Argonne NATIONAL LABORATORY

Generic Portals for Science Infrastructure (GPSI)

Thomas D. Uram, Michael E. Papka, Mark Hereld, Michael Wilde

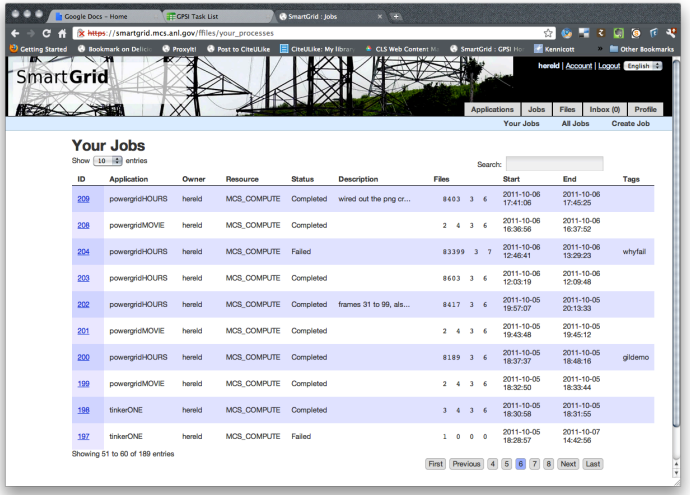
Argonne National Laboratory, University of Chicago Computation Institute

U.S. DEPARTMENT OF ENERGY

GPSI
Generic Portals for Science Infrastructure

Approach

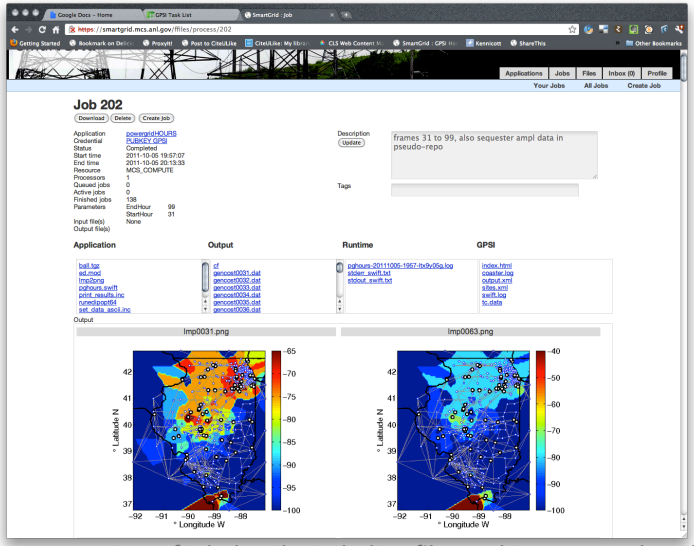
- Simplify execution and data management on compute resources
- Support reproducibility of results through execution and data provenance
- Enable collaboration among researchers working on a science campaign
- Capture details of scientists' idiosyncratic workflows



View of job history

Research Questions

- Effective scheduling of jobs across resources
- Efficient data staging to and from compute resources
- Automated data analysis and feature extraction
- Data filtering for remote analysis



View of job details, including files and images produced

GPSI: Technologies

- Django core (hence Python)
  - Django is widely used in industry, and is the most popular Python web framework
  - Python is widely used in the scientific community
  - Uses a variety of database backends (MySQL, Sqlite)
- Collection of Django apps
  - Apps exist for (nearly) anything one needs
- GPSI is one more Django app
  - Can remain small, focus on only the gateway-related functionality
- Uses Swift for describing workflow and running jobs
- GPSI exposes opportunities for customization by science domains
  - User-driven gateways

