MeanRSS throttling in PanDA

Fernando Barreiro Megino

on behalf of the PanDA team

WLCG Job Allocation and Handling, 28 November 2024

UTA

PanDA components

Production requests

Rucio

ProdSys2

Production tasks

JEDI

Jobs

PanDA server

Analysis tasks

Jobs

Harvester

Harvester

Harvester

Harvester

aCT

aCT

pilot

pilot

pilot

pilot

pilot

EGI

OSG

NDGF

HPCs

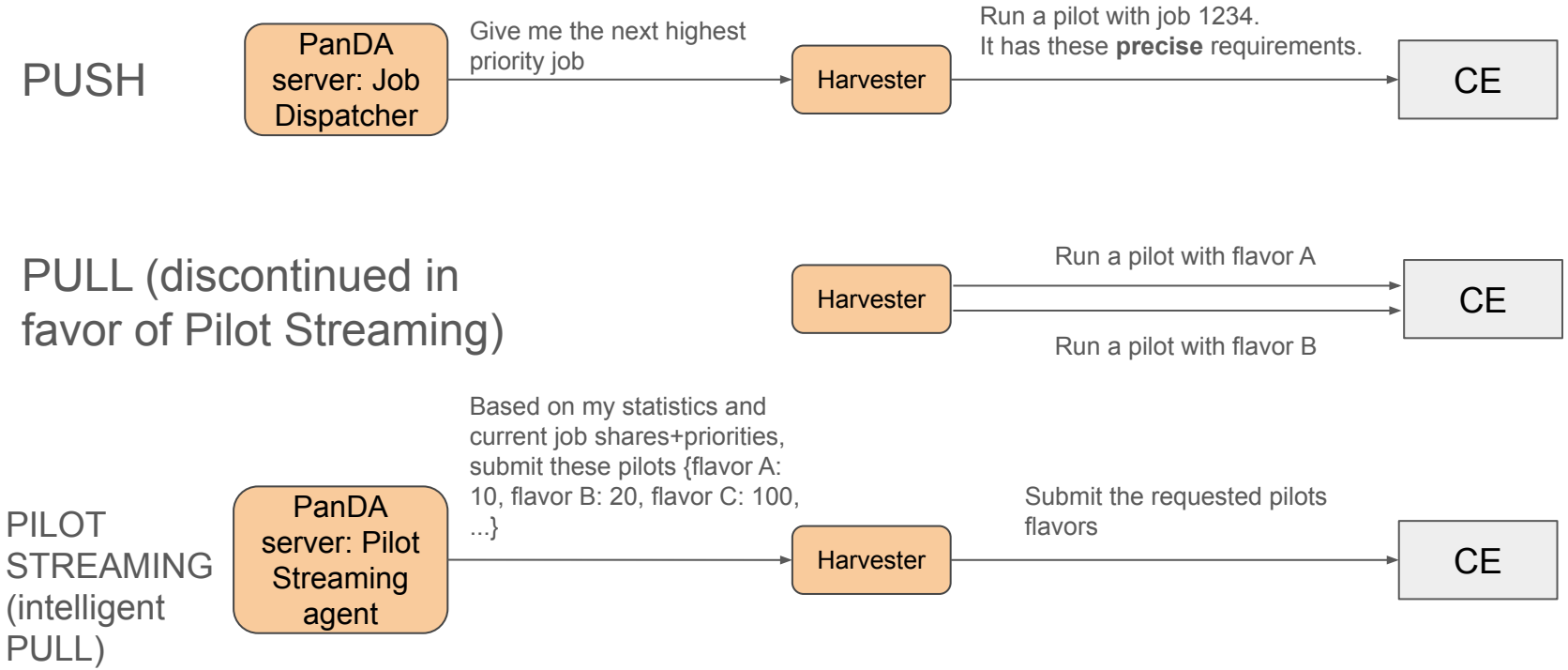
Clouds

Worker nodes

Worker nodes

Worker nodes

Pilot submission modes: push, pull, pilot streaming



Types of “flavors” for pull (we call them resource types)

- Flavors are configured in the DB and assigned to tasks/jobs at creation time
- **Historical flavors**

[single core, multi core] x [standard, high memory]

  - Standard: 2GB/core
  - High memory: the maximum defined by the queue
- **Increased granularity**

[single core, multi core] x [low memory, standard, high memory, very high memory]

