



FIFE Roadmap Update: Summer 2021

The work of many
August 19th 2021



FIFE Roadmap Workshop

- The goal of the roadmap discussion is to both inform experiments and gather feedback about strategic infrastructure changes and computing service modifications
- SCD has budgetary, security, and effort constraints that must be met
- Must also keep track of industry best practices and available tools
- Constant large meetings between experiments and service providers are not productive to developing strategy
- Feedback concerning specific enhancements, changes, and bugs are important on day-to-day basis
- Reminder: [standing FIFE meetings](#) are every other Thursday at 1400 CT. Experimenters welcome and **encouraged to attend!**



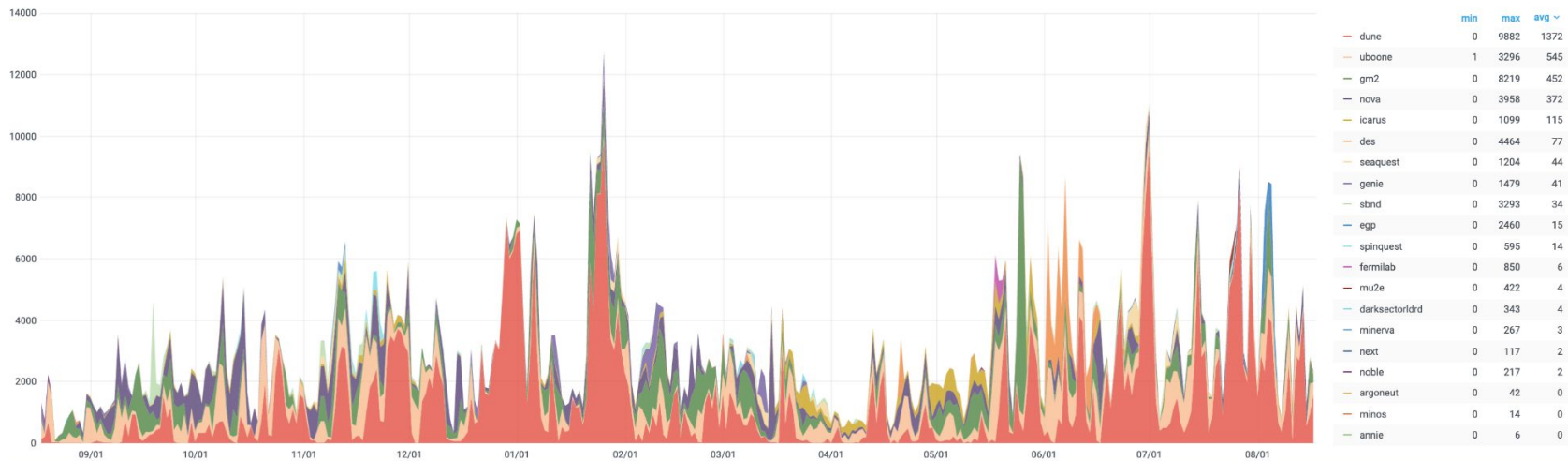
Events and trends since the 2020 workshop



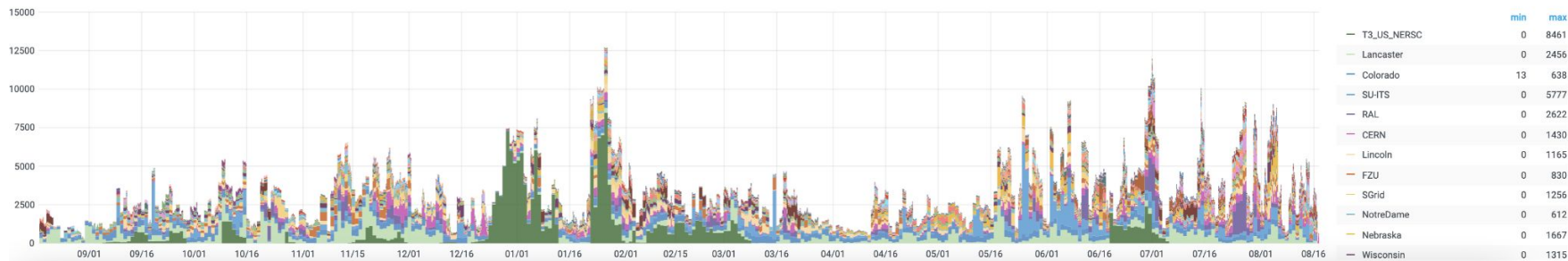
- Several important changes to jobsub and its defaults
 - SL6 reached end of life. SL6 Singularity container available but no longer updated
 - Singularity
 - Automatic singularity invocation at supported sites: problem of missing libraries on worker nodes disappears!
 - Can specify your desired container. The default is one built from the same Docker file used for the FermiGrid workers. More details [here](#).
 - Offsite running included as a default (now opt-out instead of opt-in)
 - Singularity support required on the worker node by default unless overridden (only useful for site debugging); eliminates problems with broken Singularity offsite
- Brought the [summer school](#) back after few-year hiatus
 - Slides and session recordings are at the link; share with your users!

Remote Computing resources

Offsite Running Jobs by Experiment

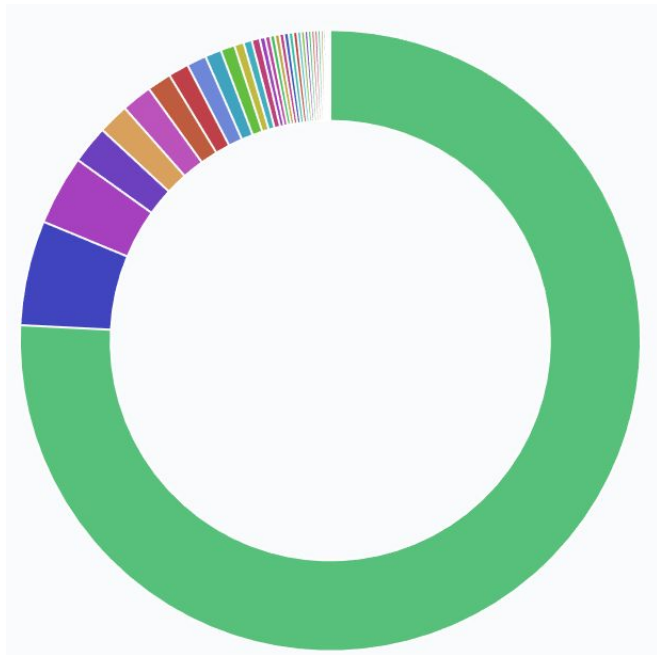


Offsite Running Jobs by Site

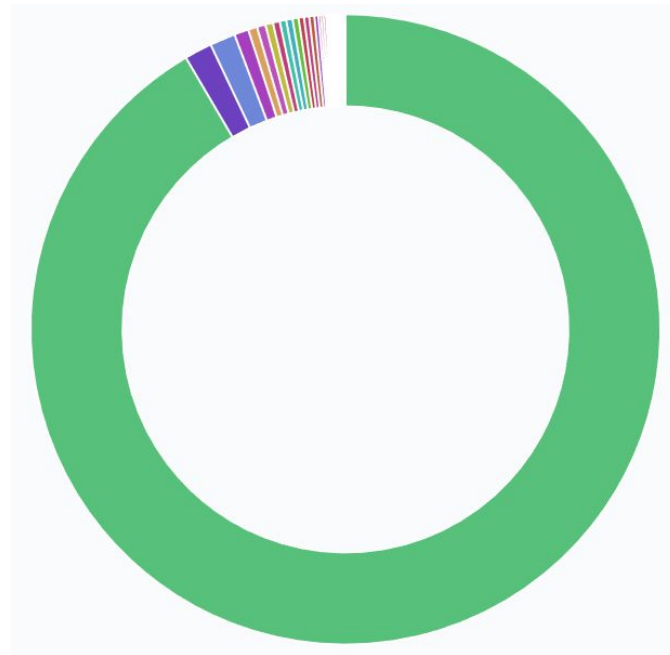


Remote Computing Resources (2)

Production: 75% FermiGrid



Analysis: 92% FermiGrid



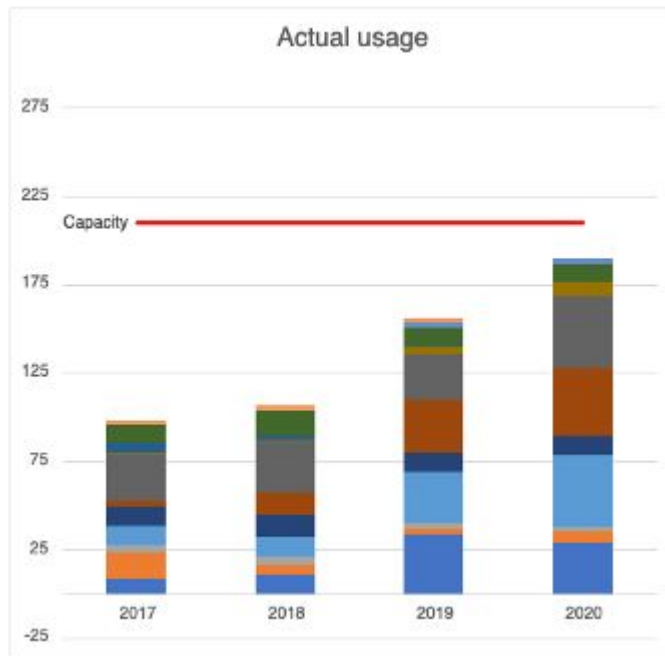
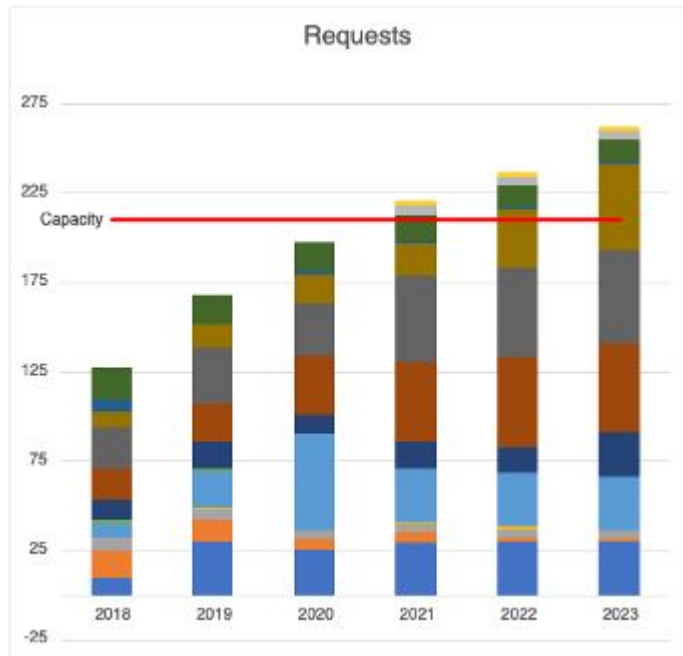
Down to 87% since making
offsite part of the defaults

For the last one year. Plots weighted by wall time, not job count

Experiment Requests and resources

FermiGrid has expanded but significant chunks are out of warranty. Can cover about 240M slot-hours, but that's over the year and expt peaks tend to overlap

Also remember slot weight: higher-memory jobs reduce the maximum number of concurrent jobs



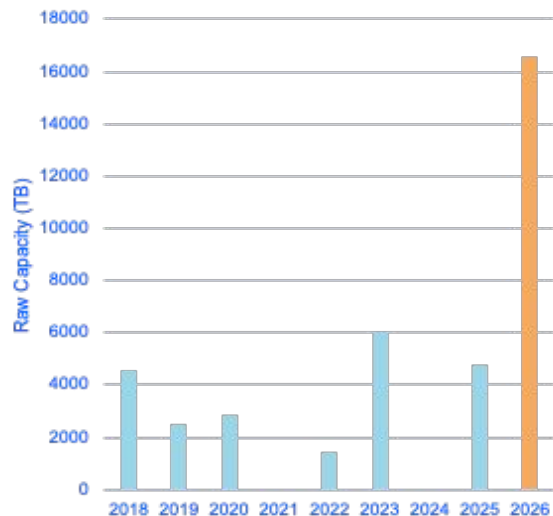
Storage Resources

Recent dCache disk purchases are not keeping up with warranty expiration

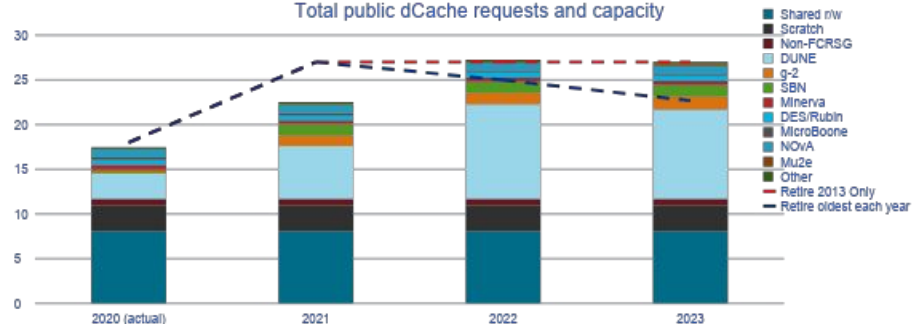
~27 PB total; ~7 PB OOW; ~3 PB more OOW this year

