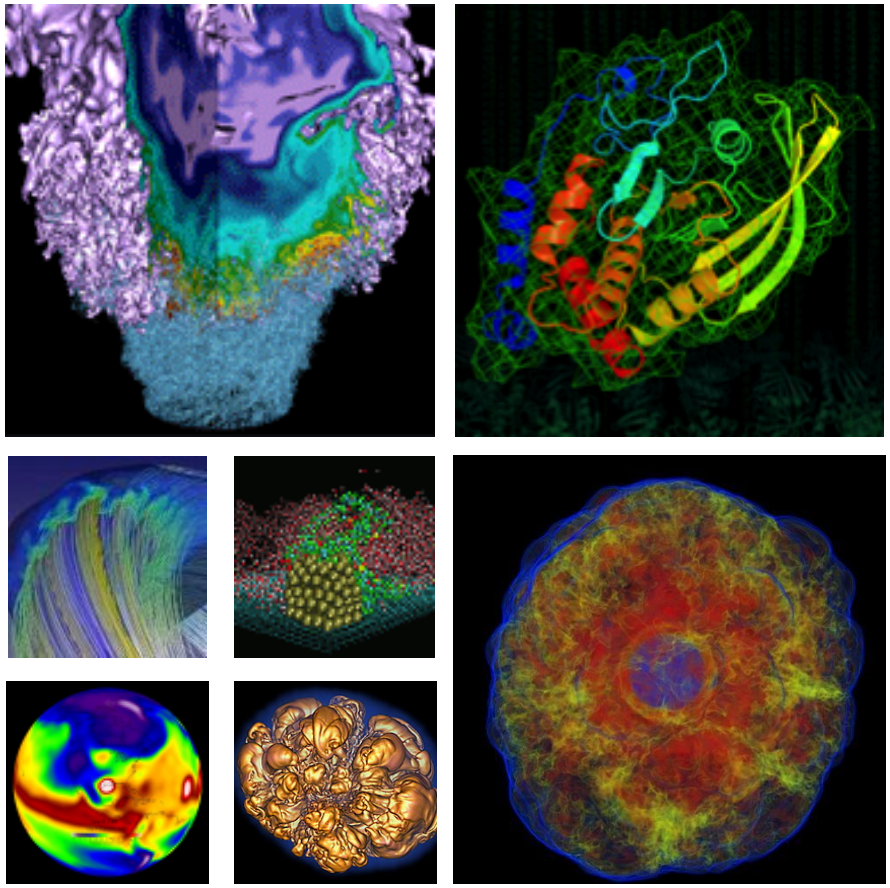
Workflow Tools at NERSC



Debbie Bard
djbard@lbl.gov
NERSC Data and Analytics Services

NERSC User Meeting
August 13th, 2015

NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

What Does Workflow Software Do?

- **Automate connection of applications**
  - Chain together different steps in a job pipeline.
  - Automate provenance tracking -> enable ability to reproduce results.
  - Assist with data movement.
  - Monitor running processes and handle errors.
  - Data processing of streaming experimental data (including near-realtime processing).
- **Workflows help work with (around?) batch scheduler and queue policies.**

Workflows are Personal

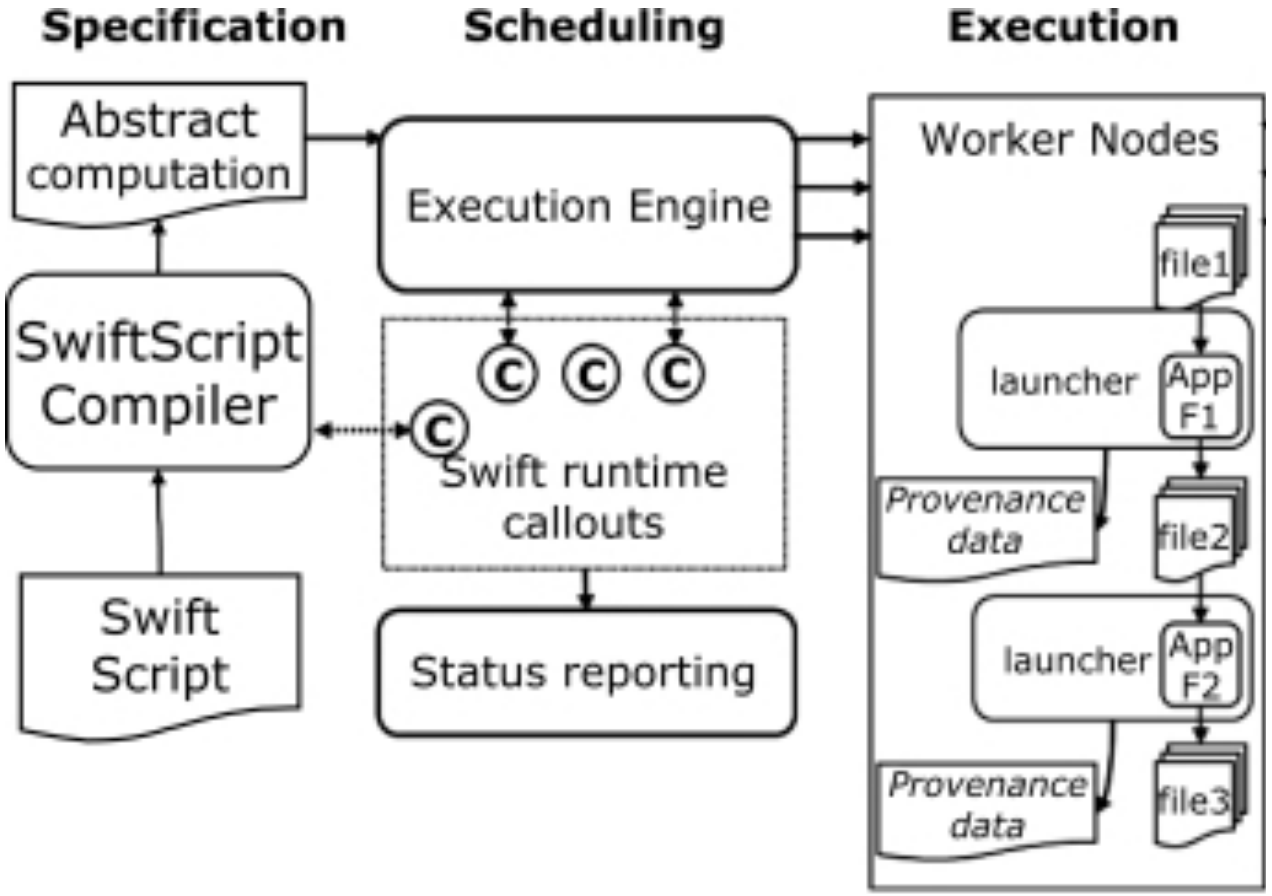
- **Many tools exist in the workflow space**
  - Google: “Scientific Workflow Software”
- **It seems like each domain has its own workflow solution to handle domain-specific quirks**
- **No single tool solves every single problem**

