Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

PanDA Overview & Status

W. Guan, E. Karavakis, Z. Yang on behalf of the PanDA team

Rubin Observatory Data Facilities & Multi Site Processing Workshop
IN2P3, Lyon, France

Wednesday 12 February 2025

@BrookhavenLab

https://pandawms.org / panda-support@cern.ch / https://panda-wms.readthedocs.io

PanDA Core News

Code Modernization Effort

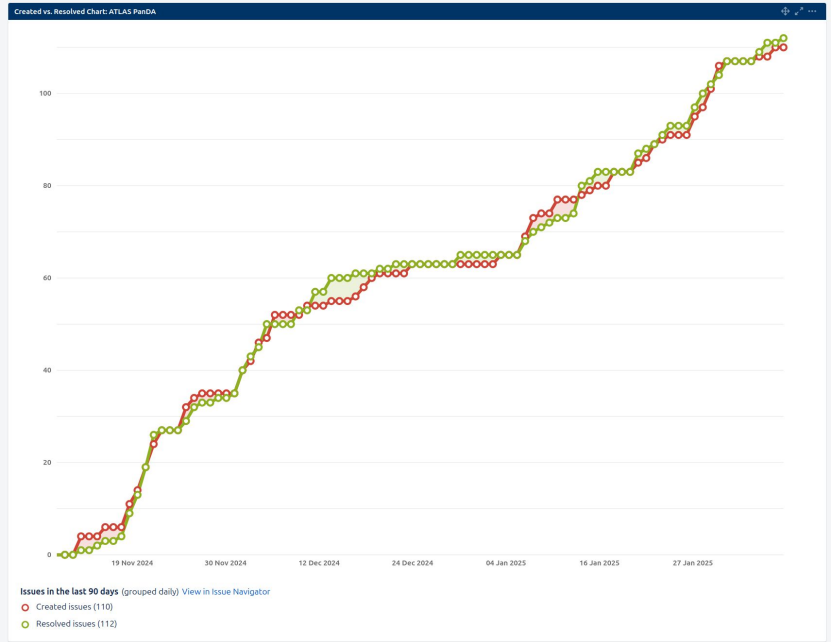
- PanDA server package started ~20 years ago
- (ATLAS) Distributed Computing has gone through many changes during that time
- Technical/infrastructure requirements have also evolved
- Several modules have become extremely complicated to understand/maintain
- Architectural choices made 20 years ago have stood the test of time, demonstrating their robustness. We need evolution, not revolution

Brookhaven National Laboratory

Code Modernization Effort (cont.)

Issue Statistics: Stream 1: All tickets (Assignee)

Assignee	Count	Percentage
Wen Guan	1	2%
Jammel Brooks	2	4%
Fa-Hui Lin	3	7%
Unassigned	3	7%
Aleksandr Alekseev	4	9%
Tadashi Maeno	4	9%
Edward Karavakis	9	20%
Fernando Harald Barreiro Megino	19	42%
Total	**45**	



- Improve and modernize our code and environment
- Overlooked area in the past, but with increased importance as we grow
- Some dedicated effort with participation across the team
- Issues range from small improvements to large code refactoring
- Significant effort throughout 2024!

Brookhaven National Laboratory

Code Modernization Effort (cont.)

- Code refactoring:
  - PEP8 compliant, adding docstrings and adding descriptive variable/method names
  - Cleaning up obsolete code, obsolete agents - modularizing and simplifying where possible
  - Remove python2 compatibility code, move from python3.6 to 3.11, use more python3 features
  - More comprehensive documentation on the agents in readthedocs

- Code refactoring for the pandaserver/dataservice directory, updating nine agents in total: DynDataDistributor, Activator, Closer, Finisher, EventPicker, EventLookupClientEI, DDMHandler, Setupper and Adder
  - Roughly speaking, jobs go through UserIF → Setupper → Activator → JobDispatcher (→Watcher) → Adder → Closer → Finisher

- Decommissioned schedconfig and deleted most of the PANDAMETA tables
- Major deletion of obsolete Harvester plugins
- Various speed improvements in JEDI → **Really** noticeable in Vera Rubin tasks (slide 14)

Brookhaven National Laboratory

Code Modernization Examples

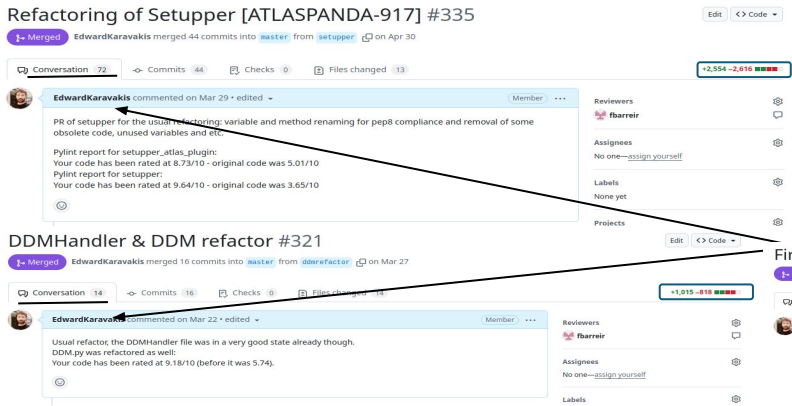

```
+            options = {
+                "runEvent": [],
+                "eventPickDataType": "",
+                "eventPickStreamName": "",
+                "eventPickDS": [],
+                "eventPickAmiTag": "",
+                "eventPickNumSites": 1,
+                "inputFileList": [],
+                "tagDS": [],
+                "tagQuery": "",
+                "tagStreamRef": "",
+                "runEvtGuidMap": {},
+                "ei_api": "",
+            }
+
+            for tmpLine in self.event_picking_file:
+                tmpMatch = re.search("^([^=]+)=(.+)$", tmpLine)
+                if tmpMatch is not None:
+                    key, value = tmpMatch.groups()
+                    if key in options:
+                        if key == "runEvent":
+                            options[key].append(value.split(","))
+                        elif key in ["eventPickDS", "inputFileList", "tagDS"]:
+                            options[key] = value.split(",")
+                        elif key == "eventPickNumSites":
+                            options[key] = int(value)
+                        else:
+                            options[key] = value
+
+            self.userDN = options["userName"]
+            self.user_task_name = options["userTaskName"]
+            self.user_dataset_name = options["userDatasetName"]
+            self.locked_by = options["lockedBy"]
+            self.creation_time = options["creationTime"]
+            self.params = options["params"]
```