Tape requests are very large - double usage by end of 2022 - but past requests have not been very accurate. **All experiments need to consider dataset lifecycles and discarding unneeded data**

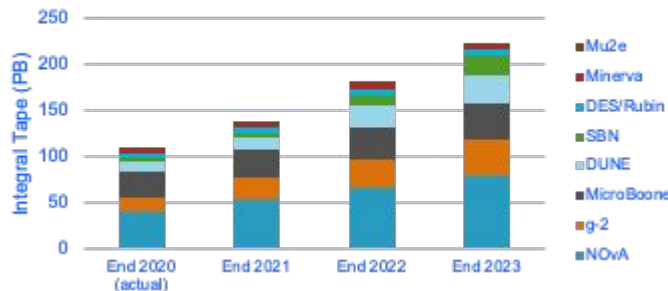
End of Warranty Dates for Public dCache Disk



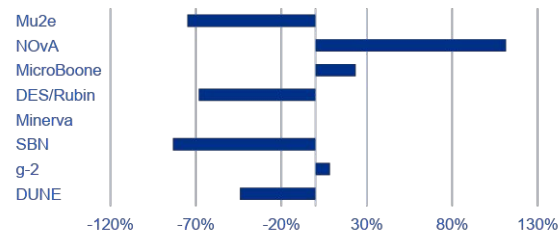
Total public dCache requests and capacity



Integral Tape Volume

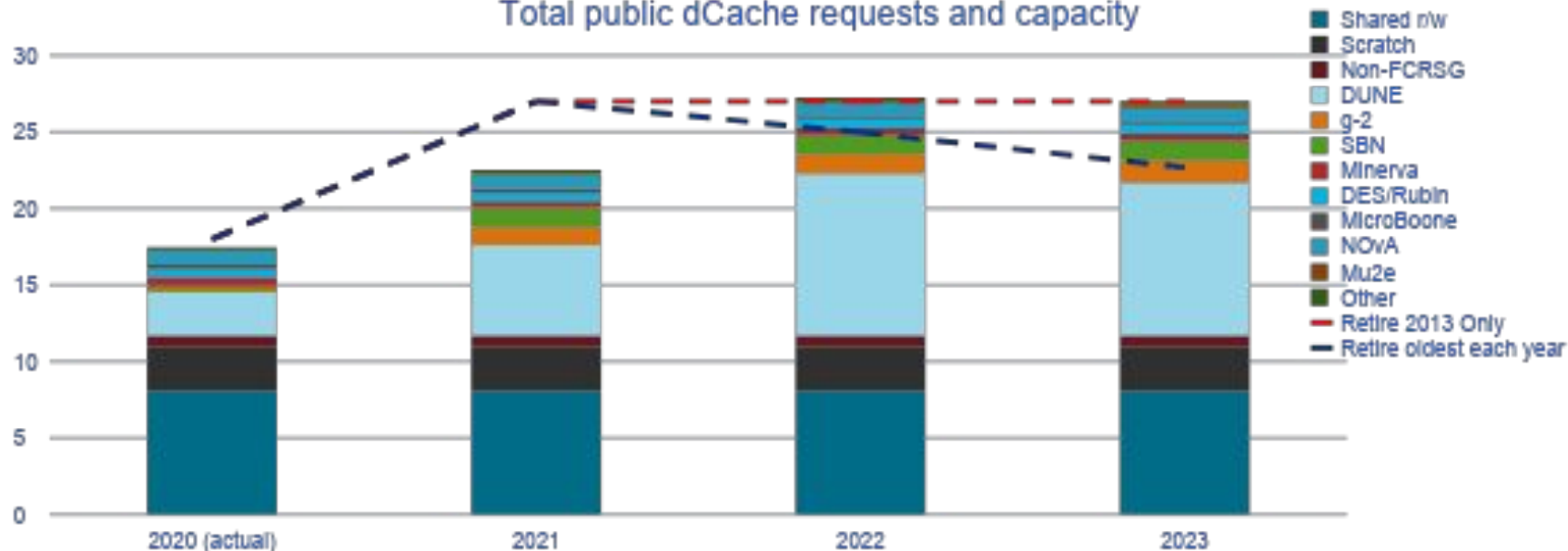


2020 Tape Actual Use Difference from Request



Total dCache requests

Total public dCache requests and capacity



- Assumes no increase in scratch or shared space
- Dashed lines show capacity (usable, no replication).
 - Assumes no additional purchases before 2023
- Red line – retire/repurpose 2013 disks only
- Blue line – retire/repurpose oldest each year

Today's Topics

- Storage
- Data management
- Metadata Catalog
- Code distribution
- Jobsub Lite
- Landscape/Monitoring
- POMS
- Project-dash-py
- Security/Federated Identity
- Elastic Analysis Facility
- Spack
- CI
- Wilson HPC Cluster
- User DOS Policy



Storage

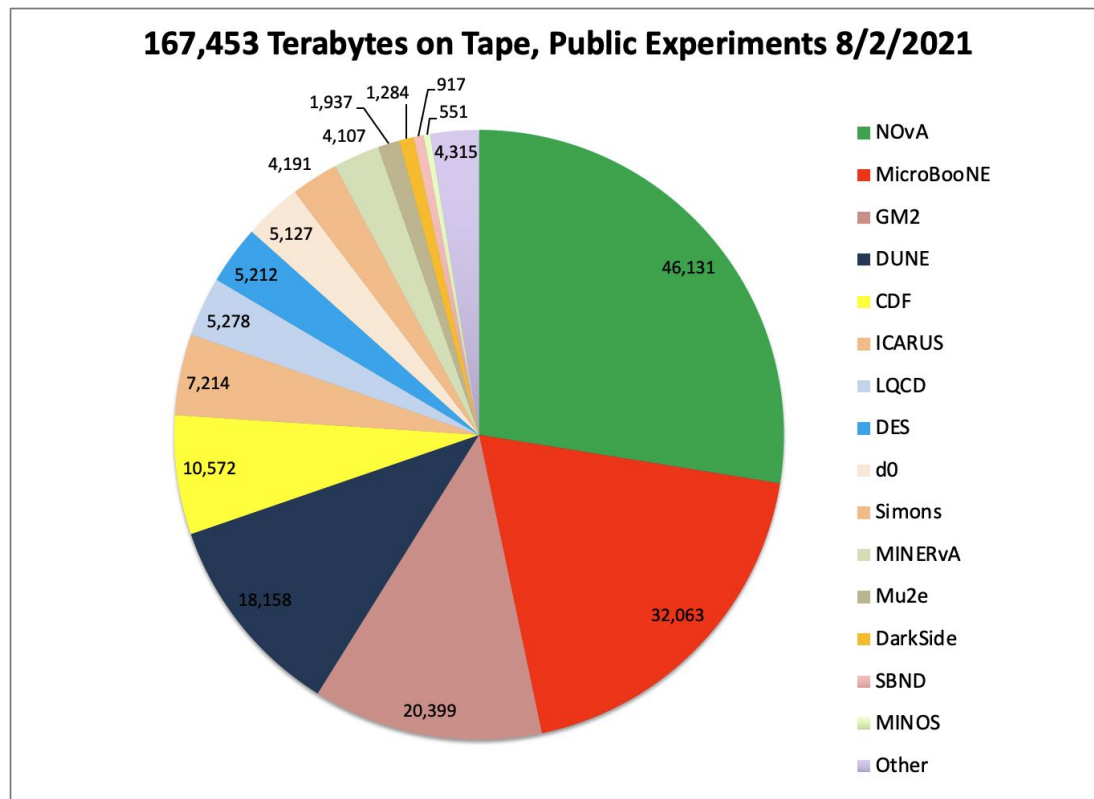
Dmitry Litvintsev

Rafael Rocha

FIFE Roadmap Workshop

Summer 2021

Current Tape Storage Usage Per Experiment



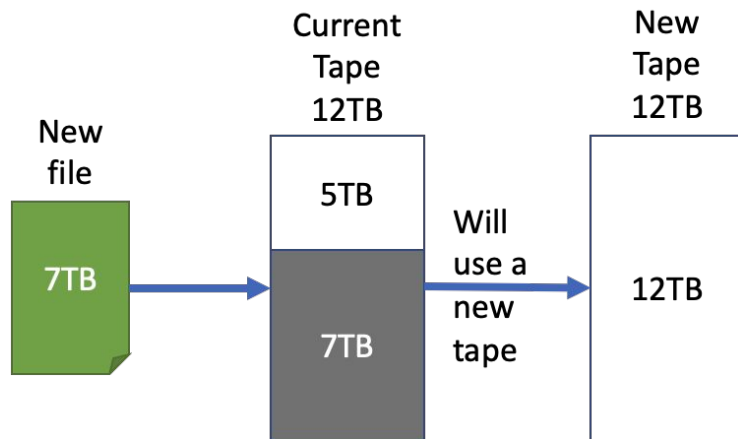
Please Do Not Write “Small Files”

- Current tapes are LTO-8 (12TB)
- Small files(SF) <0.01% of tape capacity (1.2GB)
 - SF wear out the tape faster.
 - Tape reaches EOL sooner.
 - Your information can get corrupted.
 - Causes more stress in the system for everyone.



Please Do Not Write “Small Files”

- How you can help?
 - Use dCache persistent space for continuous reading/writing.
 - Pack your files to get a bigger file.
 - Be careful of too many big files that could waste tape space.



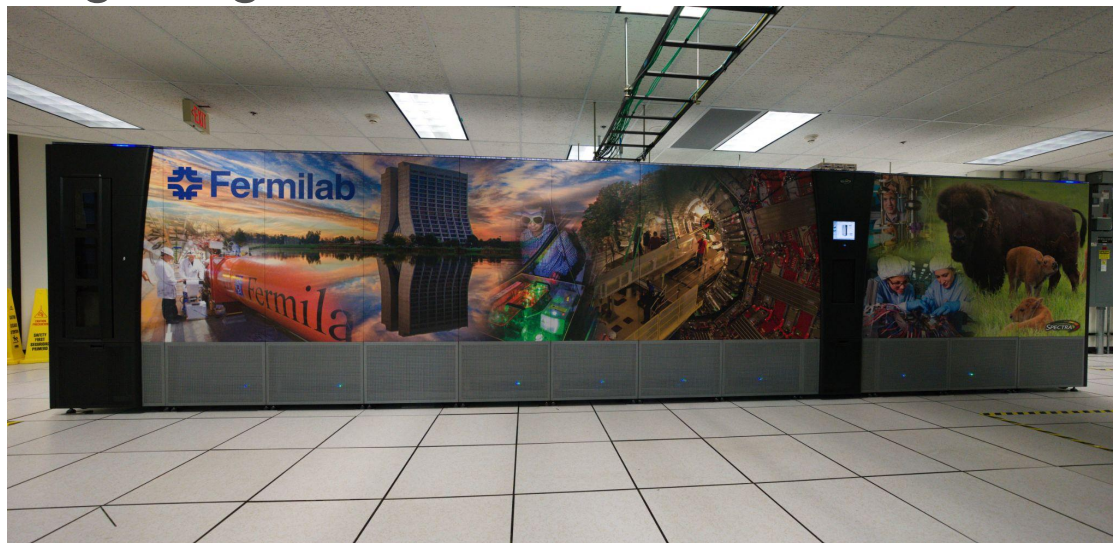
Data Migration and Tape Drive Bandwidth

- We are currently migrating T10KD tapes.
- 40PB migrated - 39PB to migrate.
- Please delete unused files by creating your own datasets lifecycle/expiration policies!
- We have 26 drives in TS4500G2.
- Drives are used for both: user and migration operations.
- Be patient when transferring to tape, plan ahead and consider the current bandwidth.



Our Plans

- Finishing Enstore integration with the new CMS tape library.
- Looking for getting a new library for public. (The actual ones are filling up)
- Looking for getting new drives for TS4500G2.



dCache News

- Growing in size:
 - 950M inode entries. Reaching 1B entries in about 1 month
 - Received 46 new pool nodes (~12PB)
- Since migration of clients to SL7 PNFS is far more stable (client nodes no longer hang requiring reboots)
 - Still more work to be done here to improve user experience

dCache in this current FY

- All dCache hosts upgraded to SL7.
- New head/door node hosts.
- New namespace (chimera) DB host and upgrade to postgresql 11.
- Allow list based firewalls configured on dCache (many thanks to SSI for assistance).
- Decommissioned "plain" FTP servers.
- JWT support in HTTP has been backported from the upstream and tested with Fermilab IDP and token infrastructure.

