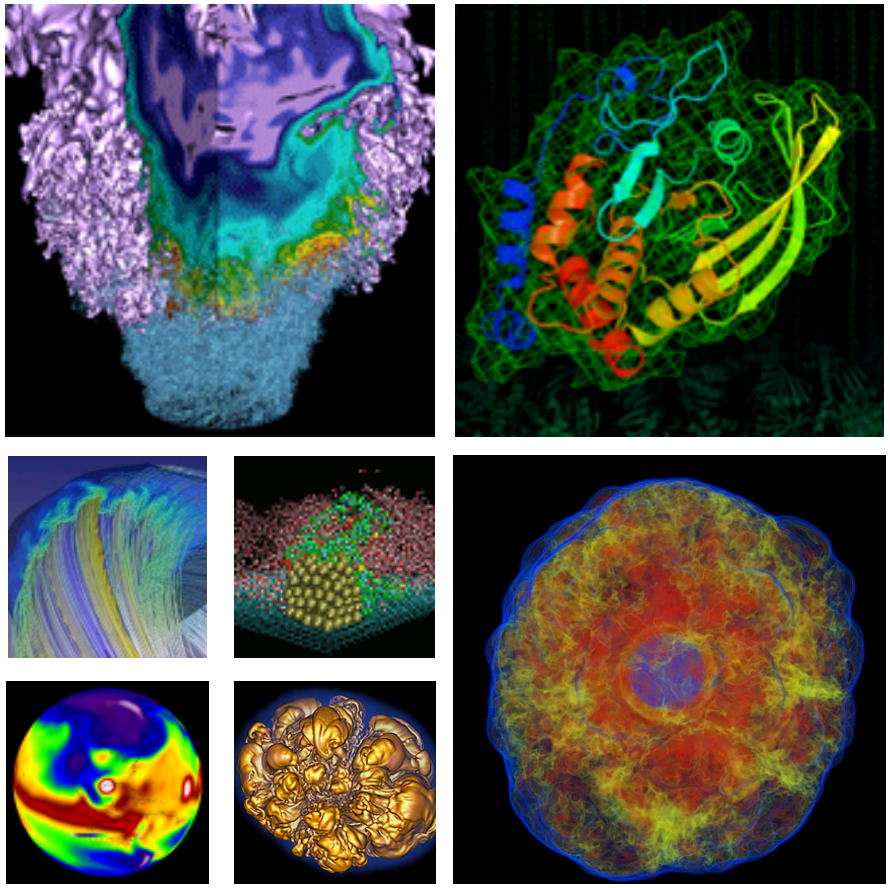
Workflow Tools at NERSC



Shreyas Cholia
scholia@lbl.gov
NERSC Data and Analytics Services

NERSC User Meeting
February, 2015

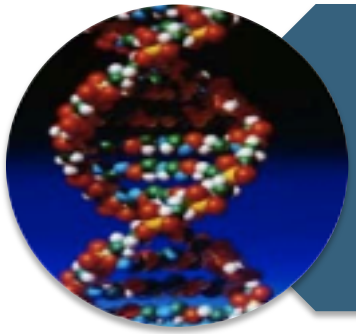
NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

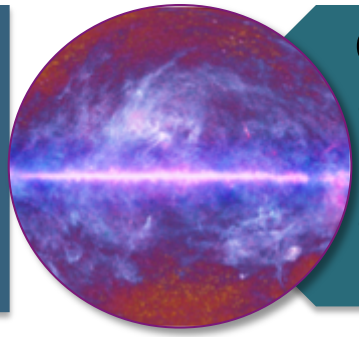
"Big Data" Challenges

Volume, velocity, variety, and veracity



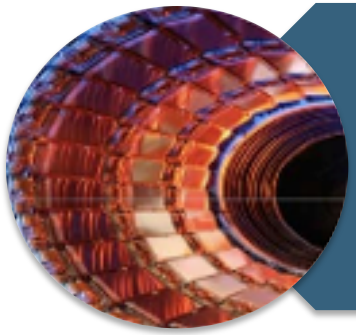
Biology

- *Volume:* Petabytes now; computation-limited
- *Variety*: multi-modal analysis on bioimages



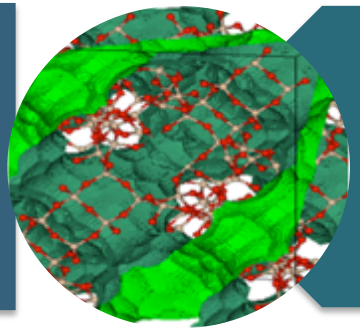
Cosmology & Astronomy:

- *Volume:* 1000x increase every 15 years
- *Variety:* combine data sources for accuracy



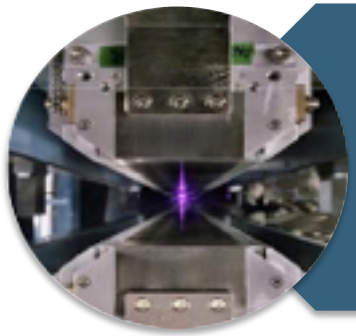
High Energy Physics

- *Volume*: 3-5x in 5 years
- *Velocity*: real-time filtering adapts to intended observation



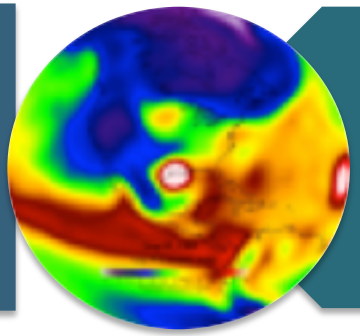
Materials:

