Freesurfer on OSG Connect: a simple submit service for computational neuroscientists

Suchandra Thapa

September 30, 2015

Summary

- There are few high-demand OSG users.
  - Few or no alternatives → high need → high motivation
  - Will invest effort to learn and overcome problems.
  - Example: Referee Consensus Solver → neuroelectric brain imaging
- Much more numerous low-demand OSG users.
  - More alternatives → reduced need → reduced motivation
  - Easily diverted by the difficulties of using the OSG.
  - Low hanging fruit solution:
    - Identify OSG compatible apps with large user communities
    - Provide turn-key tools for one app at a time.
    - Demonstration app from brain science: freesurfer
  - Use the demo app as a model for other app's
    - Turn-key user interface
    - Limited functionality

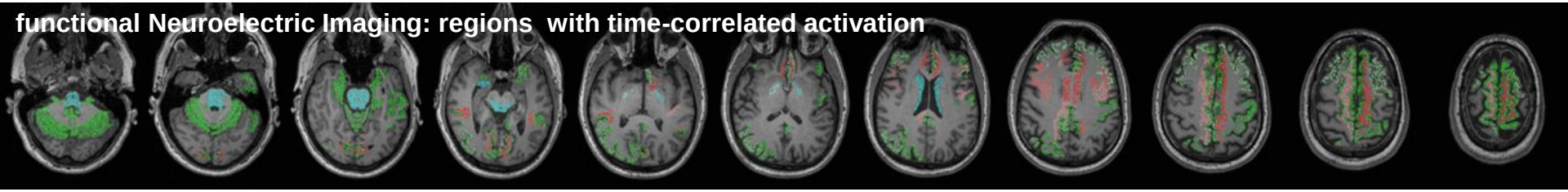
Utilizing the grid

- Access is open to those at universities, government labs, many others.
- Your need must be sufficient to motivate your start-up effort.
- The great majority of potential users have a modest need with consequently limited motivation.
- To reduce the start-up effort we are trying a simple approach:
  - Provide turn-key command line tools for one software package at a time.
  - We have selected a brain image processing application as a demonstration, Freesurfer (Martinos Biomedical Imaging Center, Harvard University).

Freesurfer User Base

The Freesurfer user base will benefit from access to the Open Science Grid:

- There is a user base of at least 3,000 world-wide. Most may be contacted through the open-subscription Freesurfer mailing list.
- Many users lack sufficient local computing resources and/or local software support expertise.
- Freesurfer execution is a high-value low-demand use of the Open Science Grid …



Freesurfer on the Open Science Grid

Freesurfer is a suite of image processing utilities that segments MRI images.
Freesurfer jobs are readily adapted for execution on the Open Science Grid:

- Each job is fully independent.
- Execution time: 16-24 hours (single core), 4-8 hours (multiple cores)
- Memory requirement: ~3 GBytes.
- Input to each job: ~10 MBytes.
- Output from each job: ~400 MBytes.

Turn-key use of the Open Science Grid

```
fSurf --runFile SubNo_01_defaced.mgz --singleCore          ;# no rush
fSurf --runFile SubNo_02_defaced.mgz --multiCore           ;#       rush
fSurf --showJobsStatus
fSurf --fetchResults SubNo_02_defaced.mgz
```

Our demonstration effort will produce turn-key command line tools:

- Submit a Freesurfer job to the grid.
- Monitor the job status.
- Fetch the results when complete.

A web-browser turn-key interface is being developed as a university computer science classroom project under the direction of Bill Laboon, author of A Friendly Introduction to Software Testing .

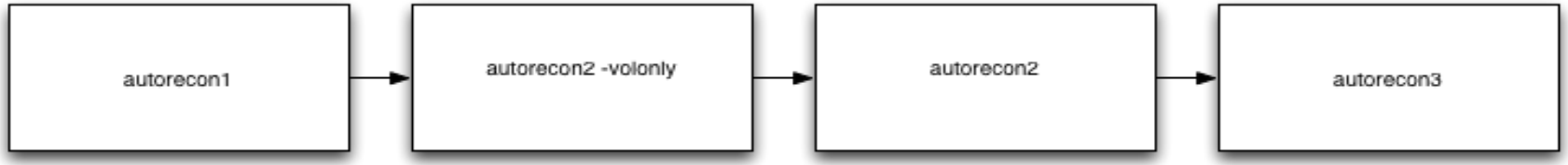
Freesurfer workflow - serial workflow

Simplest freesurfer workflow -- Single workflow running four steps *in sequence*

1. autorecon1
2. autorecon2 - volonly
3. autorecon2
4. autorecon3

Only the autorecon2 step can effectively use multiple cores

Broken up into 4 steps to avoid excessive preemptions



Current status

Replicated original workflow on COMET using Don Krieger's scripts

Installed Freesurfer 5.3.0 on OASIS

Set up pegasus workflow and DAX generation scripts
(https://github.com/OSGConnect/freesurfer_workflow/tree/master/python)