GPSI: Overall Flow

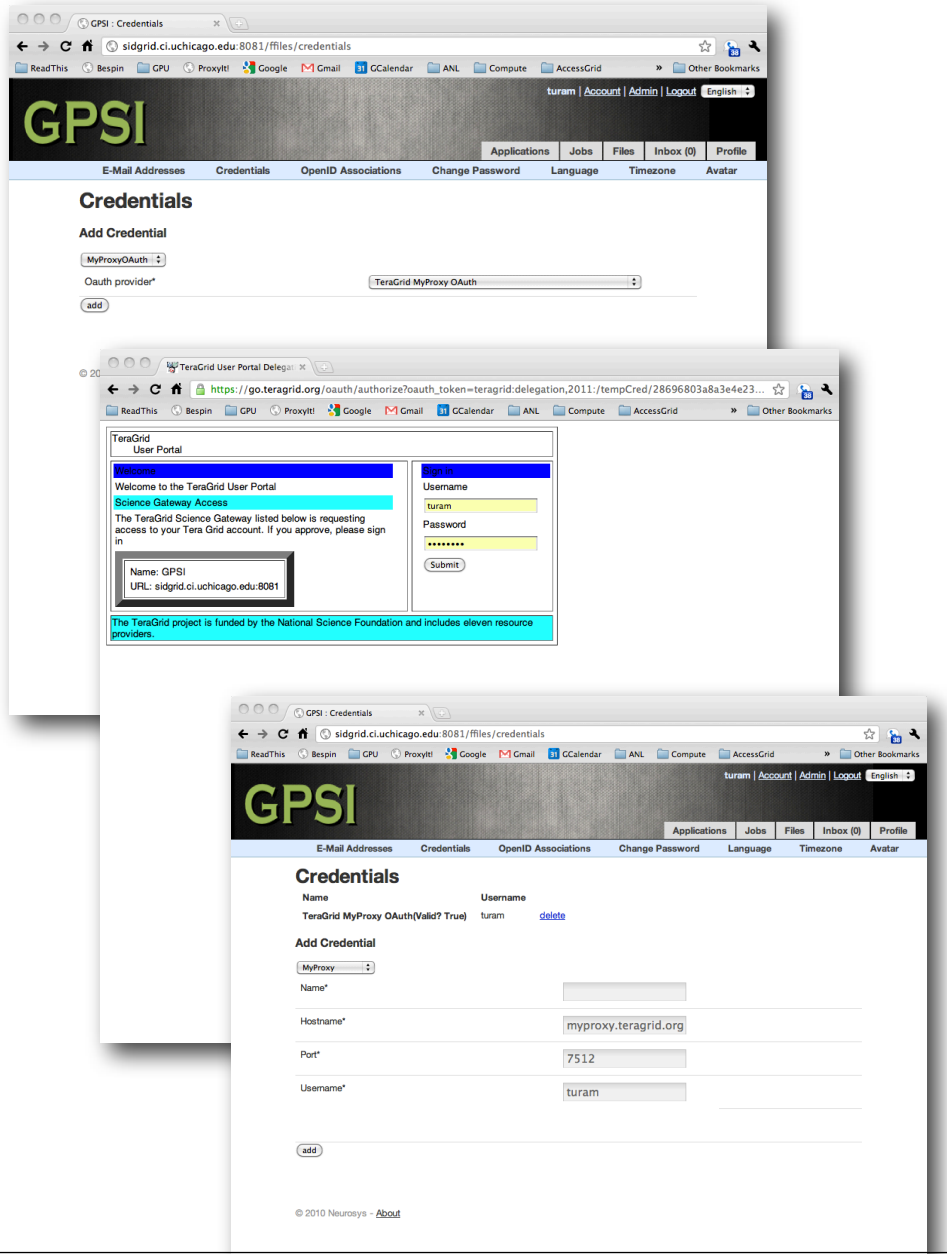
- Introduce data
- Introduce applications
- Schedule jobs
- View/Analyze results
- Reproduce results
- Download
- Share

Compute Resource Setup

- Mechanisms supported by Swift
  - File transfer: gridftp, ssh, http
  - Execution: pbs, gram, sge, ssh, ec2, FutureGrid, ...
- Have run on a variety of resources
  - Ranger, Steele, Blacklight @ ~~TeraGrid~~ XSEDE
  - Compute machines at Argonne, including Fusion cluster
  - PADS at UChicago
  - Third-party resources
- Administratively controlled (for

