User Manual

for

Programming Assignment 1

Manthan Patel
(A-20413535)
mpatel120@hawk.iit.edu

Index

1. CPU Benchmark

This benchmark has been developed in C language and used POSIX library for threading. The purpose of this benchmark is to test the cpu speed for very large number of arithmetic operations.

Commands for compilation and running project

To compile the .c file on terminal the command is -

gcc -Wall MyCPUBench.c -o MyCPUBench -lpthread

and for running after successful compilation, the code for execution is -

./**MyCPUBench <*.dat file name >**
or
./MyCPUBench

(I have kept an else part for testing constant values in main method)

Using Makefile

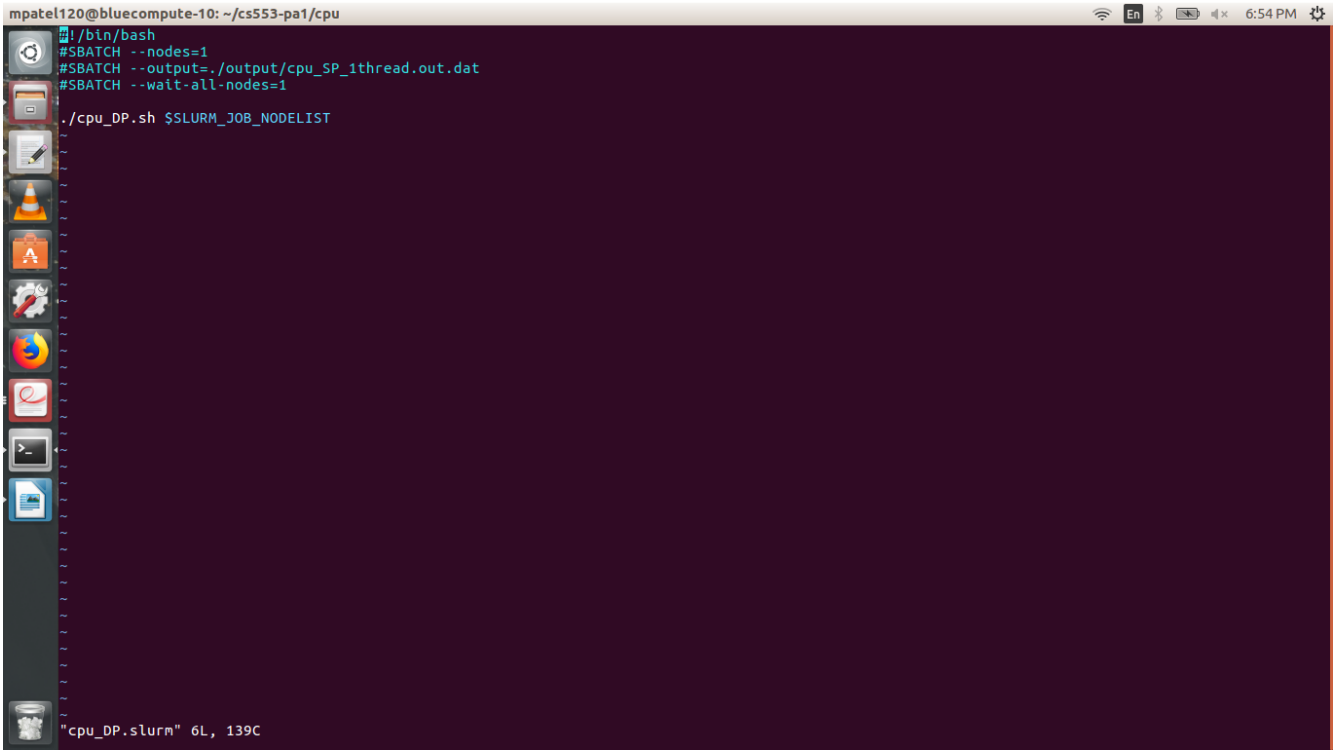
Use "**make all**" command in terminal and it will remove all .o files and executable file **MyCPUBench** . And then it will compile the code and create new **MyCPUBench** executable file.

Note: The makefile was not deleting or not creating a correct executable file. Therefore, I have made changes in makefile. Still if the output dose not come in appropriate form use the command line compile command from above.