- Supports both single job and diamond dag workflows
- Supports using single and multicore jobs

Currently testing workflows and middleware

Testing program

- Tested single core, single job workflow
  - Various OSG sites via OSG Connect
  - Ran tests 4 times on 8 sample images (32 images processed in total)
  - Average time to process a single image approximately 10 hours (36427s)
  - Min time ~ 8 hours (28696s) and a max of ~17 hours (64659s)
  - Peak memory utilization was about 2.4GB
- Scaling up to 100 jobs in various workflows:
  - single job, single core
  - single job, multicore (2, 4, 8 cores)
  - diamond dag, single core
  - diamond dag, multicore (2, 4, 8 cores)

Browser interface for Freesurfer

- Working with 2 student groups at University of Pittsburgh *mentored by Bill Laboon, author of A Friendly Introduction to Software Testing .*
- Students are doing this as part of a class on software engineering
- Providing students with requirements documentation, middleware, and technical support
  - Interaction with student groups is being handled by slack, email, and phone conversations as needed
- Students will create web interface to middleware allowing users to register for an account, update account information, upload images for processing, monitor jobs, and download results
  - Students will provide a github repo with finished code
  - Student projects written in Java

Caveats re OSG involvement in freesurfer demo

- The freesurfer demonstration effort is valuable as a use case which drives development of OSG functional capabilities:
  - Multi-core “advertisement” and job deployment.
  - Usage accounting for a non-VO user community.
  - What’s “under the hood” provides an instructive working model for use as a tutorial. This may help reluctant cpu cycle contributors get past concerns over handling their client users’ requests for support.
- The browser interface is not essential to our demonstration of Freesurfer since we will have a turn-key command line interface to our middle-ware.
- The browser interface will likely be valuable if it happens. If not, it required minimal effort from us.
- Generating a requirements document, fielding questions, and the other details attendant on working with the students have proven helpful.

Middleware for Freesurfer

Two different components:

- Command line tool to submit image for processing (for testing and benchmarking)
- Web server providing REST interface for student frontends
- Using nginx, uwsgi, and python
- Lightweight shim to pegasus tools and to access database

Upcoming work

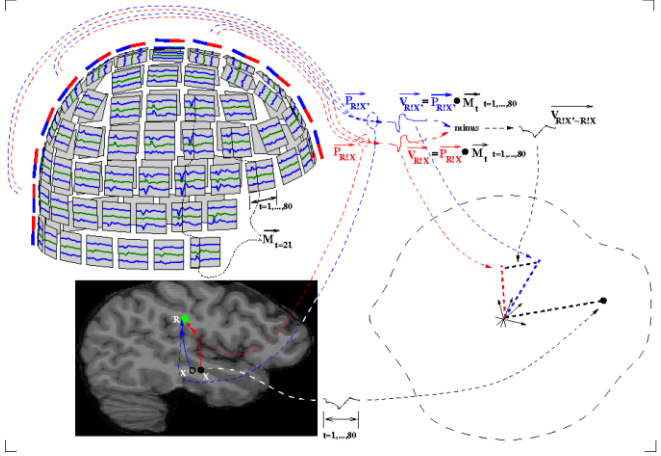
Server to provide REST endpoints for web frontends

Optimizing workflows to provide best throughput for jobs

Creating factory to submit jobs to COMET

Coordinating with student projects

Referee consensus solver: An evolving prototype on the Open Science Grid



- Referee consensus is a new approach to solving large systems of simultaneous mixed linear/nonlinear equations (ACM, Intl J Adv Comp Sci, Explanation of the figure on YouTube).
- Solution of the equations requires search of a high dimensional space.
- The discovery and use of a cost function computed in the “source” space rather than in the “measurement” space enables:
    - solution for a single source at a time,
    - efficient implementation on the grid.

Referee consensus solver applied to magnetic field recordings

Our group is using this method to solve a functional brain imaging problem.

The data are serial magnetic field measurements (1 KHz) from which neuroelectric currents within the brain are localized and measured.