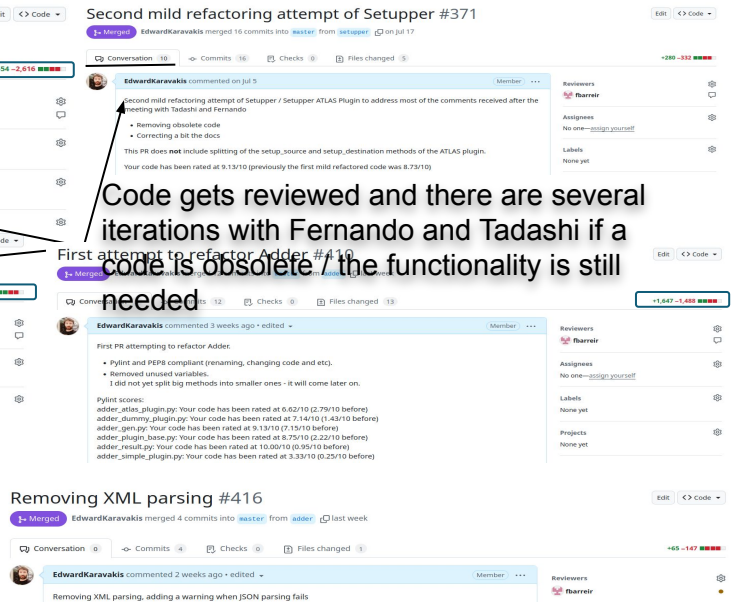
Finisher: 220 LOC - refactored: 195 LOC with around 70 being docstrings and comments - replaced XML files with JSON
...and so on for the remaining agents

DynDataDistributer: 1424 LOC - refactored: 838 LOC with docstrings like these:

```
def get_candidate_sites(self, tmp_rep_maps: Dict, prod_source_label: str, job_label: str) -> Tuple[bool, Dict]:
    """
    Retrieves candidate sites for data distribution based on certain conditions.

    This method filters out candidate sites based on the following condition:
    - The site should have a replica of the dataset.

    Args:
        tmp_rep_maps (dict): A dictionary containing dataset names as keys and their replica maps as values.
        prod_source_label (str): The label of the production source.
        job_label (str): The label of the job.

    Returns:
        tuple: A tuple containing a boolean status and a dictionary. The dictionary has dataset names as keys and
               another dictionary as values. The inner dictionary has cloud names as keys and a tuple of various
               site-related lists and values as values.
    """
```

```
def convert_evt_run_to_datasets(self, event_run_list: List, dataset_type: str, stream_name: str, dataset_filters: List,
                                ami_tag: str, user: str,
                                run_evt_guid_map: Dict, ei_api: object) -> Tuple[bool, Dict, List]:
    """
    Convert event/run list to datasets.

    Args:
        event_run_list (list): The list of run events.
        dataset_type (str): The type of the dataset.
        stream_name (str): The name of the stream.
        dataset_filters (list): The list of dataset filters.
        ami_tag (str): The AMI tag.
        user (str): The user.
        run_evt_guid_map (dict): The map of run events to GUIDs.
        ei_api (str): The EventIndex API.

    Returns:
        tuple: A tuple containing the status (bool), the result (dict or str), and the list of all files.
    """
```

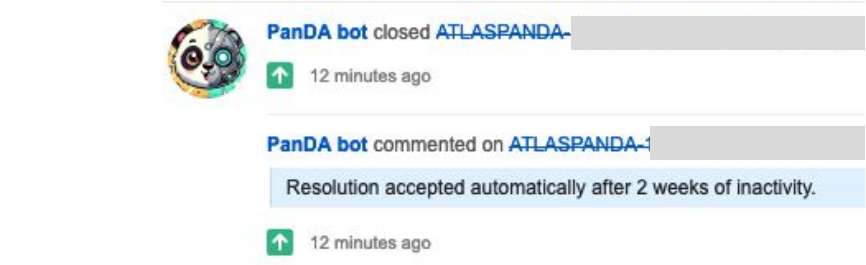


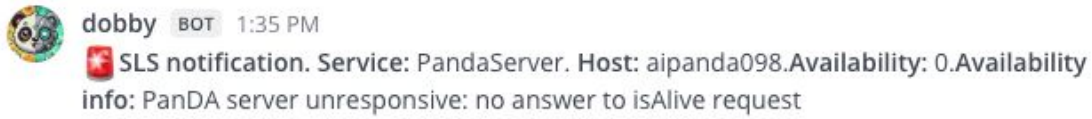


Code gets reviewed and there are several iterations with Fernando and Tadashi if a code is obsolete / the functionality is still needed

Automating some of our daily activities

- A clean JIRA for a clear mind
  - Close tickets in resolved state after 2 weeks
  - Close tickets in any state after 6 months of inactivity
- Internal MM chatops* with bot updates
  - Introductory task for Jammel Brooks (Indiana U.)
  - Optimize our infrastructure management
  - SLS notifications
  - Server upgrade notifications: help synchronize our upgrades and increase awareness





