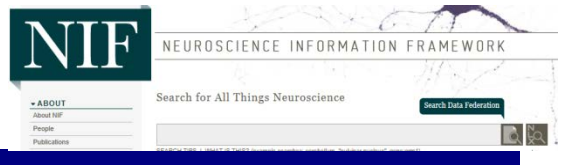
A Neuroscience Gateway: Software and Implementation

www.nsgportal.org

Subhashini Sivagnanam, Kenneth Yoshimoto, Amitava Majumdar
San Diego Supercomputer Center
Ted Carnevale
Yale School of Medicine
Vadim Astakhov, Maryann Martone, Anita Bandrowski
Neuroscience Information Framework, UCSD

Yale SCHOOL OF MEDICINE



Outline

1. Introduction – Right time for Neuroscience Gateway
2. Background – CIPRES Workbench Framework
3. Adaptation – CIPRES to NSG 
4. Summary
5. Acknowledgement

1. Introduction – Right time for Neuroscience Gateway

Growth of Computational Neuroscience

- Growth of computational modeling in neuroscience research

  Evidence:

  – *New Journals* (Neural Computation, J. of Computational Neuroscience)

  – *Increase in Modeling papers* (J. of Neurophysiology, J. of Neuroscience, Neuron - >1176 publications; ~10% annual growth)

- Trend in research proposals to NSF, NIH in US, and similarly in other countries
- Driven the evolution and refinement of powerful simulation tools
  - NEURON, GENESIS, MOOSE, NEST, PyNN, Brian etc

Research Bottleneck

- Modeling projects start "small" and many stay "small"
- Increase in development of complex models require CI, HPC
- Very few neuroscientists have access to extreme scale HPC
- Widely used simulators (NEURON, GENESIS, MOOSE, NEST, PyNN, Brian) are parallel

- Wider computational neuroscience community needs access!

We want to bring HPC to more neuroscientists

Barriers to Entry

- HPC/CI resources are available from national supercomputer centers

BUT

- Requesting time requires preliminary access to be able to write proposals every year
- Difficulty in understanding HPC machines, complex OS/software
- Need to learn policies, batch system details – different on different HPC systems
- Challenge of managing workflows involving multiple remote authentication systems
- Figuring out data transfer, output result retrieval, storage issues

Our Goals

1. Easy user interface – providing easy model upload, running of codes
2. Complete set of neuronal simulation tools – widely used by computational neuroscientists
3. Ability to easily get to the results, download results

NSG Portal

NSG Portal | The Neuroscience Gateway
A Portal for Computational Neuroscience

Home | Toolkit | My Workbench | My Profile | Help | How to Cite Us | Logout | Job Logs

Missing results? Send us the job handle, and we may be able to help.

More Information

- About Us
- Usage Statistics
- User locations
- Enabled publications
- Latest News

Welcome to NSGportal! Currently NEURON, PGENESIS, NEST, BRIAN, and PyNN codes can be run on Trestles, HPC resource at SDSC. Please login to proceed

First Time Users: Please review our usage policies

New users who are interested in getting an account should fill out the form and email it to nsghelp@sdsc.edu

NSGPortal Login

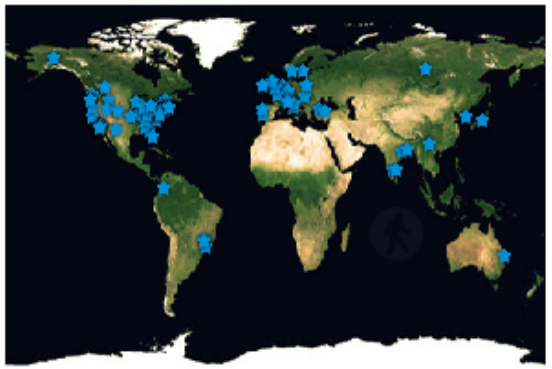
*Username:

*Password:

Login Reset

Forgot Password?

Location of users



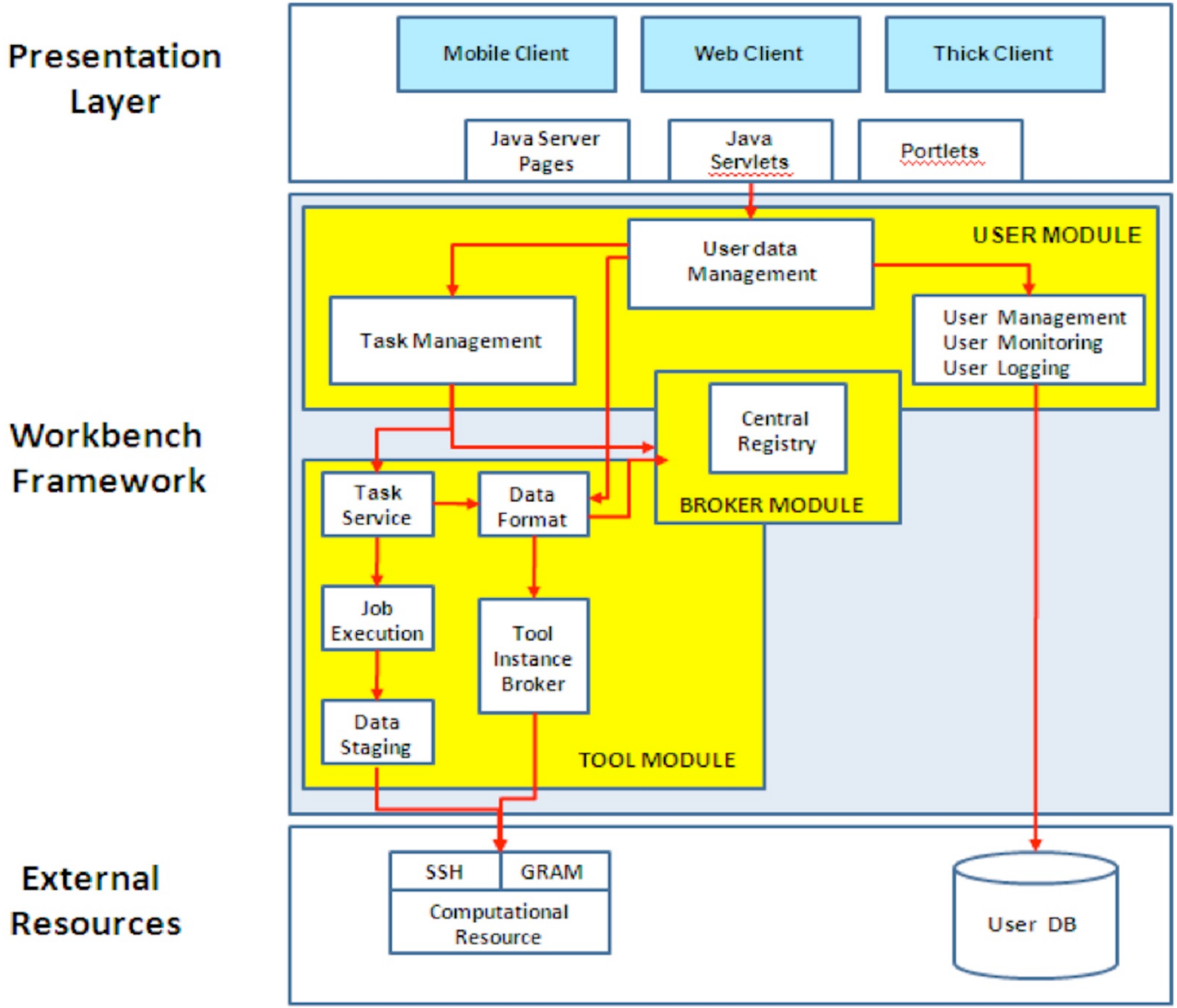
2. Background – CIPRES Workbench Framework

Background

- Why CIPRES (Cyberinfrastructure for Phylognetic Research) ?