- *Variety:* multiple models and experimental data
- *Veracity:* quality and resolution of simulations



Light Sources

- *Velocity:* CCDs outpacing Moore's Law
- *Veracity:* noisy data for 3D reconstruction



Climate

- *Volume:* Hundreds of exabytes by 2020
- *Veracity:* Reanalysis of 100-year-old sparse data

NERSC Data Portfolio

Capabilities	Areas/Technologies
Data Transfer + Access	Globus, Grid Stack, Authentication
	Portals, Gateways, NEWT
Data Processing	Workflows (qdo, Fireworks, Pegasus, …)
Data Management	Models, Formats (HDF5, NetCDF), Databases
	Storage, I/O, Movement (SRM, BestMan)
Data Analytics	Statistics, Machine Learning (R, python)
	Imaging (MATLAB, OMERO, Fiji,…)
Data Visualization	SciVis (VisIt, Paraview), InfoVis (D3, …)
Backend Infrastructure	NX Docker SDN Analytics Stack (BDAS, Hadoop) Databases (MySQL, MongoDB, SciDB, Postgres)

Why users like Data Intensive Systems

- Complex workflows (including High Throughput Computing - HTC)
- Policy flexibility
- Local disk
- Very large memory
- Massive serial jobs (~100K)
- Communicate with databases / host databases
- Stream data from Observational/Experimental Facilities
- Familiar, Easy to customize environment

Workflows and Data Intensive Science

- **Data intensive scientific computing may not always fit the traditional HPC paradigm**
  - Large numbers of tasks, low degree of parallelism
  - Job dependencies and chaining
  - Need to communicate with external datasources, DBs
- **Workflow, work orchestration: Sequences of compute and data-centric operations**

What Does Workflow Software Do?

- **Automate interoperability of applications**
  - Chain together different steps in a job pipeline
  - Automate provenance tracking -> enable ability to reproduce results
  - Assist with data movement
  - Monitor runs and handle errors
  - Data processing of streaming experimental data (including near-realtime processing)
- **Workflows help work with (around?) batch scheduler and queue policies**
- **Some types of Workflow Tasks:**
  - Bag of tasks (DAG)
  - Map-Reduce
  - In-situ
  - Tracking Provenance / Data Movement

Workflows are Personal

- **Many Tools exist in the workflow space**
  - Google: “Scientific Workflow Software”
- **It seems like each domain has its own workflow solution to handle domain-specific quirks**
- **No single tool solves every single problem**

Workflows Working Group

- **Workflows working group actively investigating breadth of technologies**
- **Build a feature matrix of workflow software**
- **Formally support 2-3 tools at NERSC**
- **Create an ecosystem to enable self-supported WF tools**
  - Databases, User defined software modules, AMQP services etc.

Workflow Software

- Fireworks
- qdo
- Tigres
- Galaxy
- Swift
- BigPanda
- Pegasus
- Taverna
- Airavata

- Orange: currently in use at NERSC

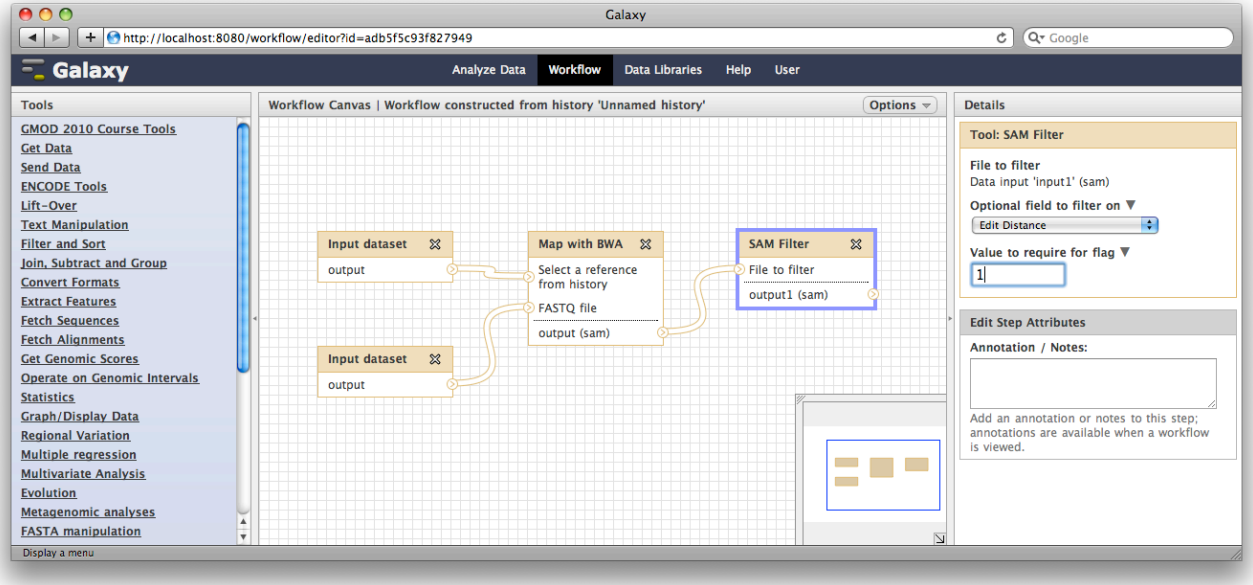
Existing Workflow Ecosystem @ NERSC

- **Science Gateways**
- **Databases**
  - Mongo, Postgres, MySQL, SQLite, SciDB
- **AMQP Services**
- **Workflow tools (self-supported)**
  - Fireworks, Tigres, qdo, Galaxy
- **High throughput batch queues**
- **NEWT REST API**
- **Globus / Data Transfer Nodes**
- **Many task frameworks**
  - MySGE, Taskfarmer
- **Other web based tools for interactive use cases**
  - iPython, R Studio, NX
- **MapReduce frameworks**
  - Spark, Hadoop

