Workflow System status

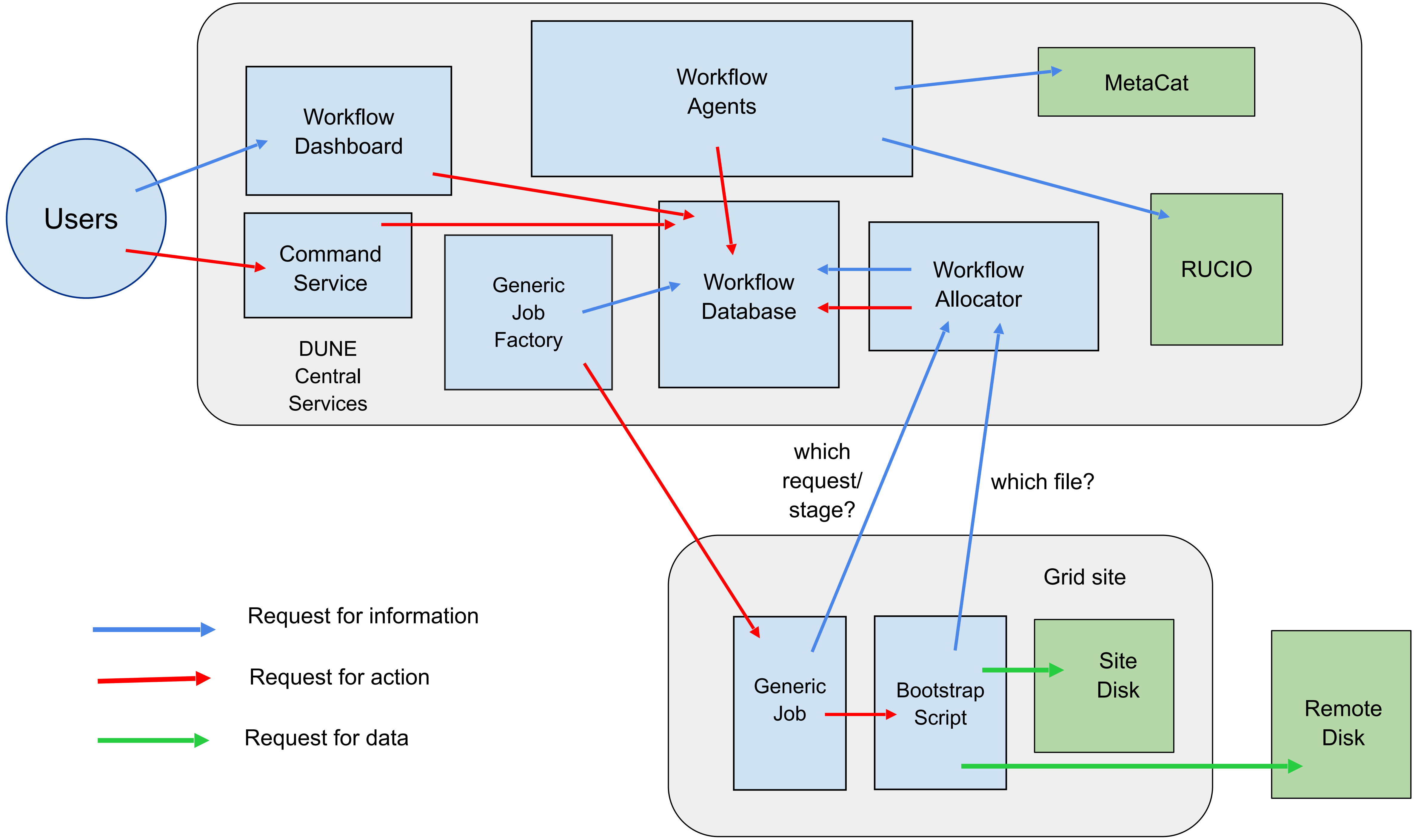
Andrew McNab

University of Manchester

Plan of this talk

- Short ten minute overview
    - Status of the components
    - Things we need from other parts of the project
- Lots and lots of backup slides
    - Mostly from the LBNC talk and its backup slides
    - Please look through them if anything isn't already clear