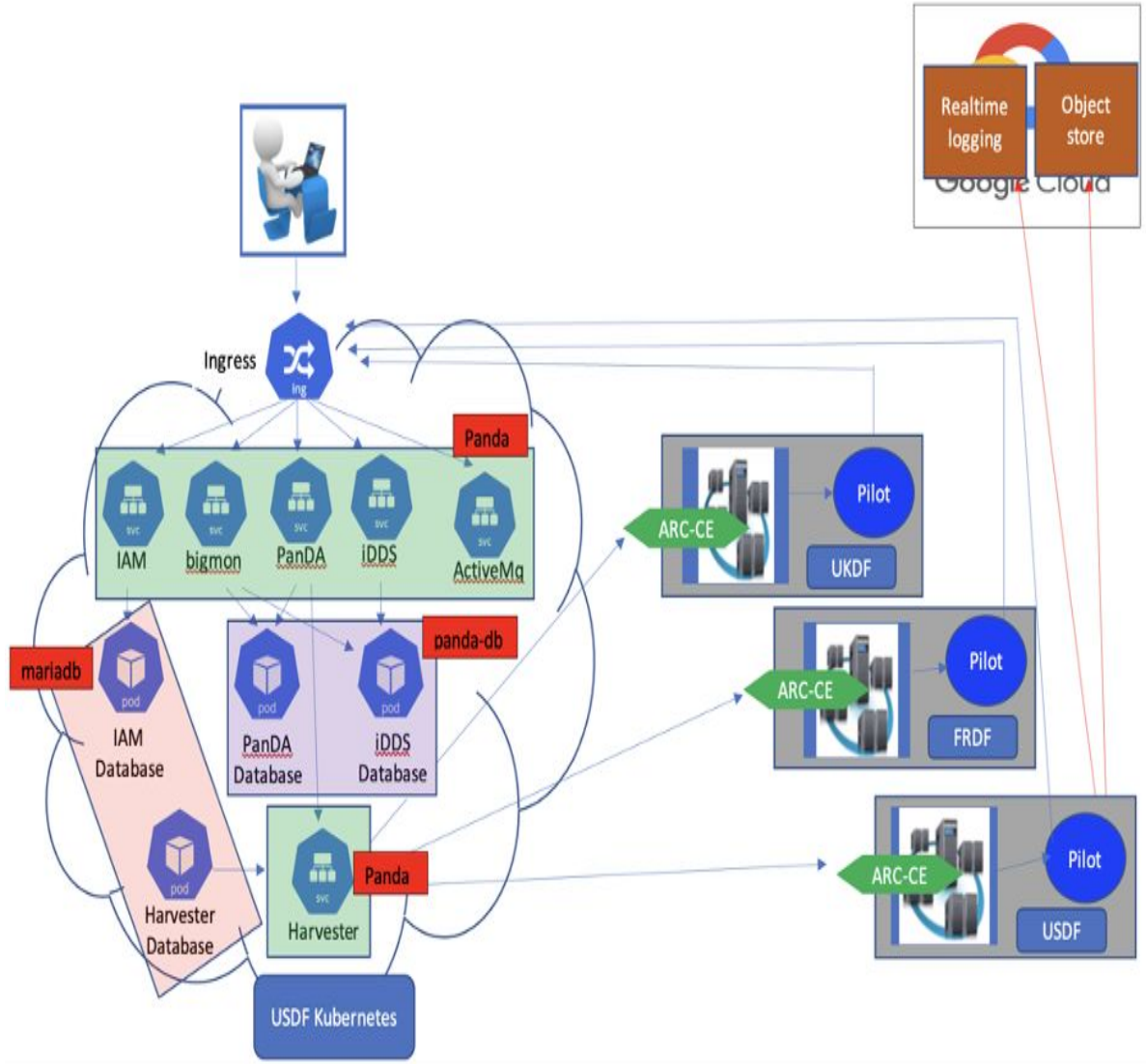
*specific to ATLAS but could be extended to other VOs if needed

Brookhaven National Laboratory

PanDA@SLAC Status

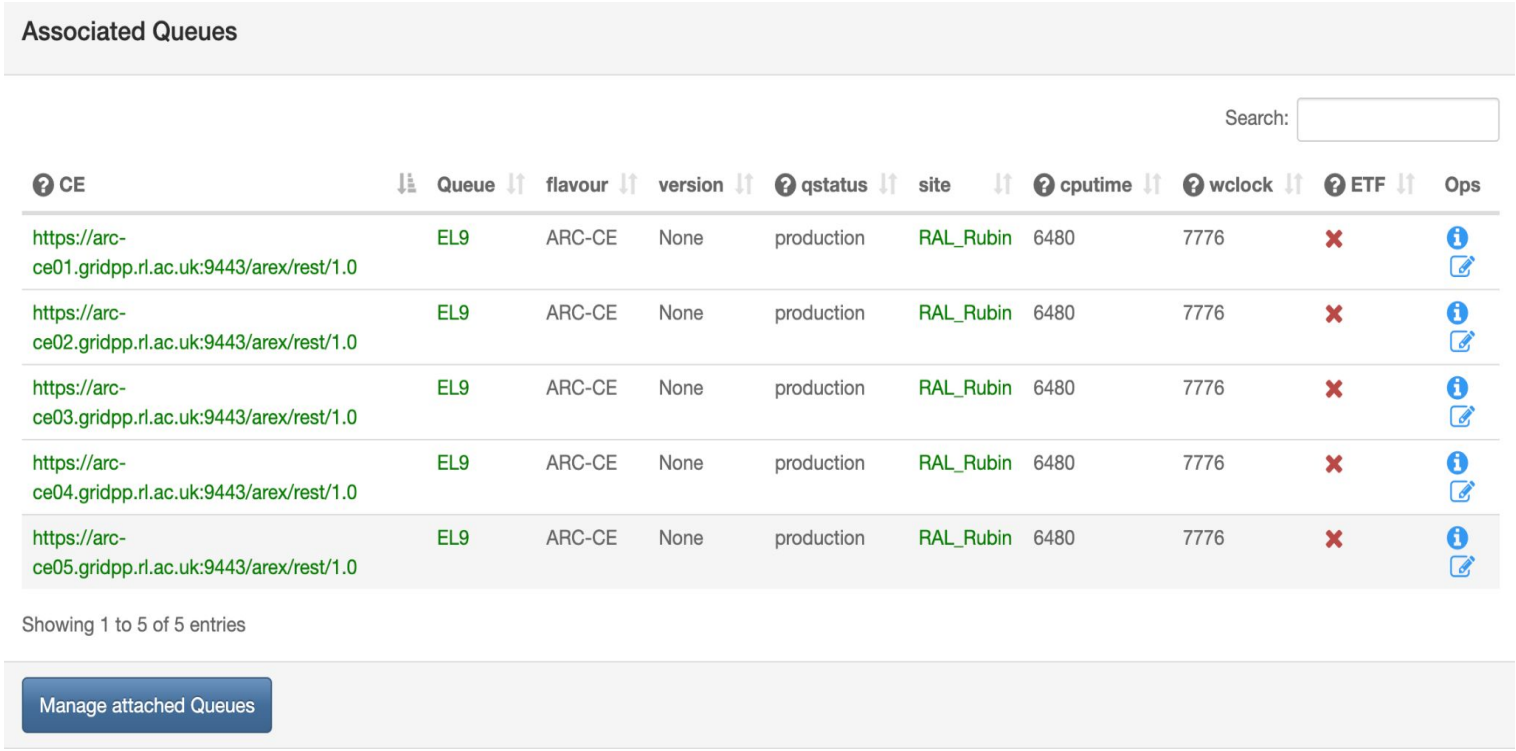
PanDA deployment@USDF k8s

- PanDA system
  - PanDA Server, JEDI, iDDS, PanDA Monitor, Harvester, Pilot
  - External
    - IAM, ActiveMQ, (CRIC)
  - Database
    - CNPG, Mariadb
  - Google realtime logging
- PanDA deployment policy
  - Integration tests on the dev PanDA cluster @USDF
  - Fix a stable version in panda-k8s github
  - Deploy a stable version to production PanDA @USDF and update documents
    - PanDA deployment change control
    - PanDA RunBook
    - Corresponding tickets