- Well-established, robust, and mature S/W
- Adapted by other gateways
- CIPRES developers and researchers are at SDSC
- Reuse of existing NSF funded software was considered a good practice
  - viewed positively by the NSF

Workbench Framework (from http://www.ngbw.org/wbframework/)



- Presentation Layer
- Accesses SDK capabilities
- Provides lightweight web access
- Architecture based on Linux, Apache Tomcat, MySQL and Java Struts2
- Allows data and task management in user-created folders
- Users can create a login protected personal account
- Users can store data, records of their activities for a defined period

- User Module
- Passes user-initiated queries and tasks to executive portions of the infrastructure
  - Done via data and task management modules
- Stores user data and task information in MySQL
- Supports individual user roles, accounts, sharing of data
- Supports selective access to tools, data sources
- Mapping user information happens in this layer
- Allows tracking the individual usage on comp resources

- Broker Module
- Provides access to all application specific information in a Central Registry
- Contains information about all data types for input, output for each application
- Concepts and concept relationships are formed in XML and read by Central Registry
- Tools, data types defined in single location
- Allows adding new tools, data types without impacting any functioning outside of the Registry

- Tool Module
- Translates submitted tasks into command lines; submits command line strings, w/ user input data to compute engines
- Handles data formatting for jobs and job staging
- Tracks which tool can be run on which resources
- Allows adding of computational resources by editing the tool resource config file
- The application can send command line scripts, receive output by any well defined protocol
  - Unix command line, web services, SSH, GRAM, gsissh

3. Adaptation – CIPRES to NSG

Adaptation of CIPRES to NSG - Effort

- Minimal effort as opposed to starting from scratch
- Contributing factor - help provided by SDSC CIPRES experts

- Set up - Tomcat, Apache, database, VM, Cloud storage (courtesy of a separate SDSC grant) - all from SDSC IT department
- Acquired NSF XSEDE startup allocation on Trestles (SDSC), Lonestar (Texas Advanced Computing Center)
- Received XSEDE community account
- Installed NEURON on Trestles

- (Have more codes/tools now – GENESIS, NEST, PyNN, MOOSE, Brian)

Adaptation of CIPRES to NSG – continued

What's different from CIPRES?

- NSG models have multiple files (e.g. NEURON models)
  - Need to accommodate directory hierarchy containing model code files
  - uuencode/uudecode to support zip file upload to the data handling function
- Requirement to compile input model through NSG
  - mod files for custom mechanisms in C++ are compiled to calculate the biophysical effect (NEURON)
  - Modifications to data staging, job submission
- Define neuronal tools in the XML format
- Automatic storing of output file in SDSC's Cloud storage from HPC resources
- Automatic deletion of user files based on time length of inactivity

Adaptation of CIPRES to NSG – continued

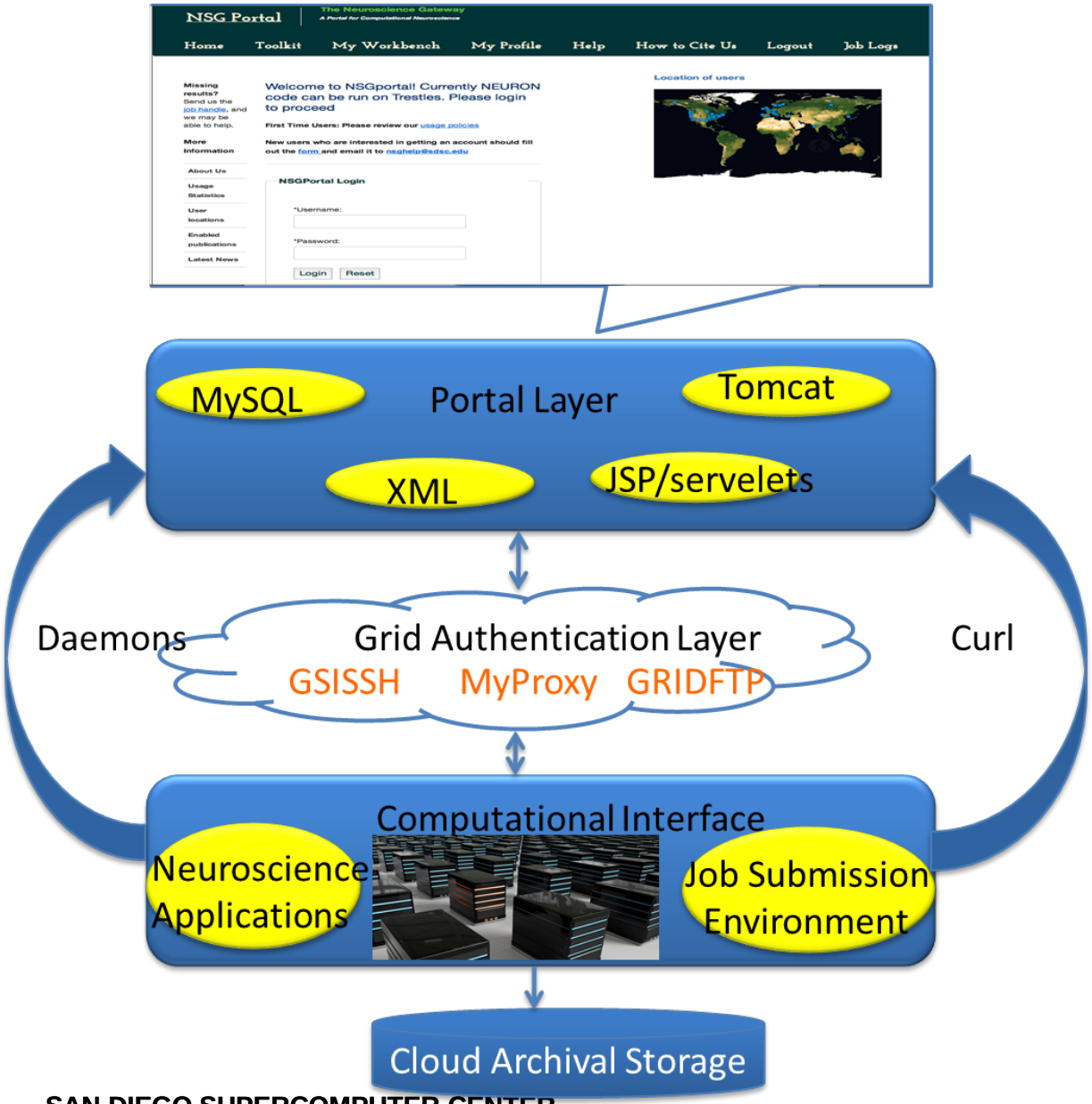
What's different from CIPRES?

