Created: 20201104
Last modified: 20210616
Editor: George

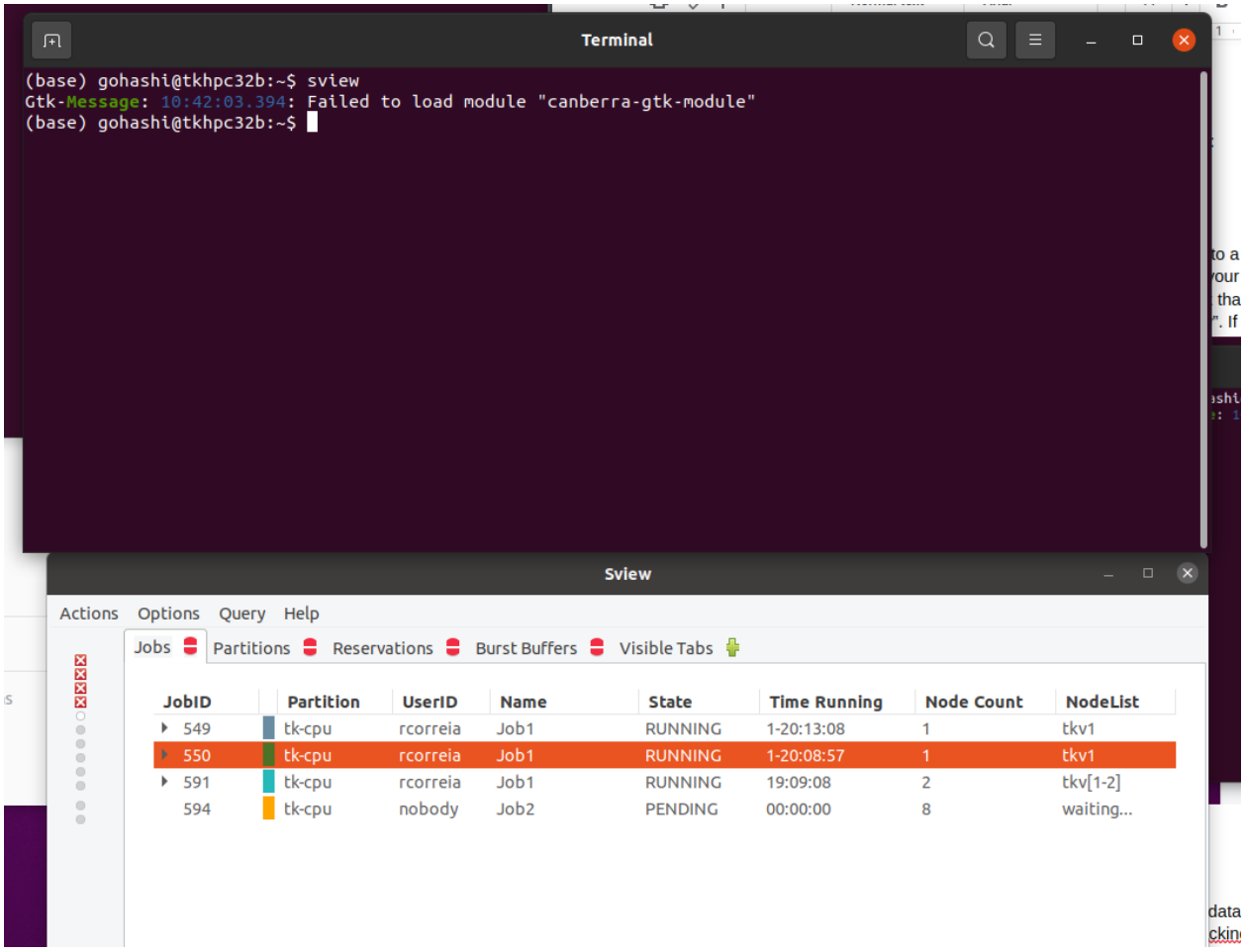
Connecting SLURM to Matlab

## Connecting to a workstation

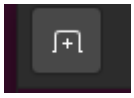
1. Follow steps here
    a. “/nfs/data1expansion/datasync3/llsm-deskew/afterAcquistion_20210617/Protocols/Connect_workstations_Remote.pdf”
2. Jump to step 1 below.