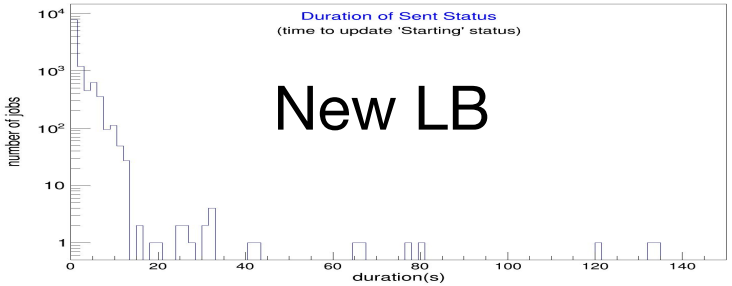
Migrate CE information to CRIC

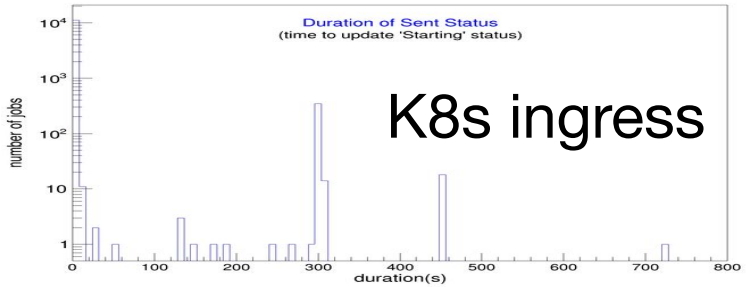
- Currently CE information is managed in a Harvester configuration file
  - When there is an update, need to update every panda queue in the production and dev PanDA
- Moving CE information to CRIC
  - Update once
  - PanDA will automatically sync info
  - https://datalake-cric.cern.ch



Load Balancing

- Wei Yang found out that the K8s ingress service served many services at SLAC causing many network related delays and timeouts
- Wei set up a separate loadbalancer to expose the PanDA on a different IP
- Took this opportunity to perform multiple tests to compare K8s ingress with separate loadbalancer
  - DC2 step1 workflow with 11461 parallel jobs in one clustered task checking time it took for jobs to update their status from **sent** to **starting** in pilot pull mode
  - Jobs in sep. loadbalancer updated their status much faster, all within 130s - jobs in k8s ingress at 5mins / ~3% jobs needed more than 5mins
  - “bps ping” tests indicated potential congestion between PanDA clients and PanDA server, likely caused by the K8s ingress - timeouts disappeared under the new LB





- Job status updates involve communication between pilot (Slurm) and PanDA server (K8s)
- New load balancer outperforms the previously used K8s ingress
- After this exercise, a K8s ingress pool was setup with more nodes

Brookhaven National Laboratory

PanDA New Features

- UCORE
  - Single core or multi-core
  - Within same PanDA queue, we can define multiple ‘pull’ resources that will be used frequently - easy to configure
  - For other, ‘Undefined’ resources, we can use ‘push’ queue
- Multi-Core
  - ‘Push’ is configured to support any number of cores and any memory requirement

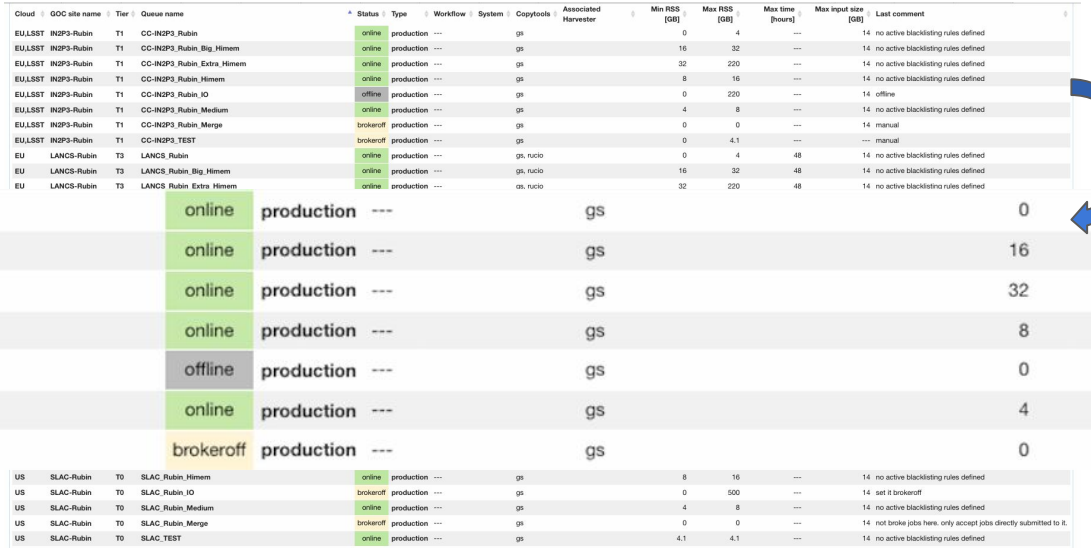
```
panda_db=> select * from resource_types;
 resource_name | mincore | maxcore | minrampercore | maxrampercore
---------------+---------+---------+---------------+---------------
 U2G           |       1 |       1 |             0 |          2000
 U4G           |       1 |       1 |          2000 |          4000
 U8G           |       1 |       1 |          4000 |          8000
 U12G          |       1 |       1 |          8000 |         12000
 U16G          |       1 |       1 |         12000 |         16000
 U20G          |       1 |       1 |         16000 |         20000
 U24G          |       1 |       1 |         20000 |         24000
 U28G          |       1 |       1 |         24000 |         28000
 U32G          |       1 |       1 |         28000 |         32000
 Undefined     |       1 |         |               |
(10 rows)
```

* We could add more entries like multi-core and different memory requirements. In the future if we have a frequent processing type, for example, with 8 cores and 32G memory, we could add it as one entry to this table

Brookhaven National Laboratory

Multi-Site Processing

- Rubin Computing Sites: USDF(SLAC), FrDF(CC-IN2P3), UKDF(LANCS, RAL)
- Workflows are splitted and submitted from USDF to all sites and processed there
- Currently ComCam@USDF, RC2, DC2, PDR2@FrDF + UKDF