Workflow System recap



Workflow components

- **Workflow Database** has request descriptions, cache of file information, dynamic status of jobs, sites/storages info
- **Info Collector** agent gets info about sites, storages, users
- **Finder** agent queries **MetaCat** for lists of files and **Rucio** for location of replicas (just once, or repeatedly for a duration)
  - **FNAL Finder** (will) handle pins of replicas on FNAL dCache
- **Generic Job Factory** agent sends out jobs to be matched
- Jobs ask the **Workflow Allocator** service what to do **now**
  - "Just in time" matching based on unprocessed file locations
- The **workflow command** contacts the **command service** to display info about requests, jobs etc, or to submit requests
- **Workflow Dashboard** presents monitoring on a website

“Just in time” FNAL Finder

- At most RSEs, found replicas become available for matching immediately
- At RSEs with needs_pins=TRUE (just FNAL dCache), an additional agent has to enable the replica for matching
- FNAL Finder uses the dCache Bulk REST API to request pins for these replicas, with an expiration time
    - Once the pin has been created and the file brought online, then the agent enables the replica for matching
    - “Being available” means having an “accessible_until” value that is in the future
    - FNAL Agent sets accessible_until = pin expiration time
- This can be repeated if the file remains unallocated
- There will be a “pin budget” to limit the number of pins requested
    - Oldest files get pinned first

Workflow To Do list

- More testing of system at scale
    - See what rate brings the services down
    - See what queries are slow and look at adding extra indexes
    - Look at how to recover if timeouts, stalled jobs etc happen
- Complete FNAL Finder
    - Basic testing done
        - Including some ad-hoc stuff via ssh/curl
- Add more monitoring pages to the web Dashboard
    - More search options ("show jobs running for me within a given date range" etc)
- Test uploads with MetaCat and Rucio clients inside the generic jobs

Workflow “Shopping List”

- We need a working Rucio client that can **upload** inside jobs on random grid worker nodes
  - Queries are fine but running up against gfal2 dependency mismatches for uploads
  - Ideally from cvmfs but a recipe would be fine too
- Same for MetaCat client inside jobs
- Need access to the dCache Bulk REST API from outside FNAL
  - This is needed by the FNAL Finder agent
- Would like to see the job scripts intended to be run during the DC so bootstrap versions of them can be produced and tested
- Need to plan keep-up strategy, using Finder refind option?

Backup

Request, stages, files, replicas

- **Requests** are high level descriptions of processing campaigns, produced by people
  - Data to be processed is defined in terms of MetaCat MQL expressions
  - MetaCat gives us a list of input files to start each request
- **Stages** are steps within a request, with one or more input files per job and an identifiable application/script to be run
- **Files** exist on one or more storages as **replicas**
  - Workflow System uses Rucio to find where replicas are
  - Files have a state within each stage they are part of:
    - Unprocessed, allocated, processed

Workflow constraints (1)

- Up to now, DUNE has used SAM to manage the progress of each "project" but we are replacing SAM with Rucio+MetaCat+????
  - **First constraint: design and create a simpler and easier to maintain Workflow System, that works with Rucio and DUNE's MetaCat**
- SAM clients built into the applications request the next file in the project the job is from
  - SAM has some limited ability to choose a suitable replica based on where the job is running
  - Can process a series of files inside a job slot, reducing set up costs and adapting to CPU / IO / time limit.
  - **Second constraint: retain the ability to use the SAM "next file" model inside applications**

Workflow constraints (2)

- As we shift processing outside Fermilab, the varied “landscape” of sites vs storages is becoming obvious
  - Studies for the CDR show that accessing data at different sites can change a job’s efficiency by an order of magnitude
  - Sites also have finite bandwidth which we can saturate in some cases with too many jobs accessing remote data
- **Third constraint: workflow matching must be able to take the location of *unprocessed* files into account.**
- There are different ways to resolve this but the solution we’re prototyping is to extend the SAM “next file” model
  - **Jobs ask a central service which workflow to work on**
  - **Then ask for unprocessed files from that workflow**

Workflow Allocator

- Jobs ask a central Workflow Allocator service which request/stage to work on
- Allocator works out which stages have unprocessed files with a replica “near enough” the job
    - Files are ranked by request priority, request date, etc
    - Stages as marked as taking data from anywhere or just “nearby” storages
    - “Just in time” matching rather than trying to plan/guess ahead
- A single SQL query does the matching
    - Assigns the job to the highest ranked eligible stage
    - Job gets a bootstrap script etc in tar file
    - Script/application requests files and best replica URLs