- **Fireworks**
- **qdo**
- **Tigres**
- **Galaxy**
- **Swift**

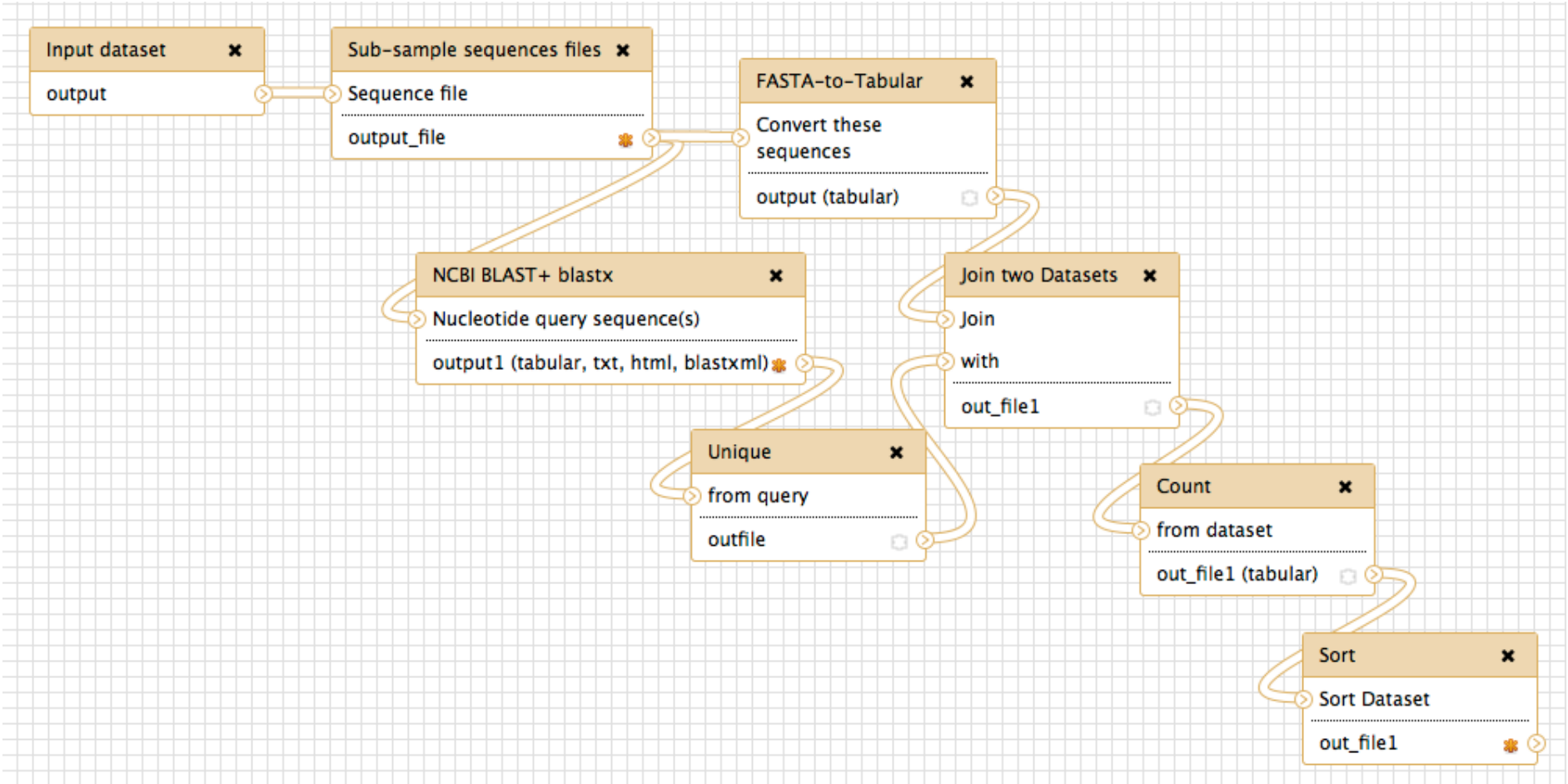
- **BigPanda**
- **Pegasus**
- **Taverna**
- **Airavata**

…….