Cloud	GOC site name	Tier	Queue name	Status	Type	Workflow	System	Copytools	Associated Harvester	Min RSS [GB]	Max RSS [GB]	Max time [hours]	Max input size [GB]	Last comment
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_Rubin	online	production	---		gs		0	4	---	14	no active blacklisting rules defined
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_Rubin_Big_Himem	online	production	---		gs		16	32	---	14	no active blacklisting rules defined
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_Rubin_Extra_Himem	online	production	---		gs		32	220	---	14	no active blacklisting rules defined
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_Rubin_Himem	online	production	---		gs		8	16	---	14	no active blacklisting rules defined
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_Rubin_IO	offline	production	---		gs		0	220	---	14	offline
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_Rubin_Medium	online	production	---		gs		4	8	---	14	no active blacklisting rules defined
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_Rubin_Merge	brokeroff	production	---		gs		0	0	---	14	manual
EU,LSST	IN2P3-Rubin	T1	CC-IN2P3_TEST	brokeroff	production	---		gs		0	4.1	---	---	manual
EU	LANCS-Rubin	T3	LANCS_Rubin	online	production	---		gs, rucio		0	4	48	14	no active blacklisting rules defined
EU	LANCS-Rubin	T3	LANCS_Rubin_Big_Himem	online	production	---		gs, rucio		16	32	48	14	no active blacklisting rules defined
EU	LANCS-Rubin	T3	LANCS_Rubin_Extra_Himem	online	production	---		gs, rucio		32	220	48	14	no active blacklisting rules defined
UK	RAL_Rubin	T2	RAL_Rubin	online	production	---		gs		0	4	---	14	no active blacklisting rules defined
UK	RAL_Rubin	T2	RAL_Rubin_Big_Himem	online	production	---		gs		16	32	---	14	no active blacklisting rules defined
UK	RAL_Rubin	T2	RAL_Rubin_Extra_Himem	online	production	---		gs		32	220	---	14	no active blacklisting rules defined
UK	RAL_Rubin	T2	RAL_Rubin_Himem	online	production	---		gs		8	16	---	14	no active blacklisting rules defined
UK	RAL_Rubin	T2	RAL_Rubin_IO	offline	production	---		gs		0	220	---	14	set offline
UK	RAL_Rubin	T2	RAL_Rubin_Medium	online	production	---		gs		4	8	---	14	no active blacklisting rules defined
UK	RAL_Rubin	T2	RAL_Rubin_Merge	brokeroff	production	---		gs		0	220	---	14	set brokeroff
US	SLAC-Rubin	T0	SLAC_Rubin_Himem	online	production	---		gs		8	16	---	14	no active blacklisting rules defined
US	SLAC-Rubin	T0	SLAC_Rubin_IO	brokeroff	production	---		gs		0	500	---	14	set it brokeroff
US	SLAC-Rubin	T0	SLAC_Rubin_Medium	online	production	---		gs		4	8	---	14	no active blacklisting rules defined
US	SLAC-Rubin	T0	SLAC_Rubin_Merge	brokeroff	production	---		gs		0	0	---	14	not broke jobs here. only accept jobs directly submitted to it.
US	SLAC-Rubin	T0	SLAC_TEST	online	production	---		gs		4.1	4.1	---	14	no active blacklisting rules defined

- Number of jobs processed in the last 12 months

Data Facility	# jobs
USDF (SLAC)	11,870,216
FrDF (CC-IN2P3)	3,012,498
UKDF (LANCS, RAL)	3,760,555 (3,759,248, 1307)

18.6M jobs in total!

Brookhaven National Laboratory

Workflow Improvements

- PanDA Server/JEDI/Harvester/Pilot/iDDS have made many improvements. The Rubin pipeline workflow can be processed more efficiently now
  - Below is the DC2 step1 pipeline with 11461 parallel jobs in 5 clustered tasks: isr, characterizeMimage, calibrate, writeSourceTable, transfromSourceTable
  - The whole workflow is processed in **~36 minutes** from start to end. It's comparable to the batch processing of a similar pipeline - used to take ~**2 hrs** at the beginning (DOMA), then **1 hr** with some improvements
  - There are no intermittent errors from e.g. infrastructure failure, so no automatic retries (enabled in PanDA and usually 3-5 attempts are applied to a failed job) were needed in this processing

Workflow ID	Task name TaskType/ProcessingType Campaign Group User Errors Logged status	Task status Nfiles	Input files Nlost Nfinish % Nfail %	Total/Remaining events	Created	Modified
160675	u_zhaoyu_test_step1_us_20240924T165424Z_03_finalJob_01_160675_10663 test Zhaoyu Yang Errors None	done 1	1 100%		2024-09-24 17:23:20	2024-09-24 17:34:58
160675	u_zhaoyu_test_step1_us_20240924T165424Z_02_visit_detector_01_160675_10662 test Zhaoyu Yang Errors None	done 11461	11461 100%		2024-09-24 16:58:45	2024-09-24 17:20:57
160675	u_zhaoyu_test_step1_us_20240924T165424Z_01_pipetaskInit_01_160675_10661 test Zhaoyu Yang Errors None	done 1	1 100%		2024-09-24 16:58:35	2024-09-24 17:00:57

Automatic Boost of Requested Memory

- The required memory to run a Rubin job varies significantly even in the same pipeline task, making it difficult to set a single memory request that suits all jobs
- PanDA now automatically increases the requested memory and retries jobs that fail due to an out-of-memory (OOM) error
  - ❏ The first attempt failed due to OOM
  - ❏ PanDA increased the memory from **8G** to **16G**, the job was rescheduled from the Medium queue to HIMEM queue and the second attempt finished successfully

PanDA ID Attempt#	Owner / VO Group	Request Task ID	Transformation	Status	Created	Time to start d:h:m:s	Duration d:h:m:s	Mod	Site	Priority	N input events (N input files)	Max PSS/core, GB	Job info
19639792 Attempt 2	Brian Yanny / wlcg	8526 18972	bash-c-enc	finished	2024-08-09 18:11:12	0:0:01:19	0:0:21:28	2024-08-09 18:34:06	SLAC_Rubin_Himem online	500	0 (0)	9.38	
	Job name: 2.2i_runs_test-med-1_w_2024_32_DM-45612_step3_group1_job_000_05_coadd_01_8526_18781.19630694 #2												
	Datasets: -												
19630694 Attempt 1	Brian Yanny / wlcg	8526 18972	bash-c-enc	failed	2024-08-09 18:03:23	0:0:04:17	0:0:03:28	2024-08-09 18:11:12	SLAC_Rubin_Medium online	500	0 (0)	7.24	**pilot, 1212:** Memory cgroup out of memory: Killed process 3719039 (python) total-vm:9272300kB, anon-rss:8063052kB, file-rss:142520kB, shmem-rss:4kB, UID:17951 pgtables:16504kB oom_score_adj:0 **trans, 137:** Unknown transexitcode error code 137
	Job name: 2.2i_runs_test-med-1_w_2024_32_DM-45612_step3_group1_job_000_05_coadd_01_8526_18781.19630694 #1												