Credential Management

- Credentials managed by user and associated with particular resources
- Credential types include
  - ssh
  - MyProxy
    - MyProxy server host, port, username and password
    - Credential renewed as needed
  - MyProxy OAuth
    - Avoids collection of MyProxy username and password by gateway
    - Have currently integrated pre-production TeraGrid MyProxy OAuth support

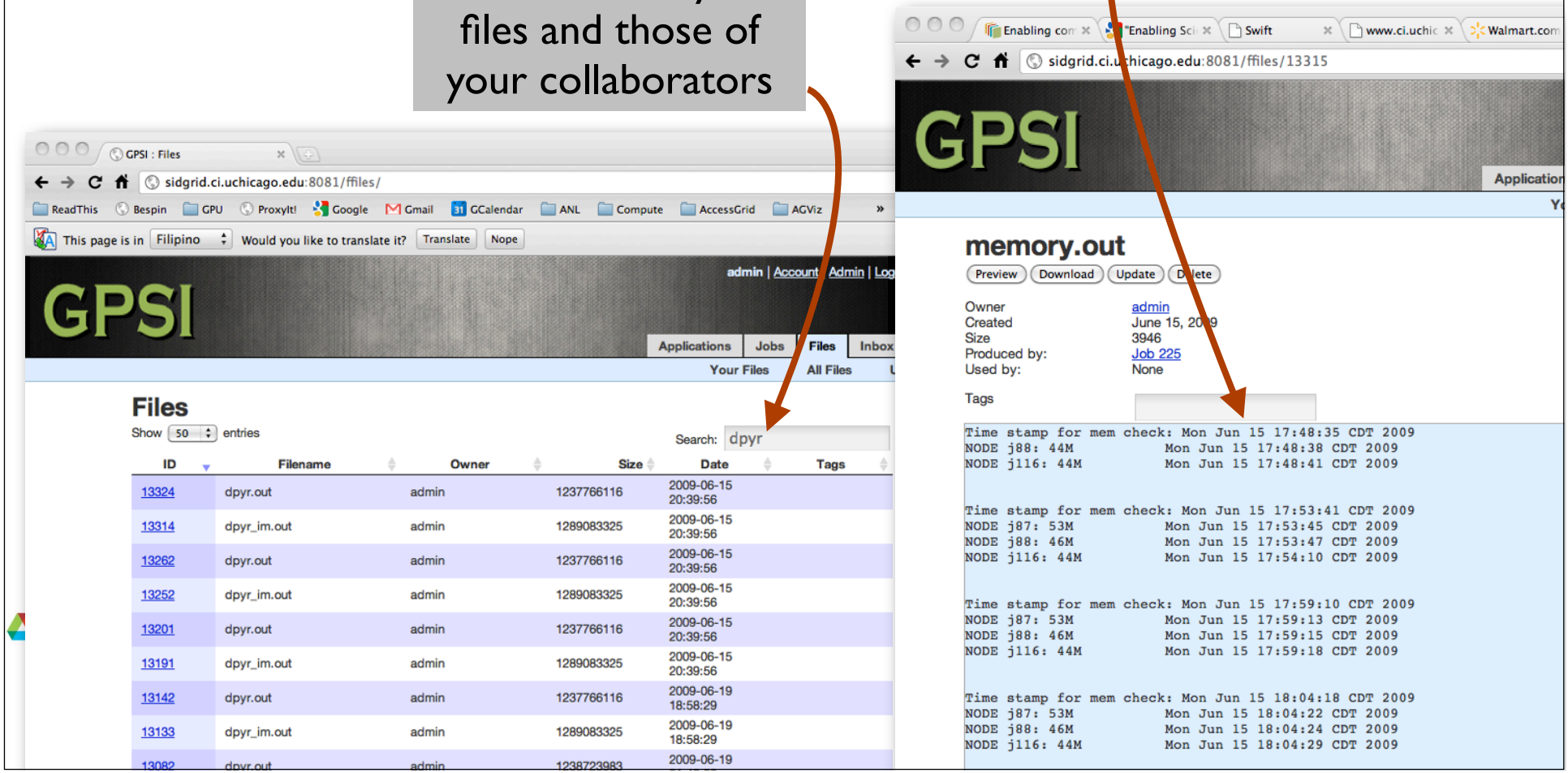


Introduce Data

