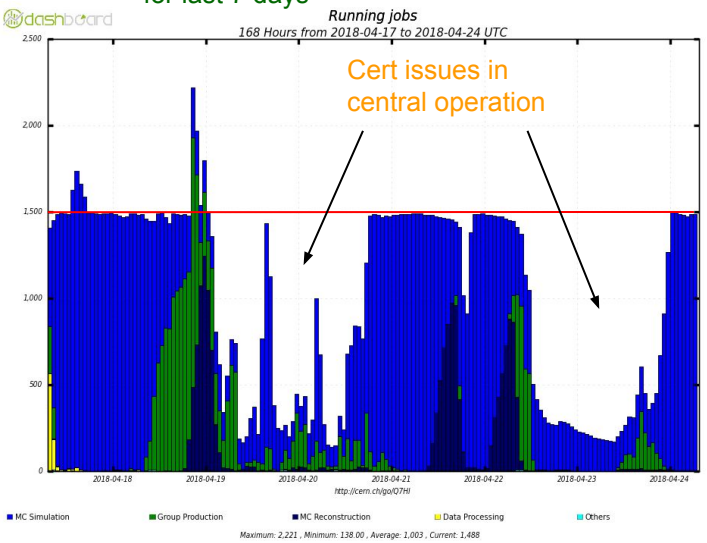
Danila's talk for details at OLCF

Pavlo's talk for beyond ATLAS

Harvester for the Grid

- Current status
  - Successfully demonstrated to submit workers (pilots) to hundreds of ATLAS queues except a couple of queues with GT5 CEs which retire soon
  - Successfully ran a few thousands of pilots concurrently at CERN
  - Testing a capability of dynamic resource partitioning at CERN and Taiwan
  - Migration for large scale production is ongoing at BNL
- Plans
  - Full migration to Harvester
  - Dynamic resource partitioning in production
  - A single submission engine
  - Consolidation of resource-specific queues
  - Better site description

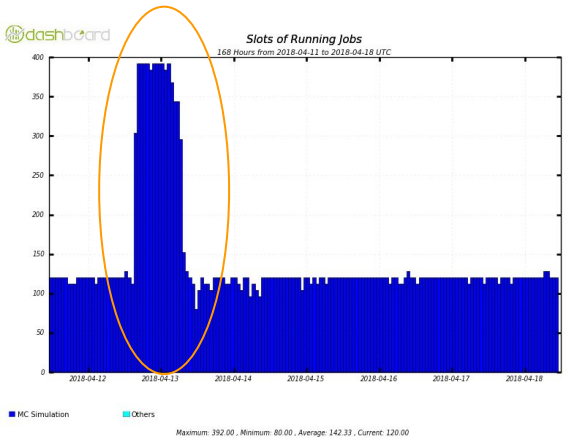
Running jobs at BNL harvester queue for last 7 days



Harvester for Cloud

- Current status
  - CERN + Leibniz + Edinburgh resources with 1.2k CPU cores in production
- Two major developments
  - Condor-based for ATLAS High Level Trigger (HLT) CPU farm, aka Sim@P1
    - With 50k cores and limited network bandwidth per node
    - HLT experts have a "button"
      - Turn it on to release the resource to PanDA when it is not used for online trigger
      - Turn it off to immediately take the resource back when it is used for online trigger
    - The resource is not always available, but it behaves like a static cluster once it is given to PanDA
    - Workload provisioning to assign enough jobs to the resource before the resource becomes available
  - Using native cloud API for GCE and EC2
    - Use-cases
      - In context of the data ocean project
        - Pure google : GCE + Google storage + GCE API
      - Openstack instance with EC2 API at Taiwan for non-ATLAS experiments
    - Embedded a mechanism in harvester for lifetime management of VMs
    - HTTP(S) based communication between harvester and workers
    - Large scale demonstration and new cloud API for bulk operation