Brookhaven National Laboratory

Clustering of Short Jobs

- Rubin pipelines consist of very short jobs, especially in step 1, where jobs typically run for less than a minute. The large number of concurrent short jobs puts significant strain on WMS
- Event Service has been integrated into Rubin-PanDA to cluster short jobs into longer ones. ES can group jobs across multiple pipeline tasks and/or parallel jobs within the same task
- For example, *isr* is usually a short job for one detector. After ES clusters 10 *isr* jobs into 1 bigger job, the processing time increased from 50 seconds to 8 minutes, significantly reducing the load on the PanDA system

Without ES

PanDA ID	Owner / VO	Request Task ID	Status	Transformation	Created Last modified	Time to start Duration [d:h:m:s]	Cloud Site	Harvester instance Worker ID	Cores	Priority	Attempt
20102499	Zhaoyu Yang / wlcg	8711 19826	finished	bash-c-enc	2024-09-03 18:01:50 2024-09-03 18:04:00	0:0:01:18 0:0:00:50	US SLAC_Rubin	SLAC_Harvester_SDF 3585476	1	500	1

With ES

PanDA ID	Owner / VO	Request Task ID	Status	Sub status	Transformation	Created Last modified	Time to start Duration [d:h:m:s]	Cloud Site	Harvester instance Worker ID	Cores	Priority	Attempt
20726429	Wen Guan / wlcg	8875 20979	finished	fg_done	bash-c-enc	2024-09-27 14:03:47 2024-09-27 14:14:21	0:0:01:40 0:0:08:49	US SLAC_Rubin	SLAC_Harvester_S3DF 3666930	1	500	1

Latest 10 event range records:

Show 10 entries Search:

File ID	Min event	Max	Status	ProcessID	Attempt	DataSet ID	ObjectStore ID	Jobset ID
68354452	10	10	finished	10	-	-	-	-
68354452	9	9	finished	9	-	-	-	-
68354452	8	8	finished	8	-	-	-	-
68354452	7	7	finished	7	-	-	-	-
68354452	6	6	finished	6	-	-	-	-
68354452	5	5	finished	5	-	-	-	-
68354452	4	4	finished	4	-	-	-	-
68354452	3	3	finished	3	-	-	-	-
68354452	2	2	finished	2	-	-	-	-
68354452	1	1	finished	1	-	-	-	-

Submit User Scripts

- During the ongoing multi-site test campaign, a request has emerged for PanDA to submit user non-pipeline scripts from USDF to remote sites
  - PanDA client provides a tool called prun for this purpose
  - Integrated prun into the Rubin pipeline middleware BPS, now part of the stack

The custom script is submitted from USDF to FrDF

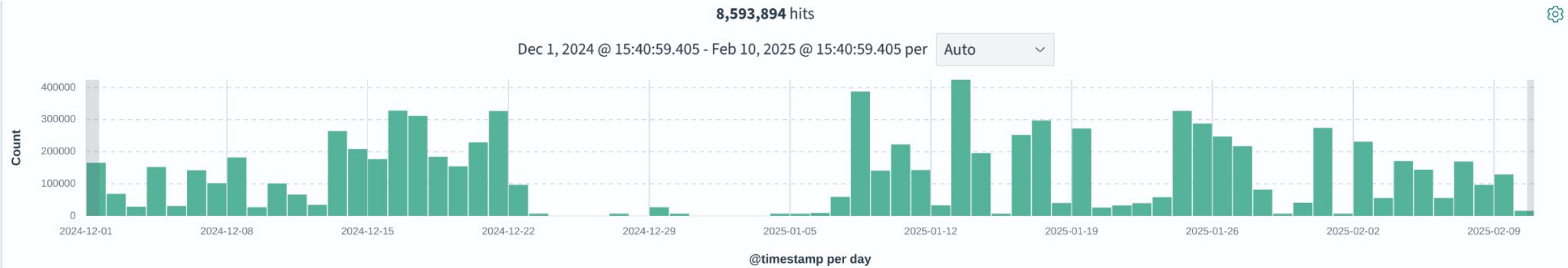
ID Parent	Workflow ID	Task name TaskType/ProcessingType Campaign Group User Errors Logged status	Task status Nfiles	Input files Nlost Nfinish % Nfail %	Total/Remaining events	Created	Modified	State changed	Priority	Nucleus
30519	10813	u_yanny_move_plots_FRDF_20250209T220440Z_01_customJob_01_10813_30005 test Brian Yanny Errors None	done 1	1 100%		2025-02-09 22:05:28	2025-02-09 22:11:23	2025-02-09 22:11:23	500	None

The script output (rucio commands)

```
DATASET: hsc_pdr2_multisite:Dataset/HSC/runs/PDR2/VVDS/w_2025_04/DM-48570/frdf/plotsru
+-------------------+---------+---------+
| RSE               |   FOUND |   TOTAL |
|-------------------+---------+---------|
| IN2P3_BUTLER_DISK |       1 |       1 |
| SLAC_BUTLER_DISK  |       0 |       1 |
+-------------------+---------+---------+
done
PANDACACHE_URL: https://usdf-panda-server.slac.stanford.edu:8443/server/panda
Download archive file from pandacache status: 0, output: True
Extracted /tmp/rubin_0I7rT5W7/PanDA_Pilot-34220960/job0.2d637cfd-3b6d-4131-a56f-0c793183eaaf.tar.gz to /tmp/rubin_0I7rT5W7/PanDA_Pilot-34220960
Removed /tmp/rubin_0I7rT5W7/PanDA_Pilot-34220960/job0.2d637cfd-3b6d-4131-a56f-0c793183eaaf.tar.gz
ret_event_status: True, with_events: False
executable command line:
./move-plots-frdf2.bash
```