Generic Jobs

- These are sent to sites via HTCondor
  - Sends the JSON request to the Workflow Allocator
  - Receives a tar file from the Allocator
    - Contains bootstrap script to run for that stage
    - Metadata about the request etc for debugging
- Generic Job runs the bootstrap script
  - In future, could be inside a container as the user
  - Bootstrap script or application requests file(s) to process
- Afterwards, Generic Job uploads the output files
  - Can be done with a higher privileged identity than user

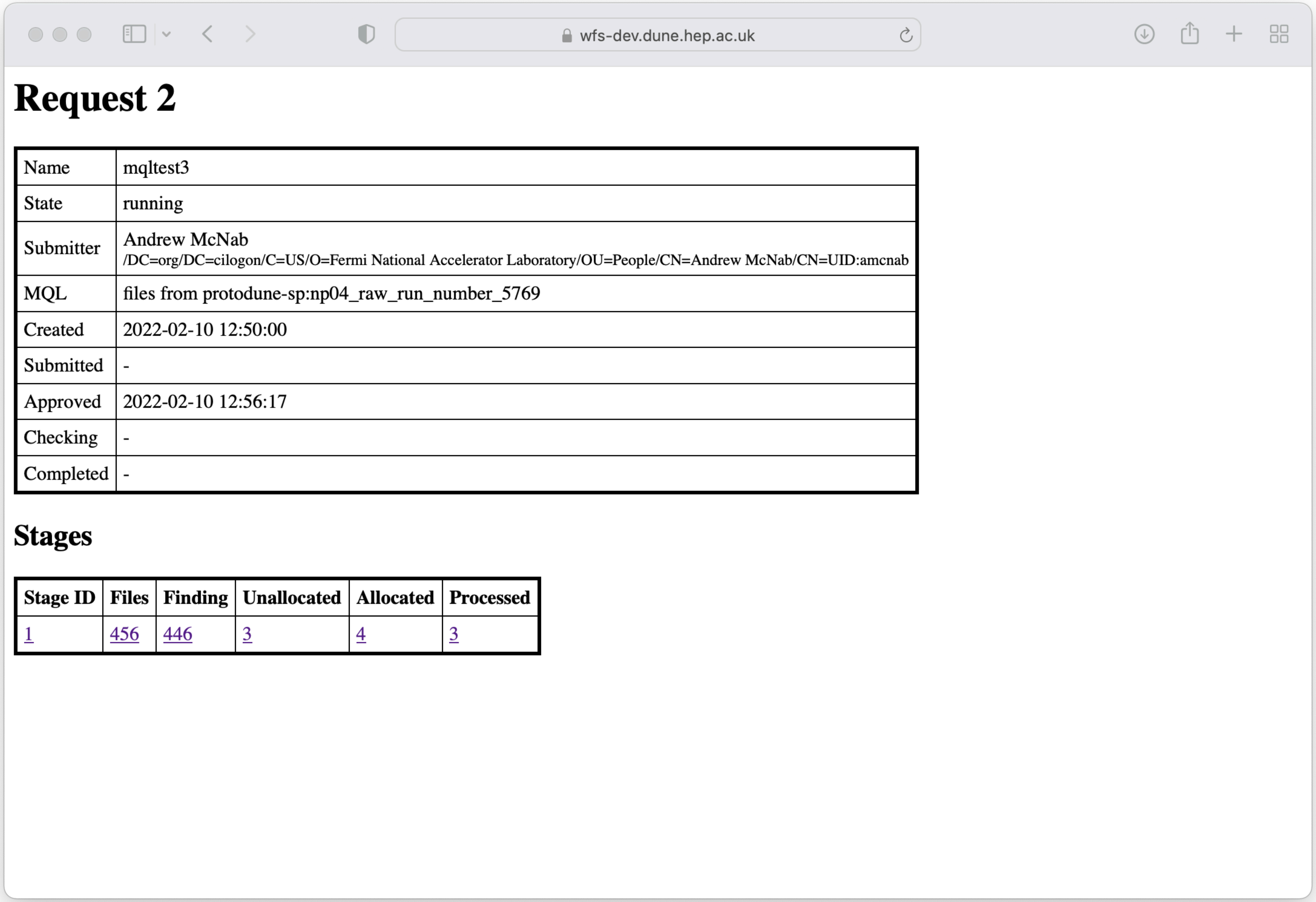
Workflow command

```
./workflow create-request --name “My request number 1” \
--mql “files from protodune-sp:np04_raw_run_number_5769”

./workflow show-requests

./workflow show-requests --request-id 2

./workflow create-stage --request-id 2 --stage-id 1 \
--max-wall-seconds 7200 --max-rss-bytes 2123456789 \
--upload-file req1—bootstrap.sh --processors 1

./workflow show-stages --request-id 2

./workflow show-files --request-id 2 --stage-id 1

./workflow show-replicas --request-id 2 --stage-id 1

./workflow approve-request --request-id 2

./workflow show-storages
```