Workflow Creation

GUI Editors

example: Galaxy



Workflows as code

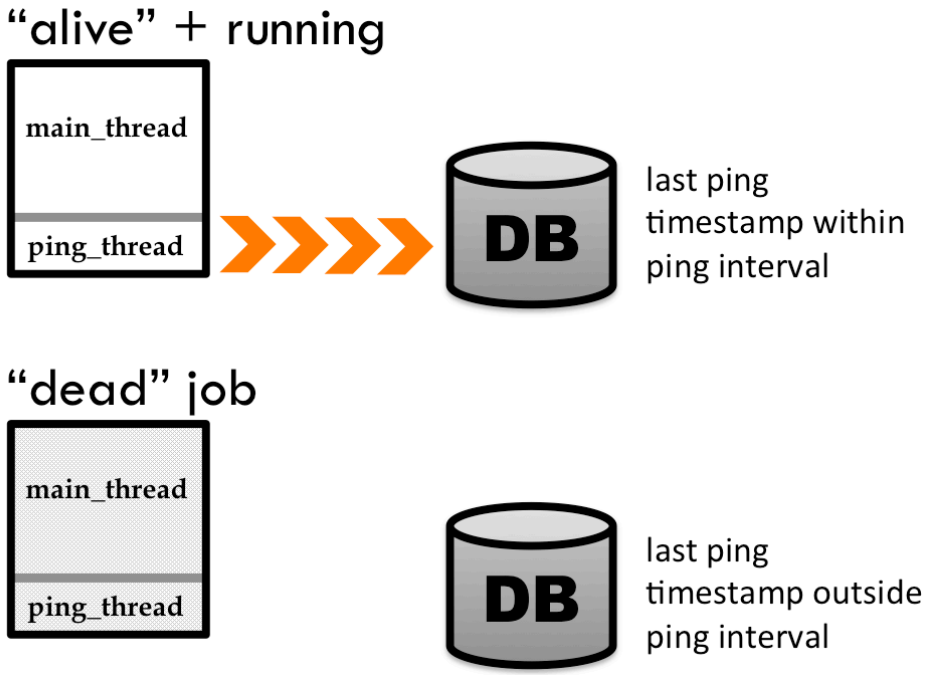
- **Swift is a workflow language (http://swift-lang.org)**

```
type file;

app (file o) simulation ()
{
  simulate stdout=filename(o);
}

foreach i in [0:9] {
  file f <single_file_mapper; file=strcat("output/sim_",i,".out")>;
  f = simulation();
}
```

- **Tigres is a Python/C library or capturing workflow constructs within your code (http://tigres.lbl.gov)**
  - Parallel computing, Split/merge, Sequences

Error Handling and Dynamic Workflows: Fireworks

- **Soft failures, hard failures, human errors**
  - "lpad rerun –s FIZZLED"
  - "lpad detect_unreserved –rerun" OR
  - "lpad detect_lostruns –rerun" OR

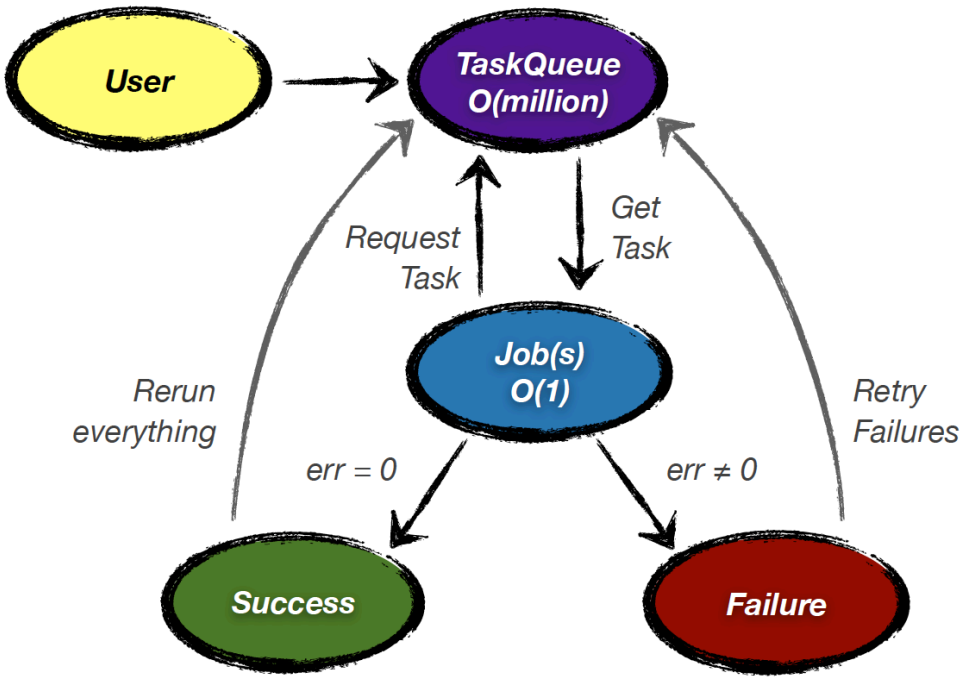


High Throughput "Bag of Tasks"