- We run dCache 5.2 which is very old. Plan to upgrade in Sep (or Oct).
- Highlights of the next release:
 - **User/Group quota implementation** (long awaited).
Will allow to simplify persistent pool space/partitioning in particular. Details will follow.
 - **SciToken/JWT profile token support for XrootD**
- Carry over from previous FY:
 - Enable IPv6 (dual stack) throughout the system
 - Configure HA (at least for the doors)



Data Management

Brandon White

FIFE Roadmap Workshop

Summer 2021



- SAM
 - Development effort on SAM has been reduced to bug fixes and user support
 - SAM storage sizes and query complexities continue to scale, pushing the limits of the system without some more innovative optimization
- Rucio
 - Production deployments of Rucio on the Fermilab OKD infrastructure are running for both DUNE and ICARUS at this time
 - Other experiments considering the use of Rucio for data curation include Mu2e and Rubin
- Fermi File Transfer Service
 - Eventually rewrite the service from scratch in Python 3 to take advantage of modern Python's built-in concurrency, and add Rucio file ingest support

- Development of SAM has been dramatically reduced in lieu of Rucio-related support and development activities
- SAM Web Client version 3.2: **Now Available**
 - Python 3 compatibility fixes
 - HTTPS only, plain HTTP support has been removed
- All SAM stations, SAM web servers, and supporting services have been migrated to SLF7 hosts
- Users can proactively use SAM snapshots to reduce long runtimes for complicated queries by transforming expensive SAM Definition subqueries into static lists of files
- Prestaging a dataset with SAM (e.g. [samweb prestage-dataset mydataset](#)) **does not reset the LRU clock** used for dCache eviction
- If you do need to reset a file's last-used timestamp, you can use a SAM run-project
 - `samweb run-project --defname=my_example_definition --schema https 'echo %fileurl && curl -L --cert $X509_USER_PROXY --key $X509_USER_PROXY --cacert $X509_USER_PROXY --capath /etc/grid-security/certificates -H "Range: bytes=0-3" %fileurl && echo'`

- Fermilab is continuing to contribute bug fixes and features to the Rucio codebase
- Support for file Quality of Service (tape staging) operations is coming to Rucio
- OKD deployments are running in production
 - DUNE has been using their production deployment for daily data management activities for a handful of months
 - ICARUS is performing integration testing on their production deployment
 - Using FTS3 deployments located at Fermilab and CERN as transfer managers
 - Working to upgrade all deployments to the 1.26 Long-Term Support series
- Token based authentication and single sign-on is coming!
- Fermilab has been collaborating with the Vera C. Rubin Observatory on integrating Rucio into their Data Backbone system
- DUNE, ICARUS, Mu2e, and Rubin will all be using Rucio

Fermi File Transfer Service

- Fermi File Transfer Service now has a pip-installable distribution
 - Install F-FTS into a Python virtual environment on SLF7 systems
 - <https://cdcv.s.fnal.gov/redmine/projects/filetransferservice/wiki/Wiki>
 - There is **no** SLF7 UPS release
 - The SLF6 version can be made to work with some effort/compatibility libs
- A Python 3 rewrite is in the future!
 - Take advantage of modern async/await syntax instead of Twisted
- Containerization of the F-FTS service is being considered also
- For the future: plan to implement a (likely renamed!) version that provides equivalent functionality for Rucio & MetaCat
 - Expect some simplification: Rucio will handle file transfers after the initial ingest.

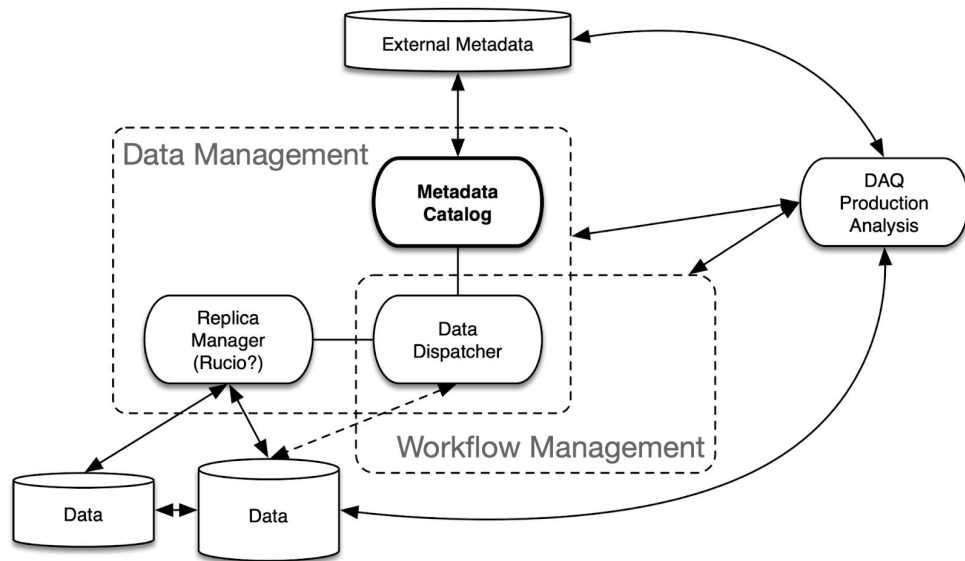
Future of Data Management

- **MetaCat**

- Will act as the new metadata catalog, replacing the metadata specific functionality that SAM provides
- Concepts and general data-model should reflect that of Rucio

- **Data Dispatcher**

- Will act as the new workflow manager, replacing the functionality that the SAM Station provides for projects and processing



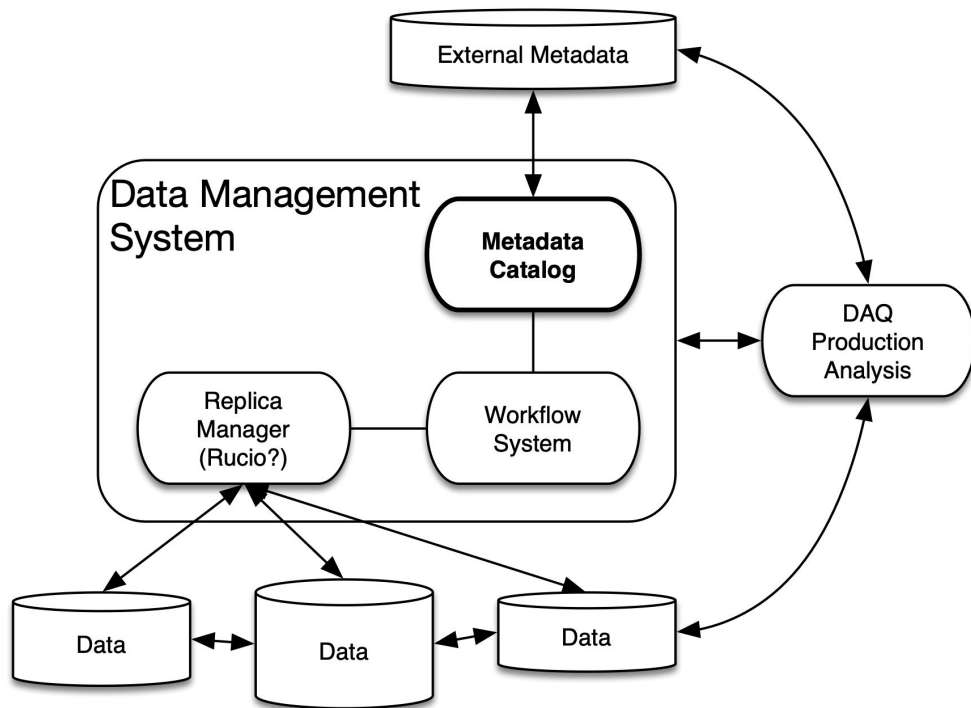


Metadata Catalog (MetaCat)

Igor Mandrichenko

Aug 2021

- Provide a metadata management component for data management systems, whether or not they use Rucio as the replica manager
- Add metadata management features missing in Rucio instead of adding them to Rucio directly
- Compatibility with Rucio at least on a conceptual, data model level



Metadata Catalog is one of 3 components of a Data Management System:

- Replica Manager (Rucio or not)
 - Where is my file?
- Workflow System
 - Next file to process?
- Metadata Catalog
 - What's in those files?

- Keep metadata about files (or objects) and datasets
- Queries
 - List of files or datasets according to the user provided criteria
 - Metadata for file
 - Metadata for dataset
 - Include external data into queries

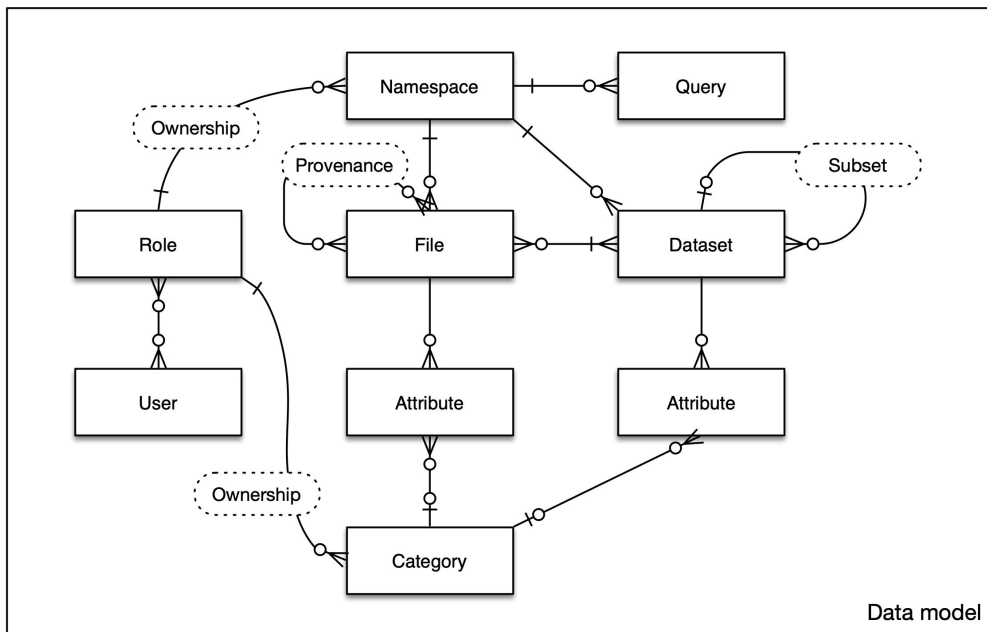
Both files and datasets can have metadata

Metadata - *any* valid JSON dictionary

Values can be

- Scalars (integer, floating point, string, boolean)
- Lists
- Dictionaries, recursively

Incorporate external data



- *Names* (file, dataset, etc.) defined within *namespaces* (Rucio term: *scope*).
- *Files* belong to *datasets* (many-to-many)
- *Dataset* may belong to a parent *dataset* (0 or 1)
 - Datasets make a tree
 - Queries can select files down the tree recursively
- *Files* and *Datasets* can have metadata attributes attached to them
- *A File* can have 0 or more *child* and/or 0 or more *parent* files - provenance

Datasets vs Queries

Dataset - collection of files

- Recorded in the database
- Files added/removed explicitly
- Has a name within a namespace

Query - *instructions* how to select files

- Recalculated every time
- Results can change at any time
- Can be saved under a name within a namespace and reused by name

Bridge:

- Query results can be saved as a new dataset or added to an existing dataset

MQL Metadata Query Language

```
files from dune:summer_2018  
  where runs[any] = 1739
```

```
children (  
  union (  
    files from mc:"summer2020%"  
      where version>"1.0",  
    files from mc:spring2020_1, mc:spring2020_2  
      where q=3.0 and t="abc"  
  )  
)
```

- File and dataset queries
- Metadata filtering by comparison ($=$, $<$, $>$, ...)
- Boolean algebra on filter expressions and queries themselves
 - Union, intersection, subtraction
- Support for arrays and dictionaries
 - “1.5 in array”, “array[any] $>$ 1.0”, “array[all] = 0”
- Value enumerations and ranges
- Support for file provenance
 - Children and parents
- Support for sub-datasets, recursively
- Asynchronous parallel queries

