Cyberinfrastructure Foundations
CSCI-830

Instructions for using RIT research Computing cluster for OpenMP, OpenMPI and CUDA C programs.

SECTION 1 : Accessing RIT research computing cluster

This section is a brief guide on connecting with RIT research computing cluster. More comprehensive information on the cluster and middleware (SLURM) can be found on the research computing's getting started link: https://wiki.rit.edu/display/rc/Getting+Started

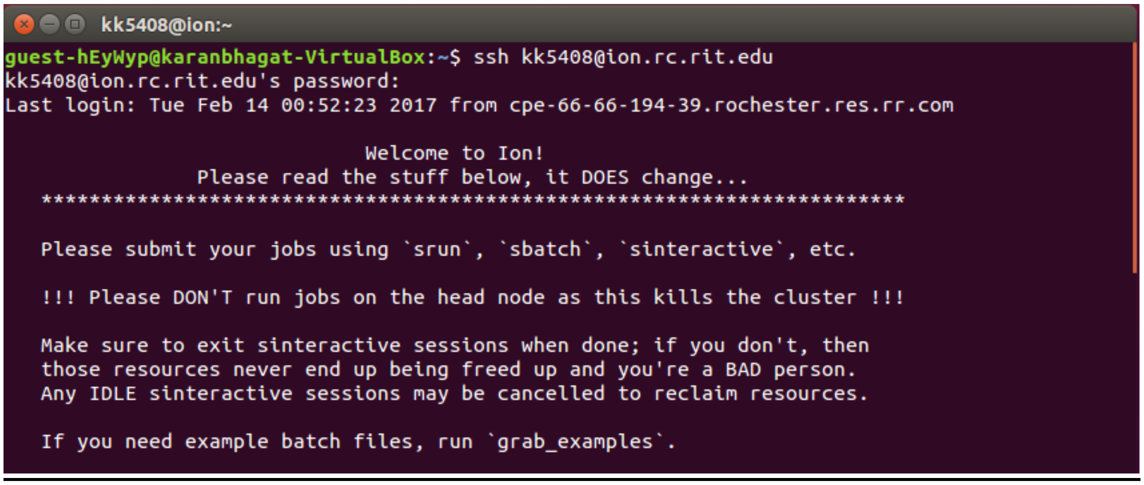
1.1 Apply for a RIT research computing cluster account.

- Apply here : http://rc.rit.edu/apply

1.2 Connecting with RIT research computing cluster using SSH.

- SSH on Linux:
  - Open terminal/command-line in any Linux distribution.
  - Start SSH session with ion.rc.rit.edu host using:

    ssh <RIT userid>@ion.rc.rit.edu
  - Enter the password associated with RIT user Id when prompted.
  - Once logged in, you should be able to see similar text as in following screenshot.



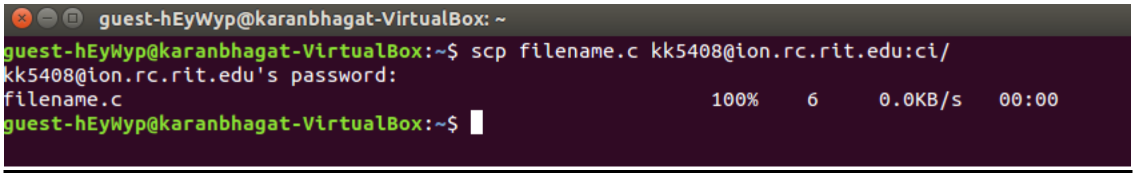
```
guest-hEyWyp@karanbhagat-VirtualBox:~$ ssh kk5408@ion.rc.rit.edu
kk5408@ion.rc.rit.edu's password:
Last login: Tue Feb 14 00:52:23 2017 from cpe-66-66-194-39.rochester.res.rr.com

                              Welcome to Ion!
                Please read the stuff below, it DOES change...
   ************************************************************************

   Please submit your jobs using `srun`, `sbatch`, `sinteractive`, etc.

   !!! Please DON'T run jobs on the head node as this kills the cluster !!!

   Make sure to exit sinteractive sessions when done; if you don't, then
   those resources never end up being freed up and you're a BAD person.
   Any IDLE sinteractive sessions may be cancelled to reclaim resources.

   If you need example batch files, run `grab_examples`.
```

- SSH on Windows:
  - Download MobaXterm for window from here: *http://mobaxterm.mobatek.net/download-home-edition.html*
  - After installing MobaXterm, start a SSH session with ion.rc.rit.edu using *session* icon on left of the tool bar.

