Zero to Neuro

Tina Huang[1], Yuchen Huang[2]
Subhashini Sivagnanam[3], Amitava Majumdar[3]
[1]Westview High School, [2]Cathedral Catholic High School,
[3]UCSD San Diego Supercomputer Center

Abstract

Using supercomputers drastically increases the efficiency of completing a task by using multiple nodes and tasks per node to spread out the work. The Neuroscience Gateway (NSG) is an effective tool for neuroscientists to gain access to supercomputers to run computational neuroscience simulations, eliminating the need to go through the supercomputer's backend to submit their scripts. We compared the steps for using NSG and going through the backend to demonstrate how the simple user interface of NSG makes it efficient and less complicated for neuroscientists around the world to further their research.

Introduction

The research intends to explore the convenience of providing gateways on supercomputers, such as the Neuroscience Gateway. Supercomputers increase the efficiency of research by assigning work to multiple cores, thus distributing the workload and decreasing the work time. The utilization of supercomputers require knowledge of operating Linux, using compiling tools and understanding schedulers and file systems. Gateways like NSG are used to make up for the lack of cyber-infrastructure knowledge for some scientists. The comparison between the NSG interface and Comet back end highlights the ease of use of NSG, which enables any scientist to make full use of the supercomputers regardless of their cyber-infrastructure background. Neuroscientists can also use computer simulated environments to recreate processes in the brain, and can increase complexity and test many variables to predict what might happen in a real life situation.

Methods/Tools

1. Comet

Comet is a dedicated eXtreme Science and Engineering Discovery Environment (XSEDE) cluster designed by Dell and SDSC delivering 2.76 peak petaflops. It features Intel next-gen processors with AVX2, Mellanox FDR InfiniBand interconnects, and Aeon storage.

2. NEURON

NEURON is a simulation environment for modeling individual neurons and networks of neurons. It provides tools for conveniently building, managing, and using models in a way that is numerically sound and computationally efficient. It is particularly well-suited to problems that are closely linked to experimental data, especially those that involve cells with complex anatomical and biophysical properties.

3. Neuroscience Gateway

The Neuroscience Gateway (NSG) project facilitates access and use of National Science Foundation funded High Performance Computing (HPC), High Throughput Computing (HTC) and cloud resources by neuroscientists.

4. Jones model

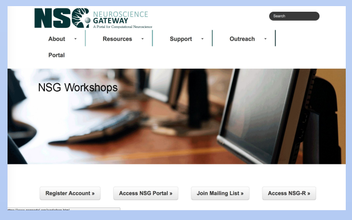
An example of biophysically realistic neural model.

Research and Results

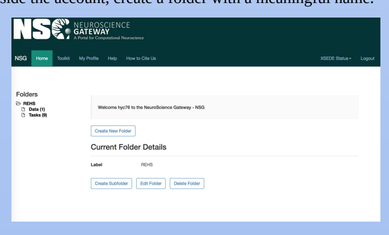
Part 1: Scaling Studies Through NSG and Comet

Access through Neuroscience Gateway (NSG) Portal

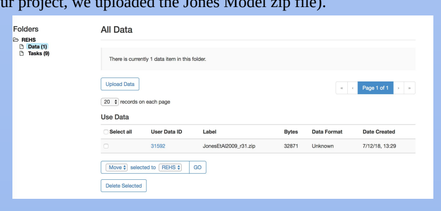
1. Access the NSG (Neuroscience Gateway) Portal [1].



2. Click "Register Account" to create an account and fill in all the information needed.
3. Once the account is created, click "Access NSG Portal" to access the login page.
4. Inside the account, create a folder with a meaningful name.

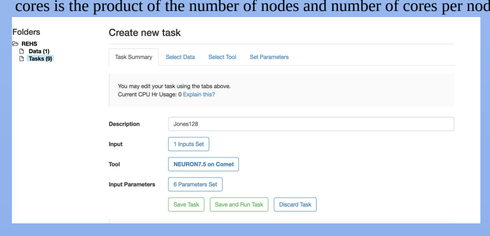


5. Inside the folder, go to "data" folder and upload zip files into the folder. (In our project, we uploaded the Jones Model zip file).

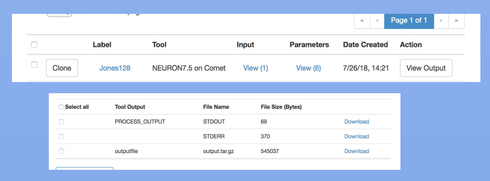


6. Once data is uploaded, go to the "task" folder and create new tasks.

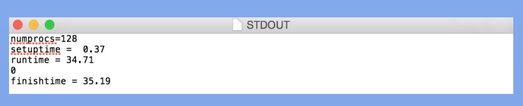
7. Modify number of cores through the parameter list, where the final number of cores is the product of the number of nodes and number of cores per node.



8. "Save and Run Task" to NSG.
9. An email would be sent informing the completion of a task.
10. Once the job is completed, view the output and download the "STDOUT" file.



11. Record the time used to complete the task.



Access through Comet Backend

1. Make a Comet account.
2. Go to Model DP [2] and download the zip file.
3. Download "WinSCP" to transfer the Jones file to your own Comet directory.
4. Log into your Comet account.
5. Drag and drop the Jones script from your computer's files into your Comet directory.
6. Download "putty" to access your Comet account, or use terminal for Macs.
7. Log into Comet [3].
8. Log in with your Comet username and password.
9. Type "ls" to display all of your files. "JonesEtAl2009" should be listed.