- User account creation via user "vetting" process in place

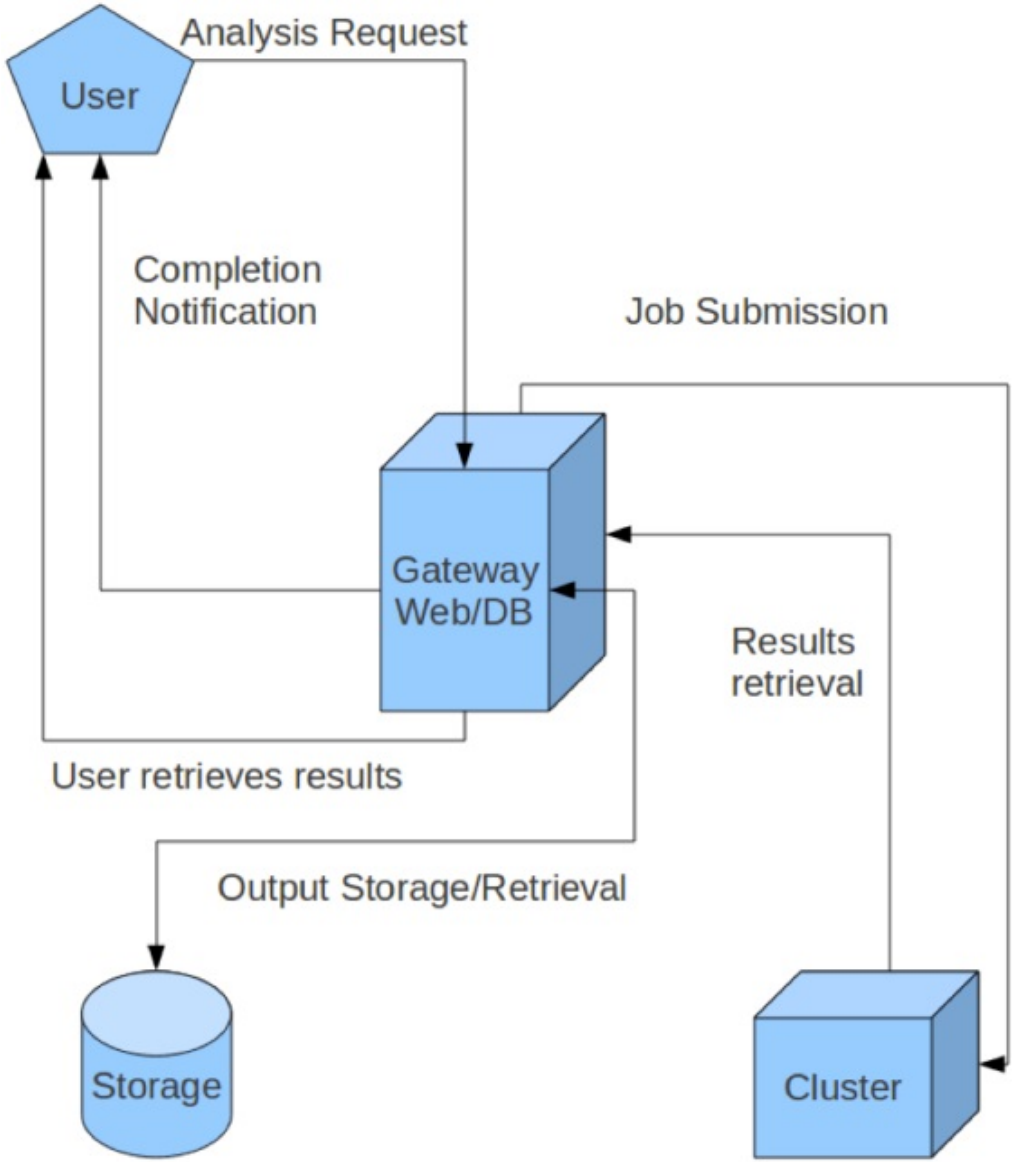
- Why the manual "vetting" process ?
  - As it is possible for user-submitted code to perform actions via HOC, C++ or shell languages (NEURON)
  - For GENESIS the interpreter has access to shell commands
  - To provide user accountability for malicious or incorrect use of HOC/MOD/SHELL languages

- Account creation process is used to verify users
  - Users submit brief contact and technical information
  - We manually "vet" by web searches
  - Then provide account creation instruction