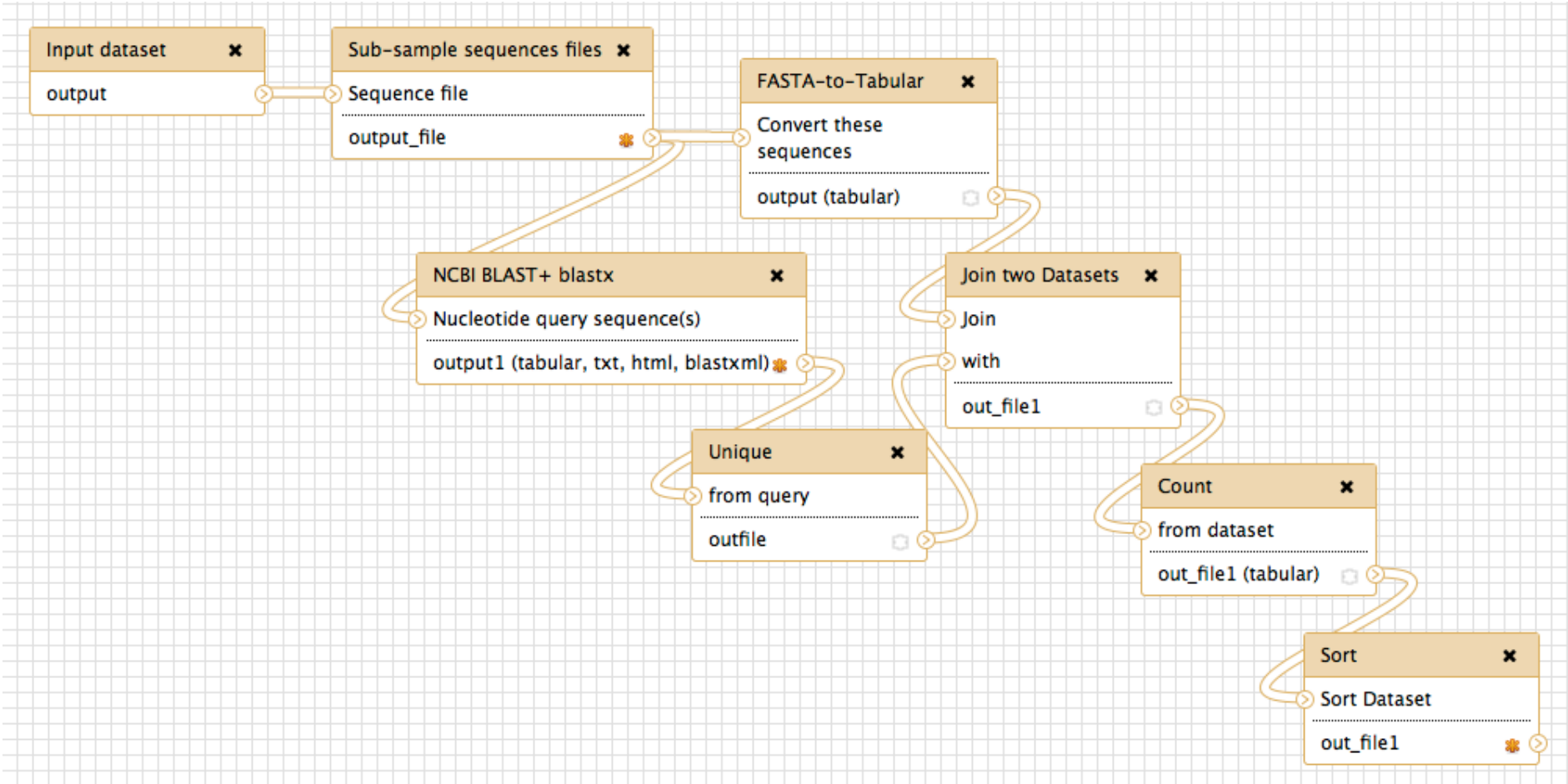
Visualising a workflow: swift



Visualising a workflow: galaxy



Workflows as GUI: galaxy



Workflows as code: Swift/Tigres

- **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

Workflows Working Group

- **Earlier this year Workflows working group investigated breadth of technologies**
- **We 'support' 2 tools at NERSC**
  - **FireWorks**
  - **Swift**
  - this doesn't mean other tools won't be used/supported at NERSC, only that DAS has specific expertise in these.
- **Create an ecosystem to enable self-supported WF tools**
  - Databases, User defined software modules, AMQP services etc.

Existing Workflow Ecosystem @ NERSC

- **Science Gateways**
- **Databases**
  - Mongo, Postgres, MySQL, SQLite, SciDB
- **Workflow tools (self-supported)**
  - Fireworks, swift, 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 tools exist in and interact with a rich environment of NERSC capabilities and services.

Workflows and Data Intensive Science

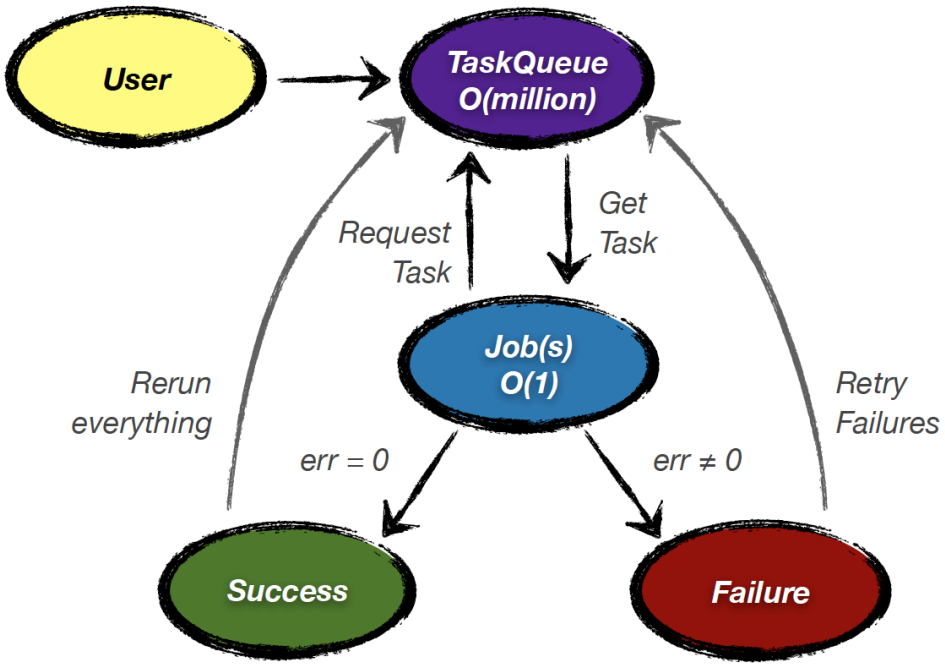
- **Data intensive scientific computing may not always fit the traditional HPC paradigm**
  - Large numbers of tasks, low degree of parallelism.
  - Job dependencies and chaining.
  - Need to communicate with external datasources, DBs.
- **Workflow and work orchestration in this context can be thought of as sequences of compute and data-centric operations.**

High Throughput "Bag of Tasks"

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

Use Case: qdo (cosmology)

