

# Operational experience at US HPCs with Harvester (ie fun with HPC's)

Doug Benjamin

Feb. 6, 2025

# Acknowledgements

These slides represent the inputs and hard work of the following people –

Lincoln Bryant

Rui Wang

Julien Esseiva

Xin Zhou (recent addition to the team)

Doug B

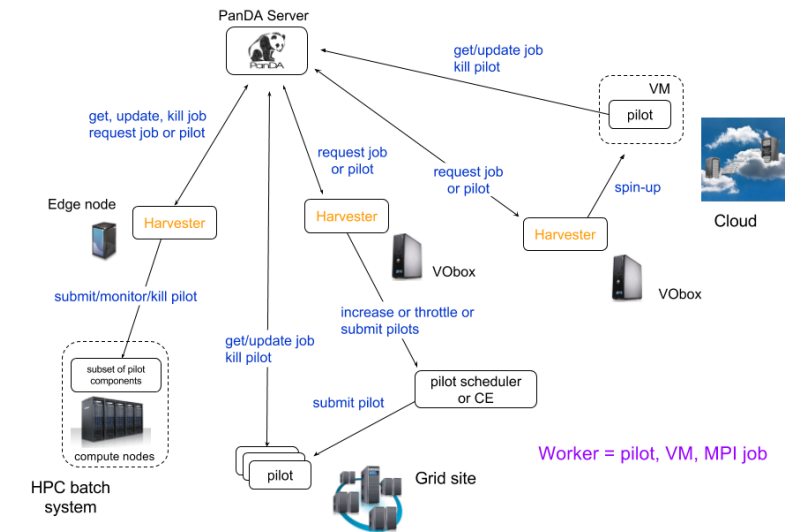
- Credit is theirs and mistakes are mine



# Harvester

- Harvester runs on Edge of the following HPC's in the US
  - TACC Frontera ( [Link to further details](#) )
  - NERSC Perlmutter ( [Link to Perlmutter details](#) )
    - On Perlmutter NERSC provides a workflow QOS. We run Harvester inside this workflow QOS
      - Special SLURM based cron mechanism
      - Run workflow job on a login node under common collaboration account (usatlas)

[Link to Harvester documentation](#)

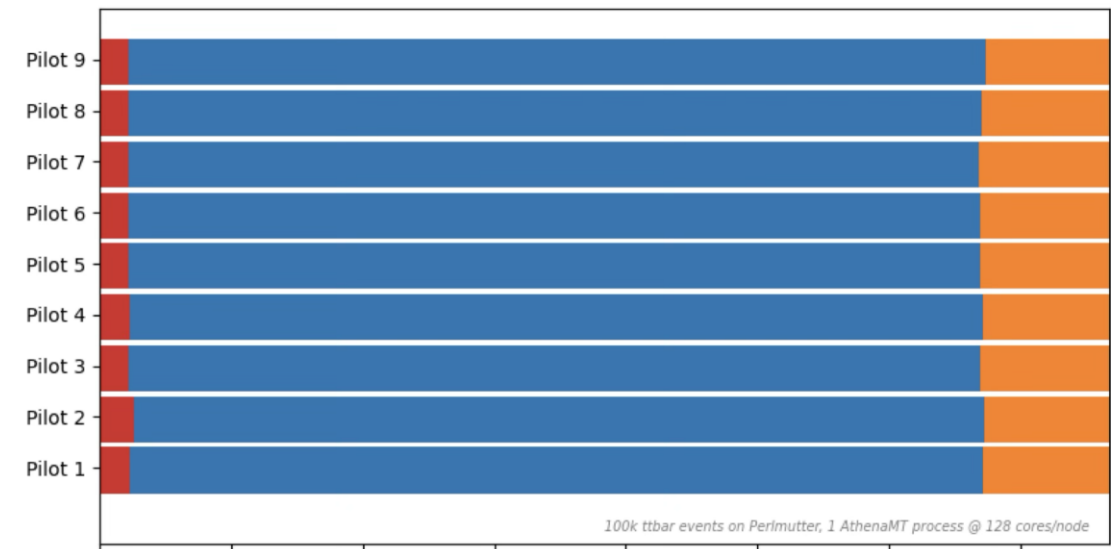


# Harvester details

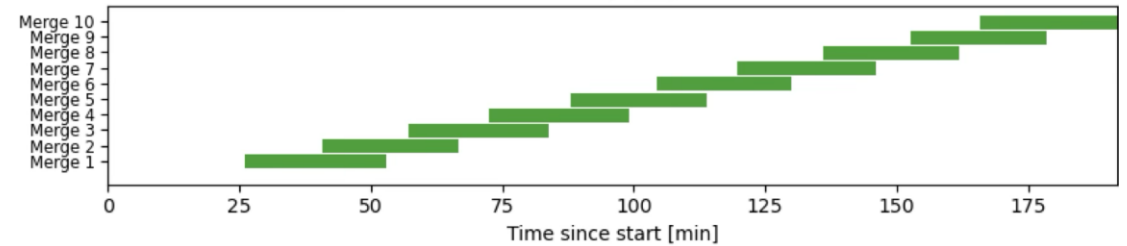
- Tokens used for authentication in communication with PanDA servers (Xin Zho)
- Multiple queues at each HPC
  - Multicore, single core , test and gpu queues
  - Multiple pilots per node
    - Use specialty pilot wrappers at NERSC and TACC
- Slurm submitter plugin used to start work (initially written by DB)
  - Depending on specific queue – submit Slurm jobs multi node for fixed period of time (for example - at NERSC 5-10 nodes for 12-24 hours)
- TACC needs a lower release of urlib compared to the default from harvester
- Data transfer between BNL (WLCG) and HPC's
  - TACC uses Globus (via Globus plugins – written by DB) ( [Globus Data Transfer](#) )
  - NERSC uses Rucio tools to upload and download data to BNL