- Upload relevant files, or ingest on server-side

Search across your files and those of your collaborators

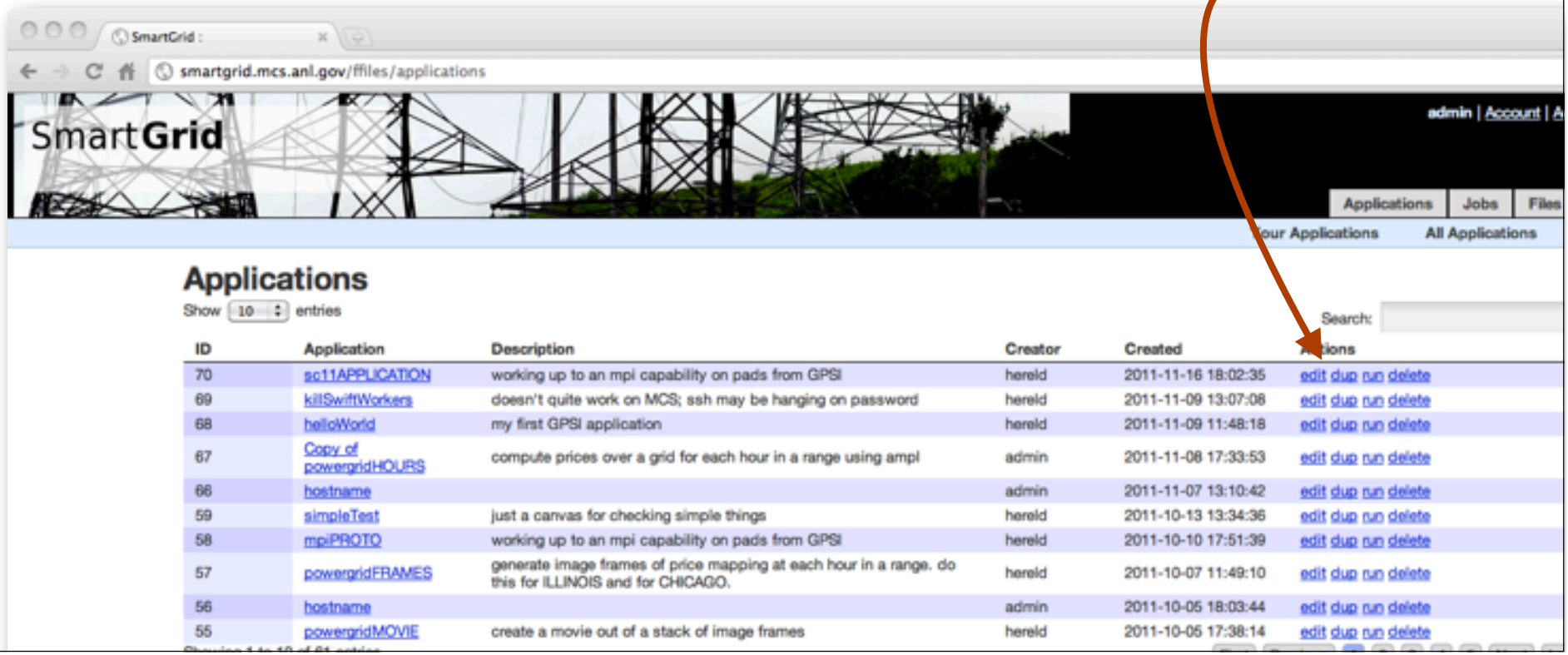
Preview files inline



Introduce Applications

- Upload relevant files
- Application definition relies on required applications residing on compute resource
- Edit Swift script within GPSI

Edit, duplicate, run, delete jobs



Swift

- Language for parallel scripting
- Execution engine with interfaces to various schedulers