qdo Model

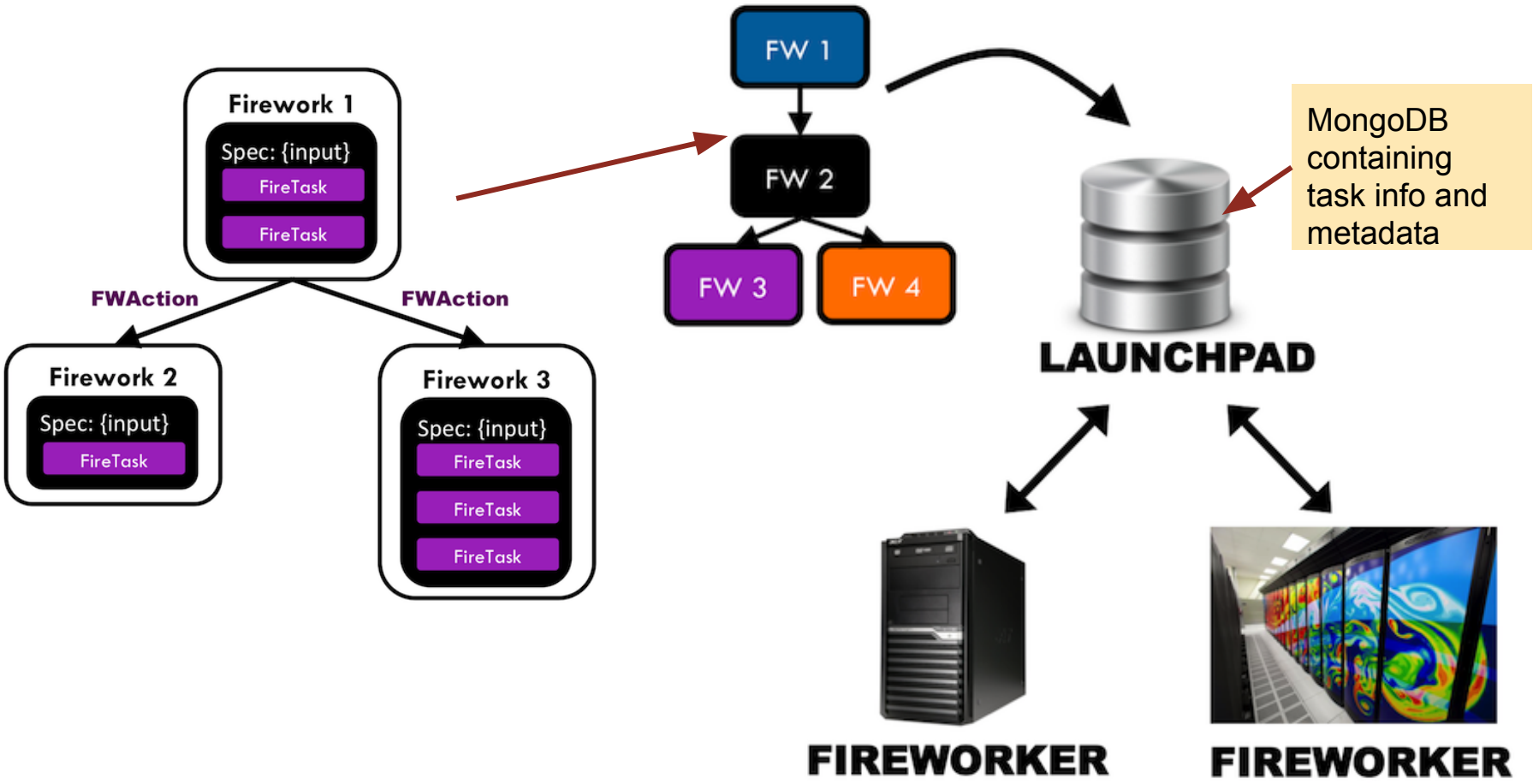


- qdo is specifically designed to package up multiple small tasks into one batch job.

qdo examples

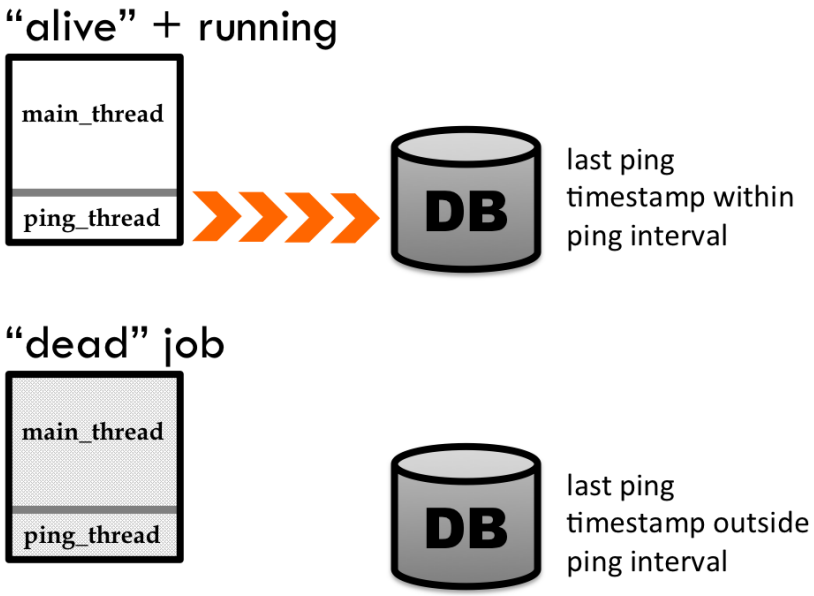
```
#- Command line
qdo load Blat commands.txt      #- loads file with commands
qdo launch Blat 24 --pack       #- 1 batch job; 24 mpi workers

#- 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)
```

Use case: Fireworks (material science)



Fireworks: Error Handling and Dynamic Workflows

- **Can specify action based on soft failures, hard failures, human errors**
  - "lpad rerun –s FIZZLED"
  - "lpad detect_unreserved –rerun" OR
  - "lpad detect_lostruns –rerun" OR



Many task/MR frameworks

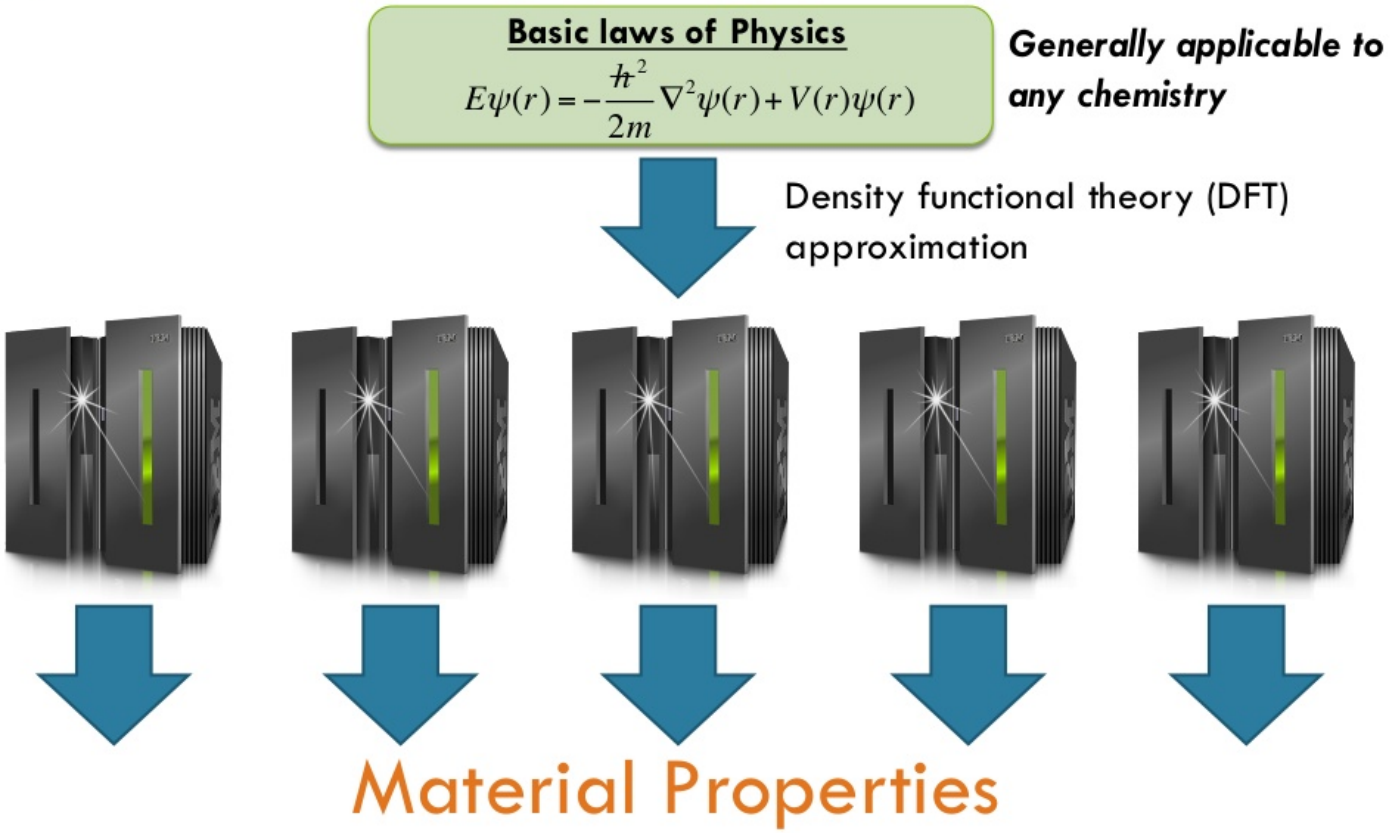
- **Repeatedly perform tasks on a large dataset**
- **Map => perform an operation across a large set i.e. map a task across the dataset**
- **Reduce => collect and reduce the results from map operation**
- **Split the data across nodes and run task on each node**
- **Typically does not require much cross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