Functional Diagram of NSG Architecture

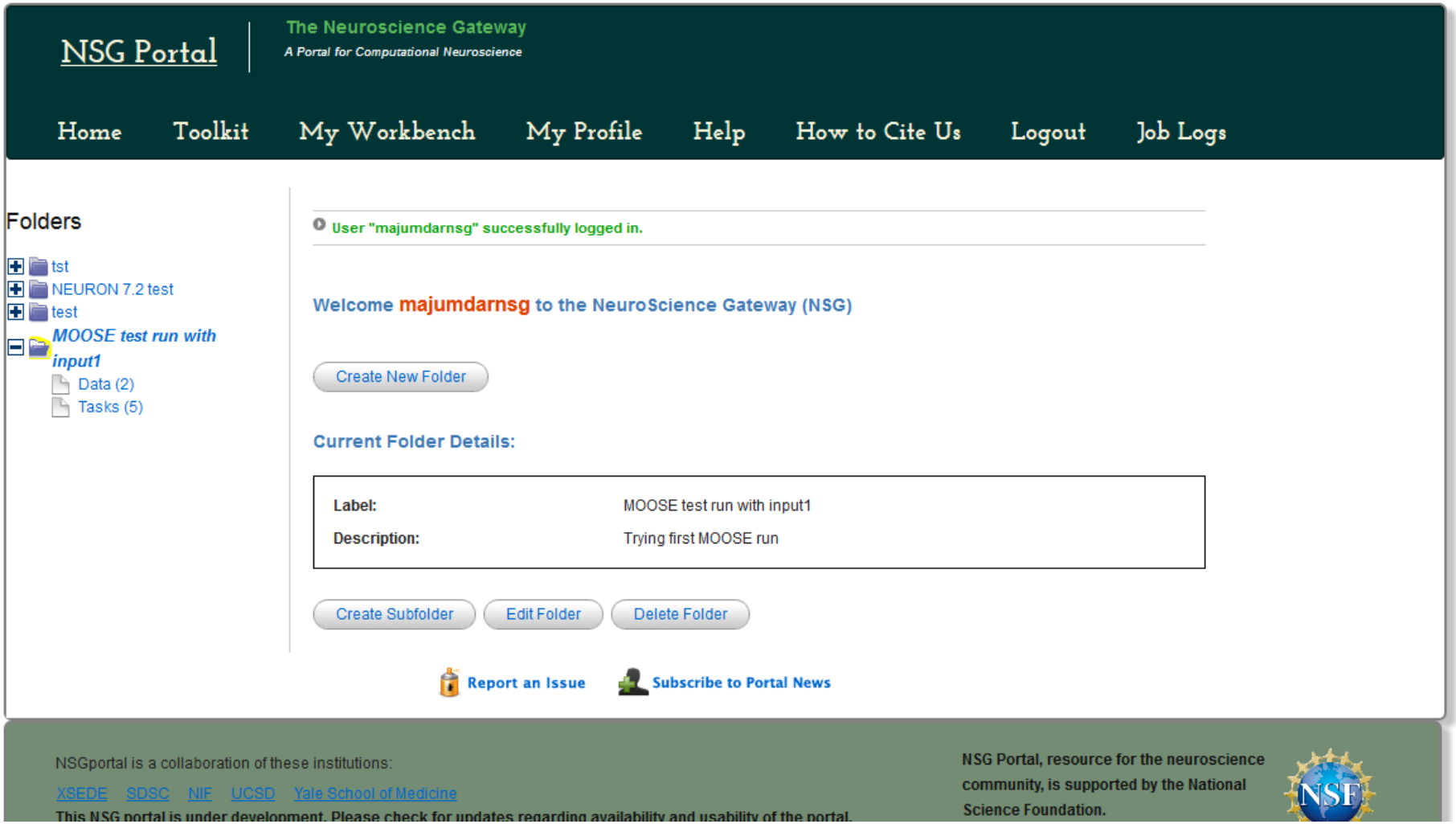


Users View of NSG



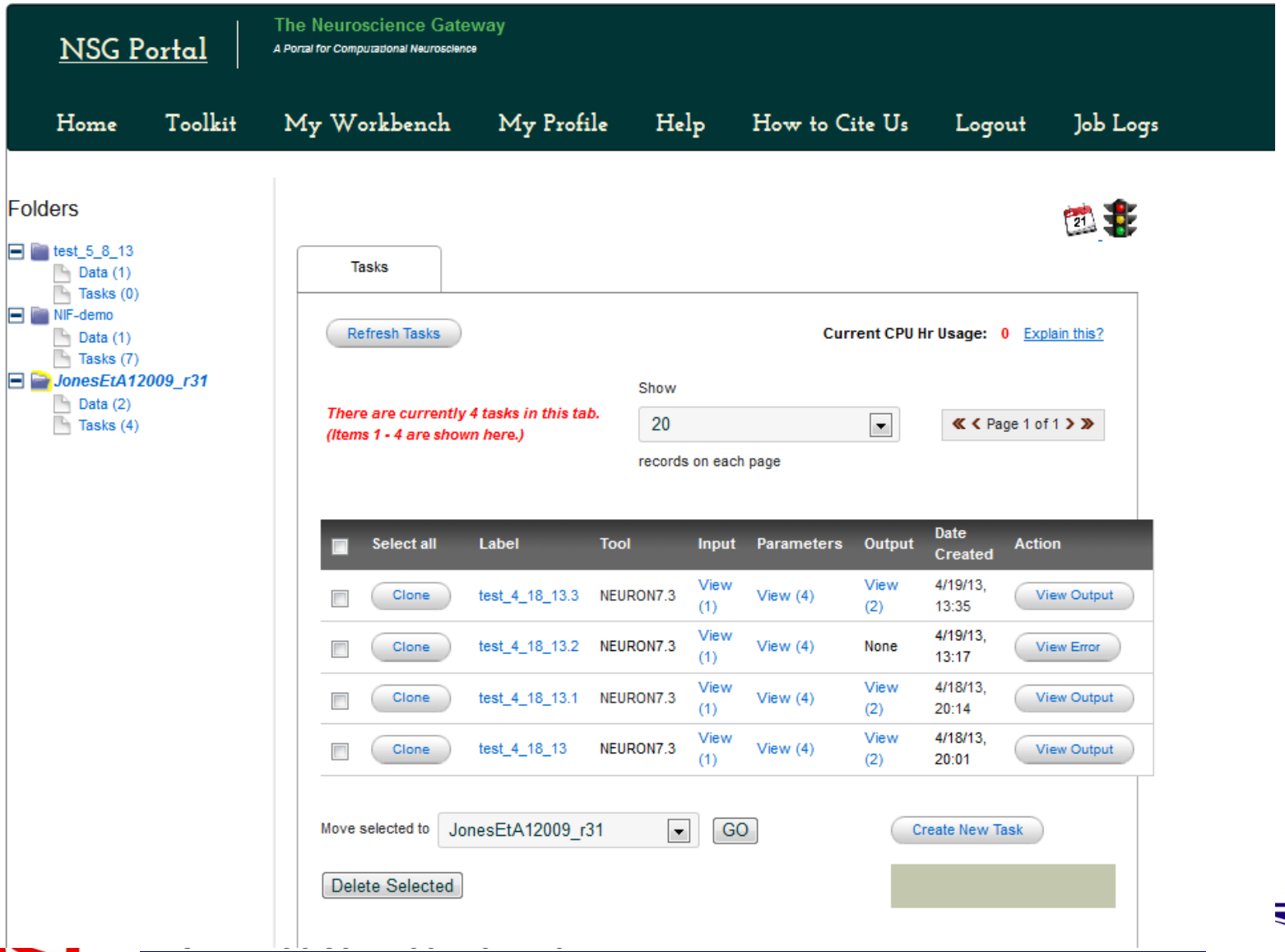
Users View of NSG – continued

Screen after portal log in