- **Need to process large numbers of smallish tasks repeatedly**
- **Typical queue policies work against you**
  - too much time lost waiting
  - Batch system not set up of lots of little tasks
- **Instead use a workflow system**
  - to queue up tasks
  - to launch long running workers to consume these tasks
- **Can**
- **Examples: qdo and fireworks**

Use Case: qdo (cosmology)

qdo Model



qdo example

```
#- Command line
qdo load Blat commands.txt      #- loads file with commands
qdo launch Blat 24 --pack       #- 1 batch job; 24 mpi workers

#- Python
import qdo
q = qdo.create("Blat")
for i in range(1000):
   q.add("analyze blat{}.dat".format(i))

q.launch(24, pack=True)

#- Python load 1M tasks
commands = list()
for x in range(1000):
    for y in range(1000):
        commands.append("analyze -x {} -y {}".format(x, y))

q.add_multiple(commands)   #- takes ~2 minutes
q.launch(1024, pack=True)
```

Workflows: Data Management

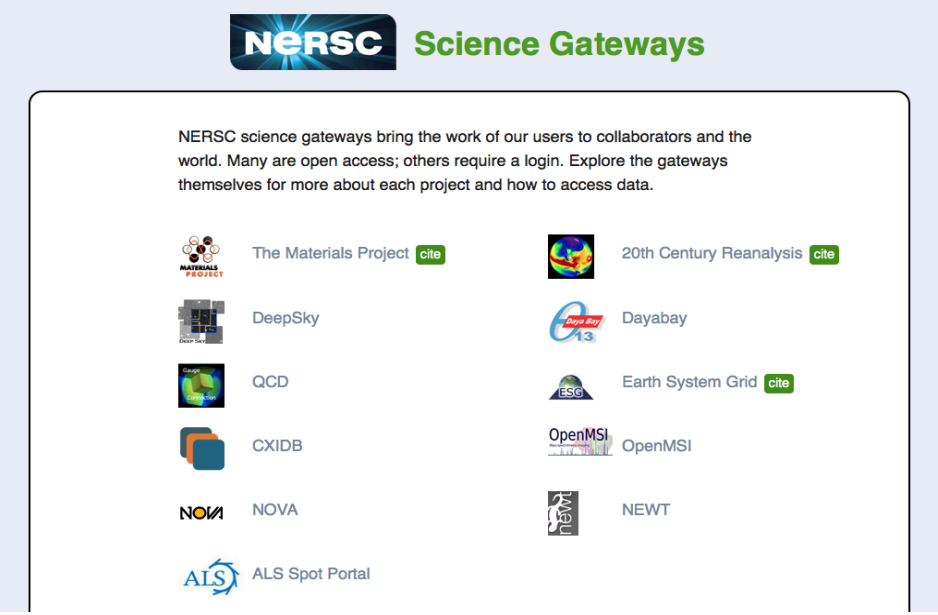
- Workflows often have a data staging component to deal with pulling data from remote locations
- The SPADE tool can help you manage file transfers with Globus/GridFTP and trigger jobs after the fact
- There is a transfer queue on NERSC systems called "xfer" which lets you queue up transfers after your job (typically archive to HPSS using HSI)

Batch Queues

- NERSC has support for serial and high throughput queues well suited to jobs that need many task computing
- Reservations available for special needs
- Consider using job packing options in various workflow tools to optimize for HPC queue infrastructure

NERSC Science Gateways

- **Web portals that allow you to interface with your data and computation at NERSC**
- **Provide an intuitive web interface to drive your workflows**