Batch Queues

- **NERSC has support for serial and high throughput queues well suited to jobs that need many task computing**
  - Cori Serial queue designed specifically for these use cases.
- **Reservations available for special needs.**
- **Consider using job packing options in various workflow tools to optimize for HPC queue infrastructure**
  - also for packing single-core jobs into a multi-core node.

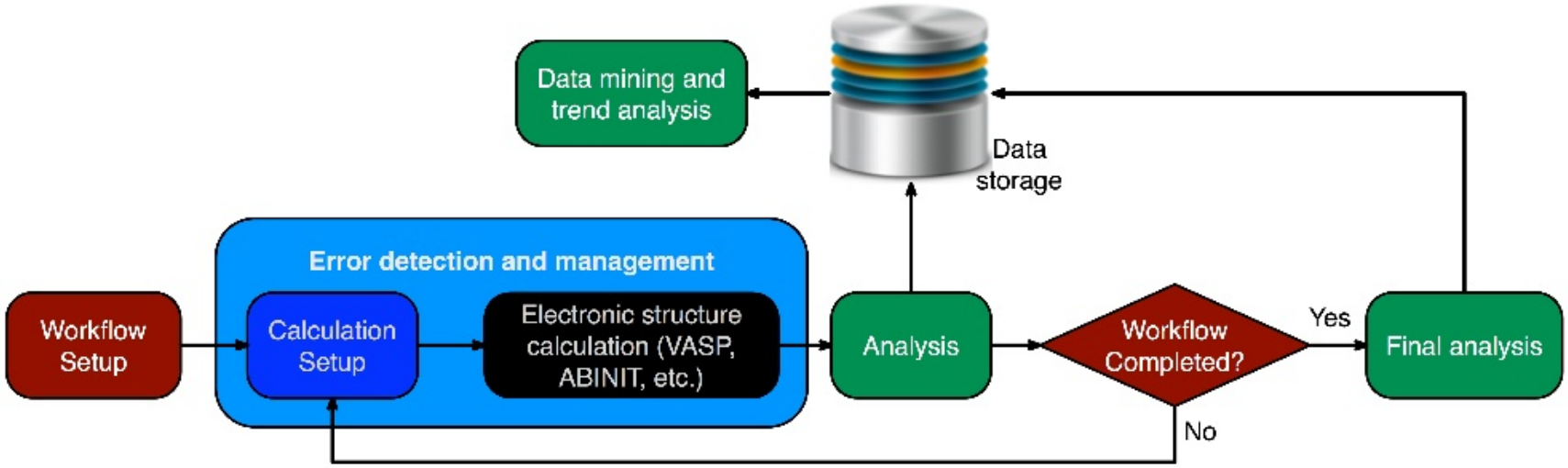
Use Case: Materials Project

- **Simulate properties of all possible materials.**



Use Case: Materials Project

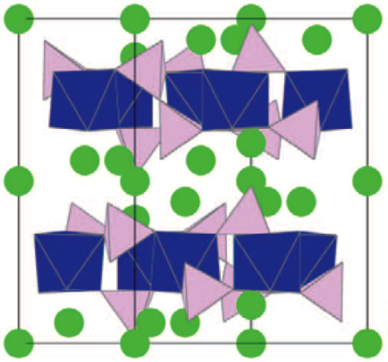
- **Simulate properties of all possible materials.**



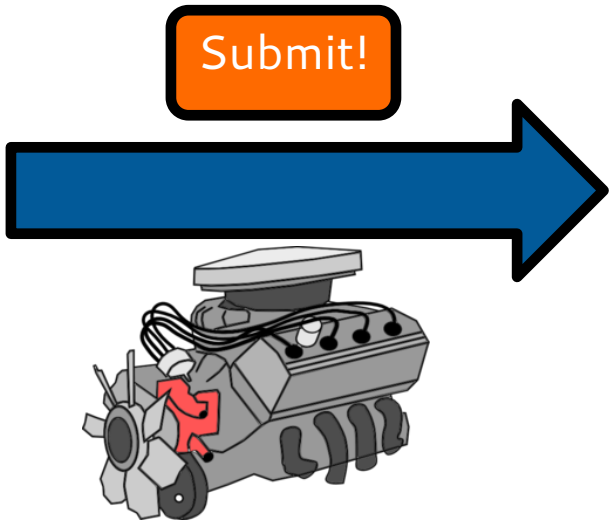
Use Case: Materials Project

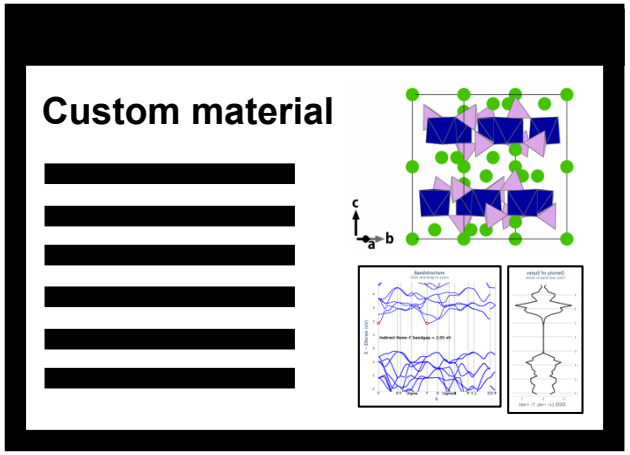
- **Tasks submitted to Fireworks MongoDB via API/python script etc.**
- **MongoDB keeps a list of tasks to be run.**
- **Fireworks submits workers to NERSC queues.**
- **Workers pull jobs from MongoDB.**
- **Fireworks manages job orchestration**
  - Retry on failure
  - File transfer
  - Job Dependencies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

Materials Project Workflow

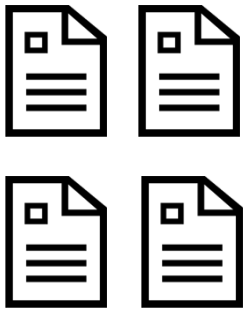


input: A cool material !!