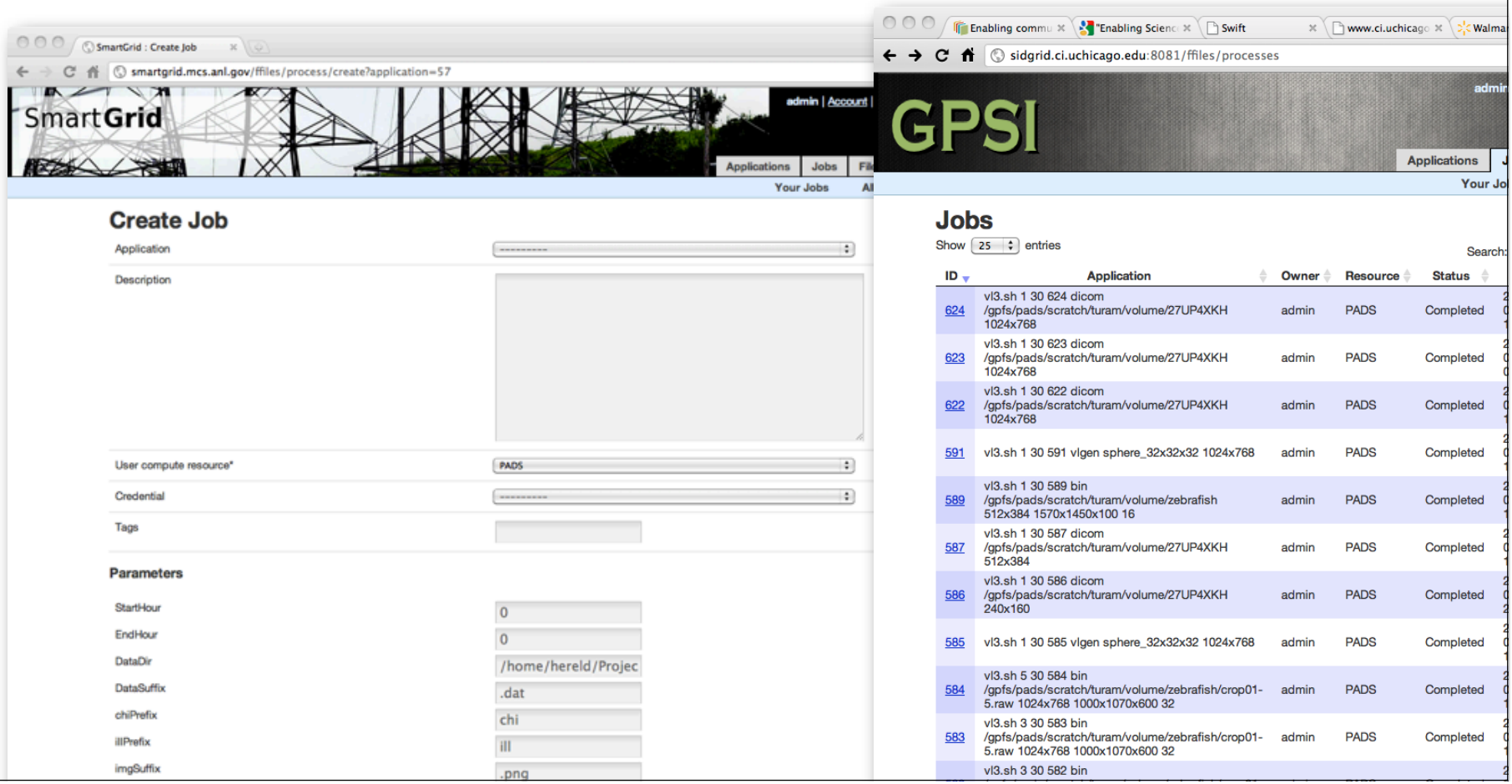
```
# simple script to execute hostname N times
type file;

# Swift app definition; each call will schedule a job
(file t)do_hostname() {
    app {
        hostname stdout=@filename(t);
    }
}
foreach i in [0:50] {
  file outfile <concurrent_mapper;prefix="stdout",suffix=".txt">;
  outfile = do_hostname();
}
```