  - *Low memory: 1GB/core*
  - *Standard: 2GB/core*
  - *High memory: 3GB/core*
  - *Very high memory: the maximum defined by the queue*

CRIC queue configuration

- New field to define the **target mean memory** for the queue, which reflects the available HW: **meanRSS**
- This provides more flexibility for the maximum allowed memory maxRSS
  - It used to be set to the available HW or slightly above (usually 2-3 GB/core)
  - Now it can be set higher and PanDA will throttle high memory jobs to stay under meanRSS

Example for BNL

32	maxrss	overwritten	48000	= 6GB/core	The maximum RSS, in MB, available to the slot with corecount cores, i.e. 16000 for corecount=8 and 2GB/core. This can be larger than the physically available RSS/core (meanrss) because Panda will throttle to maintain the available mean
35	meanrss NEW	overwritten	2700	= 2.7GB/core	The mean hardware memory/core in MB for this resource
36	minrss	inherited	0		The minimum RSS, in MB, available to the slot with corecount cores. Used to partition a cluster

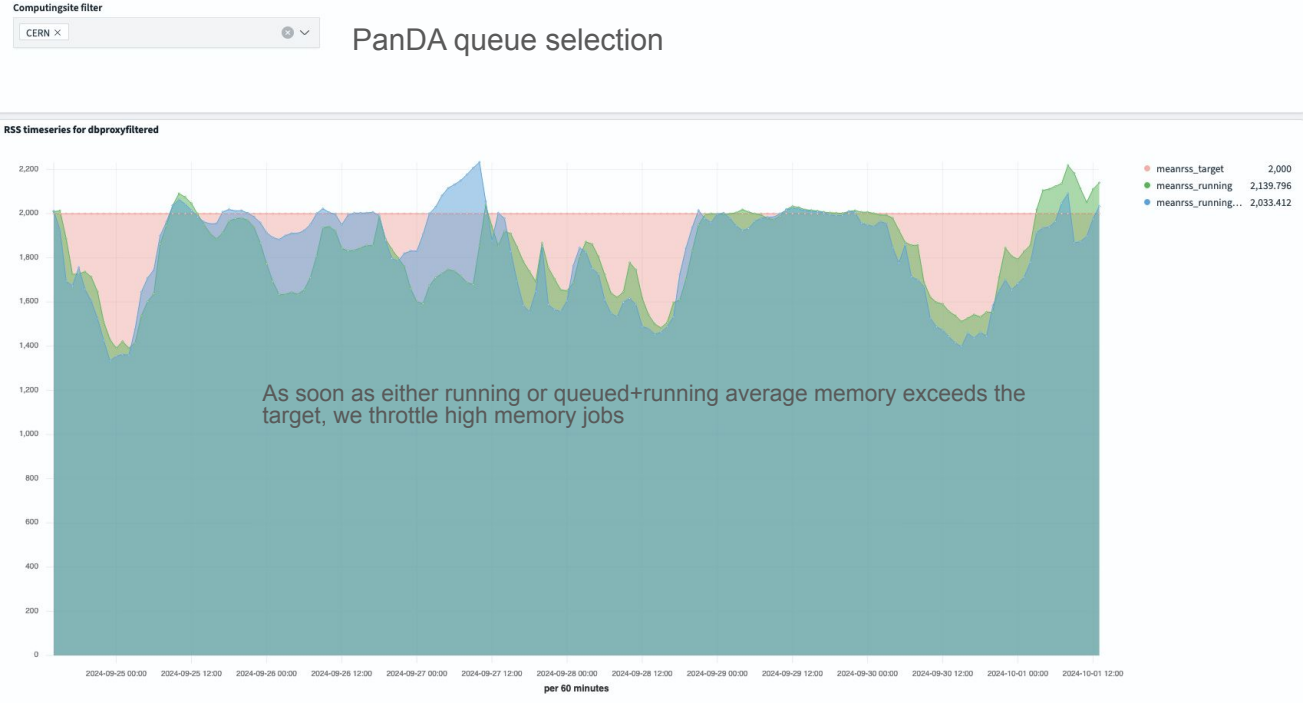
Memory throttling

- PanDA server generates snapshots with statistics at job and pilot level
- Push case
  - Works with statistics at (detailed) job level
  - Implemented in the Job Dispatcher of PanDA server
  - When Harvester asks for a job, the Job Dispatcher will validate the memory utilization and if necessary only return jobs with memory < meanRSS
- Pilot Streaming case
  - Works with statistics at (coarse) pilot flavor level
  - Implemented in the Pilot Streaming agent in PanDA server
  - As soon as the memory for the pilots at a site is exceeded, Pilot Streaming will only consider jobs with memory < meanRSS for the next cycle
- In both cases, for earlier throttling we use the higher value or running or running+queued pilots/jobs