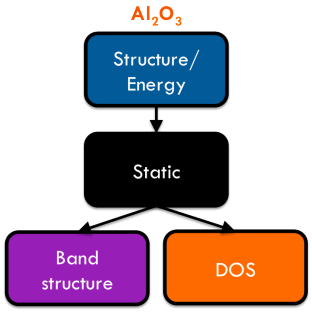




output: Lots of information about cool material !!

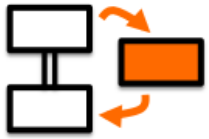


Input generation (parameter choice)

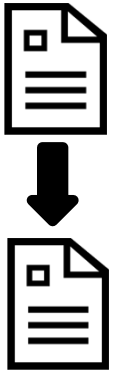


Workflow mapping

Supercomputer submission / monitoring



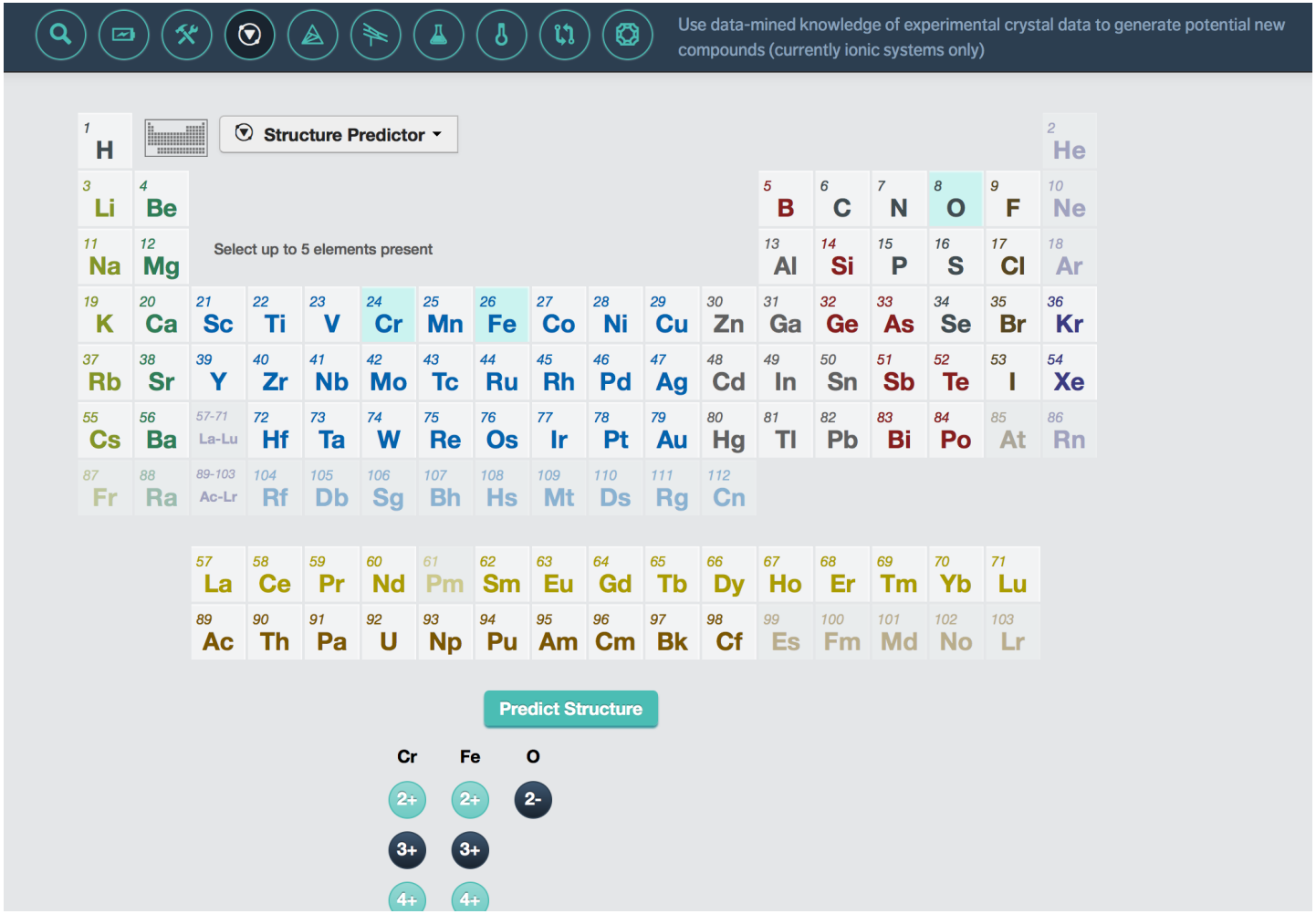
Error handling