For info on Swift, see http://www.ci.uchicago.edu/swift

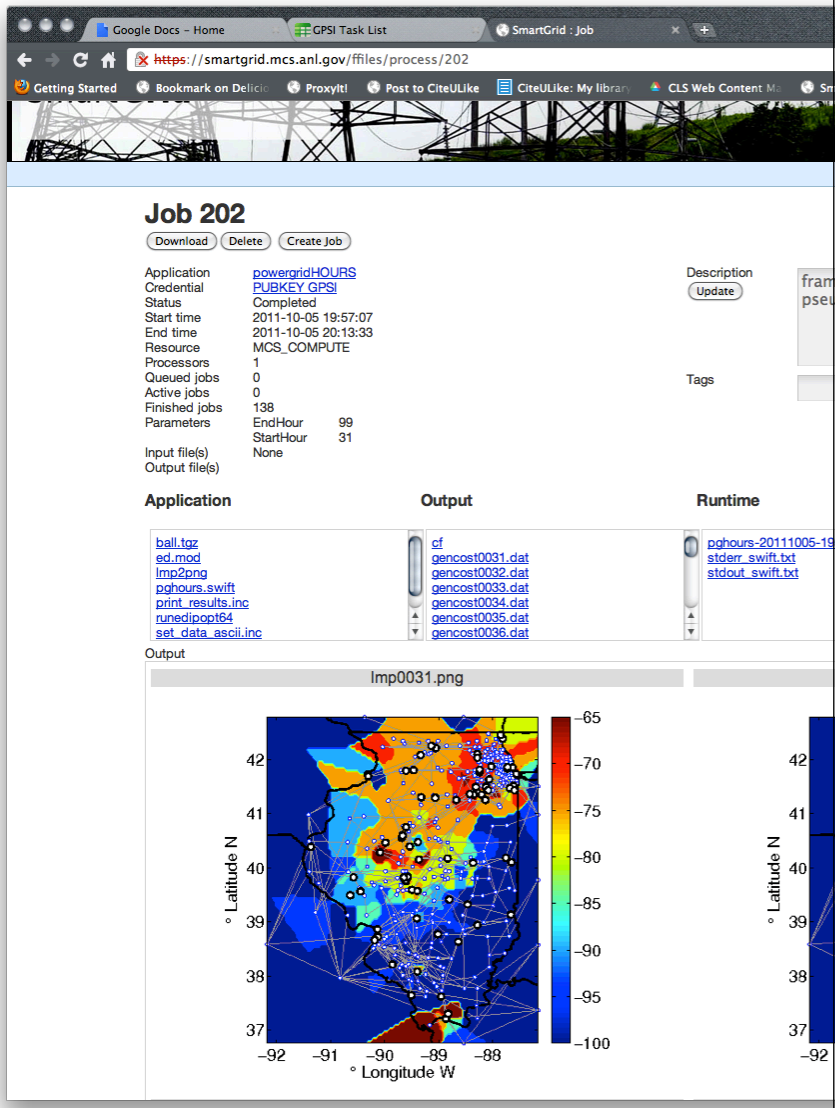
Execute Jobs

- Web form generated by introspecting Swift script, including default values
- Uses server-side proxy certificate for single sign on