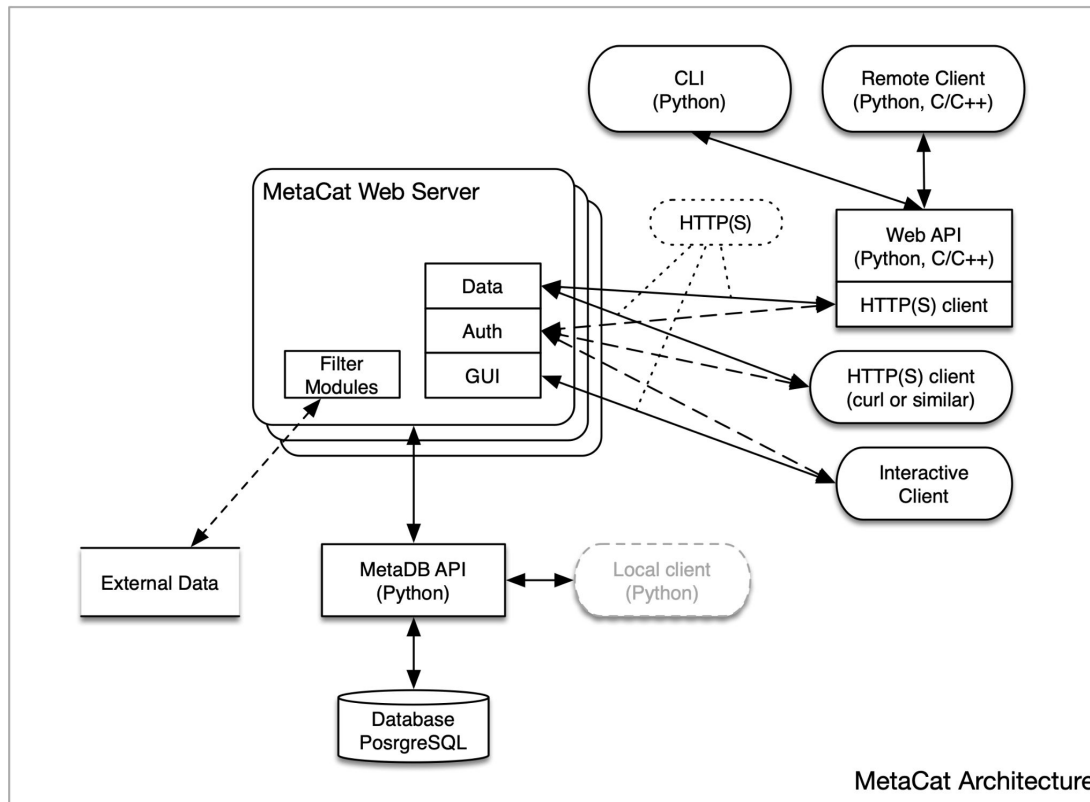
Provide a mechanism to use external data to select files for processing without copying the external data into MetaCat DB

- Implementation: user code plugged into MetaCat engine, which makes external data appear as if it was native
- Examples: runs database, conditions database - not necessarily relational data
- Use metadata stored in the external data source in the query

Since a filter is an arbitrary user code, it can be used to implement logic beyond MQL capabilities

Examples:

- Sample the query results
 - Select every n-th file
 - Select 10% of files randomly
- Merge results of 2 queries by picking random files from each of them



Typical web service application

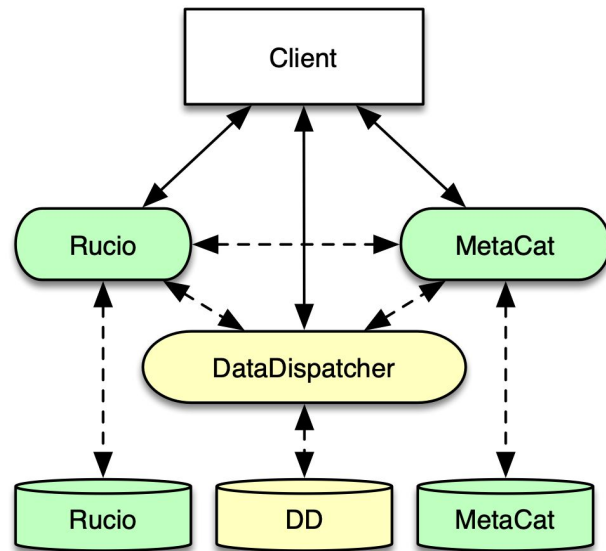
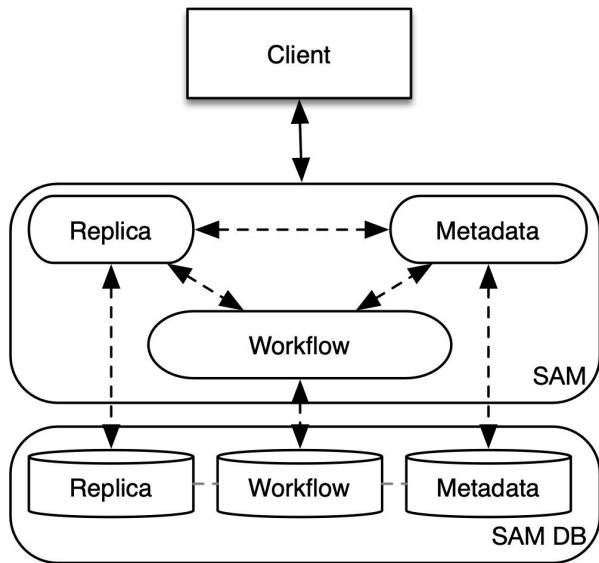
3 independent servers:

- GUI
- Data (REST)
- Authentication
 - Containerized (podman)
 - X.509+proxies support

Can run as multiple redundant servers

- Procedure to copy ProtoDUNE metadata from SAM DB to MetaCat exists.
- ~10M files in the latest snapshot
- Goal: migrate to MetaCat for ProtoDUNE 2
 - ProtoDUNE MetaCat GUI:
https://metacat.fnal.gov:9443/dune_meta_demo/app/gui/index

Production Deployment



Data Dispatcher would roughly replace SAM Station functionality
Currently, main driver/target: DUNE/ProtoDUNE

Paper:

<https://cdcv.s.fnal.gov/redmine/attachments/download/64700/MetaCat%20CHEP%202021%20paper%20v5.pdf>

Presentation slides:

<https://docs.google.com/presentation/d/17inkkOch1MyKQmy6gEO6iCVOXI5EgjtaS1bcKeRzINc/edit?usp=sharing>



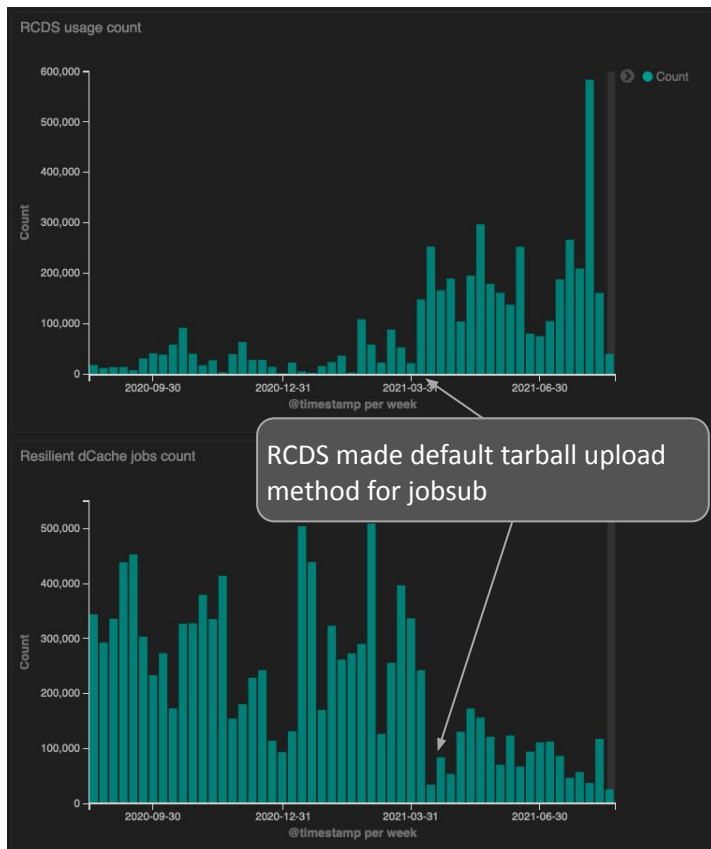
CVMFS (user code distribution)

Shreyas Bhat, Dennis Box
FIFE Roadmap Workshop
2021

Rapid Code Distribution Service (RCDS) via CVMFS

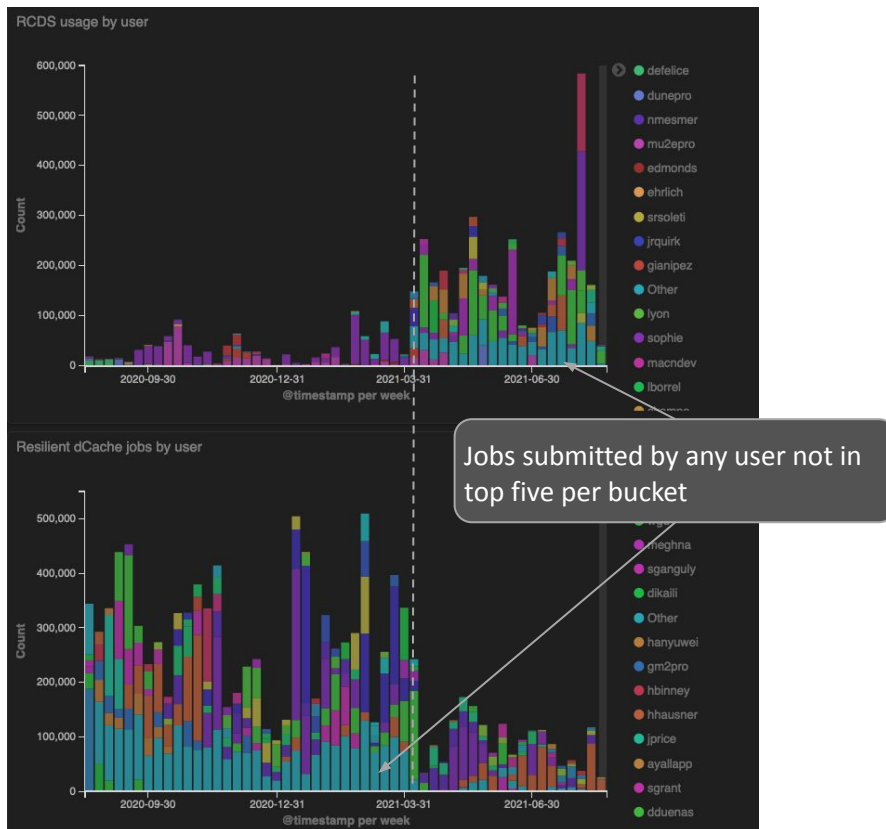
- Support for RCDS was main feature added to jobsub last year
- With jobsub 1.3.3 (released April 2021), RCDS became default method for distributing tarballs
- Some things MAY have changed for your workflow: As always, submit test jobs first!
- Opt-out for experiments is possible on case-by-case basis. Contact FIFE support to discuss
 - For individual submissions, can add *--use-pnfs-dropbox* to your jobsub_submit commands, to replicate old resilient dCache behavior, but this is NOT RECOMMENDED

RCDS usage via jobsub in the past year (1)



- People are using RCDS!
- Top graph: Count of jobs using RCDS for the last year
- Bottom graph: Count of jobs using resilient dCache for the last year

RCDS usage via jobsub in the past year (2)



- Lots of users!
- Same graphs as last slide, just broken down by user
- Vertical dashed line shows RCDS cutover
- Share of jobs not submitted by top five users decreased, after RCDS cutover: so only a few people are major users of resilient dCache

How to use it

- jobsub_submit API unchanged
- Access *unwound* tarball via softlink (that points to CVMFS)
 - We expect the tarball to be available to worker nodes two to five minutes after submission
- e.g. If your tarball was named mytar.tar, then in your job, you'll access the contents of the tar using \$CONDOR_DIR_INPUT/mytar
- And that's it!
- Details here:
[https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki/Rapid Code Distribution Service via CVMFS using Jobsub](https://cdcvs.fnal.gov/redmine/projects/jobsub/wiki/Rapid_Code_Distribution_Service_via_CVMFS_using_Jobsub)

How it works (high level)

- Submit job specifying your tarball
- Remember to put **only code** in tarball (no flux files, etc.)
 - NOT meant for production code. We mean user analysis code
- Jobsub client submits a publish request with tarball to RCDS publish server
- Tarball is published, unwound in CVMFS, and replicated across the OSG
- If files not yet available, Jobsub wrapper will wait until they are before starting user job
- Job can access it using softlink as explained on previous slide

Issues with RCDS default rollout

- All was good...except
- There was a bug in 1.3.3 that broke the `-f dropbox://` flag for submission if NOT uploading a tarball
- Three experiments brought this to our attention soon after 1.3.3 was released
- Workaround: use `--use-pnfs-dropbox` flag to replicate old resilient dCache behavior
- Fixed in 1.3.4 (released yesterday)



Jobsub

Shreyas Bhat, Dennis Box, Marc Mengel, Kevin Retzke
FIFE Roadmap Workshop
Summer 2021

- Issue
 - Current jobsub code base supports many outdated use cases, along with new features and is thus cumbersome to maintain.
 - Substantial rewrite required to support tokens.
- Solution
 - Refactor jobsub into a new server-less architecture that interacts with separate schedd, much like CMS
 - Also, like CMS, allow users to use condor commands directly if they choose or want access to advanced features
 - Native token support
- Status
 - Basic jobsub_submit client working and in testing
 - Token auth work in progress
 - Expected available for user testing this year.