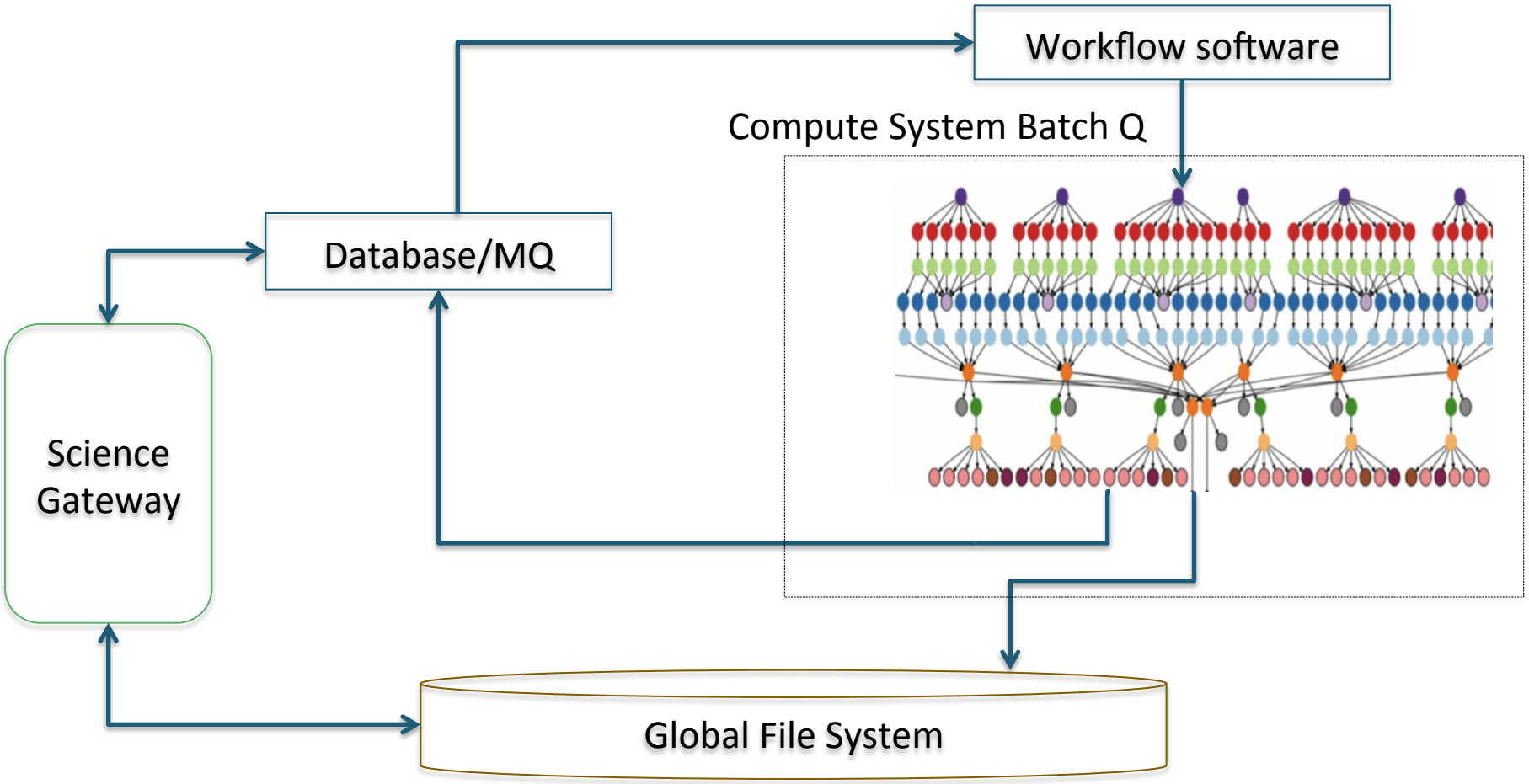
Science Gateway Services

- **Simple data publishing capabilities**
  - Data in /project/projectdirs/yourproj/www visible on web
- **Rich web interfaces and complex portals**
  - Full stack web applications in Python/PHP/Javascript etc.
- **Backend databases and message queues**
  - Connect web apps to backend services
- **NEWT API to access NERSC resources**
  - Interface with NERSC using an HTTP API (files, jobs, command, auth, NIM etc.)

Many task/MR frameworks

- **Repeatedly perform tasks on a large dataset**
- **Map => perform an operation across a large set i.e. map a task across the dataset**
- **Reduce => collect and reduce the results from map operation**
- **Split the data across nodes and run task on each node**
- **Typically does not require much cross node communication**
- **Frameworks at NERSC**
  - Spark
  - Hadoop
  - MySGE
  - Taskfarmer

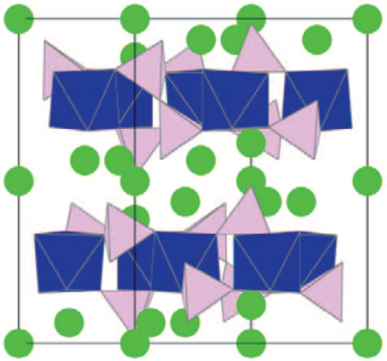
Tying it all together



Use Case: Materials Project

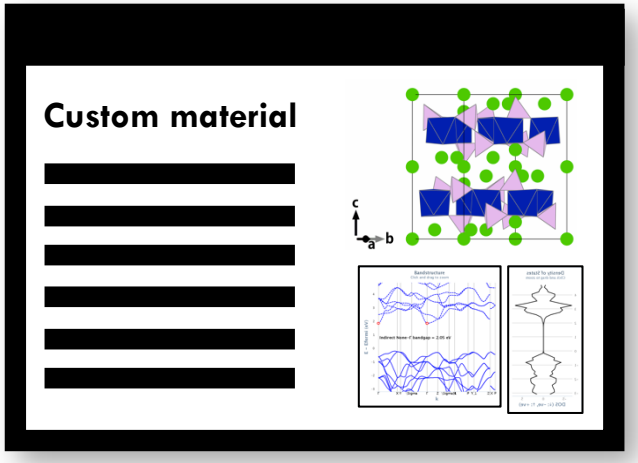
- **Tasks submitted to Fireworks MongoDB via REST API/Gateway**
- **MongoDB keeps a list of tasks to be run**
- **Fireworks submits workers to NERSC queues.**
- **Workers pull jobs from MongoDB**
- **Fireworks manages job orchestration**
  - Retry on failure
  - File transfer
  - Job Dependencies
  - Flow control for subsequent jobs
  - Duplicate management

Materials Project Workflow

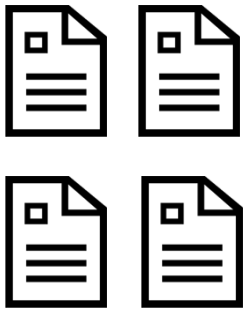


A cool material !!

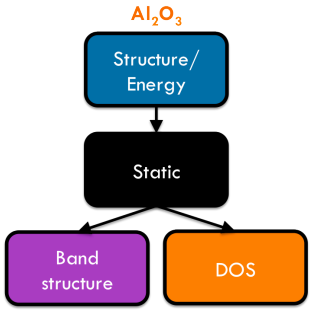
Submit!