1.3 Copying files to RIT research computing cluster using SCP.

- SCP on Linux:
  - Open terminal/command-line in any Linux distribution.
  - Use following command to copy file from local to remote host. CI is a directory under home directory. CI is just an example and would not necessarily present under your home directory.

    scp filename.c <RIT user Id>@ion.rc.rit.edu:CI/
  - Enter the password associated with RIT user Id when prompted.
  - You should see following output if the file successfully copied to remote host.



```
guest-hEyWyp@karanbhagat-VirtualBox:~$ scp filename.c kk5408@ion.rc.rit.edu:ci/
kk5408@ion.rc.rit.edu's password:
filename.c                                            100%    6     0.0KB/s   00:00
guest-hEyWyp@karanbhagat-VirtualBox:~$ 
```

- SCP on Windows
  - Download WinSCP for windows using following link: https://winscp.net/eng/download.php
  - In the login window, select *New Site* menu item from the left panel. Enter the hostname as ion.rc.rit.edu, and RIT user Id and password.
  - This will open a local and a remote host's file explorer view in side by side arranged panels.

SECTION 2 : Running programs on the RIT research cluster

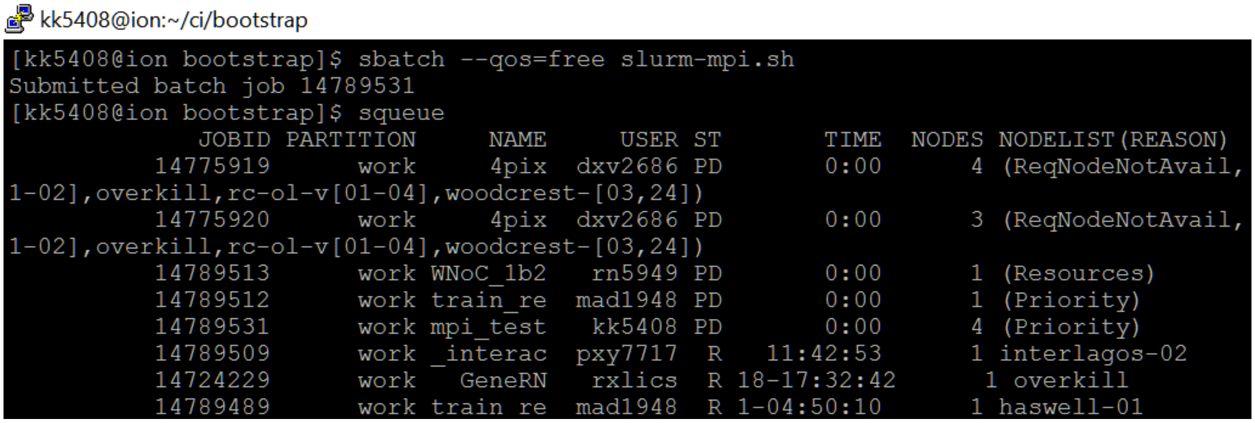
The SLURM middleware for the cluster accepts jobs submitted using sbatch shell scripts. SBatch shell scripts contain instructions for the SLURM. Every program should be run using sbatch shell scripts. Please DO NOT run any program directly on the master node which is ion.rc.rit.edu.

All the instructions under this section uses sbatch files for running all kind of programs. These sbatch files are created by tailoring example sbatch files available on RIT research computing's remote host. SSH into ion.rc.rit.edu and use command *'grab-examples'* to create a directory with example sbatch files under your home directory.