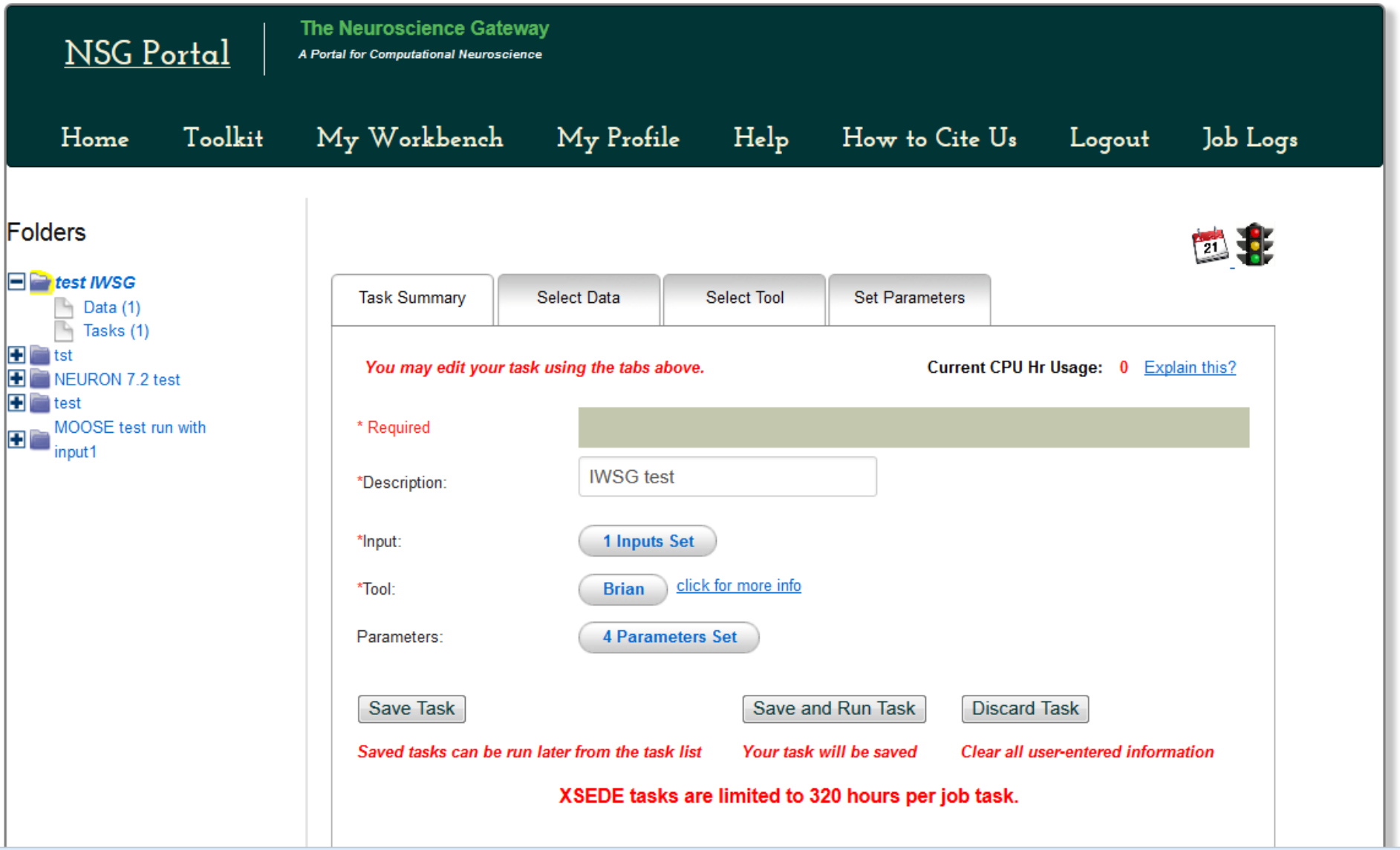
Users View of NSG – continued

Can clone task - reproducibility and data validation



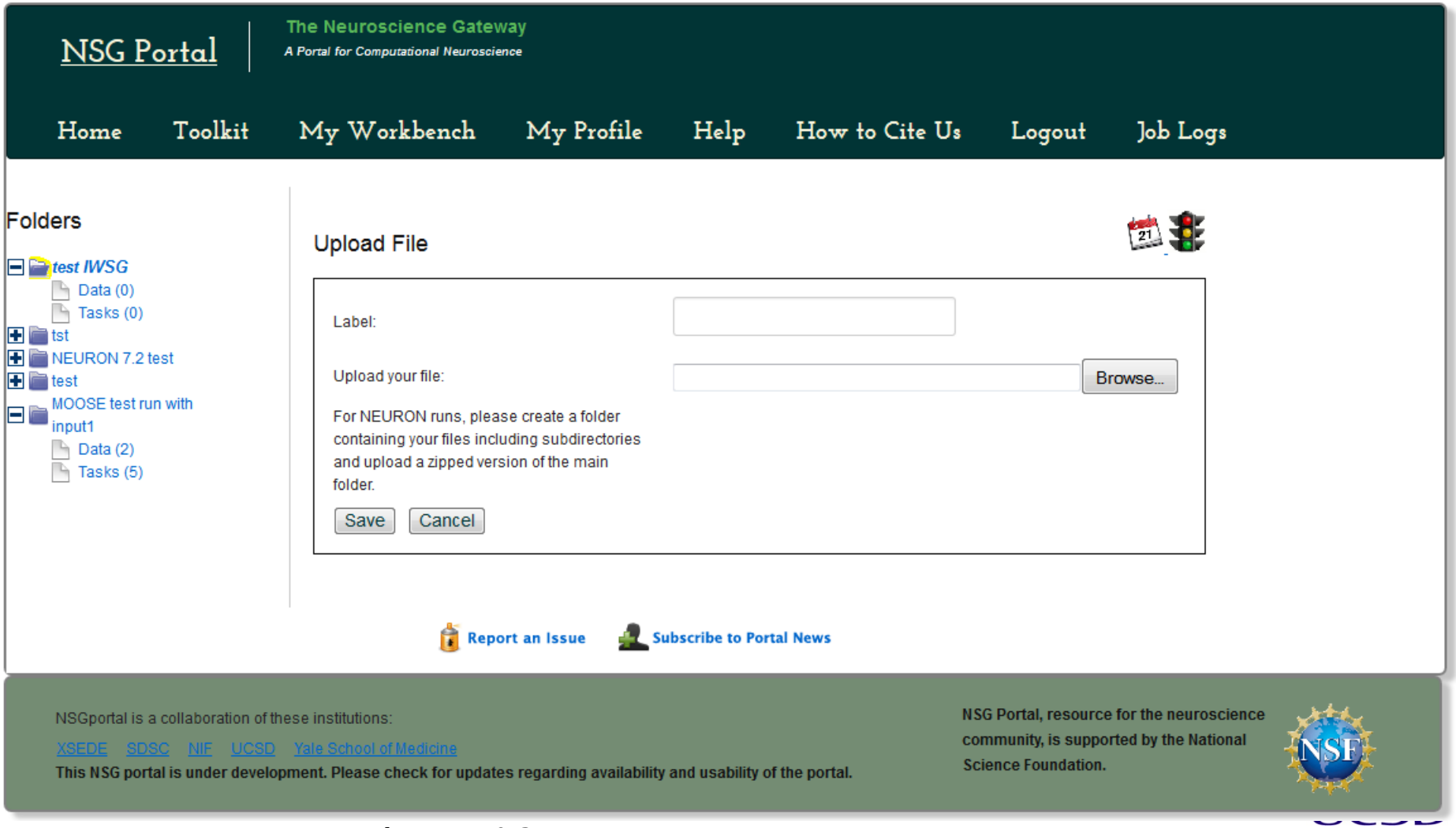
Users View of NSG – continued