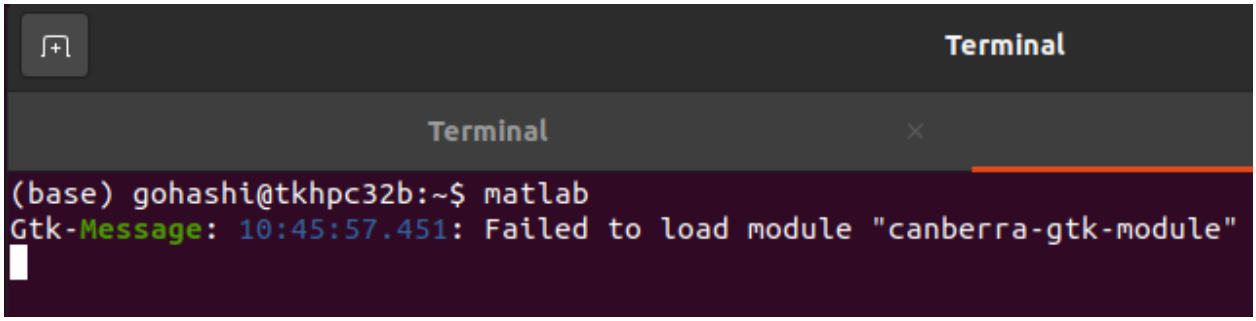
## Using a local Computer

1. Log into a Linux/Ubuntu/Unix computer connected to the Harvard network. You should have your own account and password sent from sbgrid
2. Check that there is slurm in your network. Open terminal and execute the command “sview”. If successful, a new window should pop-up.

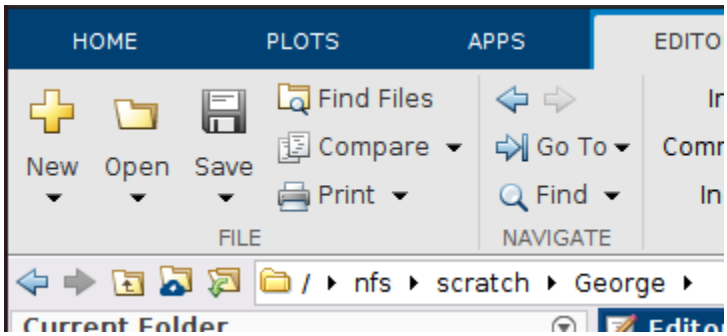


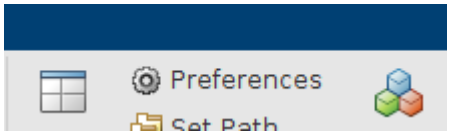
3. If you (user) do not have the SlurmProfile configured in your Matlab, please follow steps below. Otherwise proceed to step 15.

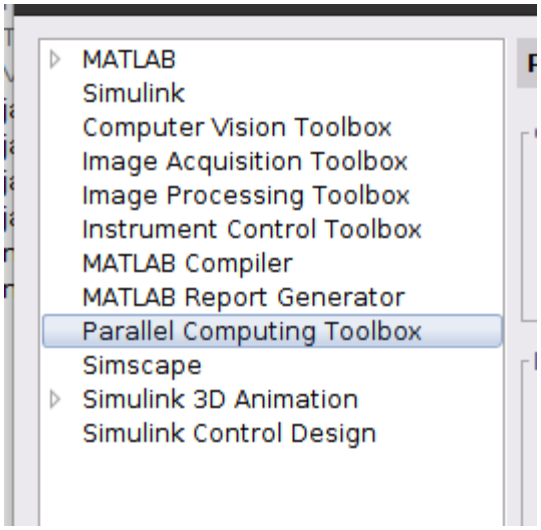
4. On terminal, click the new tab icon (  ) or “ctrl + t” to open a new tab. If you are remotely connecting, make sure that you are connected to the desired remote computer. If your new tab loads the terminal of your local computer, follow “Connecting to workstation” protocol above and connect to the remote workstation
5. Execute the command “matlab”. Matlab window should pop up