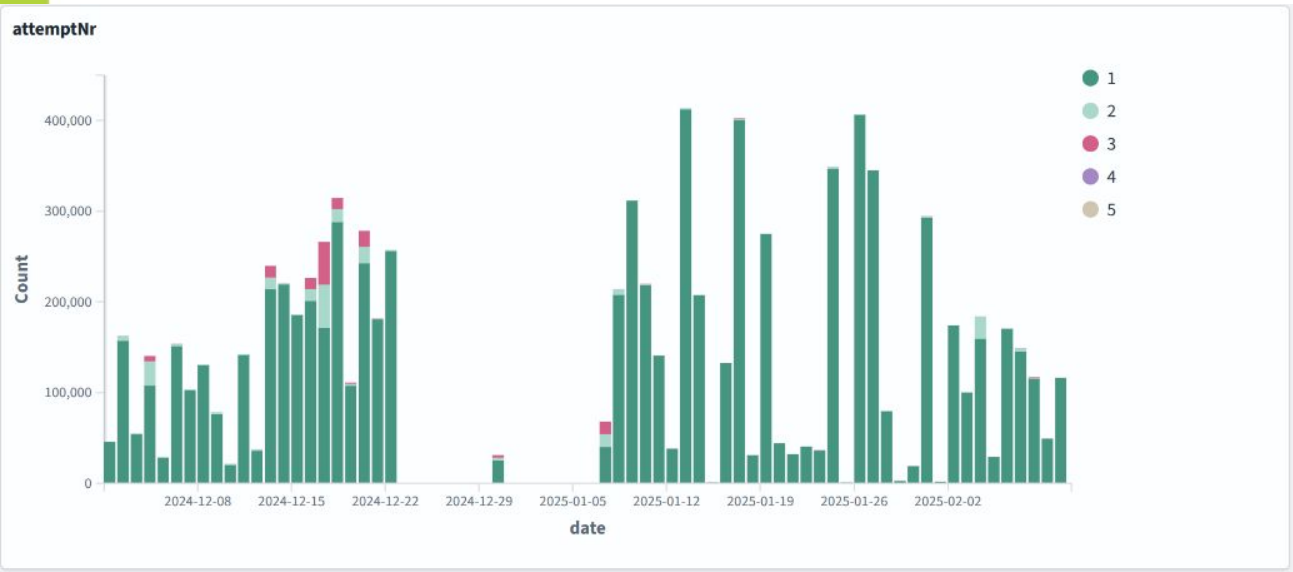
Job Monitoring Info on OpenSearch

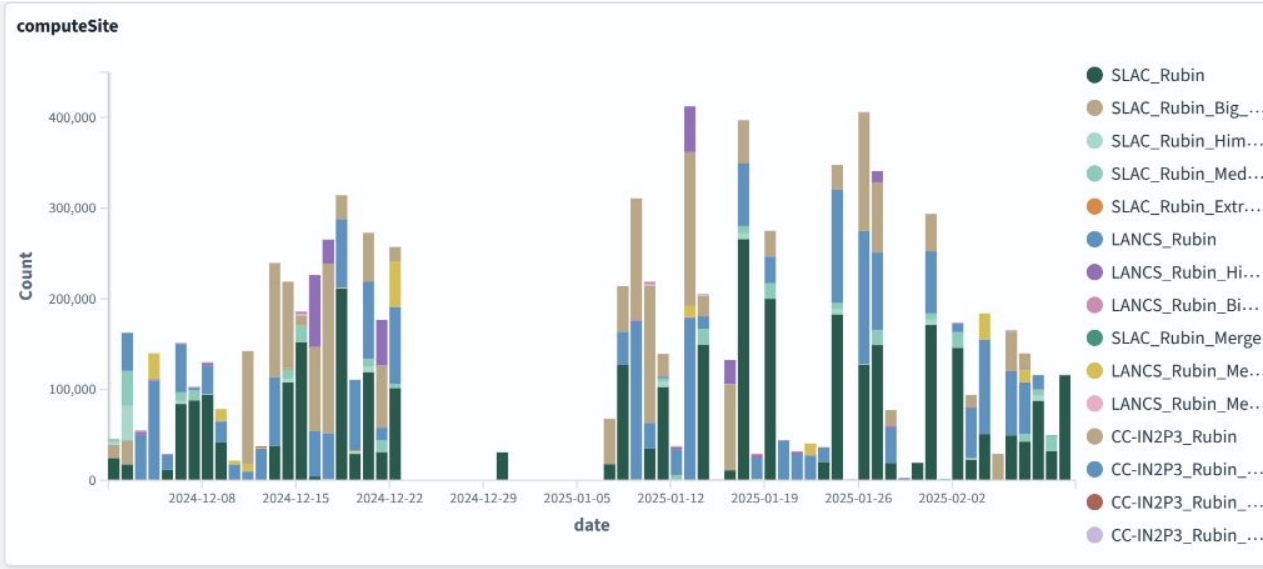
- To monitor computing resource usage, the Rubin Campaign Management team has requested dashboards for historical job data
- PanDA already uses OpenSearch for job monitoring in ATLAS, making it straightforward to extend OpenSearch for Rubin jobs
- The script is currently running on a temp pod to push job-related info into OpenSearch