Tests with production jobs

- Adapted production jobs to work with the Workflow System
- Workflow Database populated with file names and locations
  - Initially from SAM but now from Rucio
- Input files already located at FNAL, CERN etc
- Generic jobs sent to the sites
- Jobs start up and talk to the Workflow Allocator
  - Get bootstrap scripts and files
  - Run processing as normal
- As more components have been added, more of the set up is done automatically by agents rather than by ad-hoc scripts

Workflow Dashboard



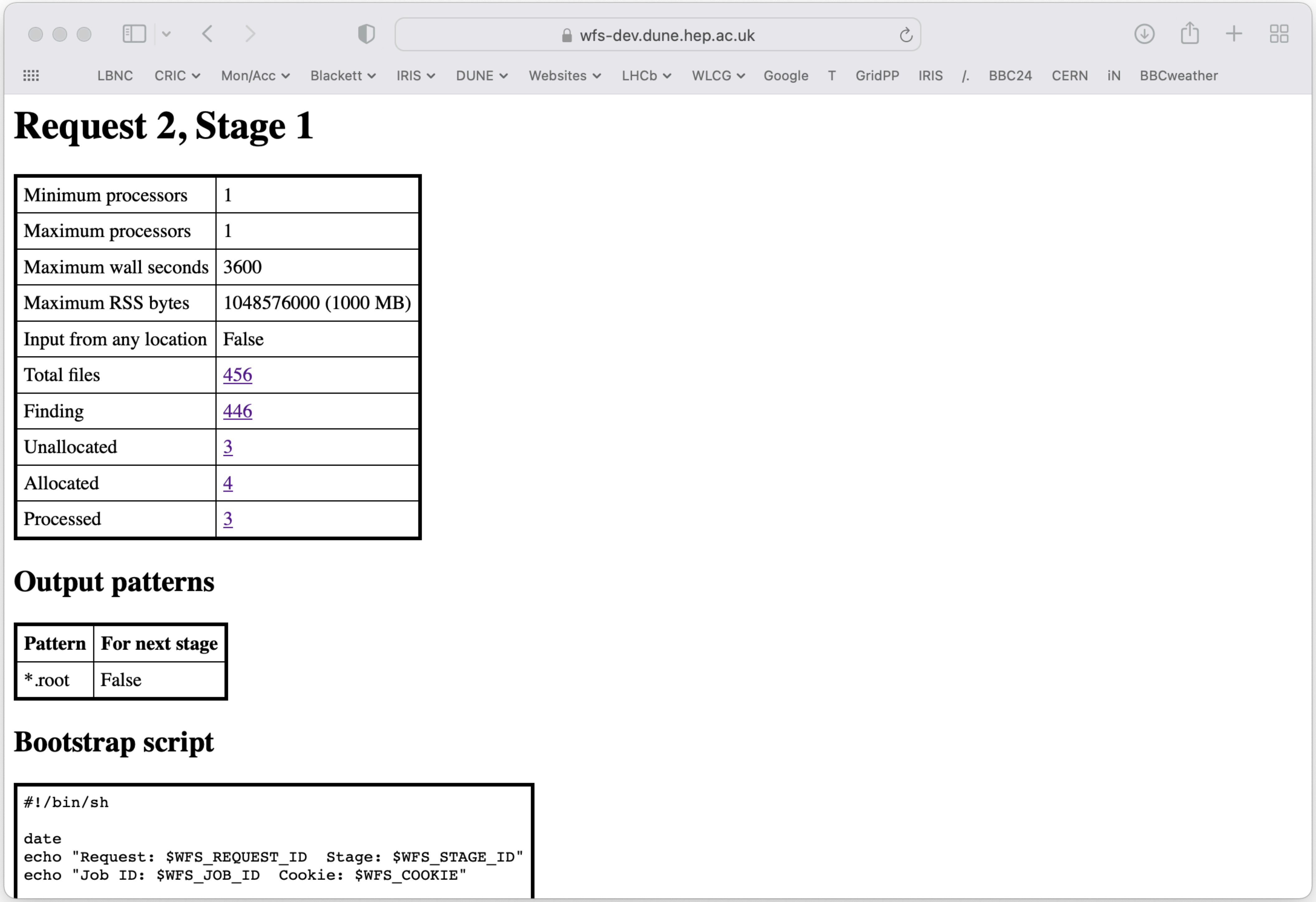
Request 2

Name	mqltest3
State	running
Submitter	Andrew McNab /DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Andrew McNab/CN=UID:amcnab
MQL	files from protodune-sp:np04_raw_run_number_5769
Created	2022-02-10 12:50:00
Submitted	-
Approved	2022-02-10 12:56:17
Checking	-
Completed	-

Stages

Stage ID	Files	Finding	Unallocated	Allocated	Processed
1	456	446	3	4	3

Workflow Dashboard



Request 2, Stage 1

Minimum processors	1
Maximum processors	1
Maximum wall seconds	3600
Maximum RSS bytes	1048576000 (1000 MB)
Input from any location	False
Total files	456
Finding	446
Unallocated	3
Allocated	4
Processed	3