- Create new task (job); specify tool, parameters



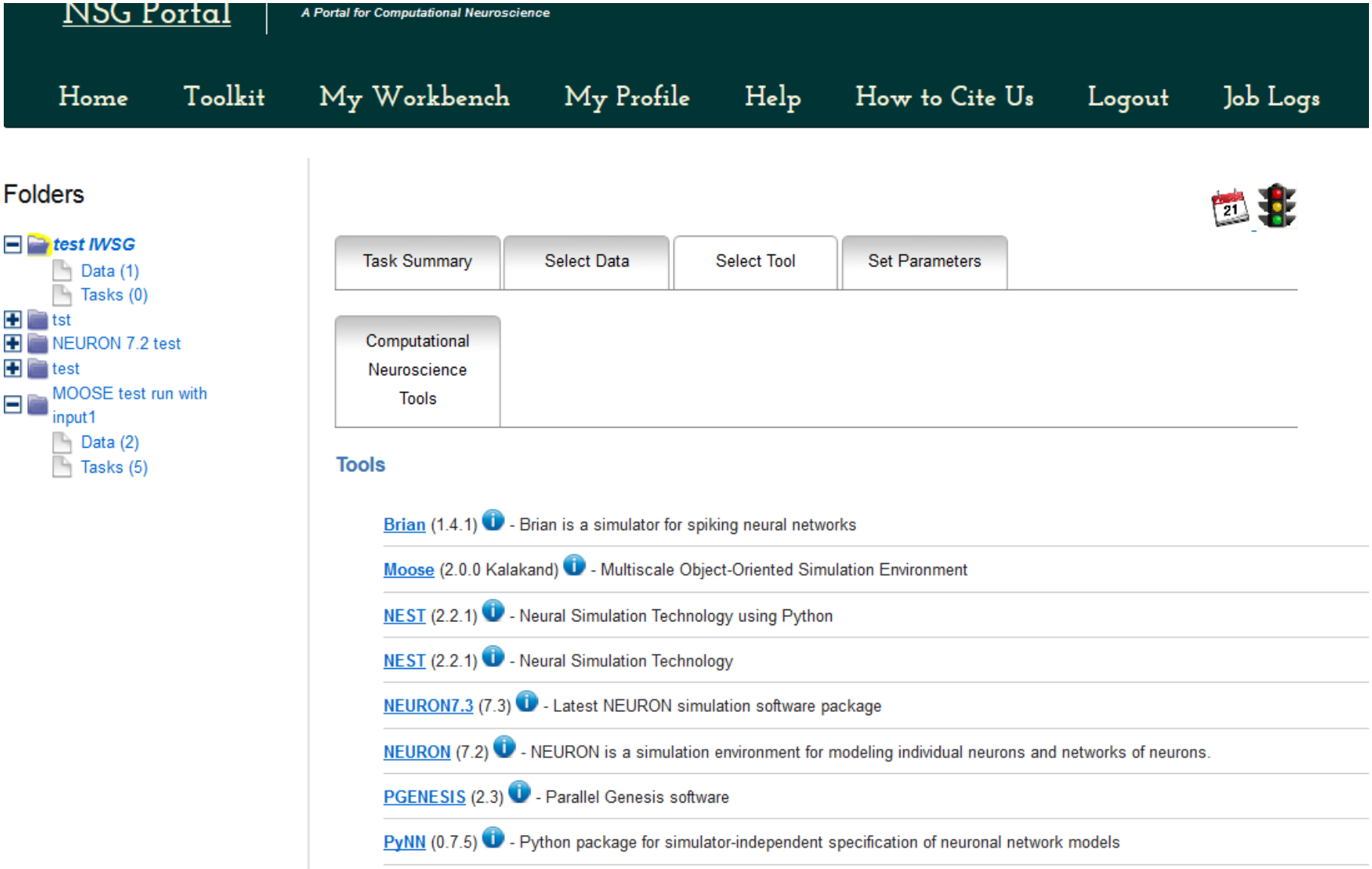
Users View of NSG - continued

- Input/output
  - Zipped directory
  - Single file
  - Save folders
  - Multiple data sets