All the sbatch shell scripts, C files and CUDA files used in the following sub sections can be downloaded from the following link: https://www.cs.rit.edu/~jmk/cisc830/rcinstructions/programs.html

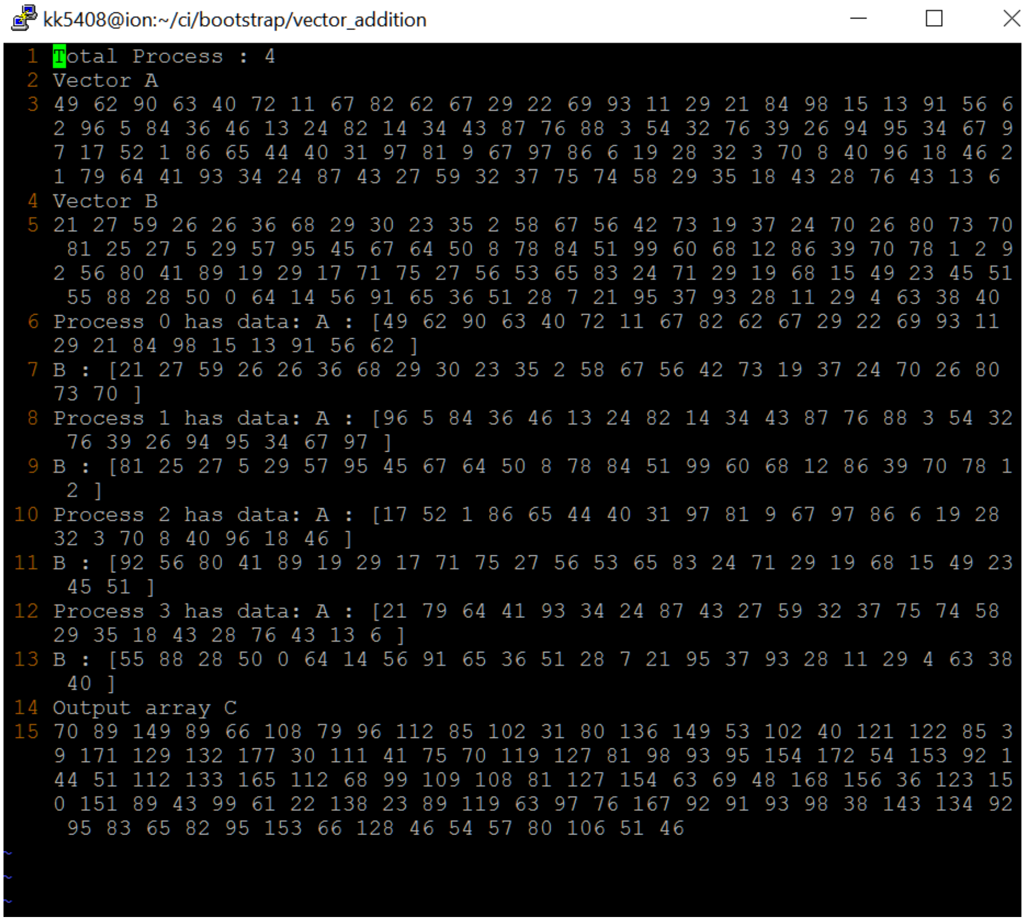
2.1 Multi node program using C and OpenMPI.