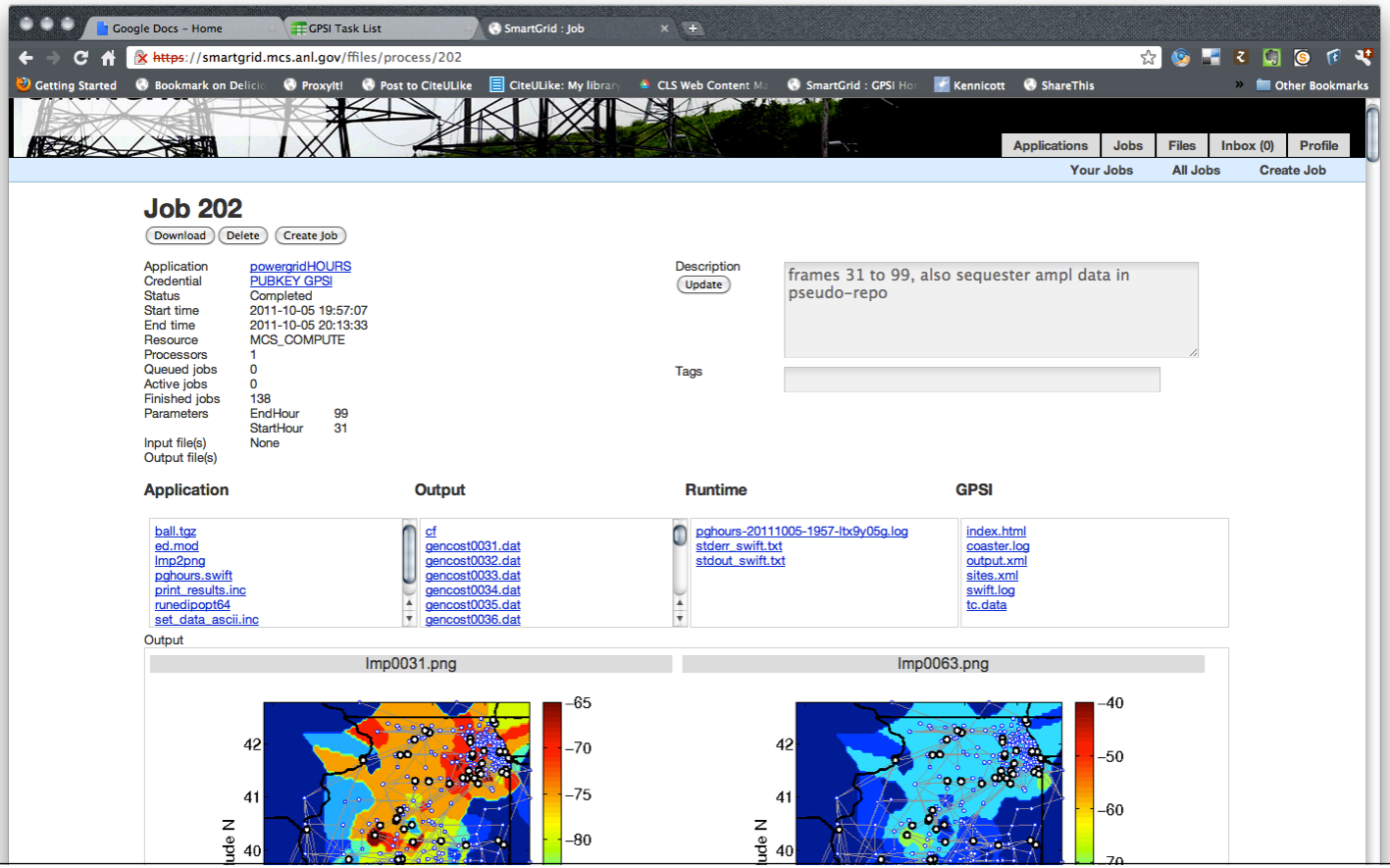
Generate, view, download results: Collaborate

- Well-known data types are presented graphically among job output
- Applications control which job outputs are included and how they're represented
  - XML designation of output files to include
  - XSLT transform of file input to representative HTML
- Users view each other's results, can run each other's apps