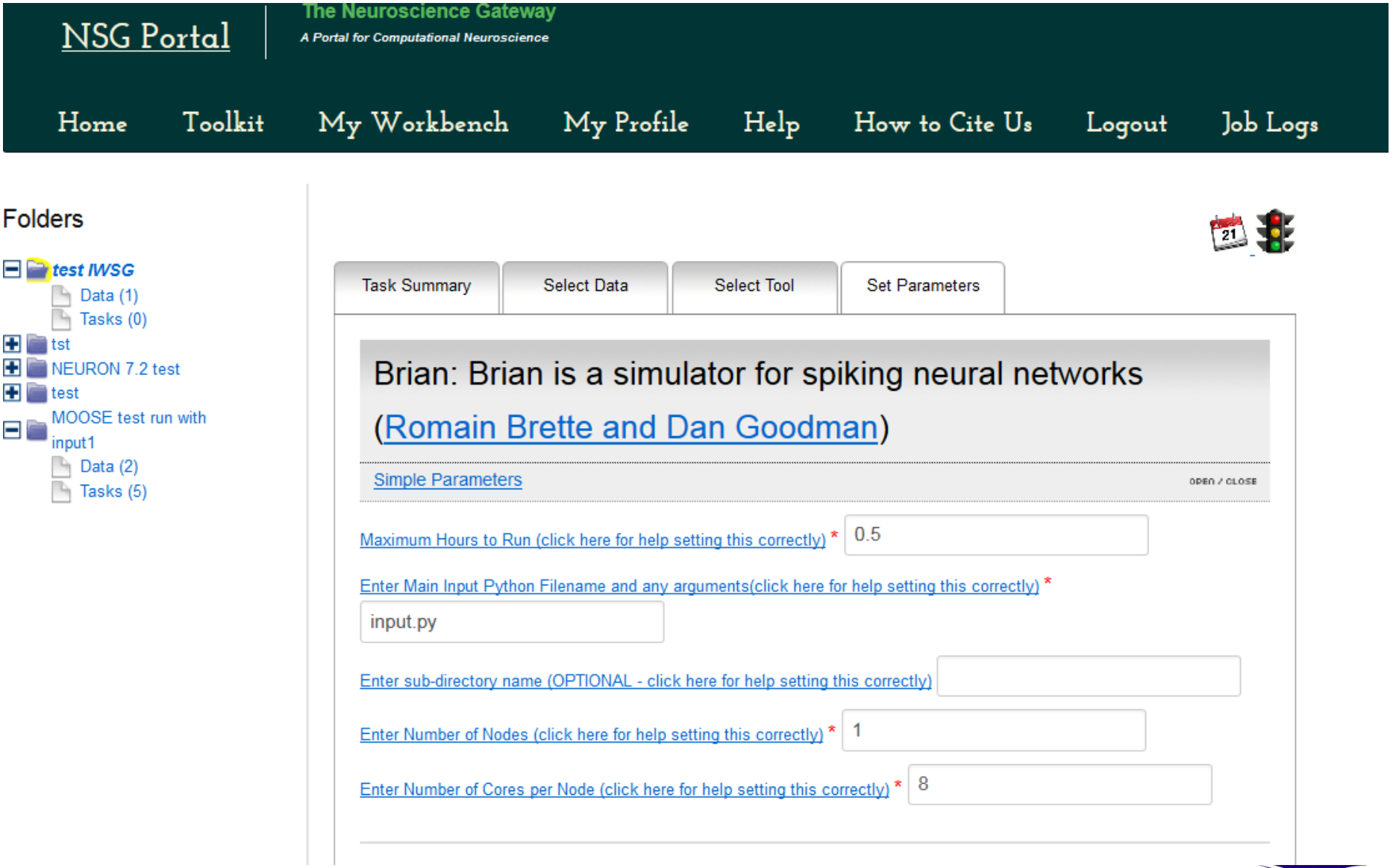
Users View of NSG – continued

- Select a Tool



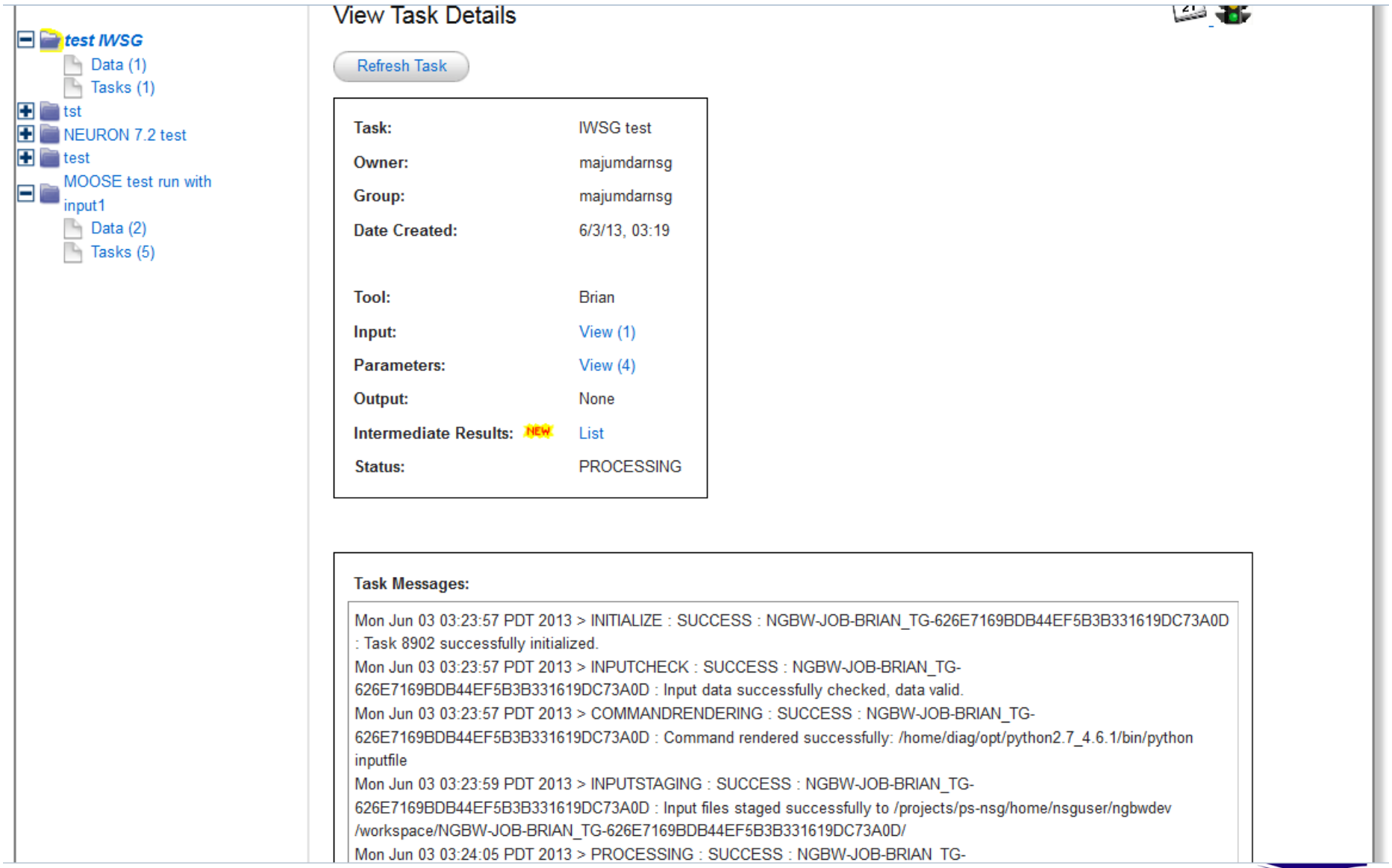
Users View of NSG – continued

- Advanced users – set parameters



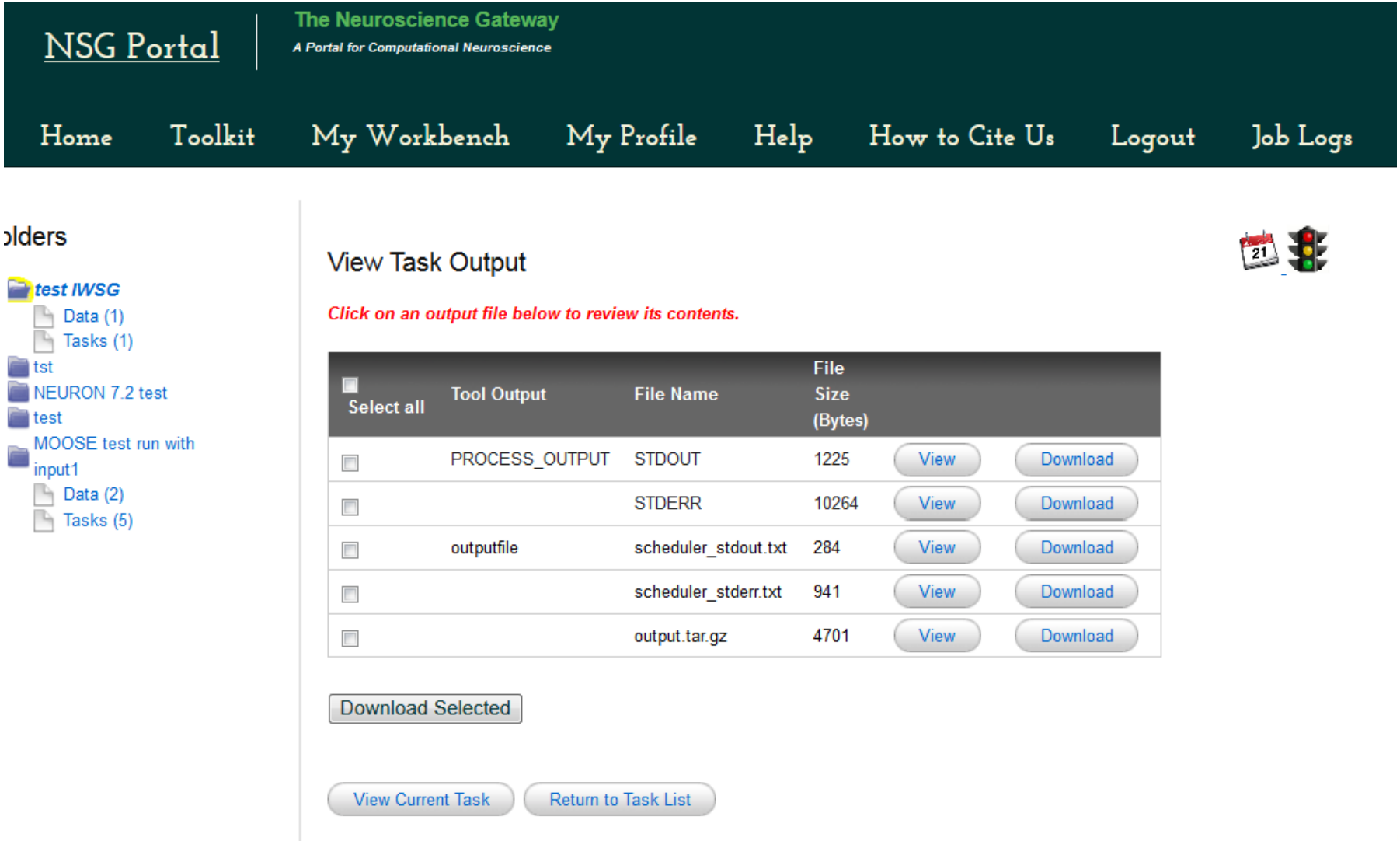
Users View of NSG – continued

- Monitor task progress



Users View of NSG – continued

Download output



Operational Features

- Two daemons – loadResultsD and recoverResultsD run as nightly cron to retrieve any failed output transfer

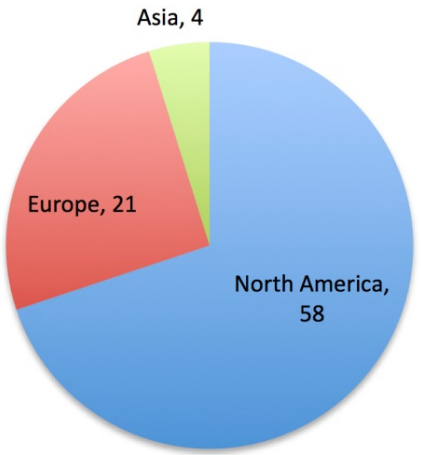
- Every job has unique job handle associated with the user – stored in DB
- Nightly cron to gather usage from TGCDB and update DB

Policies

- Any researcher, student from anywhere can use NSG
- Will go through manual "vetting" process – we verify user's existence
- Current early production state – each user limited to 5000 core hours/year – will increase as we have more users, usage
- If more than 5K/year needed, user's can use their own allocation via NSG

Early Use

- Opened for friendly users around Dec, 2012
- Received 50,000 hours of core hours on SDSC Trestles and TACC Lonestar machine
- 50K on Trestles was used up by end of Feb
  - We received supplemental 200K on Trestles
  - Another 60K has been used up in the last few weeks
  - Will write yearly XSEDE allocation proposal by the October deadline
- Currently about 80 users – usage picking up slowly
  - We made it more robust, adding hardwar
  - Now doing outreach to users
  - XSEDE-wide tutorial in March, 2013
  - SFN demo/outreach in 2012 and in Nov, 2013
  - Summer projects by HS and UG students



Current Issues – Being Addressed

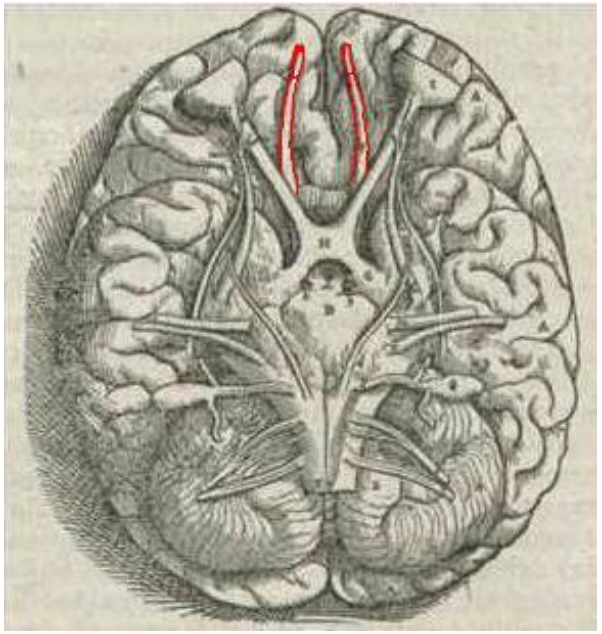
- Data transfer issue
  - GridFTP occasionally fails during data transfer between VM home directory and HPC resource
  - Alternate implementation – gsissh/scp

- NFS home dir issue on SDSC Trestles machine