Lots of information about cool material !!

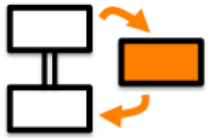


Input generation (parameter choice)

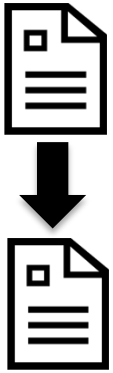


Workflow mapping

Supercomputer submission / monitoring



Error handling



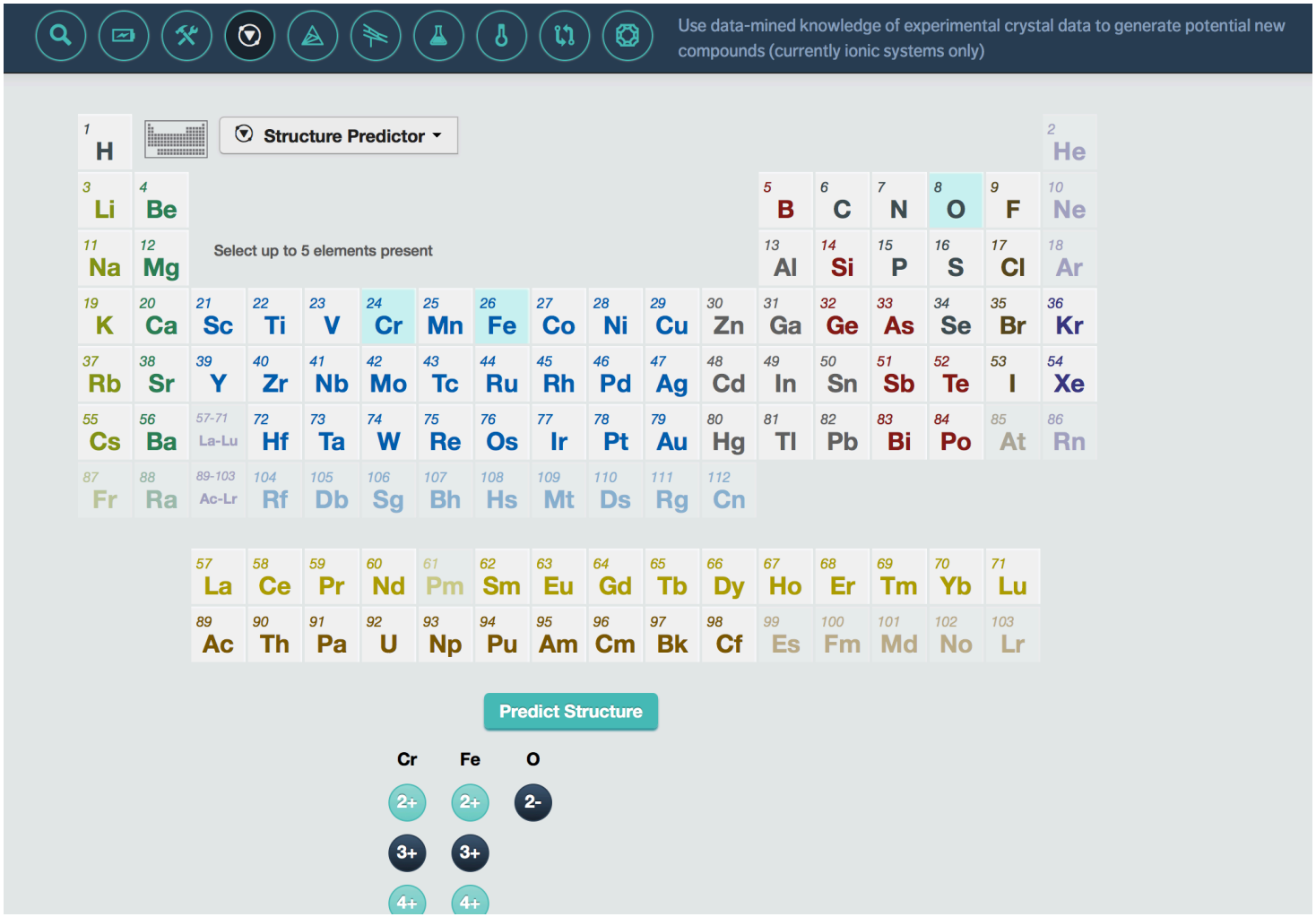
File Transfer

File Parsing / DB insertion

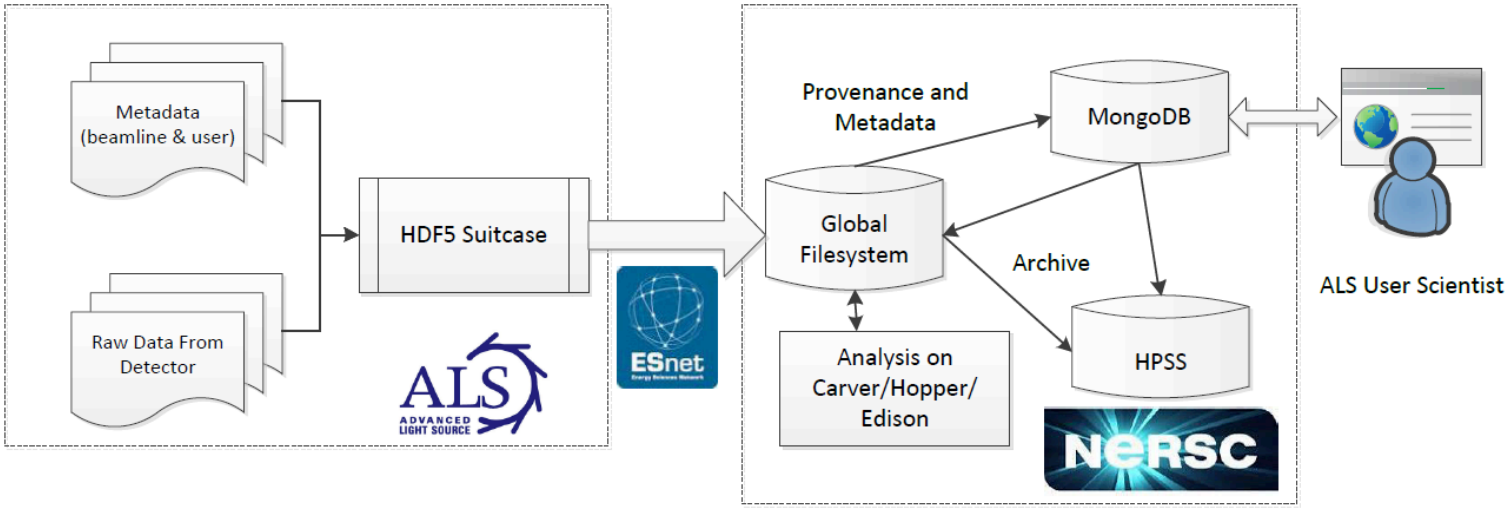
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Materials Project Gateway