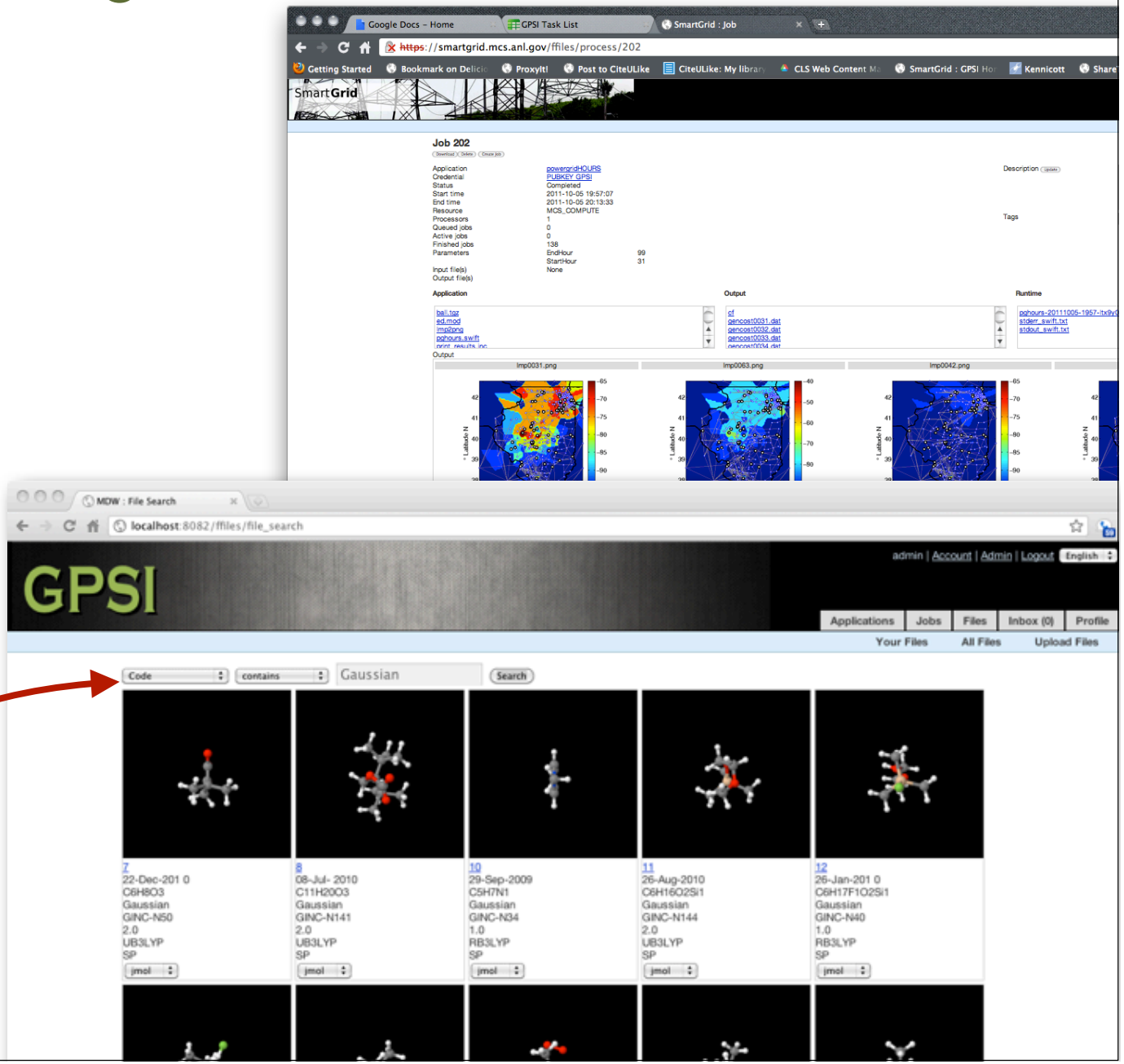
provenance

- Job details captured in one place
  - input, options, compute resource, credential
- Jobs can be rerun from within GPSI, using the original scenario



Where we're using it

- SmartGrid
- Materials
- Phylogenetics



Triple-store for metadata integration and searching

Future Work

- Integrated analysis
- Dynamic provisioning
- SCS integration
- Globus Online integration
- Data cataloging

Acknowledgements

- Michael E. Papka, Mark Hereld, Joe Insley, Eric Olson, and the Futures Laboratory at Argonne National Laboratory
- Mike Wilde and the Swift team
- Nancy Wilkins-Diehr, Suresh Marru, TeraGrid/XSEDE Gateways program
- Jim Basney, Jeff Gaynor for MyProxy OAuth
- NSF Award SCI-0503697
- XSEDE Science Gateways program