SLURM Job command

The command for slurm job is -

sbatch <.slurm file name>



(SLURM file for CPU DP code)

I have written one "run_all_cpus.sh" file having command for all the batch files to run.

2. Disk Benchmark

This benchmark has been developed in C language with the help of POSIX threads for multi threading. This benchmark is for testing the speed for disk operations(read-write sequential/random) and the disk latency.

Commands for compilation and running project

To compile the .c file on terminal the command is -

gcc -Wall -o MyDiskBench MyDiskBench.c -lpthread

and for running after successful compilation, the code for execution is -

./ **MyDiskBench <*.dat file name >**
or
./ MyDiskBench

(I have kept an else part for testing constant values in main method)

Using Makefile

Use "**make all**" command in terminal and it will remove all .o files and executable file **MyDiskBench** . And then it will compile the code and create new **MyDiskBench** executable file.

Note: If the makefile dose not help in getting proper output. Please try to remove MyDiskBench executable and run the above compilation command.

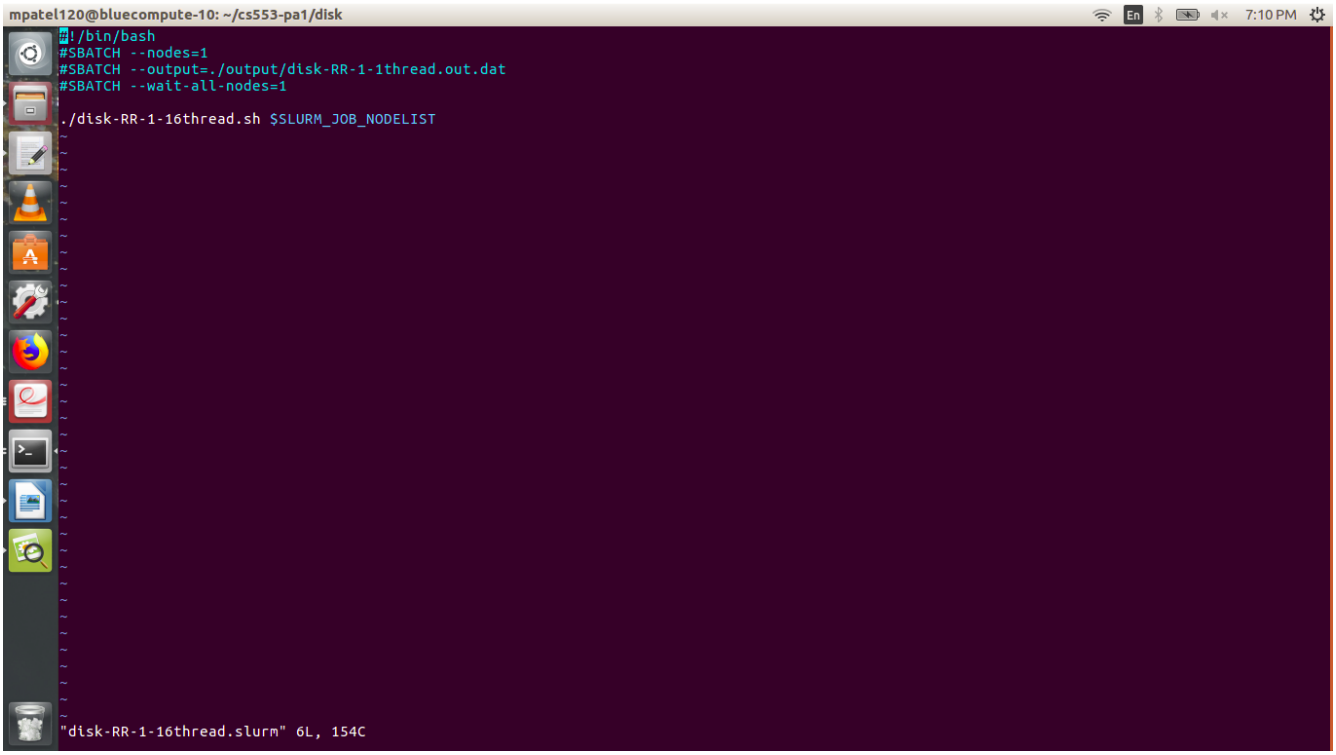
SLURM Job command

The command for slurm job is -

sbatch <.slurm file name>

for example:

sbatch disk-RR-1-1thread.slurm



(SLURM file - disk-RR-1-16thread.slurm **)**

I have written one "run_all_jobs.sh" file having command for all the batch files to run.