# Event Service

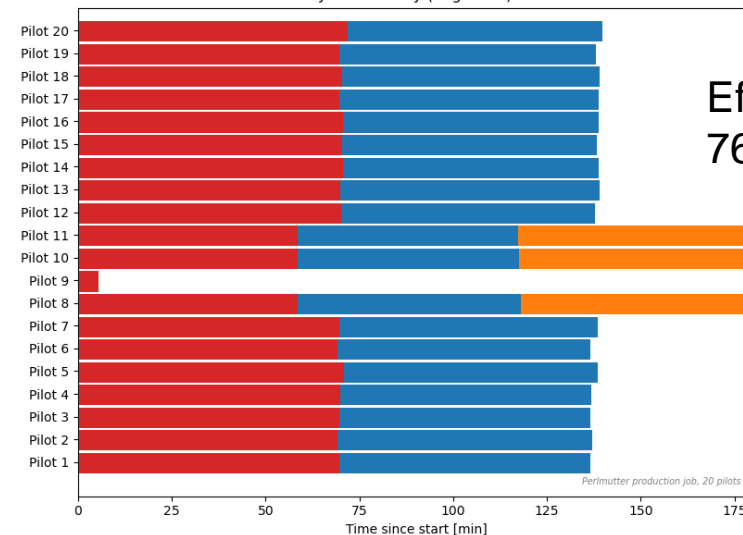
- Event Service – event level simulation – developed by Julien Esseiva (LBNL)
  - Base on [Ray.IO](#)
  - One process handles communication w/ Harvester via filesystem and hands out events to simulate
  - One process process does the merge step
  - In these examples 3 hour Slurm job – 5 nodes



Job Efficiency (avg: 0.87)



Job Efficiency (avg: 0.76)



Effcy-76%

Standard Running Multiple pilots per node Multi-jobs per pilot

# cvmfs / cvmfsExec

- cvmfsExec at TACC - Information Courtesy – Rui Wang
  - Leftover files from the terminated jobs when running cvmfsExec locally.
    - They are hard to clean by the new jobs, which causes failures.
    - Lincoln Bryant wrote a clean script to delete them, which had been launched in Slurm after the pilot job.
  - Run cvmfsExec inside the container helps isolate the cvmfs caching
    - container-in-container: outer container for cvmfsExec; inner container from pilot-wrapper based on releases
    - The path of the shared tmp filesystem changes on some nodes at TACC, which causes the mounting fail
    - TACC limits the max\_user\_namespace to 2000 in the middle of 2024, which prevents running Atlas container after cvmfsExec been executed
  - TACC is using self-extract cvmfsExec inside of centos7/el9 right now without the inner container
    - More stable against software changes
- cvmfs at NERSC
  - Older version – 2.11.5
  - Issue with max number of single core pilots on node (currently limited to 32)
    - Need to set - *CVMFS\_CACHE\_REFCOUNT=true*

# Further enhancements / Improvements

- One Harvester on the Edge of a HPC does not scale
  - Need to move job submission to centralized Harvester
  - HPC's developing API for submitting jobs
    - NERSC two solutions
      - Superfacility API ( [Link to documentation](#) )
      - Globus Compute ( [Link to Documentation](#) )
        - Active project with HEP-CCE to make Globus Compute work for NERSC
    - NSF's HPC's (TACC Stampede3, SDSC Expanse, PSC Bridges-2)
      - Globus Compute
      - We have small allocation on SDSC Expanse and PSC Bridges-2 (50k cores each)
    - ALCF – Globus Compute
  - Rucio + Globus integration
    - Initial implementation - Matt Snyder (BNL) and Doug Benjamin
    - HEP-CCE project w/ Doug B. to test implementation at scale
      - Going slowly need to jump start activity

# Conclusions

- Harvester on the Edge has been highly successful for ATLAS
  - It allows us to use more restricted sites
  - Allowed for new simulation workflows (Event Service)
  - Many Harvester instances at each HPC not really scalable
    - Common API's exist for job submission (Globus Compute)
    - US DOE HPC's working towards common API
    - NSF HPC's have common API
  - Common tools exist for data motion (Globus Data Transfer)
    - Rucio + Globus exists but needs to be tested at scale
  - Need to focus on efficiency (more compute per allocation hour)