Landscape

Kevin Retzke

FIFE Roadmap Workshop

Summer 2021



- No major service outages in over a year!
 - Only non-Landscape maintenance (network, databases) and site outages
- New data sources online or work in progress:
 - FTS3 events
 - Lens & Jobview request tracing
 - Enstore tape details
 - CERN Elasticsearch
 - HEPCloud Decision Engine metrics
 - SAM Station and API uptime

Landscape Meta Monitoring

User statistics similar to last year:

- 500-1500 dashboard views per day
- 60-80 users per day

Most popular dashboards:

1. POMS Stage
2. User Batch Details
3. Job/DAG Cluster Summary
4. Experiment Overview
5. Experiment Batch Details

Dashboards that should be more popular:

1. Why Are My Jobs Held?



<https://landscape.fnal.gov/monitor/d/000000319/landscape-viewership>

- Tightening security
 - Public (no SSO or certificate auth) Grafana going away (this doesn't affect many people).
<https://landscape.fnal.gov> will redirect to fifemon.
 - Direct Elasticsearch query endpoint (<https://fifemon-es.fnal.gov>) may be removed or restricted (this also shouldn't affect many people). If you do use it, try to use Lens instead -
<https://landscape.fnal.gov/lens>.
- Grafana 8 with several new/improved visualizations
 - **Dashboard refresh/makeover**
 - Currently testing in ITB (<https://landscapeitb.fnal.gov/monitor/>, on-site/VPN only)
- Elasticsearch and Kibana 7
 - Or OpenSearch (fully open-source fork <https://opensearch.org/>)
 - Better security and **more/improved visualizations**
- Improved data processing pipeline/platform
 - Probably based on Flink (<https://flink.apache.org/>)
 - **Enrich and aggregate all the things!**



POMS

Marc Mengel and Kevin Retzke

POMS

Summer 2021



POMS: Production Operations Management System

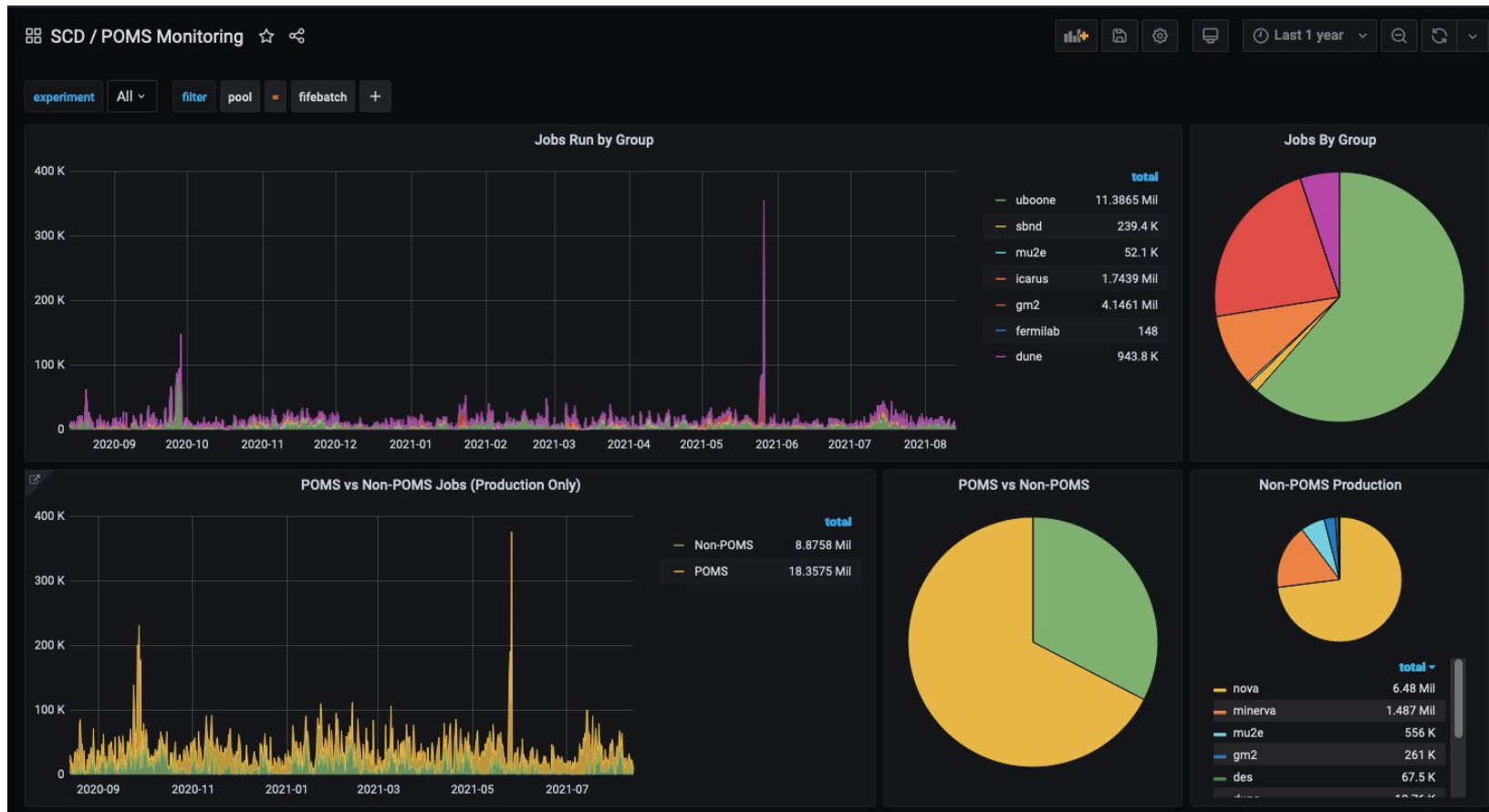
Provides an easy-to-use-environment to:

- Keep track of all the scientific production **and analysis** in one place
- Diagnose failed jobs in a few clicks and recover them
- Monitor the running workflow, parses jobs, files and logs and populate them into a database.

In the last calendar year, POMS

- production
 - launched over 40k production submissions
 - totalling over 18M jobs (**over 2x 2020!**) for
 - 6 experiments
- analysis
 - launched nearly 1k submissions
 - many using “project_py”
- Increased use of:
 - automated recovery
 - multi-stage campaigns.

Recent Usage: (cont)



POMS -- Upcoming

New release 4.3.0 coming soon

- Campaign level status page
 - Campaign-wide statistics
 - Overall file processing status
- Campaign level recovery
 - Resubmit from beginning
 - files that didn't make it all the way through
- Token support



Project-py

Vito Di Benedetto

FIFE Roadmap Workshop

Summer 2021



- **Project-py** aims to be a POMS CLI for Analysis users
 - It is a command line tool that helps to submit and monitor jobs
 - Uses same workflow configuration files used by POMS
- Support for most of common functionalities available in LArBatch/project.py (remember those are NOT supported by SCD)
 - provides a tool to convert LArBatch XML to POMS config files, tough **it is preferable to start with POMS based config files**
- It has been tested with a set of job workflows provided by SBND, uBooNE and DUNE
- The code base has been migrated to python3 (with back compatibility to python2)
- In the last release many improvements and bug fixes to address issues reported by users have been included

Project-py usage example

- Convert XML

Project.py **--convert_xml** <UserAnalysis>.xml

This creates campaign config files `${USER}_<UserAnalysis>.ini` and `${USER}_<UserAnalysis>.cfg`

- It is preferable to start directly from POMS based config files,
some templates are available [here](#)

- Create a Campaign in POMS

Project.py **--create_campaign --ini_file** `${USER}_<UserAnalysis>.ini` **--cfg_file** `${USER}_<UserAnalysis>.cfg`

- Submit jobs for a stage

Project.py **--submit --campaign** <campaign_name> **--stage** <stage_name>

or

Project.py **--submit --stage_id** <stage_id>

(more details are available in [Project-py wiki](#))

Project-py usage example (cont)

- Check submission status

Project.py `--check --submission_id` <submission_id>

- Provides:

- job summary info
 - With options to provide jobs failure/held info
- SAM project info
- Output SAM dataset info

(more details are available in the dedicated section of [Project-py wiki](#))

- SAM dataset merge and cleanup

Project.py `--merge --defname` <UserSAMdefinition> [`--mergetype` <ntuple|hist>]
to merge ntuples/histos from non Art ROOT output

Project.py `--clean --defname` <UserSAMdefinition>
to delete files, undeclare locations and delete SAM dataset

(more details are available in the *Misc* section of [Project-py wiki](#))

Contacts and documentation

- Project-py is designed as a POMS command line tool for analysis users to submit grid jobs and check their status
- Please contact us (vito@fnal.gov) or open a [redmine issue](#) for questions and feature requests
- Testers are welcome
- [Project-py wiki](#)