3. Memory Benchmark

I have implemented this benchmark to test the RAM read/write operation speed using different block size (1KB, 32KB,etc.). This benchmark is written in C language with POSIX library for threading.

Commands for compilation and running project

To compile the .c file on terminal the command is -

gcc -Wall MyRAMBench.c -pthread -o MyRAMBench

and for running after successful compilation, the code for execution is -

./ **MyRAMBench <*.dat file name >**
or
./ MyRAMBench

(I have kept an else part for testing constant values in main method to reduce command line arguments)

Using Makefile

Use "**make all**" command in terminal and it will remove all .o files and executable file **MyRAMBench** . And then it will compile the code and create new **MyRAMBench** executable file.

Note: If the makefile dose not help in getting proper output. Please try to remove MyRAMBench executable and run the above compilation command.

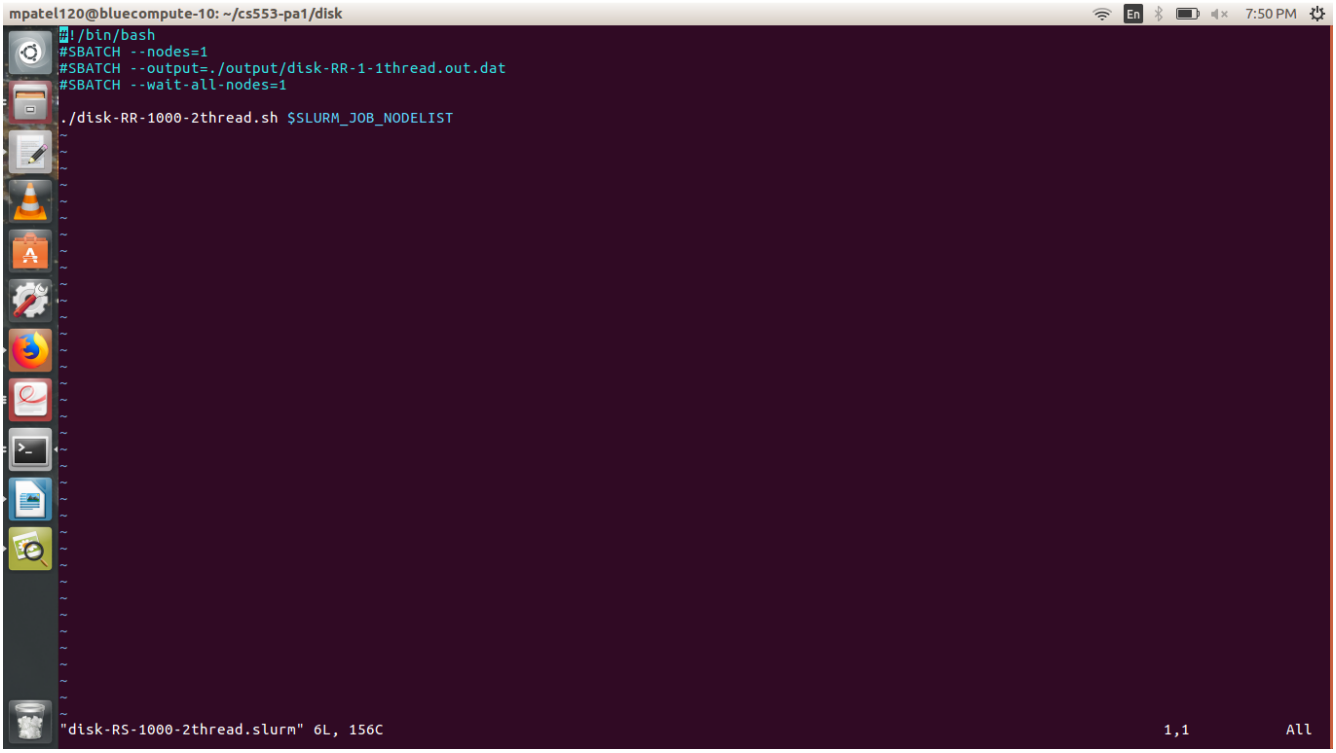
SLURM Job command

The command for slurm job is -

sbatch <.slurm file name>

for example:

sbatch memory-RWR-1.slurm



(SLURM FILE - disk-RS-1000-2thread.slurm)

I have included one **"run_all_batch.sh"** file which has all the commands for job creation. It is executable,too.