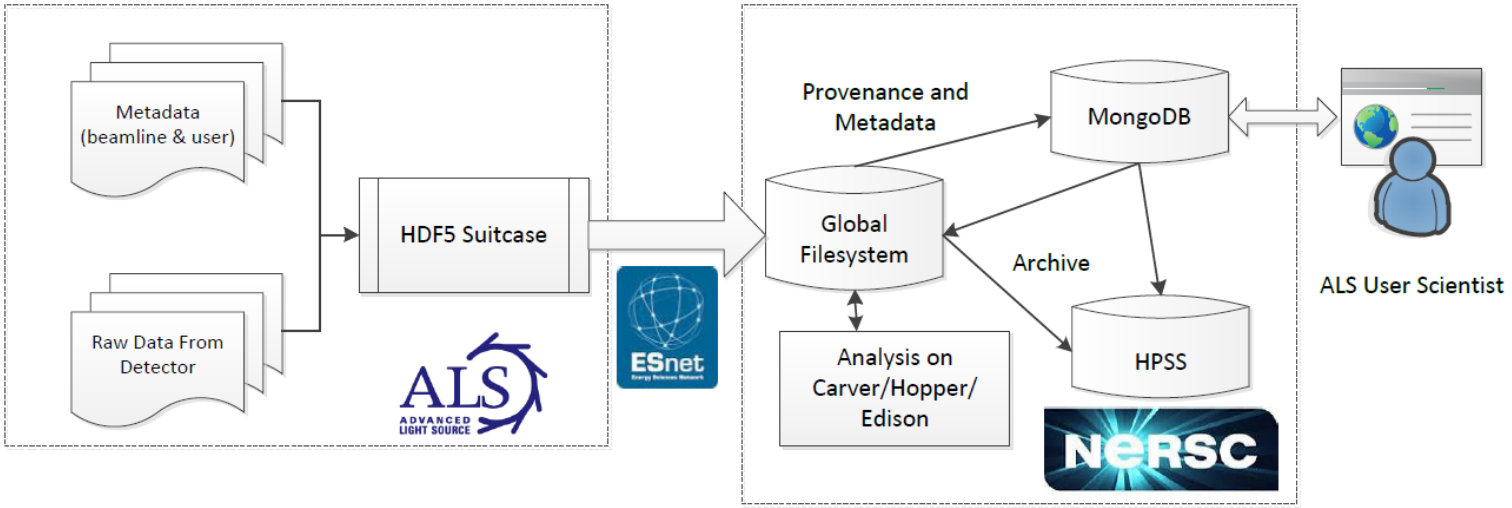
File Transfer

File Parsing / DB insertion

Materials Project Gateway

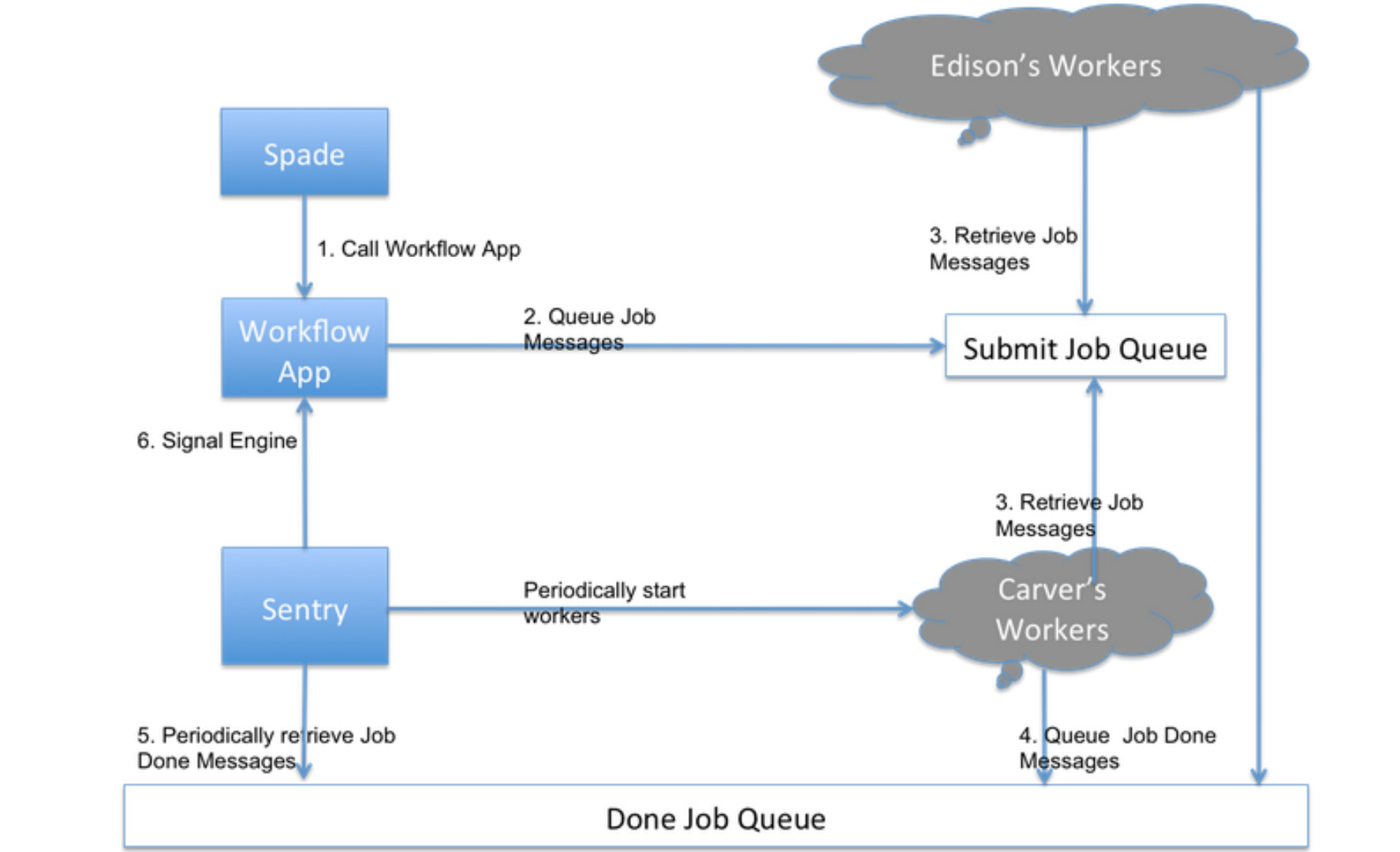


Use Case: SPOT Suite

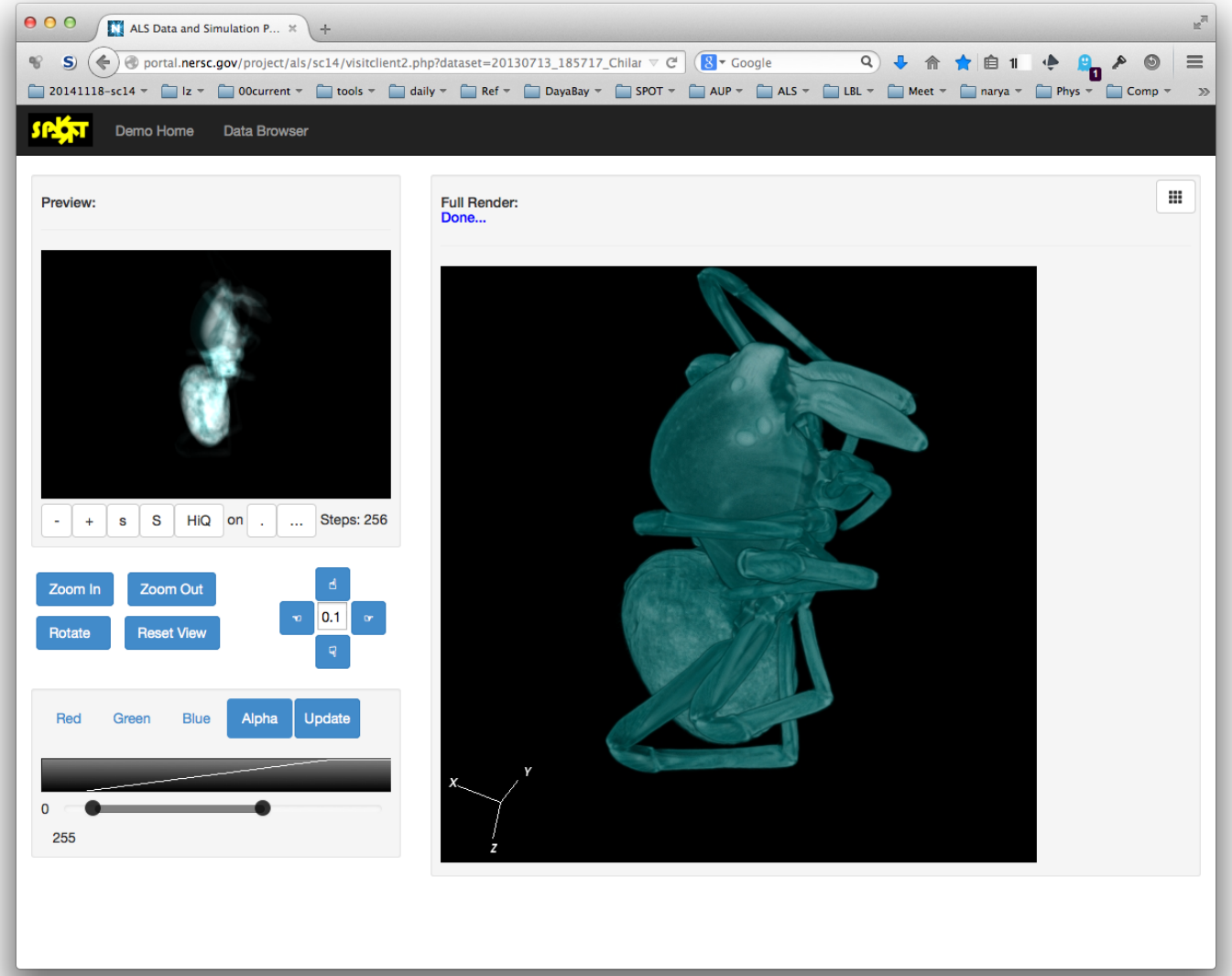


- Collect Data from Beamline
- SPADE/Globus to move data to NERSC
- Trigger Analysis at NERSC via AMQP
- View Jobs and Results on Science Gateway