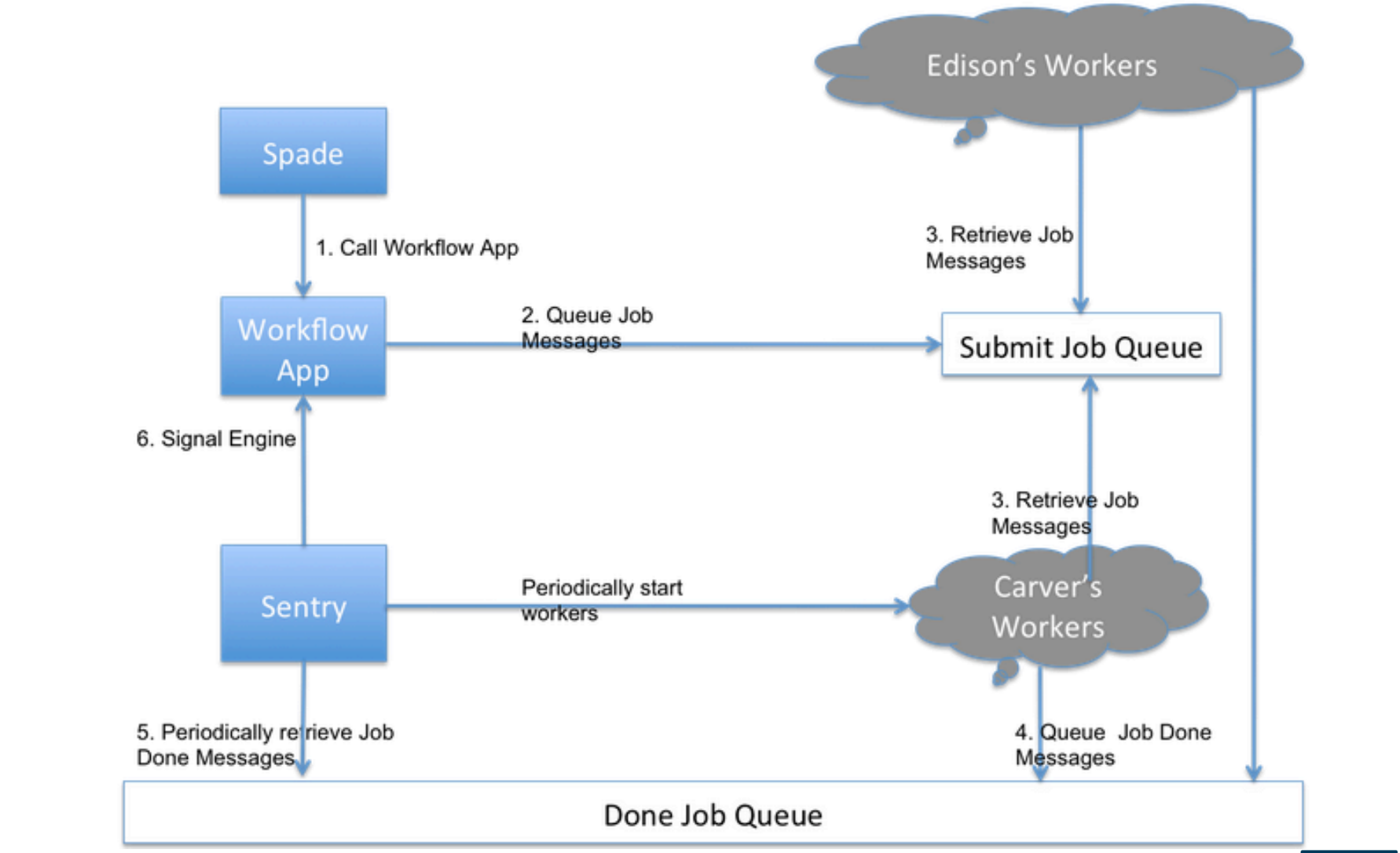


Use Case: SPOT Suite

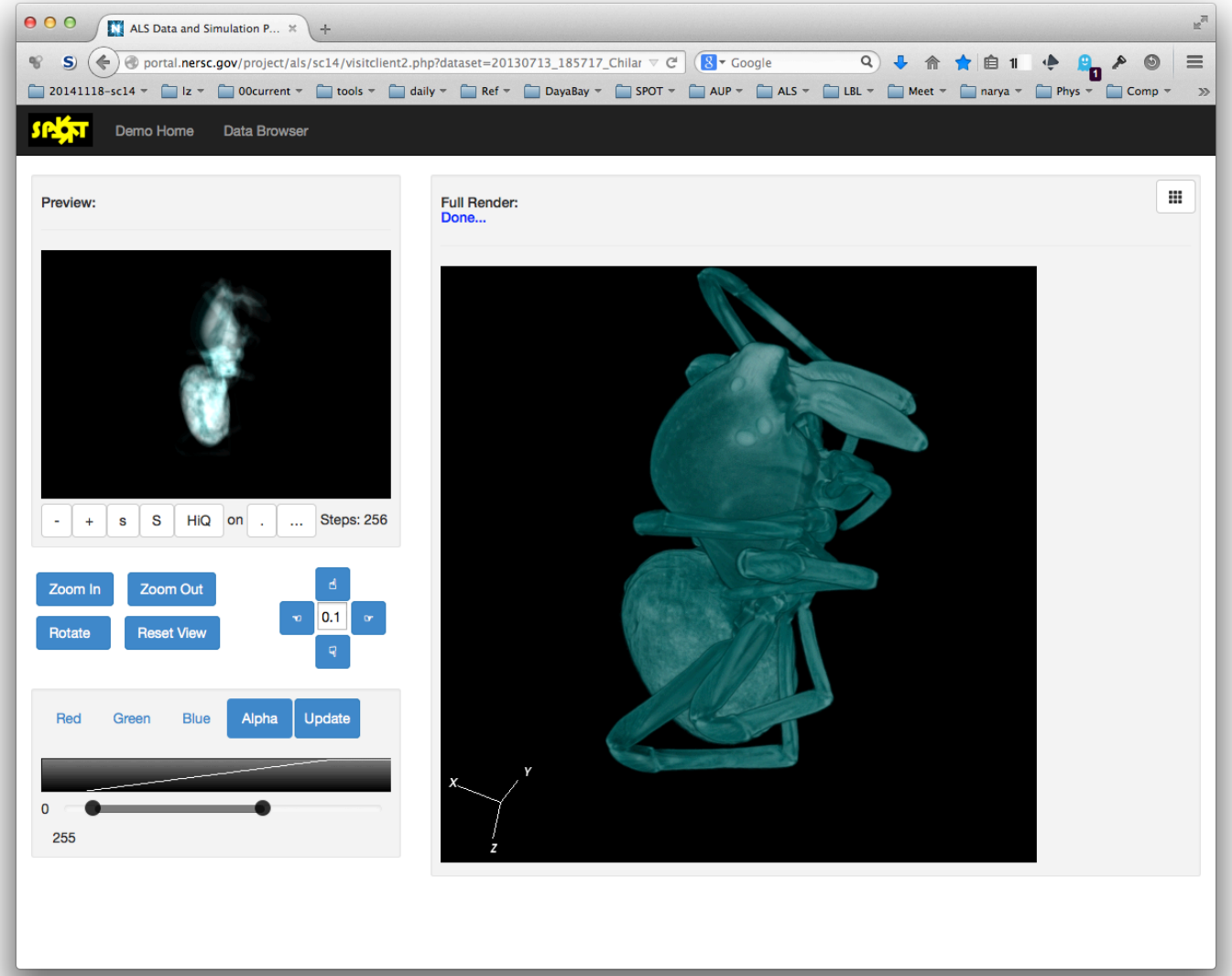


- Collect Data from Beamline
- SPADE/Globus to move data to NERSC
- Trigger Analysis at NERSC via AMQP
- View Jobs and Results on Science Gateway
- Track Provenance and Metadata via MongoDB

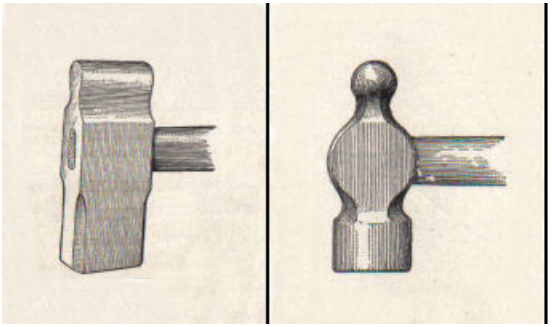
Use Case: SPOT Suite Workflow



SPOT Suite Gateway



Finding the Right Hammer



- Workflow tools have lots of features but there is no one size-fits-all
- NERSC is building expertise in classes of workflow tools and will help guide you towards the right tool for your job
- Consider stitching together a couple of different tools to make it all work

Engagement

- **Enabling science in a scalable manner**
  - Build re-usable workflow components that can be used across domains.
  - Support a 2 to 4 classes of workflow tools
  - Create an ecosystem of services to enable new tools
  - Engage with domain specific science to address specific needs. Each project will have its own requirements. Bring those requirements to the table and we can evolve our ecosystem to meet your needs.

Thank you.