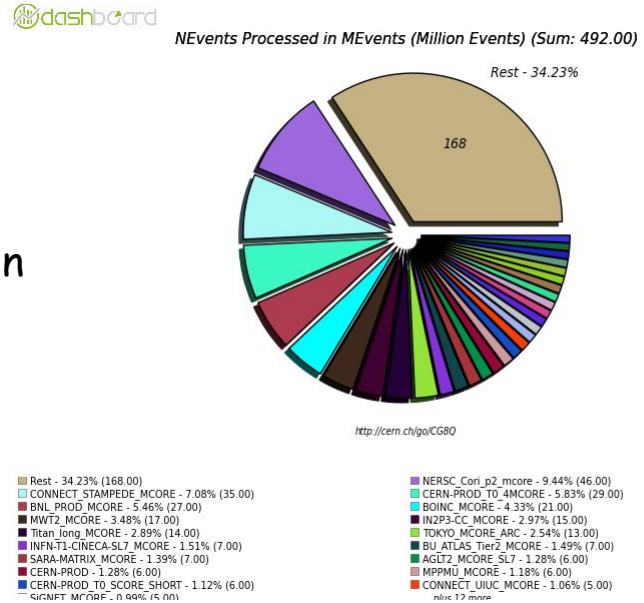
Manual mini ramp up exercise at GCE



Harvester for HPC 1/5

- Current status
  - Theta/ALCF
    - In production with ManyToOne which combines many PanDA jobs to a single MPI payload
    - Bulk data transfers with Globus online
    - A separate queue for Yoda (Event Service at HPC) tests
  - Titan/OLCF
    - In production for ALCC
    - Individual file transfers with rucio client-based Pilot Data API
    - Non-harvester for backfill
  - Cori/NERSC
    - In production
    - Bulk data transfers with ATLAS data management system (rucio)
    - Running out of allocation as it runs very well, and will switch to backfill soon
  - KNL/BNL
    - In production but the resource availability is intermittent
    - Providing test beds for HPC+CE
  - ASGC
    - In production for non-ATLAS VOs
  - EU or NSF HPCs
    - Under discussion

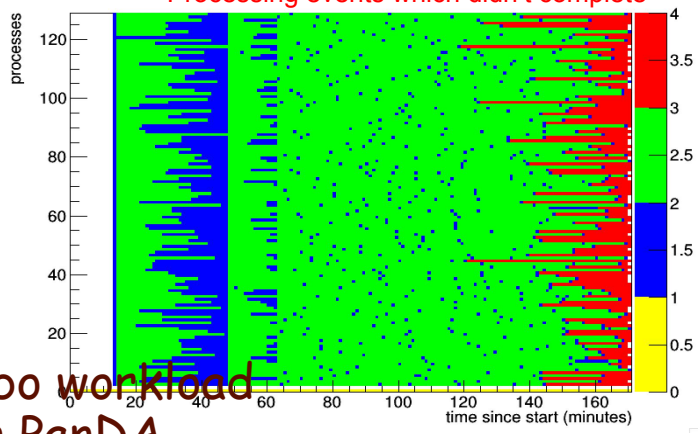
Contributions to ATLAS MC simulation production for last 7 days



Harvester for HPC 2/5

- Current and future developments
  - Yoda/Jumbo jobs at Theta
    - Goals
      - To give large workloads to large payloads
      - To use walltime as much as possible w/o micromanagement of execution time or in preemptable queues
    - Processing at Thena done while Merging at other (grid) resource still to be done
    - Further optimization
      - Shorter initialization
      - Optimal payload size
      - Many small payloads sharing a jumbo workload
      - Smart brokerage and scheduling in PanDA
      - Optimization for IO (details in Danila's talk)

Yoda job at Theta with 127 nodes
Idle, Processing events which completed,
Processing events which didn't complete



  - A common operational model among HPCs
    - A single payload based on pilot 2.0
    - In US, all DOE HPCs running with harvester, while NSF HPCs to be migrated
    - For non-US HPCs, harvester + CE (next page) could be used