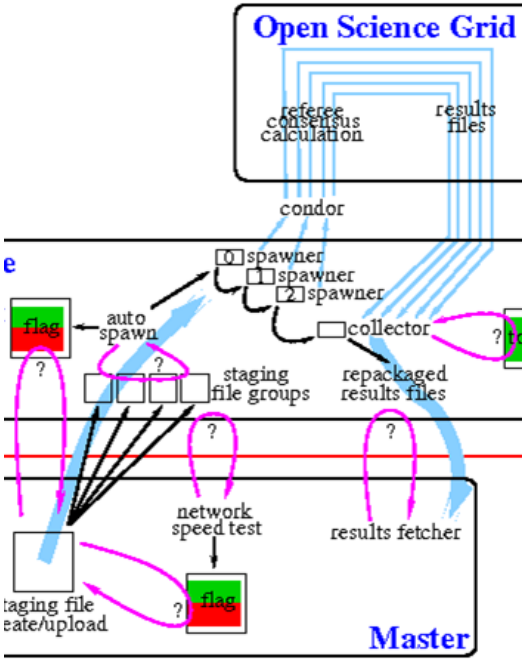
Primary data processing using the referee consensus solver requires ~50 cpu hours / second of data. Hence ours is a high-value high-demand use of the Open Science Grid.

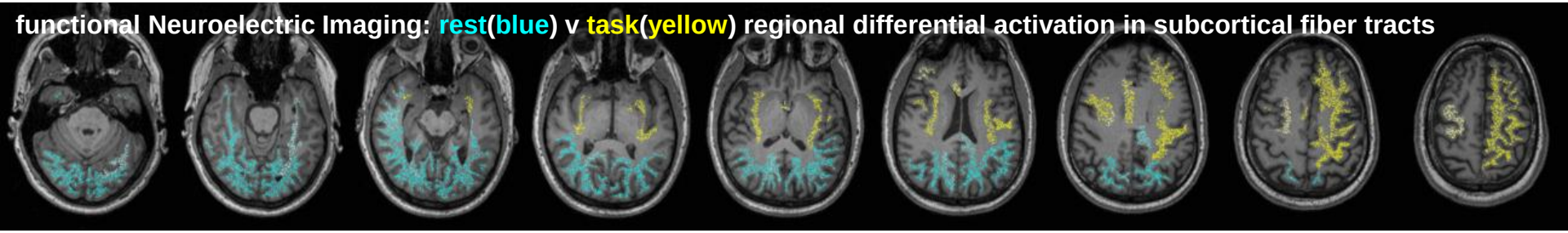
Referee consensus solver on the Open Science Grid

High computational need motivated us to create a robust and efficient work flow. This was aided by our experience in creating and using real-time monitoring applications for high-risk surgical procedures.

Our effort used 40 million hours in the past 12 months:

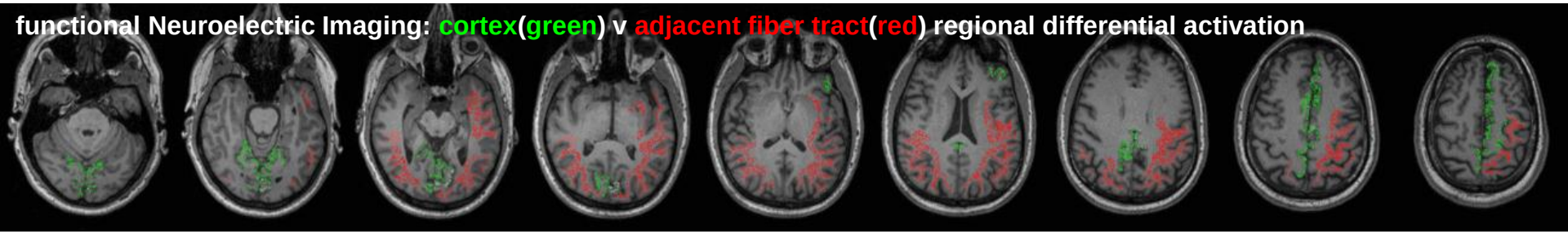
- Job execution time: 30 min - 6 hours.
- Memory requirement: ~400 MBytes.
- Input to each job: ~100 MBytes.
- Output from each job: ~50 KBytes.





High-value high-demand use of the Open Science Grid

- Our functional neuroelectric imaging effort is entirely dependent on the calculations performed on the Open Science Grid.
- The method applied to neuromagnetic field recordings extracts ~1000 times the information obtainable in any other way.
- Comparable performance is expected for its use in identifying and removing atmospheric noise from ground-based microwave telescope images.
- Our results demonstrate the first functional imaging of differential activation within the brain's "white matter," the fiber tracts through which neural signals propagate.



Planned applications for referee consensus on the grid

- Current dipole lists and regional differential activation maps for each recording will be placed online for open subscription access.
- Human Connectome Project cohort (n=68) MEG recordings:
  - 68 neurologically normal volunteers
  - eyes open rest + 3 behavioral task
  - ~80 minutes recorded per volunteer : ~20,000,000 cpu hours
- Cambridge (England) Center for Ageing and Neuroscience MEG recordings:
  - 656 neurologically normal volunteers
  - eyes closed rest + 2 behavioral tasks
  - ~20 minutes recorded per volunteer: ~45,000,000 cpu hours
  - Recordings repeated 1-2 years later in ~250 volunteers: ~ 20,000,000 hrs