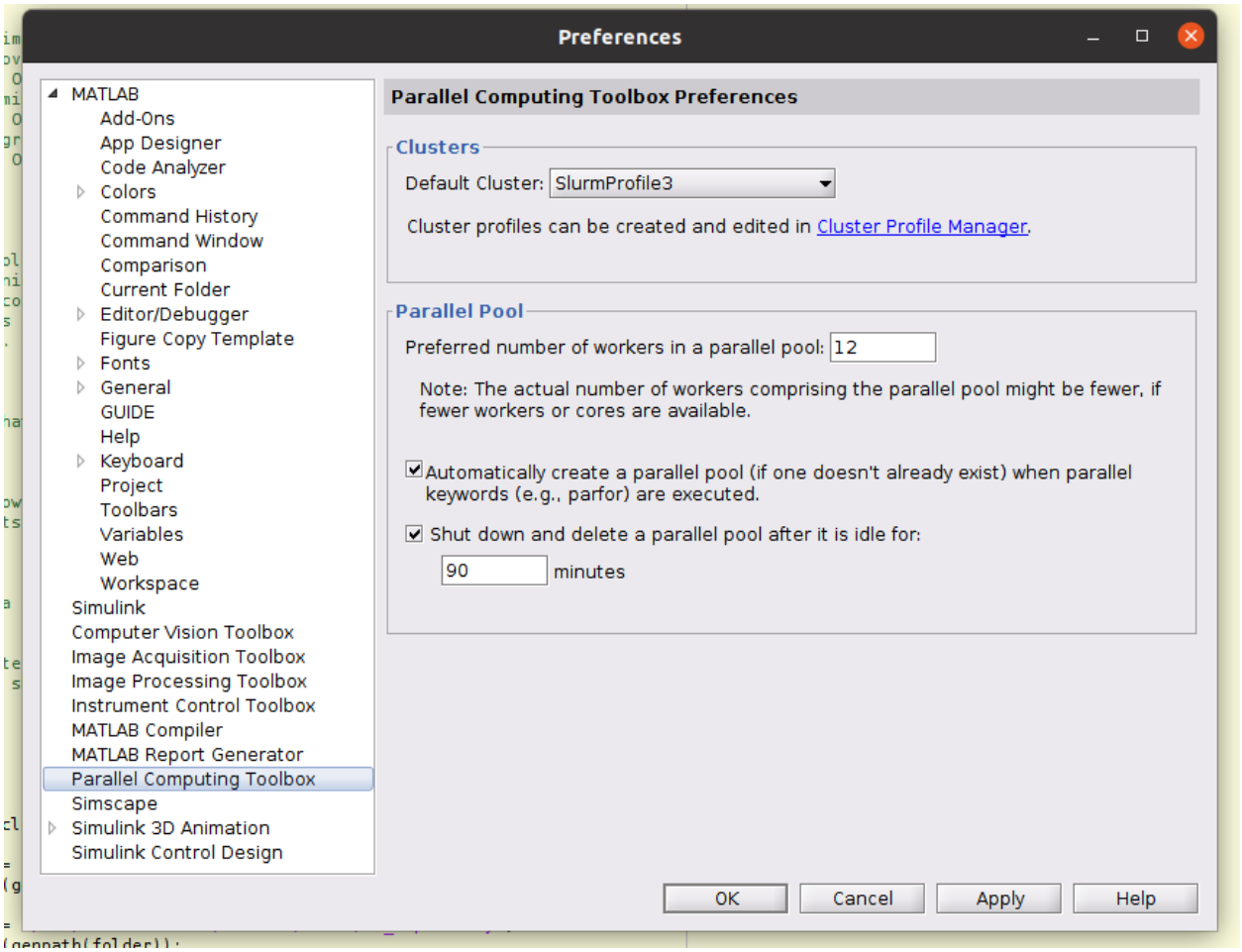

6. On matlab, use the file explorer and navigate to your scratch directory

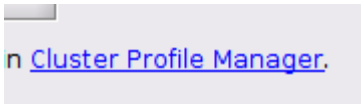


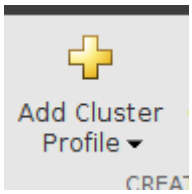

7. On matlab, go to Home (  ) > preferences (  ) >

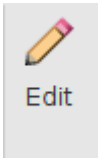
Parallel Computing Toolbox ( 

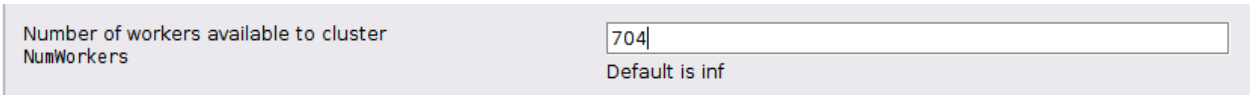
)