Harvester for HPC 3/5

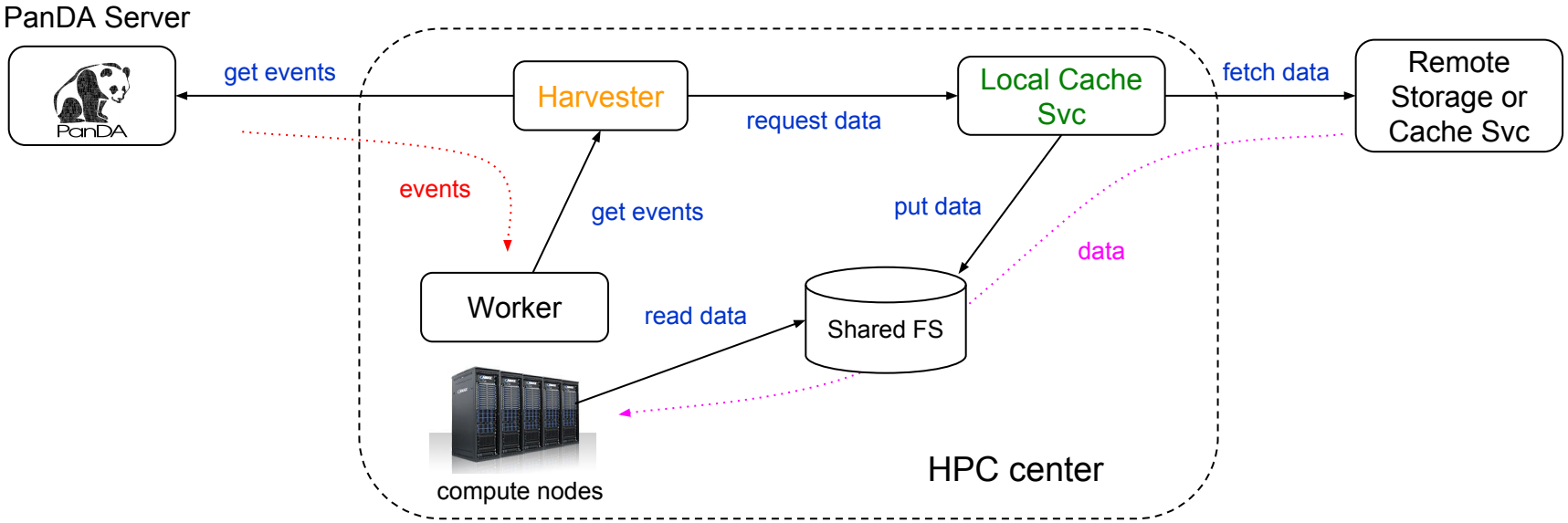
- Current and future developments
  - HPC + CE
    - Rather straightforward if the HPC can be used as a large batch cluster which doesn't have strong constraint and/or preference on payload size
      - E.g. normal PanDA jobs are running at many EU HPCs through ARC CE w/o any special optimization
      - No difference from grid resources especially if outbound network connection is available from compute nodes
    - More tricky if advanced workflows like ManyToOne, OneToMany, jumbo jobs are required, which is typically the case for large HPCs, e.g. larger payloads get higher priorities
      - Tight communication between harvester and workers
      - What's available as a communication channel?
        - Through CE, message bus, …
    - HTCondor-CE and ARC CE have been deployed at BNL/KNL for testing

Harvester for HPC 4/5

- Current and future developments
  - Backfill
    - The ultimate goal for ATLAS is Yoda+Pilot2.0+Harvester, and thus a capability to dynamically change payload size (the number of jobs for each worker) will not be used for ATLAS production in the future
    - However, it is still useful for other experiments which don't have event service or Yoda-equivalent, and it is essentially a mechanism to collect realtime information from HPC for optimal payload scheduling
    - To implement the mechanisms in harvester which is currently used for ATLAS Titan backfill
    - To be decide which HPC is used for development
      - E.g. BNL/KNL, NERSC, Titan?
  - Container integration
    - Theta/ALCF and Cori/NERSC are using containers in production
    - To be used at other HPCs as well
    - To define naming convention, distribution scheme, and contents for images

Harvester for HPC 5/5

- Current and future developments
  - Caching
    - Currently all files in the task have to be transferred beforehand for a jumbo job
    - Would be nice to have a local cache service at HPC to avoid full data prestaging
      - Compute nodes could directly request data to the Svc, or harvester could request data on behalf if direct connection to the Svc is unavailable from compute nodes
    - Could leverage ongoing developments for prefetcher + Event Streaming service in ATLAS



Summary

- Many development activities in parallel for various resources
  - Weekly roundtable in WFMS meeting
  - F2F meeting in ATLAS Software and Computing workshop every 3~4 months
- Already in production for various resources
  - Introducing commonality layer in monitoring and operational experience
  - Further optimization
- A lot of challenges to come for HPCs as well as other resources