  - Automount issue or home dir unavailable due to heavy load
  - Alternate – NSG's own project space as Trestles home dir
    - Use this for input staging, job submission, output staging retrieval

Example Parallel Performance

- Using the NEURON7.3 parallel code
- For 500 mitral cells and 10000 granule cells a large-scale model of the olfactory bulb – small test case run for parallel scaling
- Ran for 40,000 time steps to simulate 1 sec
- https://senselab.med.yale.edu/modeldb/ShowModel.asp?model=144570

# of Gordon cores	Time (sec)	Speedup
48	1947	
96	1016	1.91
192	485	4.01
384	249	7.81
768	163	11.94



Future Development

- Computational neuroscience requires
  - Interface with model websites/databases (modelDB, neuroConstruct)
  - Output results sharing (NIF, Wholebrain catalogue)
- Will be implemented using the REST API

- Any enhancement or modification we do for NSG, will be contributed back to CIPRES software for future use

4. Summary

NSG – Enabling Computational Neuroscience

- Providing all the benefits of a science gateway
  - Eliminating technical and administrative barriers to access HPC/CI

- Enable neuroscience education and research for institutions with less resources (and unable to bring up wet labs)

- *Democratize Computational Neuroscience Research and Education*

5. Acknowledgement

- NSF collaborative grant:
  DBI 1146949 (PI A. Majumdar, SDSC, Co-PI Martone, NIF, UCSD)
  DBI 1146830 (PI T. Carnevale, Yale School of Medicine)
  Collaborative Research: ABI Development: Building A Community Resource for Neuroscientists

- Mark Miller, Terri Schwartz, SDSC for CIPRES software (based on which NSG is implemented) and help with implementation
  - XSEDE ECSS support

- SDSC Internal (CID) grant – S. Sivagnanam, K. Yoshimoto

- XSEDE – www.xsede.org
  - Providing HPC resources for the NSG

Thank You