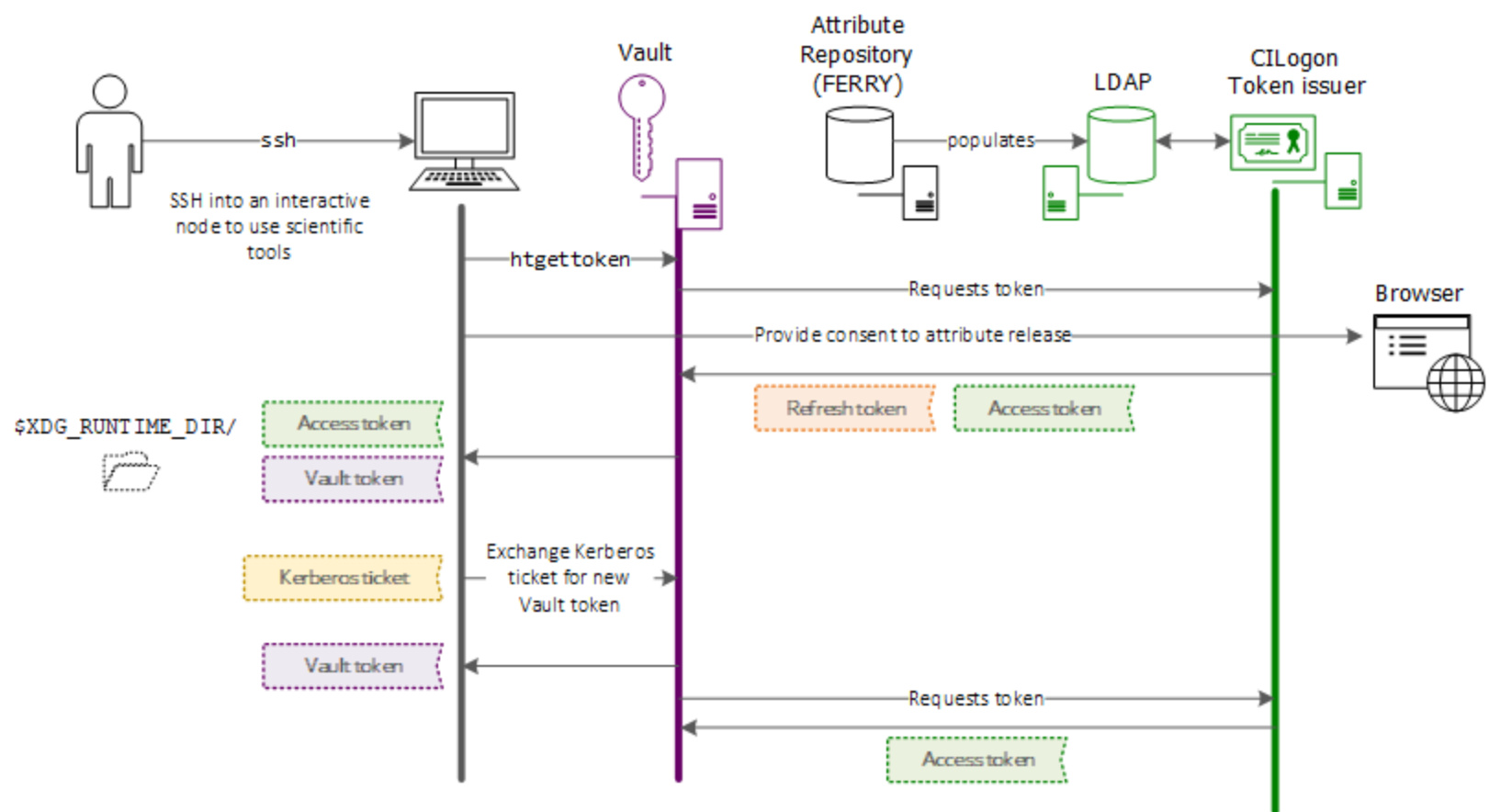


Fermilab Federated Identity Project

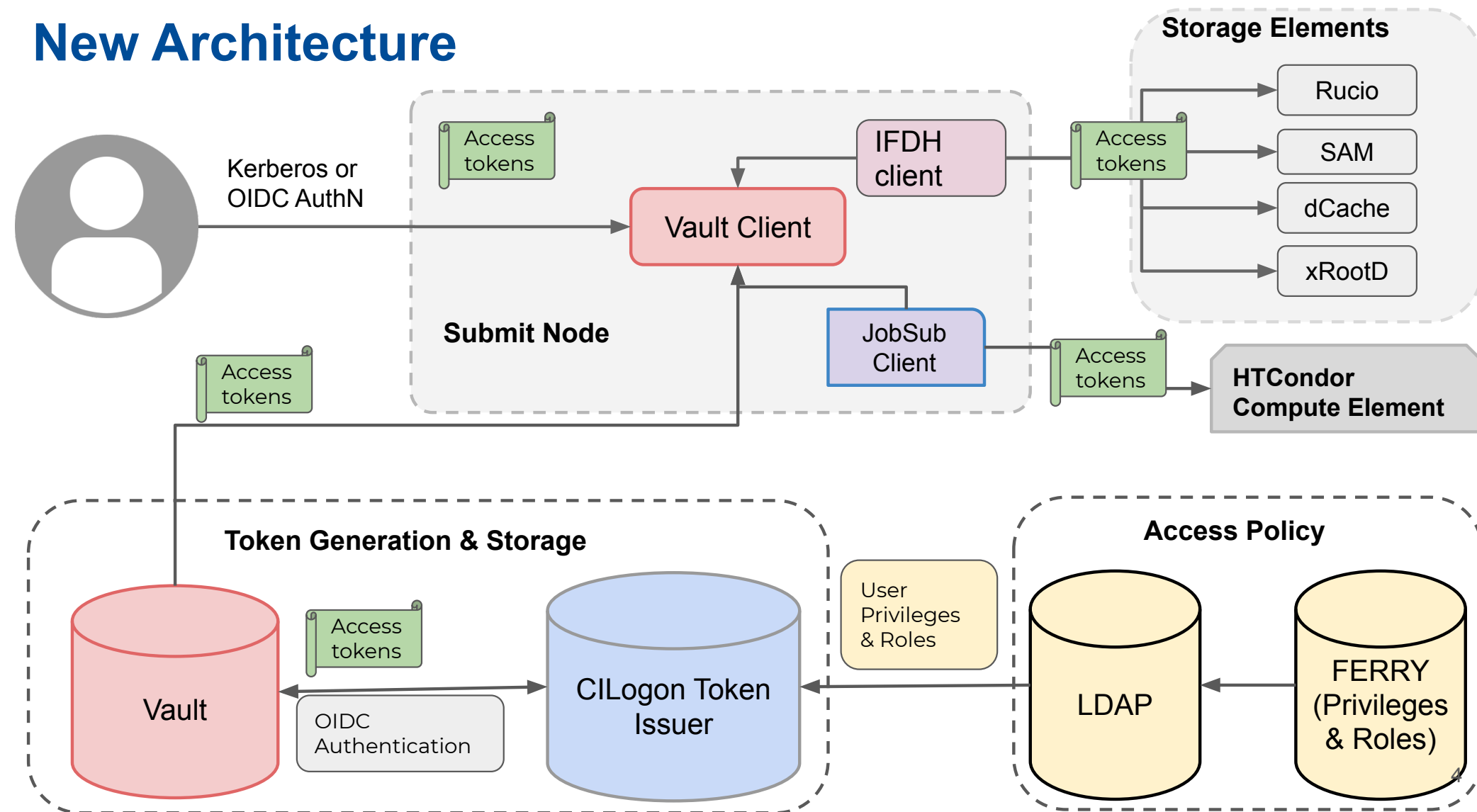
Mine Altunay
FIFE Roadmap Meeting
August 19, 2021

Federated Identity Project at Fermilab

- The goal of Federated Identity Project is to:
 - Provide our collaborators with access to lab's resources by using credentials issued by their home institutions.
 - Provide Fermilab users with access to external resources by using FNAL issued credentials.
- WLCG-led effort in significant collaboration with Open Science Grid (OSG) and aligned with InCommon Federation.
 - The goal is to have all scientific resources accessed by federated credentials.
 - Moving away from X.509 certificates towards OAuth based JWT Tokens
- Our non-WLCG VOs are also part of the project and adapts the same federation technologies.
- There are two phases of the project.
 - We are in Phase 1, where we design and build the architecture and will only allow access to users with Fermilab accounts.
 - In Phase 2, we will develop policy and procedures to extend the federated access to users at our partner institutions.



New Architecture



New Architecture

- Vault, CILogon and Ferry+LDAP are the main components of our architecture.
- CILogon is our Token Issuer.
- Vault is our token repository and also a client to CILogon Token Issuer to generate and receive tokens.
- Vault Client (htgettoken) is at the submit node and retrieves generated tokens from the Vault repository. If there is no stored token, Vault generates a new one and sends it to the Vault Client.
- Ferry is the central database where we store each users' privileges and roles (access policy). These privileges are sent to LDAP so that CILogon Token Issuer can read them programmatically and generate tokens accordingly.

Access Tokens

- Lists all the privileges a user can have.
 - Makes life very easy for resource owners since all access privileges are explicitly listed.
- Operating in a capability-based world, where each capability is an access privilege given to the user.
- Different from the certificate based access where we used groups for mapping users.
- Group concept is still alive, but evolved into capability set.

Group/Role	Capabilities
/dune	storage.read:/dune storage.write:/dune/home/joe
/dune/pro	storage.read:/dune storage.write:/dune/data
/microboone	storage.read:/microboone storage.write:/microboone/joe

Capability Set and Tokens

Scope Request	Claim Result
scope=wlcg.capabilityset:/microboone	"scope": "storage.read:/microboone storage.write:/microboone/"
scope=wlcg.capabilityset:/dune	"scope": "storage.read:/dune storage.write:/dune/home/"
scope=wlcg.capabilityset:/dune/pro	"scope": "storage.read:/dune storage.write:/dune/data"
scope=wlcg.capabilityset:/dune/pro storage.read:/dune/data	"scope": "storage.read:/dune storage.write:/dune/data storage.read:/dune/data"

- Experiments should determine which groups/capability sets are needed and which privileges should be given to each group.
 - A user cannot access a file unless she has that privilege listed in her access token.
 - A simple, short document is ideal for data entry.
- Capability sets and the privileges will be entered into Ferry. Ferry will pass this information to Token Issuer to insert into the tokens.

What is new for the experiments?

- Tokens are hidden from the end user just like certificates are, so our users will not be impacted much.
- Instead of cigetcert, we will provide a new client: htgettoken i.e. the Vault client.
 - Works similar to cigetcert. Uses a kerberos ticket to create an access token. If user does not have a Kerberos ticket, it can also do a Services username/password authentication.
 - In the very initial step, the user must approve the token creation via browser. This only happens once in token lifetime.
 - As long as user keeps submitting a job or access files and has a valid Kerberos ticket, htgettoken continues to renew the access token. If user has no activity for 1 month, htgettoken stops renewing.
- Vault server acts as a token repository and provides htgettoken with fresh access tokens.

What is new for the experiments?

- Robot certs and managed proxies will be replaced by robot tokens:
 - From user viewpoint, there is no impact on the user.
 - Just like a robot cert, robot tokens will keep renewing itself. Renewal will be based on kerberos tickets.
 - Cron script will invoke htgettoken every 7 days and renew the access token via Kerberos ticket.
- In the very initial step of creating the robot token, the owner of the robot token must authenticate herself once to the Token Issuer via browser and approve the token.
 - This only happens once in token lifetime.
 - Similar to initial manual process of obtaining robot certs.
- Service certificates are beyond the scope of this project and will continue to be used as before.

Current Status

- Majority of our software development and installation is complete.
- We are in the testing phase:
 - We did extensive tests with HTCondor and new tokens. Proved that tokens can be authorized and authenticated by HTCondor. Our jobs are successful so far.
 - We tested dcache and proved that it works with tokens.
- Now opening the testing phase to our experiments and their service providers.
 - All experiments and service providers can now start their own tests.
 - Our tests so far focused on running basic jobs with tokens. Experiments have more complicated job submission scenarios that must be tested directly by experiment members.

Next steps for experiments

- Determining the groups and the privileges given to each group.
 - Access to storage will be severely impacted unless correct privileges are given.
 - A short document listing all the groups and their privileges will be necessary.
- Testing their unique job submission or data access scenarios with tokens.
 - We have a kick off meeting on August 20 (Friday) with service providers to start their testing efforts. If interested, please join us.
 - We will provide documentation and help setting up HTCondor with tokens.
- Give us feedback on missing features or requests.