- Track Provenance and Metadata via MongoDB

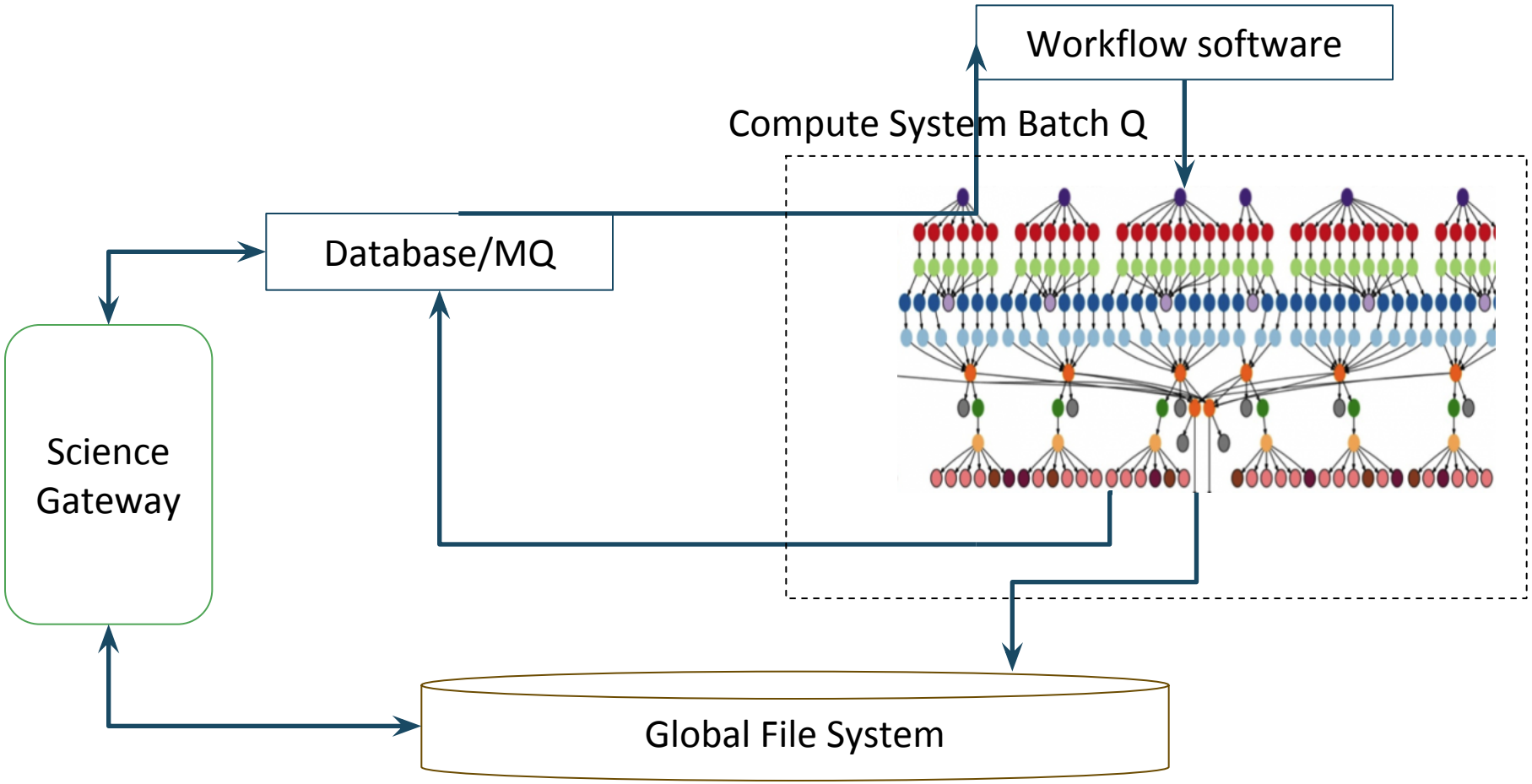
Use Case: SPOT Suite Workflow



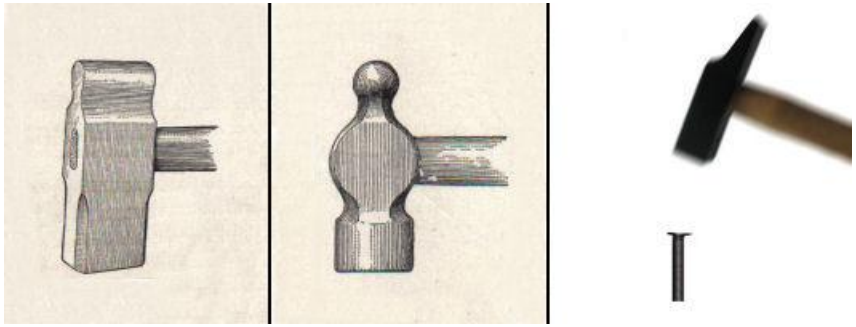
SPOT Suite Gateway



Tying it all together



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

Thank you.

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.