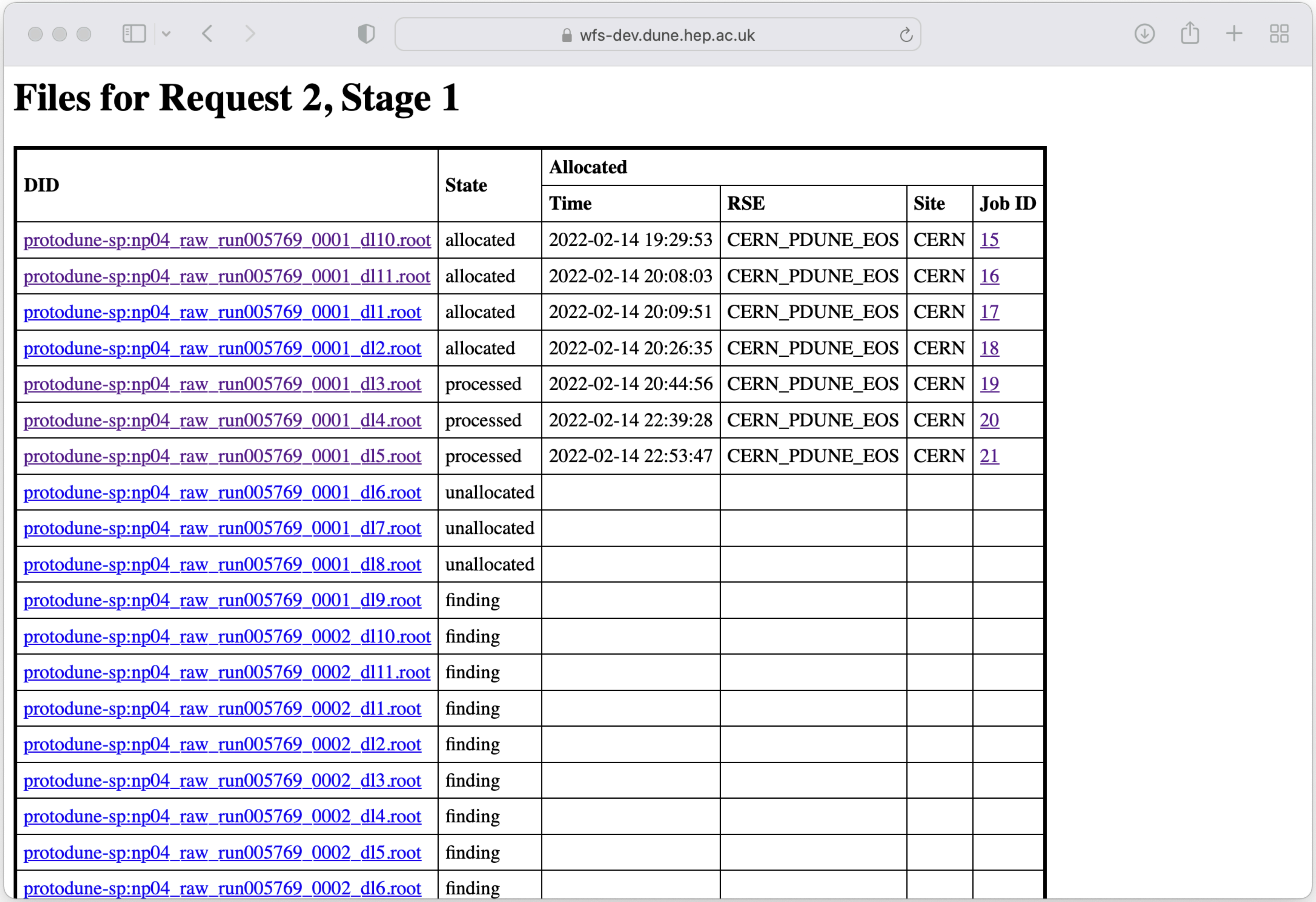
Output patterns

Pattern	For next stage
*.root	False

Bootstrap script

```
#!/bin/sh

date
echo "Request: $WFS_REQUEST_ID  Stage: $WFS_STAGE_ID"
echo "Job ID: $WFS_JOB_ID  Cookie: $WFS_COOKIE"
```