- Copy slurm-vector-addition-mpi.sh and mpi_vector_addition.c files to the remote host (ion.rc.rit.edu) using SCP (as discussed in section 1.3).
- SSH into the remote host and use the following command to start the program:
  sbatch --qos=free slurm-vector-addition-mpi.sh
- Use command *'squeue'* to check if your job is scheduled. Look for your RIT user ID under "USER" column of the output. Output of *'squeue'* should be similar to the following screenshot.



```
[kk5408@ion bootstrap]$ sbatch --qos=free slurm-mpi.sh
Submitted batch job 14789531
[kk5408@ion bootstrap]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          14775919      work     4pix  dxv2686 PD       0:00      4 (ReqNodeNotAvail,
1-02],overkill,rc-ol-v[01-04],woodcrest-[03,24])
          14775920      work     4pix  dxv2686 PD       0:00      3 (ReqNodeNotAvail,
1-02],overkill,rc-ol-v[01-04],woodcrest-[03,24])
          14789513      work WNoC_1b2   rn5949 PD       0:00      1 (Resources)
          14789512      work train_re  mad1948 PD       0:00      1 (Priority)
          14789531      work mpi_test   kk5408 PD       0:00      4 (Priority)
          14789509      work _interac  pxy7717  R   11:42:53      1 interlagos-02
          14724229      work   GeneRN   rxlics  R 18-17:32:42      1 overkill
          14789489      work train re  mad1948  R 1-04:50:10      1 haswell-01
```

- Use *'squeue'* to see if the job is running or is done. If the job is not listed under *'squeue'* anymore that means the job is completed.
- For slurm-mpi.sh sbatch file, output of the job can be found in mpi_test.output file and the error output in mpi_test_error.output file. You should be able to see following output under mpi_test.output file. Note: names of the nodes and ranks of the tasks can be different for each run of the program.

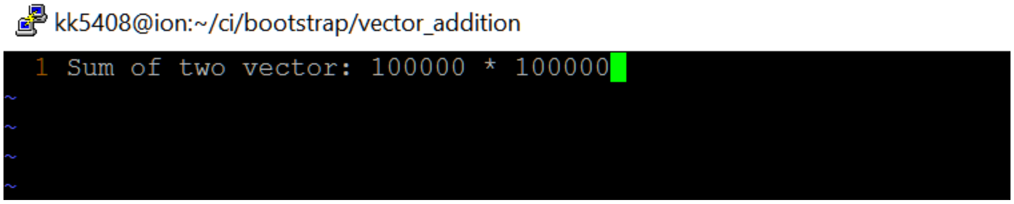


```
Total Process : 4
Vector A
49 62 90 63 40 72 11 67 82 62 67 29 22 69 93 11 29 21 84 98 15 13 91 56 6
2 96 5 84 36 46 13 24 82 14 34 43 87 76 88 3 54 32 76 39 26 94 95 34 67 9
7 17 52 1 86 65 44 40 31 97 81 9 67 97 86 6 19 28 32 3 70 8 40 96 18 46 2
1 79 64 41 93 34 24 87 43 27 59 32 37 75 74 58 29 35 18 43 28 76 43 13 6
Vector B
21 27 59 26 26 36 68 29 30 23 35 2 58 67 56 42 73 19 37 24 70 26 80 73 70
 81 25 27 5 29 57 95 45 67 64 50 8 78 84 51 99 60 68 12 86 39 70 78 1 2 9
2 56 80 41 89 19 29 17 71 75 27 56 53 65 83 24 71 29 19 68 15 49 23 45 51
 55 88 28 50 0 64 14 56 91 65 36 51 28 7 21 95 37 93 28 11 29 4 63 38 40
Process 0 has data: A : [49 62 90 63 40 72 11 67 82 62 67 29 22 69 93 11
29 21 84 98 15 13 91 56 62 ]
B : [21 27 59 26 26 36 68 29 30 23 35 2 58 67 56 42 73 19 37 24 70 26 80
73 70 ]
Process 1 has data: A : [96 5 84 36 46 13 24 82 14 34 43 87 76 88 3 54 32
 76 39 26 94 95 34 67 97 ]
B : [81 25 27 5 29 57 95 45 67 64 50 8 78 84 51 99 60 68 12 86 39 70 78 1
 2 ]
Process 2 has data: A : [17 52 1 86 65 44 40 31 97 81 9 67 97 86 6 19 28
32 3 70 8 40 96 18 46 ]
B : [92 56 80 41 89 19 29 17 71 75 27 56 53 65 83 24 71 29 19 68 15 49 23
 45 51 ]
Process 3 has data: A : [21 79 64 41 93 34 24 87 43 27 59 32 37 75 74 58
29 35 18 43 28 76 43 13 6 ]
B : [55 88 28 50 0 64 14 56 91 65 36 51 28 7 21 95 37 93 28 11 29 4 63 38
 40 ]
Output array C
70 89 149 89 66 108 79 96 112 85 102 31 80 136 149 53 102 40 121 122 85 3
9 171 129 132 177 30 111 41 75 70 119 127 81 98 93 95 154 172 54 153 92 1
44 51 112 133 165 112 68 99 109 108 81 127 154 63 69 48 168 156 36 123 15
0 151 89 43 99 61 22 138 23 89 119 63 97 76 167 92 91 93 98 38 143 134 92
 95 83 65 82 95 153 66 128 46 54 57 80 106 51 46
~
~
~
```