Aleksandr Alekseev

Monitoring

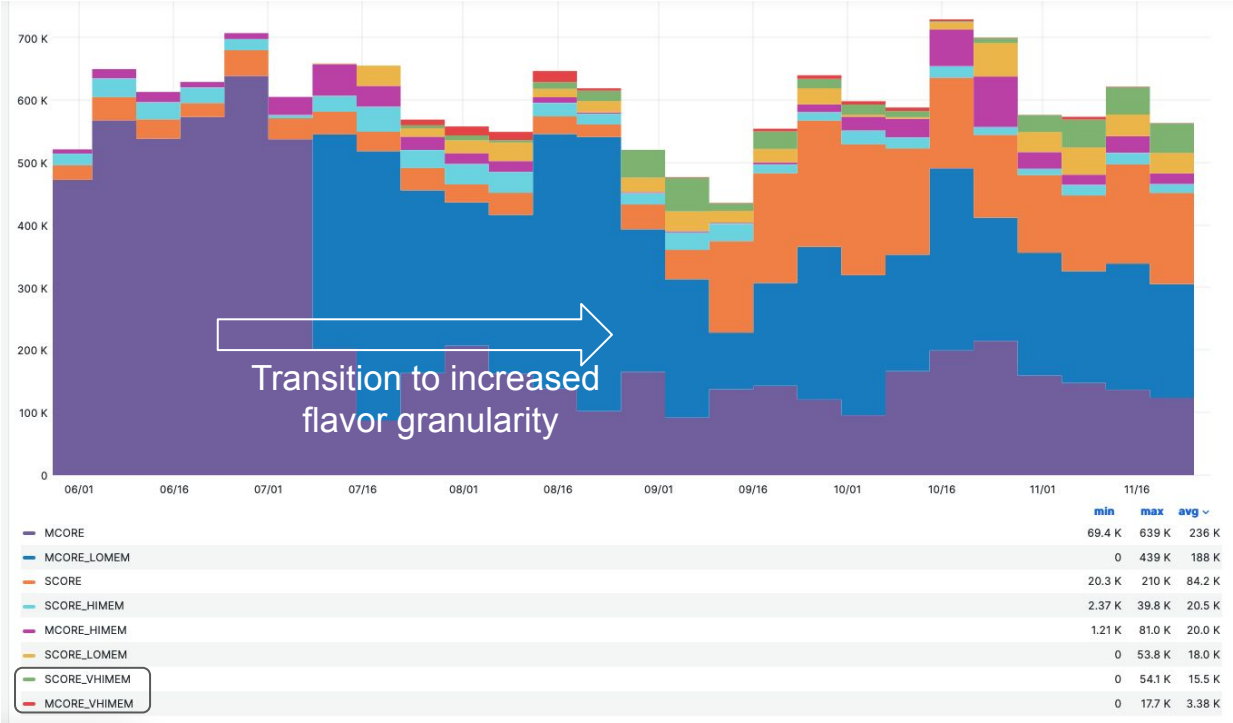
- Our monitoring is based on a filebeat-logstash-OpenSearch pipeline
  - We can plot “anything” that goes through the PanDA server logfiles



Ivan Glushkov
Rod Walker
Timo Wilkens

Results

- ADC ops team is updating maxRSS and meanRSS cloud by cloud
- Even with the migration on-going, there are >50k cores for very high memory jobs



Conclusions and observations

- Memory requirements for ATLAS workloads evolve and differ with time
- Development tries to balance out the low and high memory jobs: increase the cores available for very high memory jobs without sites having to change their HW specifications
- Important campaigns, e.g. for Sherpa evgen, are now making good progress without depending on a few sites with high memory nodes
- Memory throttling does not solve inefficient scheduling: if the batch system sends all very high memory jobs to the same node, some cores will be left idle
    - Site and batch tuning is required - e.g. US sites are in contact with HTCondor team
    - Rod knows much more on the topic