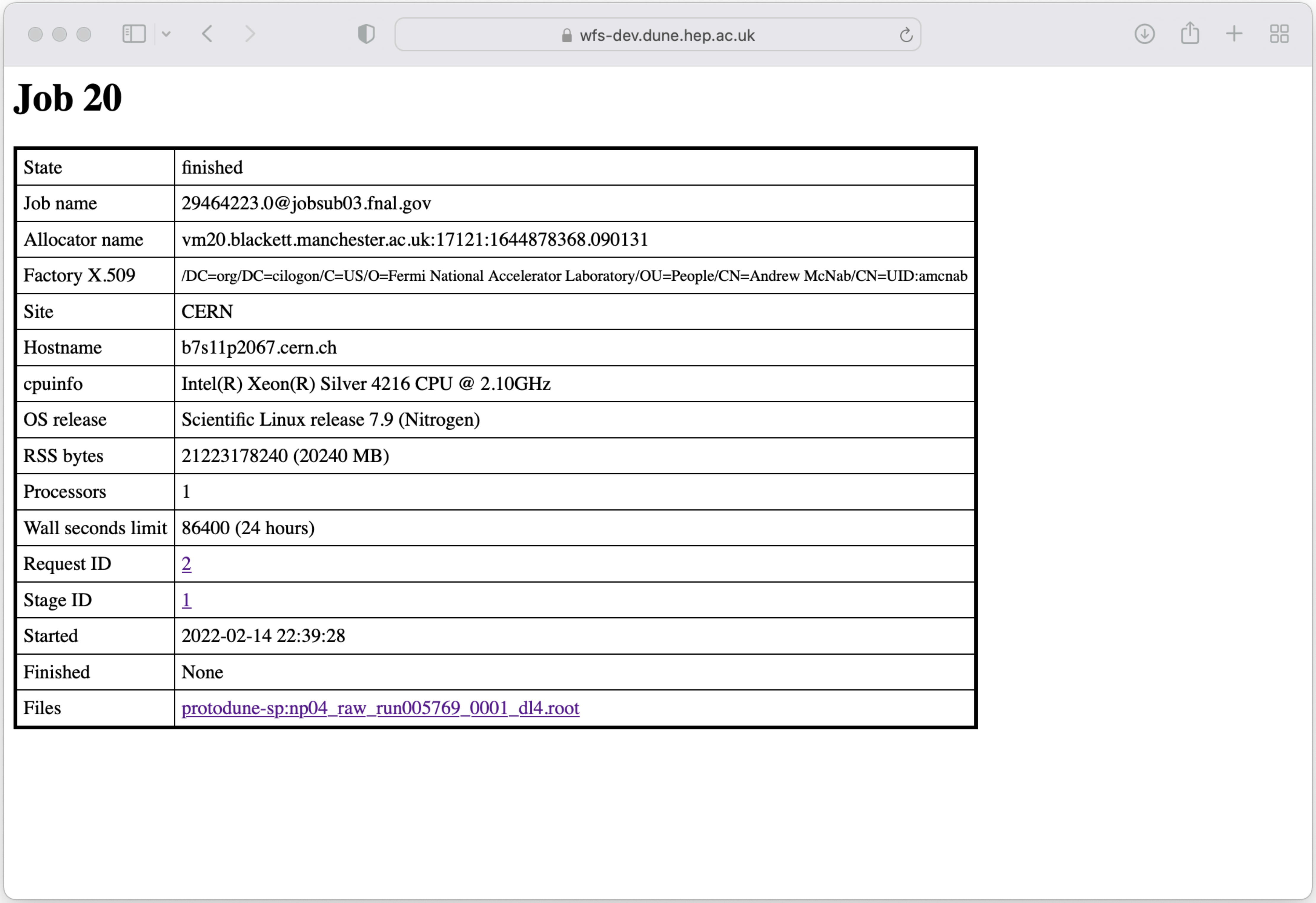
Workflow Dashboard



Files for Request 2, Stage 1

DID	State	Allocated			
		Time	RSE	Site	Job ID
protodune-sp:np04_raw_run005769_0001_dl10.root	allocated	2022-02-14 19:29:53	CERN_PDUNE_EOS	CERN	15
protodune-sp:np04_raw_run005769_0001_dl11.root	allocated	2022-02-14 20:08:03	CERN_PDUNE_EOS	CERN	16
protodune-sp:np04_raw_run005769_0001_dl1.root	allocated	2022-02-14 20:09:51	CERN_PDUNE_EOS	CERN	17
protodune-sp:np04_raw_run005769_0001_dl2.root	allocated	2022-02-14 20:26:35	CERN_PDUNE_EOS	CERN	18
protodune-sp:np04_raw_run005769_0001_dl3.root	processed	2022-02-14 20:44:56	CERN_PDUNE_EOS	CERN	19
protodune-sp:np04_raw_run005769_0001_dl4.root	processed	2022-02-14 22:39:28	CERN_PDUNE_EOS	CERN	20
protodune-sp:np04_raw_run005769_0001_dl5.root	processed	2022-02-14 22:53:47	CERN_PDUNE_EOS	CERN	21
protodune-sp:np04_raw_run005769_0001_dl6.root	unallocated				
protodune-sp:np04_raw_run005769_0001_dl7.root	unallocated				
protodune-sp:np04_raw_run005769_0001_dl8.root	unallocated				
protodune-sp:np04_raw_run005769_0001_dl9.root	finding				