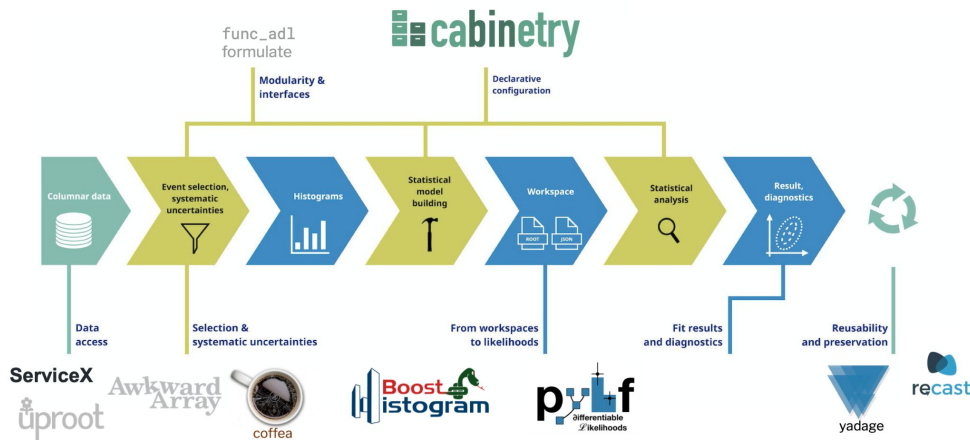


Elastic Analysis Facility

Burt Holzman
Summer 2021

Elastic Analysis Facility- Motivation

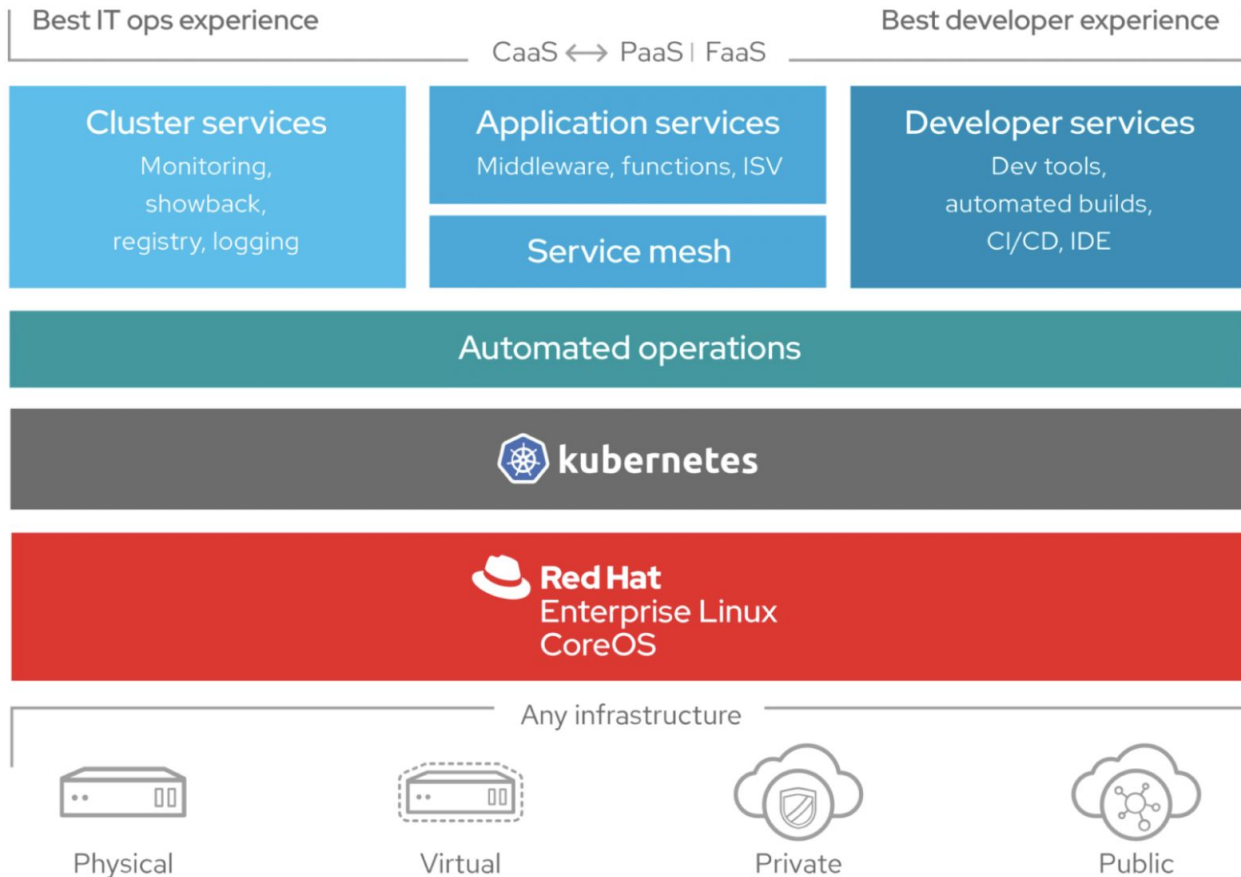
- Your typical HEP analyzer wants to run **interactive jobs** with
 - **Low-latency** access to CPU resources
 - Fast access to a significant amount of **disk**
- There is an evolving HEP python analysis ecosystem forming around this model



Elastic Analysis Facility Goals

- As a foundation layer, use a **containerized infrastructure** (Kubernetes-based). Stable, in-production, operated and supported by routine SCD Facility operations staff. “Routine” corresponds to following a set of documented procedures with responsibilities rotated on a regular basis with staff.
- A **Jupyterhub** deployment that provides a number of different environments with LDAP authentication for Fermilab users. Allow users to customize the user-space environment.
- Support “typical” (**row-wise**) analysis needs
- Support a columnar analysis low latency application (e.g. Apache Spark, **Dask**, ...)
- **Document and package** the Elastic AF so it can be deployed elsewhere (e.g. at a US CMS Tier 2)

Elastic AF: Layered Architecture



- **JupyterHub** (local workers)
 - Alpha-testing; targeting **October** for General Availability
 - Curated notebook environments; **Binder** support to come
 - Access to **GPU**s (NVidia K80s, T4s)
 - **Dask** (parallel execution) workers limited to cluster
- **JupyterHub** (remote workers)
 - In R&D; targeting **November** for GA
 - **Dask** workers running in batch systems (e.g. Fermigrid)
- **ServiceX**
 - Working through use cases for multi-tenant facility
- Additional services deployed
 - Ancillary: Harbor, Argo
 - **AI/ML**: Triton, Determined

What's next?

- This is still an R&D project
 - Timelines are estimates
 - Iterative process to improve services
- We will reach out to experiments for volunteers soon!
- Contact Burt Holzman (burt@fnal.gov) with any questions



Spack at Fermilab

Marc Mengel, Stephen White
Summer 2021



Outline

- What is Spack?
- Why Spack?
- Usage
- Dual mode for migration
- Infrastructure
- Demo
- Adding a new package
- Present/Future

What is Spack?

- A package manager which has been picked to replace UPS
 - SCD support for UPS is ending, experiments will need to switch
- like Homebrew for Mac but for supercomputer centers
 - good at forcing builds to use special optimization flags
 - lots of recipes for multithreading libraries, etc.
- supports combinatorial versioning like UPS:
 - multiple packages x versions x os-es x build variations
- Lots of [documentation](#) and [videos](#)
- Local [spack-infrastructure](#) tools

Why Spack?

Of multiple package managers

- best supercomputer build support
- meets our ups/upd requirements
 - setup package area anywhere
 - superuser not needed
 - multi-package/platform/os/variability
- wins politically as well
 - First choice by other labs..

Using Spack (for UPS users)

- start by sourcing a setup script
- "spack find" vs "ups list"
 - "spack list" lists recipes, not inst. packages
- "spack load" vs "setup"
- "spack buildcache install" vs "upd install"

However

- no concept of NULL flavor (yet).
- no concept like ups chains -> "current", "test", etc.
- very different specifier syntax
 - UPS: fred v1_0 -f Linux64bit+3.10-2.17 -q purple
 - Spack: fred @1.0 arch=linux-scientific7-x86_64 +purple

Dual mode for migration

Impossible to move everything at once, so we need both at once:

- ups_to_spack script
 - So we can "spack load" ups packages
- ups module plugin for spack
 - So we can ups “setup” spack packages
- unified-ups style layout
 - users used to looking in filesystem can still find things.

- site binary/source cache
 - essentially fnkits replacement
 - documented ([draft](#))
- /cvmfs common area
 - usable from gpvm nodes
 - usable from grid jobs.
 - Mix of native spack packages, redeclared ups packages
- spack-load in fife_launch/fife_wrap (v3_3)
 - Access packages via Spack rather than UPS
- Documentation in progress on Redmine under spack-infrastructure
- Slack channel in FNAL Computing now exists (#spack)

Demo

```
$ ssh dunegpvm09.fnal.gov
dunegpvm09:~$ . /cvmfs/fermilab.opensciencegrid.org/packages/common/setup-env.sh
dunegpvm09:~$ spack arch
linux-scientific7-broadwell
dunegpvm01:~$ spack find os=fe
==> 160 installed packages
-- linux-scientific7-i386 / gcc@4.1.1 -----
bash@2.02                jobsub_tools@1.3.1.1.4        shrc@1.10
bash@2.03                jobsub_tools@1.3.2.0          tcsh@6.06
cern@2005                perl@5.8                     tcsh@6.09
cvs@1.12.13p1            postgres_client@9.1.5.p2.6    upd_libs@5.8.5
jobsub@1.2.g1            postgres_client@9.1.6.p2.7    ups@5.1.4
...
```

Demo (cont)

```
dunegpvm09~$ spack load ifdhc
```

```
==> Error: the constraint '['ifdhc']' matches multiple packages:
```

```
  aqrb432 ifdhc@1.8.8%gcc@4.1.1 arch=darwin-yosemite-x86_64
```

```
  2vhmc2n ifdhc@1.8.8%gcc@4.1.1 arch=linux-scientificfermi5-x86_64
```

```
...
```

```
  242t7pk ifdhc@2.5.2%gcc@4.1.1 arch=linux-scientific7-x86_64
```

```
  zrbvkik ifdhc@2.5.2%gcc@4.1.1 arch=linux-scientificfermi6-x86_64
```

```
  w7nu37c ifdhc@2.5.2%gcc@4.1.1 arch=linux-scientificfermi6-x86_64
```

```
==> Error: In this context exactly one match is needed: please specify your  
constraints better.
```

```
dunegpvm09:~$ spack load ifdhc/242t7pk
```

```
dunegpvm09:~$> which ifdh
```

```
/cvmfs/fermilab.opensciencegrid.org/products/common/prd/ifdhc/v2_5_2/Linux64bit  
-3-10-2-17-python26/bin/ifdh
```

Demo (cont)

```
dunegpvm09:~$> spack buildcache list
==> Finding buildcaches on https://spack-cache-1.fnal.gov/binaries/
-- darwin-mojave-x86_64 / clang@10.0.0-apple -----
bzip2@1.0.8                lua@5.3.5                  python@2.7.16
environment-modules@4.3.1  lua-luafilesystem@1_7_0_2  readline@7.0
...
-- linux-scientific7-x86_64 / gcc@4.8.5 -----
autoconf@2.69              libsigsegv@2.11            pkgconf@1.6.1
automake@1.16.1            libtool@2.4.6              py-altgraph@0.16.1
bzip2@1.0.8                libxml2@2.9.9              py-macholib@1.11
...
-- linux-scientificfermi6-x86_64 / gcc@4.4.7 -----
bzip2@1.0.8                lua@5.3.5                  python@2.7.16
```

Adding a new package to Spack

Spack uses “recipes”, similar to those used by Homebrew to define how packages are built.

These recipes are grouped into repositories; the main one is named “builtin” and comes with Spack; others such as “fnal_art” are recipes local to Fermilab, etc.

You can make a new spack package by using **spack create** and giving it a URL from which to download a package’s source, for example a release download link from github.

Adding a new package (cont)

```
$ spack create https://github.com/jupp0r/prometheus-cpp/archive/v0.12.1.zip
==> This looks like a URL for prometheus-cpp
==> Found 10 versions of prometheus-cpp:
    0.12.1 https://github.com/jupp0r/prometheus-cpp/archive/v0.12.1.zip
    0.12.0 https://github.com/jupp0r/prometheus-cpp/archive/v0.12.0.zip
==> How many would you like to checksum? (default is 1, q to abort) 2
==> Fetching https://github.com/jupp0r/prometheus-cpp/archive/v0.12.1.zip
##### 100.0%
==> Fetching https://github.com/jupp0r/prometheus-cpp/archive/v0.12.0.zip
##### 100.0%
==> This package looks like it uses the cmake build system
==> Created template for prometheus-cpp package
==> Created package file:
/scratch/spack/spack/v0.16.0.fermi/NULL/var/spack/repos/local/packages/prometheus-cpp/package.
py
```

Adding a new package (cont)

Then you get put in your favorite editor to fix up the recipe:

```
from spack import *

class PrometheusCpp(CMakePackage):
    """FIXME: Put a proper description of your package here."""
    homepage = "https://www.example.com"
    url      = "https://github.com/jupp0r/prometheus-cpp/archive/v0.12.1.zip"

    version('0.12.1', sha256='3f9e623381a81d99e3a61053b0e671e9b5db209588a9364a980c237a19149150')
    version('0.12.0', sha256='a605904a2d40bc823bb121b2d25eb26b61065e29f0baaee6590b8058808a2cef')

    # FIXME: Add dependencies if required.
    # depends_on('foo')

    def cmake_args(self):
        # FIXME: Add arguments other than CMAKE_INSTALL_PREFIX and CMAKE_BUILD_TYPE
        # FIXME: If not needed delete this function
        args = []
        return args
```


- Art currently builds under Spack and UPS/mrb
- Larsoft is in progress, will build under both shortly.
- Integration with our CI packages is underway.
- New Spack “concretizer” (hopefully) deals better with old builds versus updated recipes/spack releases , and just plain runs faster.
- CVMFS areas moving to blended Spack/ups
- Ups/fnkits support
 - only for “legacy” experiments
 - For (tbd) specified period
- We are working with experiments to migrate their software distribution
- Tutorials will be given on request.