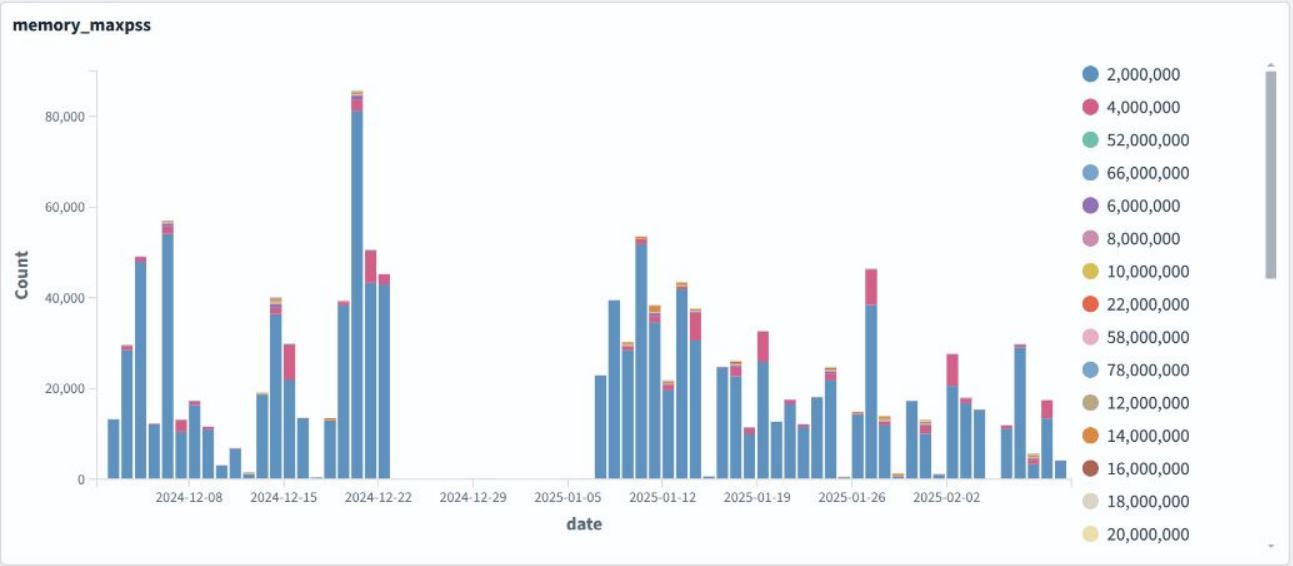


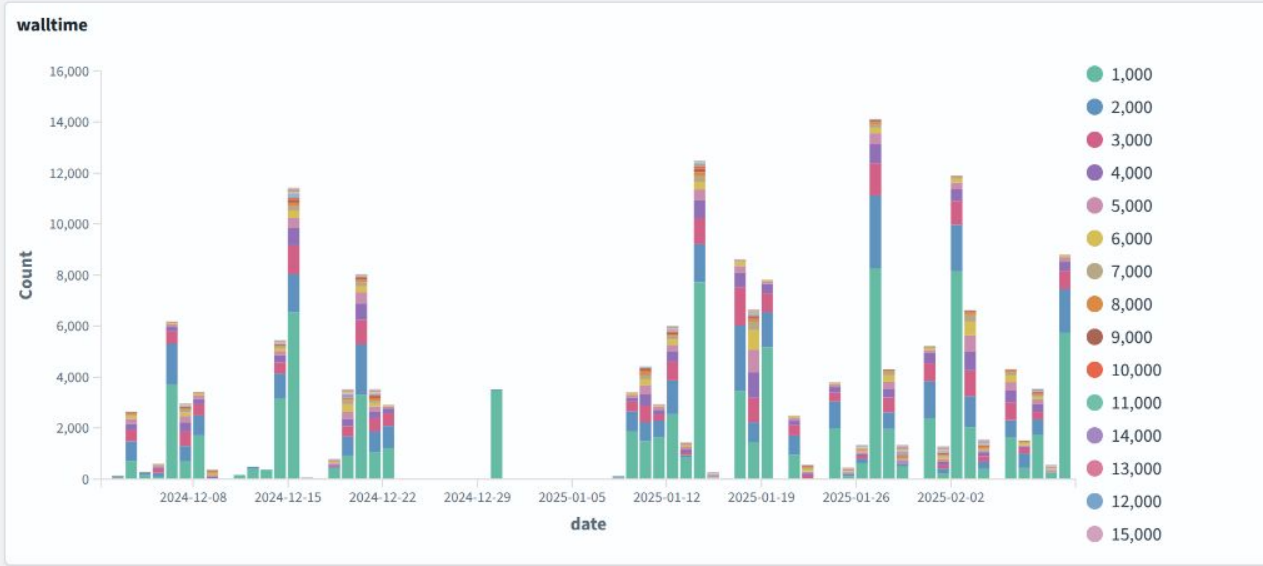
Time	_source
Feb 10, 2025 @ 09:52:03.143	pandaid: 34,364,108 modificationtime: Feb 10, 2025 @ 09:52:03.143 jobdefinitionid: 0 schedulerid: harvester-SLAC_Harvester_S3DF pilotid: https://storage.googleapis.com/drp-us-central1-logging/logs/SLAC_Rubin/PandaJob_34364108/pilotlog.txt|SLURM|PR|3.8.2.8 creationtime: Feb 10, 2025 @ 09:51:14.252 creationhost: - modificationhost: sdfmilan005 atlasrelease: transformation: https://storage.googleapis.com/drp-us-central1-containers/bash-c-enc homepackage: - prodserieslabel: pandatest prodsourcelabel: managed produserid: Huan Lin assignedpriority: 500 currentpriority: 500 attemptnr: 1 maxattempt: 1 jobstatus: finished jobname: LSSTComCam_runs_DRP_DP1_w_2025_06_DM-48810_hips_goodSeeing_20250210T144926Z_01_pipetaskInit_01_10828_30059.34364108 maxcpucount: 0 maxcpuunit: - maxdiskcount: 300 maxdiskunit: MB ipconnectivity: yes minramcount: 3,600 minramunit: MB starttime: Feb 10, 2025 @ 09:51:34.

OpenSearch Dashboards









Conclusions

- Part of PanDA's codebase around for almost 20 years - while significant progress has been made to modernize the PanDA codebase and infrastructure: **continuous** effort is **essential**
- **Enhancing** development practices
  - Peer reviews for pull requests when possible/needed
  - Future-proof the codebase, ensuring it is maintainable and scalable
  - Introducing development tools to ensure code consistency and quality
- **Various** improvements specifically to Vera Rubin Observatory:
  - Multi-site processing with PanDA has been **integrated** into the Rubin pipeline middleware
  - Improvements on many PanDA components make the Rubin workflows run more **efficiently**:
    - A medium-sized workflow now achieves a wall time comparable to batch processing for a similar workflow
  - OOM errors are handled better in PanDA
    - Failed jobs due to low memory are automatically retried with boosted memory
  - Event Service is integrated into Rubin-PanDA to cluster short jobs into longer ones, significantly reducing the load on the PanDA system
  - Pool of K8s ingress nodes significantly improved issues related to the network
  - Job Monitoring on OpenSearch is being deployed at USDF. Information extracted from the PanDA database and ingested into Opensearch. Dashboards will be built for historical job data monitoring either in the OpenSearch Dashboard and/or Grafana

Thank you for your attention!

-

Questions?

Brookhaven National Laboratory

Backup Slides

PanDA Pilot Site Environment

- PanDA system
  - PanDA service deployed on USDF k8s
  - PanDA pilot site environment deployed on cvmfs
- PanDA Pilot environment on cvmfs
  - /cvmfs/sw.lsst.eu/linux-x86_64/panda_env/
    - Conda: dependency packages such as Google-logging
    - Pilot wrapper
      - Site specific configuration LSST_LOCAL_PROLOG
      - `Eg: export LSST_LOCAL_PROLOG=/sdf/home/l/lsstsvc1/.lsst/local_prolog.sh`
    - Pilot code and pilot configuration
    - CRIC
    - (Rucio)
  - Version control
    - https://github.com/lsst-dm/panda-conf/tags

Brookhaven National Laboratory

PanDA Robot Token

- A token that can be automatically refreshed for a long period (eg: one year)
  - First initial token (requiring users to sign in on IAM)
    - A normal user ID token (7 days), used to authenticate to different services
    - A long term refresh token (eg: one year), kept privately, can be used to refresh the user ID token. With a cron script, the refresh token can be used to automatically generate the user ID token
- Purpose:
  - This token can be used for other daemon services to access the PanDA system
    - CM Service, Monitors?

Brookhaven National Laboratory