protodune-sp:np04_raw_run005769_0002_dl10.root	finding				
protodune-sp:np04_raw_run005769_0002_dl11.root	finding				
protodune-sp:np04_raw_run005769_0002_dl1.root	finding				
protodune-sp:np04_raw_run005769_0002_dl2.root	finding				
protodune-sp:np04_raw_run005769_0002_dl3.root	finding				
protodune-sp:np04_raw_run005769_0002_dl4.root	finding				
protodune-sp:np04_raw_run005769_0002_dl5.root	finding				
protodune-sp:np04_raw_run005769_0002_dl6.root	finding				

Workflow Dashboard



Job 20

State	finished
Job name	29464223.0@jobsub03.fnal.gov
Allocator name	vm20.blackett.manchester.ac.uk:17121:1644878368.090131
Factory X.509	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Andrew McNab/CN=UID:amcnab
Site	CERN
Hostname	b7s11p2067.cern.ch
cpuinfo	Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
OS release	Scientific Linux release 7.9 (Nitrogen)
RSS bytes	21223178240 (20240 MB)
Processors	1
Wall seconds limit	86400 (24 hours)
Request ID	2
Stage ID	1
Started	2022-02-14 22:39:28
Finished	None
Files	protodune-sp:np04_raw_run005769_0001_dl4.root