Continuous Integration Service

Vito Di Benedetto
FIFE Roadmap Workshop
Summer 2021



Continuous Integration overview

- Continuous Integration service is used by many FIFE experiments and by some SCD projects to test their code
- Allows to use a pretty customizable workflow configuration to test code
- Provides a dashboard with plenty of information useful to:
 - help to check and track code issues
 - monitor resource usage for the set of tested stages
- Provides support for CI validation workflows running jobs on the grid
 - helps to monitor status of physics performances and computing resource usage across code releases/branches

- **Improved GitHub integration**

- GitHub integration uses a Jenkins GitHub plugin, flexible and easy to customize
- Support for Pull Request (PR) development model
- CI builds are triggered by comment in PR or by commit to specific branches
 - PRs across different repositories can be tested together
- PR status and link to CI dashboards are provided through PR comments

- **CI Validation improvements**

- By default code to test is deployed through CVMFS via jobsub RCDS
- Jobs are configured to use a Singularity image, default image is fnal-wn-sl7, but a custom image can be used.
- CI validation results are presented in a more user friendly way

Improved GitHub Integration

Currently is used by SBN/SBND/ICARUS,
NOvA is adopting it

Comment in the PR to trigger the CI build

Comments are added to the PR with links to the CI dashboard to check detailed CI build status

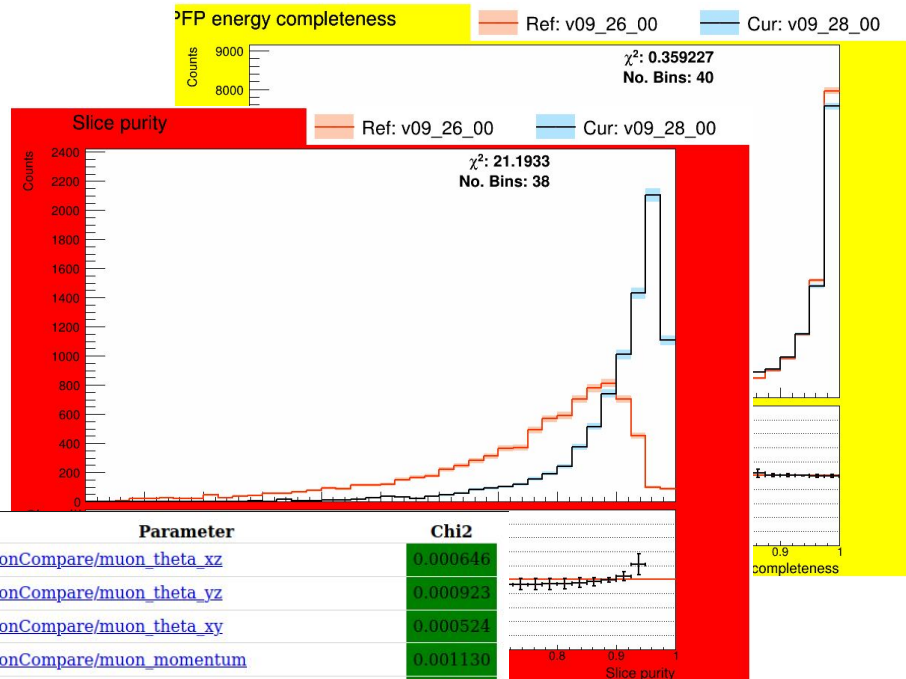
PR status is updated accordingly to the CI build status

The screenshot displays a GitHub Pull Request (PR) interface. At the top, a comment from 'vitodb' (15 hours ago) says 'trigger build'. Below it, a comment from 'FNALbuild' (15 hours ago) states 'CI build for LArSoft is in progress -- details available through the CI dashboard'. Another comment from 'FNALbuild' (15 hours ago) says '✓ CI build for LArSoft Succeeded -- details available through the CI dashboard'. A third comment from 'FNALbuild' (15 hours ago) states 'CI build for SBND is in progress -- details available through the CI dashboard' and 'parent CI build details are available through the CI dashboard'. A fourth comment from 'FNALbuild' (14 hours ago) shows a failed build: '✗ CI build for SBND Failed at phase ci_tests SBND - ignored warnings for build -- details available through the CI dashboard' and 'For more details about the failed phase, check the ci_tests SBND phase logs'. Below the comments, a section titled 'Add more commits by pushing to the feature/vitodb_test-CI-GitHub-Integration branch on vitodb:sbndcode' shows a summary of checks: 'Some checks were not successful' (1 failing and 1 successful checks). The checks listed are: 'continuous-integration/jenkins SBND' (failing) and 'continuous-integration/jenkins LArSoft' (successful). At the bottom, a green checkmark indicates 'This branch has no conflicts with the base branch' and a button to 'Merge pull request' is visible.

CI Validation improvements

This is actively used by SBND

- CI validation compares results of tested code against a reference evaluating a χ^2
- The CI allows two χ^2 thresholds to rate different level of comparison goodness
- Color background of comparison plots depends on where the χ^2 is located wrt thresholds
- A table with χ^2 values, whose background depends on the χ^2 wrt thresholds, provides a link to corresponding plots to easily access the one of interest



Sbndcode Release	Parameter	Chi2
v09_26_00 vs v09_28_00	muonCompare/muon_theta_xz	0.000646
v09_26_00 vs v09_28_00	muonCompare/muon_theta_yz	0.000923
v09_26_00 vs v09_28_00	muonCompare/muon_theta_xy	0.000524
v09_26_00 vs v09_28_00	muonCompare/muon_momentum	0.001130
v09_26_00 vs v09_28_00	muonCompare/muon_energy0	0.001026
v09_26_00 vs v09_28_00	muonCompare/muon_length	0.000870
v09_26_00 vs v09_28_00	muonCompare/muon_completeness	0.123343
v09_26_00 vs v09_28_00	muonCompare/muon_purity	1.967958
v09_26_00 vs v09_28_00	muonCompare/muon_length_metric	0.100966
v09_26_00 vs v09_28_00	muonCompare/muon_theta_xz_high_quality	0.014533

Continuous Integration Roadmap

- Working with SBND/ICARUS/DUNE to improve/extend CI validation features
- Allow use of experiment code release from CVMFS
 - makes it easier to generate validation reference and compare different code releases
- Improve and extend GitHub integration
- Add support for Spack

Wilson HPC Cluster

Joe Boyd, Glenn Cooper, Ken Schumacher, Jim Simone, *et al.*

<https://computing.fnal.gov/wilsoncluster/>



Wilson cluster

The Wilson Cluster (WC) provides High Performance Computing (HPC) services to the Fermilab community

- Tightly-coupled multi-core [parallel computing](#) (e.g. [MPI](#) + [OpenMP](#))
- [GPU-accelerated](#) (parallel) computing (e.g. [cuda](#), [TensorFlow](#), [PyTorch](#))
- Low-latency high-bandwidth (40 Gbps) QDR [Infiniband](#) network fabric
- [Lustre](#) distributed parallel file system
- Optional [Singularity HPC containers](#) for portability and reproducibility

WC batch compute worker nodes

Worker type	count	cpu	RAM [GB]	gpu
cpu	89	dual 8-core 2.6 GHz E5-2650	128	
Volta gpu	4	dual 20-core 2.5 GHz 6248	188	2 x V100
Pascal gpu	1	dual 8-core 1.7 GHz E5-2609	768	8 x P100
	1	dual 14-core 2.4 GHz E5-2680	112	2 x P100
Kepler gpu	6	dual 8-core 2.6 GHz E5-2650	128	4 x K40
“Summit”-like	1	dual 16-core 3.8 GHz Power9	1088	4 x V100

- Low latency QDR (40Gbps) Infiniband network interconnect
- 334 TB Lustre distributed parallel parallel file system

Wilson cluster use cases

- HPC application and workflow development
 - “On ramp” for campaigns that can be moved to larger HPC centers (NERSC, ALCF, OLCF)
 - Access to a Power9 compute node that is similar to a Summit gpu worker
 - Test portable containerized applications
 - WC has better turn-around time for small test jobs than the big centers!
- Training deep learning networks on one or more gpus
- Run small-scale HPC production on tens of cores / a few gpus per batch job
- When you need to do some HPC computing “now”
 - “Kick the tires”: rapidly test new ideas
 - Learn HPC programming: [C++17 pSTL](#), data parallel C++ ([DPC++](#)), [Julia](#), cuda
 - We aim to keep the barriers to getting started on WC low!

Getting started on the Wilson Cluster

Everyone in the Fermilab community already has access to the Wilson cluster as a member of their virtual organization. These VO-wide “opportunistic” accounts have lower limits on compute resources.

Groups that have specific scientific or engineering objectives that require higher levels of computing resources may [request a project](#) on the Wilson HPC facility. Approved projects are granted higher priority access to cluster resources.

The two categories of WC accounts

Attribute	VO-wide opportunistic	approved project
batch jobs	opportunistic (limited)	regular
storage quotas (default)	100GB Lustre only	1TB Lustre + 200GB in /work1; more if needed
account lifetime	forever?	review and renew yearly
user support	need to contact their own computing coordinator first to get help	create an INC or REQ via SNOW directed to WC support staff
statistics	~38 VOs; ~6,900 users	35 projects (COB Aug 10)

Groups that have specific scientific or engineering objectives that require high-performance computing resources are asked to [request a project](#) for their work on the Wilson HPC facility. Approved projects are granted higher priority access to cluster resources.



User activity mitigation policy

Ken Herner

FIFE Roadmap Workshop

Summer 2021

When you don't just affect yourself...

- From time to time we see workflows that are extremely inefficient and/or cause disruptions to one or more services to the point of affecting other users or experiments. In our experience this has never been intentional.
- While you may just see an annoying low efficiency email, others may be experiencing disruptive timeouts and/or unresponsive.
 - Low efficiency, especially in high-memory jobs, also blocks slots longer than necessary
- There is a need for a clear, enforced policy to handle these situations
- Sometimes people respond when FIFE personnel inquire what's going on, sometimes not. The policy also covers that case.
- The policy does not apply to cases where a service outage is not traceable to a specific user or experiment activity.

Reminder of the policy


- The full policy is available at [CS-doc-7045](#)
- Broken into two sections: production and non-production activities
- Activities causing emergent situations (e.g. bringing down all of dCache) and non-emergent problems treated with different initial engagement
- Spells out everyone's role in the process (end user, experiment support, service providers, etc.)
- Mitigation actions range from email to holding jobs to bans in the most severe (repeat offender) cases
- Actions tracked via SNOW ticket with experiment support personnel always copied
- **Intent is not to punish, but constructively engage in a timely manner**

Activities leading to mitigation

- Workflows that cause a degradation, outage, or denial of service to a shared resource. These include overloading a database, causing a large transfer queue on one or more dCache pools, excessive disk IO that impacts the entire machine, excessive memory usage beyond the job's request that results in entire glideins being killed. *(Obviously, these are our main concern)*
- A failure rate above 50 percent on jobs that result in a waste of over 1,000 slot-weighted hours of wall time in a 24-hour period. **Jobs finishing with a non-zero exit code are considered failed jobs, even if they are successful from the experiment's point of view.**
- Analysis jobs that consume more than 1,000 slot-weighted hours of wall time and are below the FIFE efficiency policy thresholds (defined elsewhere) in CPU, disk, or memory efficiency (whether or not the jobs are successful) for 3 or more days in a 7-day period (including weekends and/or laboratory holidays).
- Overall production output that is in violation of the efficiency policy for more than 24 hours and consumes more than 1,000 slot-weighted hours.
- Any job that violates the Fermilab security policy in any way is subject to immediate removal, and the user who submitted the job is subject to the consequences outlined in the policy.

- All types of response begin with contacting the user(s) in question, including a SNOW ticket assigned to experiment support
- **Non-emergent situations**
 - Contact user/group as described above. If no reply within one business day, proceed to emergent situations responses.
- **Emergent situations**
 - Official production activities: experiment quota will be ramped down to sustainable level until service is restored
 - User analysis activities: user's jobs for accounting group in question will be held
- Responsibilities for fixing problems are unchanged



SNOW form for quota adjustment

 > Modify Quota on FermiGrid

Request to increase (maybe just temporary) total number of batch slots/create accounting group/modify accounting allocation within quota



Requested for

Kenneth Herner




Affiliation (Experiment)

E-1071 DUNE(Dep Deep Underground Neutrino Experiment)



* Action requested

Grid Usage Suspension



Current number of job slots

3000


* Number of job slots being requested:

1

☒ Temporary?

* Date until this request will be valid

2021-07-02



* Reason

This is a test!

Reinstatement and additional steps

- If a user runs afoul of the ban, open a ticket to FIFE support asking for reinstatement for testing fixes
 - User can send small batches of jobs for testing to make fixes are working. Do not abuse this!
 - FIFE support reserves the right to require additional scale testing (since some things only show up at large scale)
- **Repeat violators** (especially if just after reinstated for testing) and consistently unresponsive users are subject to extended bans across all accounting groups. **Experiment management may be notified as needed to correct the behavior.** Hopefully none of that will ever be necessary!

- Policy has been in place for about one year
 - We have (temporarily) banned a total of **three** people so far
 - One for not responding, and two for continuing to submit the problematic workflows after being advised not to (**bans requested by the experiment!**)
- Users: make sure your official lab email is one you check, or set up forwarding of your fnal.gov email address
- Experiments: make sure the experiment support SNOW group is monitored
 - Experiments can use any means they wish for working on an issue, but please make a final record in SNOW
- **Remember: this is not to punish, but to get your attention so we can solve problems and prevent them in the future.**



Wrap-up and next workshop

Wrapping up today's workshop

- As always, thank you for attending. The slides will remain public and we will post the meeting recording later.
- Our next workshop will be in 2022
- Expect a workshop survey in the coming days
 - Questions for attendees on the format and future formats will be on there
 - Is this a good format? Would more frequent but narrowly-focused workshops be better, for example?

Another workshop in the near future

The [Snowmass](#) process is restarting soon

There will be a [Snowmass Day](#) on Sep. 24

Slack channel: snowmass2021.slack.com

A final word

Experiment efforts do not go unnoticed...lots of great science has come out in the past year!