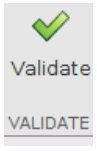


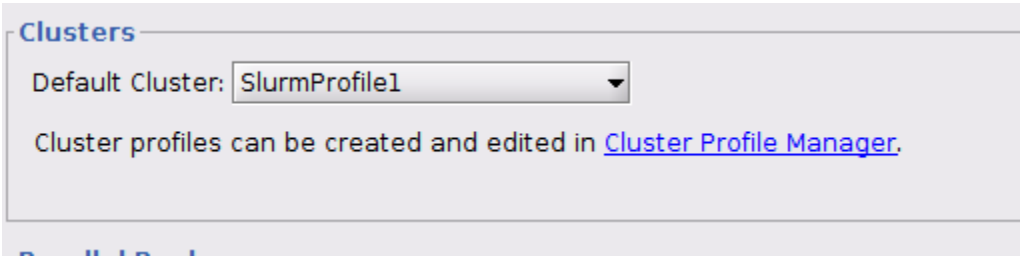
8. Click on cluster profile manager ( 

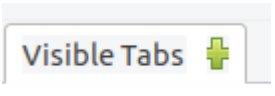
)
9. Click add cluster profile ( 

) > slurm
    a. If you see a pop-up, just click Ok. You should have the Matlab parallel server
10. A new cluster profile is now added (SlurmProfile1). Click on the cluster and then click edit ( 

)
11. Under “Number of workers available to cluster NumWorkers”, enter 704 ( 

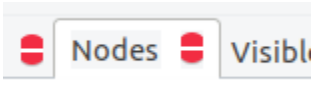
)
12. Click done. Make sure you are in a directory with write permissions (e.g. your scratch directory). Then click validate ( 

). This can take from minutes to hours.
    a. If the validation is taking 5+ minutes and you passed the job test (createJob), you should be good to go. You can stop.
13. Close the set-up window and on the preferences window, change the default cluster to the SlurmProfile you created. In my case, it is SlurmProfile1 ( 

)
14. Click Apply, and then close the window
15. On matlab, under command window, type in parpool(N) where N is the number of workers you wish to have (max of 704).
    a. To check the number of available workers, check the sview window (from step 2)
        i. On sview, click visible tabs ( 

) >

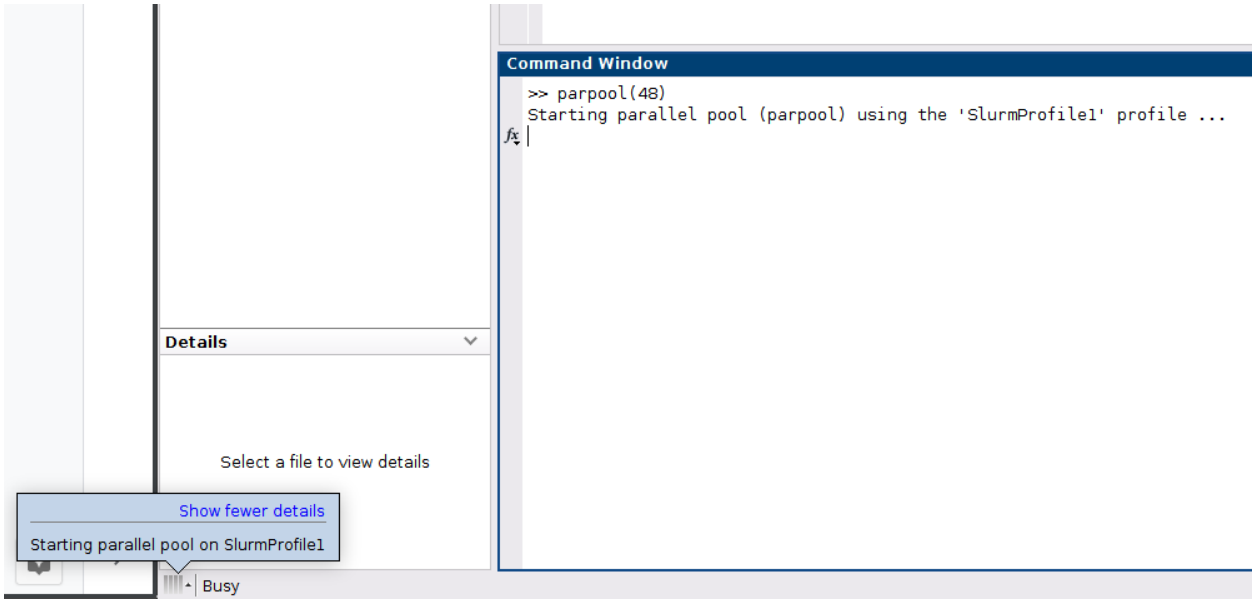
Nodes. You generated a new tab called Nodes. Click on this tab (