10. Enter the "JonesEtAl2009" directory by typing "cd JonesEtAl2009/".
11. Create or edit a pre existing script and call it "Jones_script". This will be used to submit the Batch.hoc file to run the program.
12. Type "vi Jones_script" to edit the script. Type "i" then enter to insert and edit the information.
13. Set the job and output name to a meaningful name to keep track of varying nodes and tasks. Change the mail-user to your own email to receive notifications of when the submitted task is finished. Set the number of nodes and ntasks per node both to 1. Add in "module purge", "module load intel", "module load openmpi_ib", and "ibrun ~nsguser/applications/neuron7.5single/installdir/x86_64/bin/nrniv -mpi Batch.hoc" at the end of the program.

14. Hit the "esc" key, then type "ZZ" to exit and save the edited script.
15. Enter "sbatch Jones_script" to submit the job.
16. There should be a new line that says "Submitted batch job ___" with a string of numbers.
17. Check your email to see when the submitted job is completed, then type "ls" to find the output file. It should be listed as the name you've given the output file, for example "1node1task".
18. Type "cat ___" with your output file name to see the contents of the output file.
19. Record the finish time time as a data point. This time includes the set up time and the run time.

20. Repeat steps 12 through 19 with varying numbers of nodes and tasks per node, making sure to also update the job and output names accordingly to keep track.

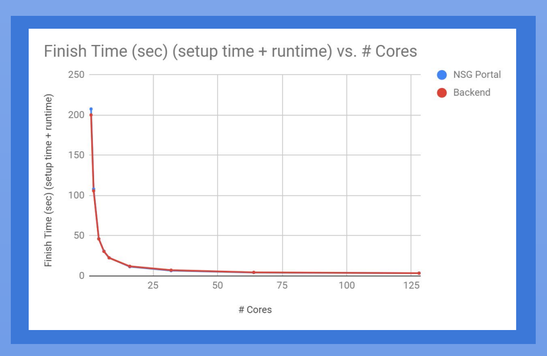


Figure 1 Comparing the finish times using the NSG portal and through the backend of Comet using the Jones model [2].

Part 2: Modeling Action Potential Using Python and NEURON

We accessed the NEURON + Python Basics page [4] and created a simple neuron with a passive cell membrane using Python code. We inserted an alpha synapse, set up recording variables, then ran the simulation using NEURON's software.

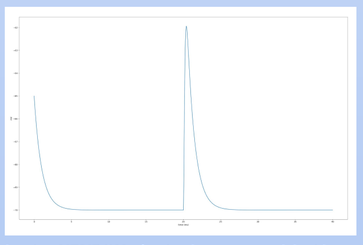


Figure 2 Action potential graph using NEURON of a simple neuron model with a passive cell membrane.

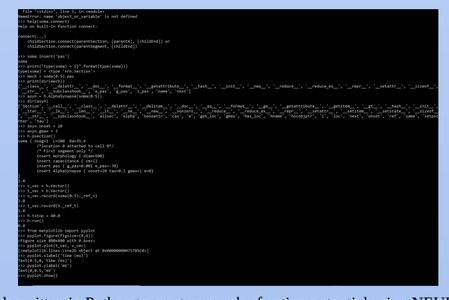


Figure 3 Code written in Python to create a graph of action potential using NEURON.

After familiarizing ourselves with the NEURON software and design, we continued the ball and stick tutorial to create a model of a soma connected to a dendrite. We added biophysical mechanisms, and then stimulated the neuron by sending a current pulse through the distal end of the dendrite. Varying the number of segments on the dendrite resulted in different outputs.

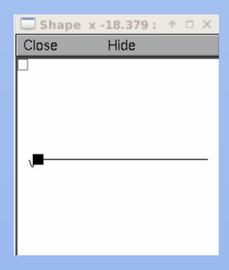


Figure 4 A simple ball and stick model of a neuron with a soma and dendrites.

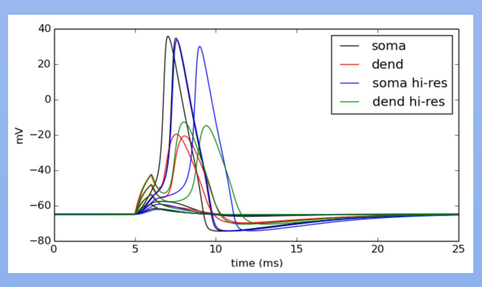


Figure 5 Measuring the action potential along various parts of the neuron. The number of segments was also another factor that modified the output.

Summary

After 2 months of research on NSG and Comet, we have learned that gateways greatly increase the efficiency of executing computational neuroscience tasks. Neuroscientists can easily use simulation environments, such as NEURON, to model brain structures and activities, such as action potentials in neurons. Modeling real life processes in hypothetical environments can push the limits of experiments that can be accomplished in laboratories.

Future Work

We plan to continue our work by exploring how to model more complex functions of the brain using NEURON. Our first step would be to create a model of a network of cells that can run on a parallel machine, then investigate how different neurotransmitters affect synapses.

References

[1] https://www.nsgportal.org/index.html
[2] https://senselab.med.yale.edu/ModelDB/ShowModel.cshtml?model=136803
[3] https://www.sdsc.edu/services/hpc/hpc_systems.html#comet
[4] https://www.neuron.yale.edu/neuron/static/docs/neuronpython/firststeps.html#what-is-neuron