2.2 GPU program using C and CUDA.

- Copy slurm-vector-addition.sh and vector_addition.cu files to the remote host (ion.rc.rit.edu) using SCP (as discussed in section 1.3).
- SSH into the remote host and use the following command to start the program.
  sbatch --gres=gpu slurm-vector-addition-gpu.sh

- Use command *'squeue'* to check if your job is scheduled. Look for your RIT user ID under "USER" column of the output.
- Check for output in gpu_test.output file. Output should be similar to the following screen shot.

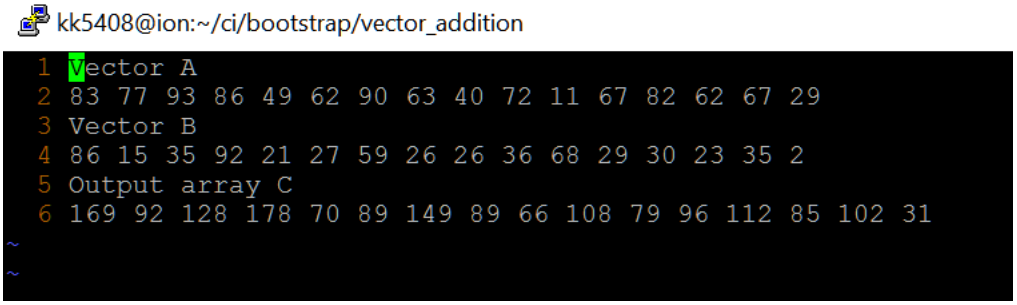


kk5408@ion:~/ci/bootstrap/vector_addition

```
Sum of two vector: 100000 * 100000
~
~
~
~
```

2.3 Multi-core program using C and OpenMP

- Copy slurm-vector-addition-omp.sh and omp_vector_addition.c files to the remote host (ion.rc.rit.edu) using SCP (as discussed in section 1.3).
- Use the following command on remote host to start the program.
  sbatch --qos=free slurm-vector-addition-omp.sh
- Use command *'squeue'* to check if your job is scheduled. Look for your RIT user ID under "USER" column of the output.
- Check for output in omp_test.output file. Output should be similar to the following screen shot. Note: order of the threads' numbers can be different for each run of the program.



kk5408@ion:~/ci/bootstrap/vector_addition

```
Vector A
83 77 93 86 49 62 90 63 40 72 11 67 82 62 67 29
Vector B
86 15 35 92 21 27 59 26 26 36 68 29 30 23 35 2
Output array C
169 92 128 178 70 89 149 89 66 108 79 96 112 85 102 31
~
~
```

SECTION 3 : Further Readings

This document discusses instructions specifically for running OpenMP, OpenMPI and CUDA programs. If interested, other useful information and tutorials regarding the SLURM can be found using following links:

- SLURM commands:

https://rc.fas.harvard.edu/resources/documentation/convenient-slurm-commands/

Further Information on accessing RIT research cluster

- Connecting using SSH https://wiki.rit.edu/display/rc/Connecting+to+Research+Computing+Systems+with+SSH
- File Management https://wiki.rit.edu/display/rc/File+Management+-+Getting+your+files+to+and+from+RC+systems