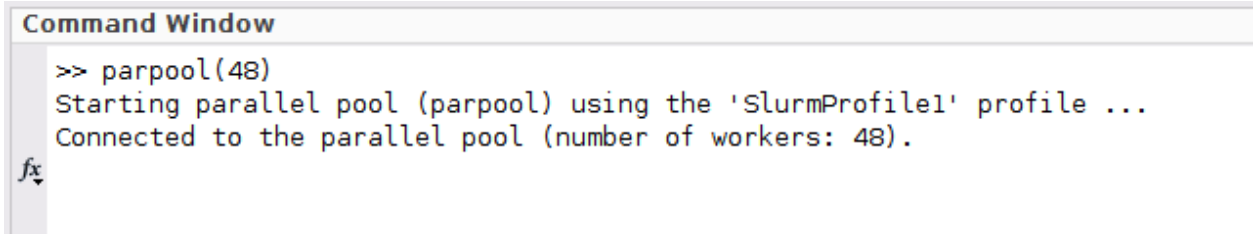

)

1. You should be able to see the workers that are busy and workers that are not busy

b. If someone is using some workers and you need them, your program will wait until someone stops using those workers.
c. If there are enough workers for your request, then matlab will check the connection and configures the nodes for your reservation. You will see the reservation of the nodes will be ready in roughly 1 min to 15 min.

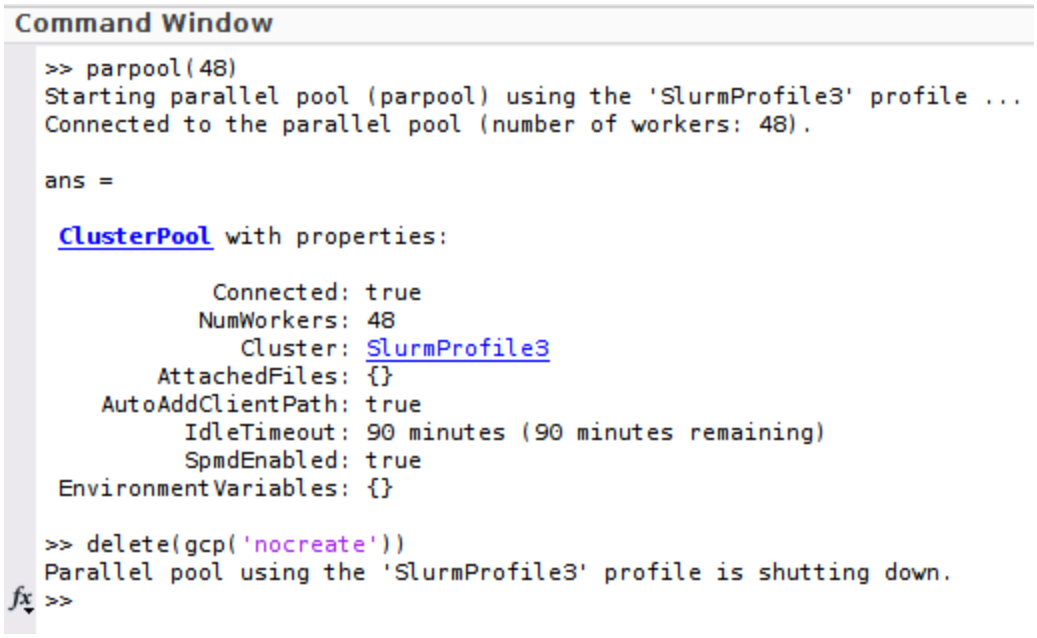



i. Once successful, you will see the following




~~Now you can run matlab programs using the cluster.~~

Once the above message is shown, then you have validated that the number of requested workers can be reserved. Execute (copy/paste) “delete(gcp('nocreate'))” to terminate your reservation.




## Basics

Cluster

- ~~We have 8 tkv computers, with 24 CPUs each, each with 2 threads = 384 workers~~
  - ~~Basically, threads are the ones that will take over matlab submission jobs~~
- As of today (June 16th, we have 704 workers)

Slurm

- A job scheduling software that will combine workers from one machine to another machine into a pool.

Matlab parallel computing

- Matlab connects to the cluster profile you created (SlurmProfile1), servers as the client to slurm (server), and then slurm will reserve resources for the request (allocate CPU), sends the signal back to Matlab saying ok, request complete, or no, can't do
- Once matlab gets the request from the server, then those allocated CPUs are locked, and no other process can take over those nodes until you end the matlab program (close the program) or by default 30min