4. Network Benchmark

This benchmark is used for testing of network speed and latency of round trip time (RTT) for a ping to given host. This benchmark has been written in C language and has included socket programming (using both UDP and TCP protocols) and POSIX library for multi threading.

Network Benchmark has main two part: client and server for which there will be a requirement of 2 concurrent nodes for each(client and server). Therefor, this benchmark runs a little differently. Also, the job creation will use 2 tasks and 2 nodes for running the benchmark.

A) TCP Network Benchmark

Commands for compilation and running project

To compile the .c file on terminal the command is -

gcc -Wall -o MyNETBench-TCP MyNETBench-TCP.c -lpthread

After successful compilation, the code for execution on one terminal or node is -

./ **MyNETBench-TCP server <*.dat file name >**

and on other node or terminal

./ **MyNETBench-TCP client <*.dat file name > localhost**

(There will be total 4 command line arguments(including ./**MyNETBench-TCP**) for client and 3 for server.)

Using Makefile

Use "**make all**" command in terminal and it will remove all .o files and executable file **MyNETBench-TCP**. And then it will compile the code and create new MyNETBench-TCP executable file.

Note: If the makefile dose not help in getting proper output. Please try to remove MyNETBench-TCP executable and run the above compilation command.

SLURM Job command

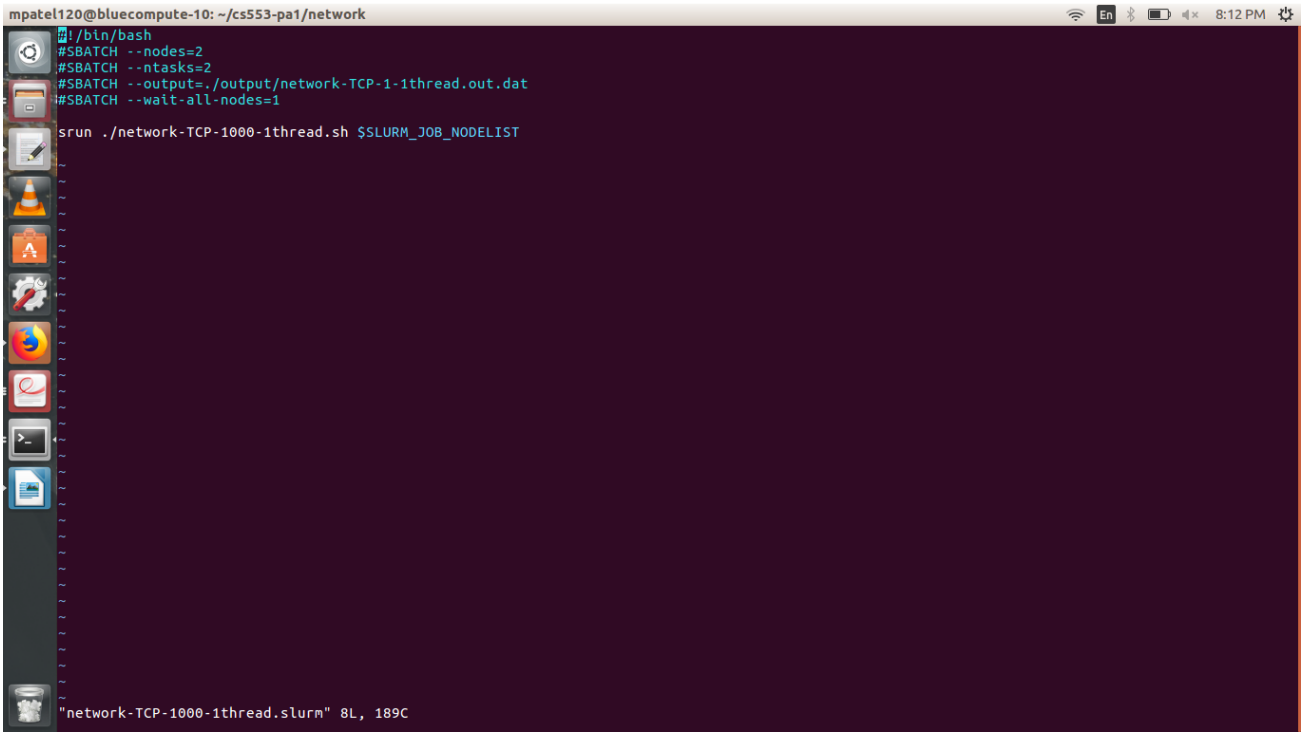
The command for slurm job is -

sbatch <.slurm file name>

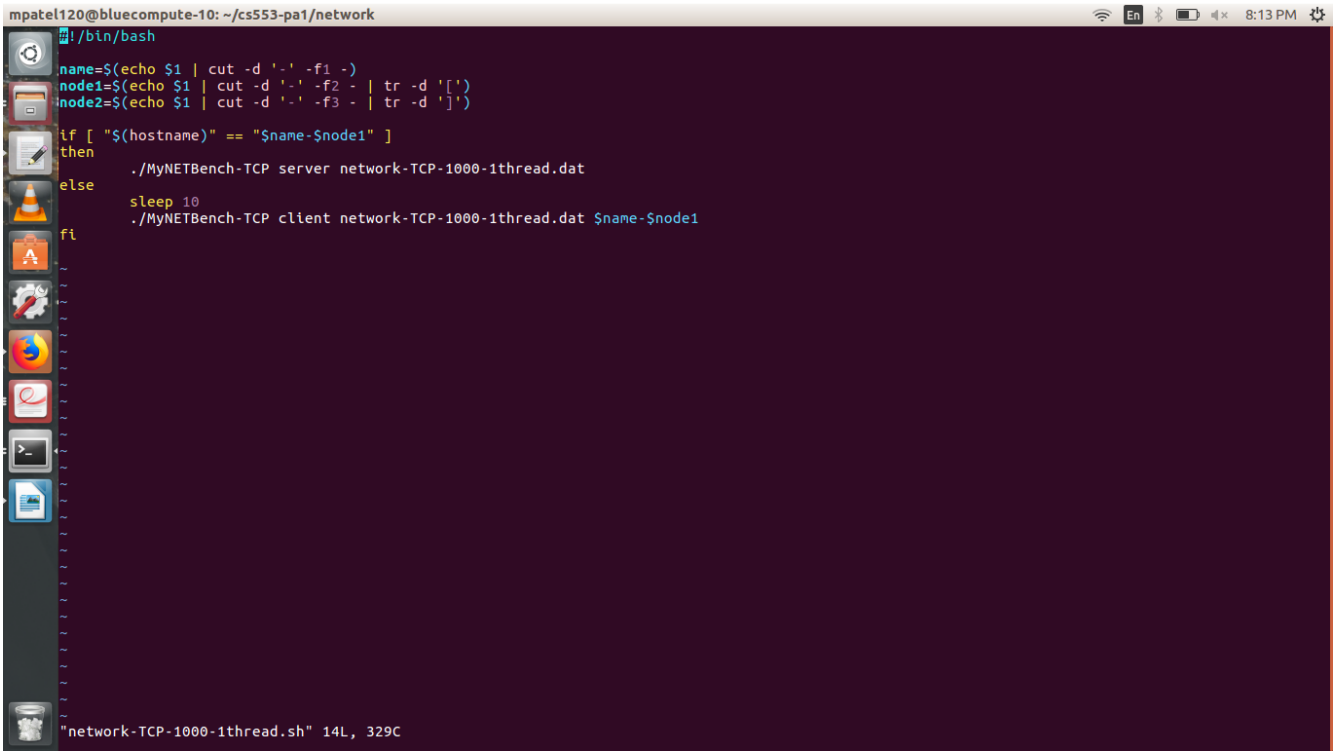
for example:

sbatch network-TCP-32000-1thread.slurm

The SLURM file and .sh files are different as these files will request 2 nodes for running server and client on each node. The code for SLURM file is given below -



The code for .sh file can be written as below -



Also, for simplification I have made one command runAllBatch.sh file which has included all TCP and UDP batch assignment command.

B) UDP Network Benchmark

Commands for compilation and running project

To compile the .c file on terminal the command is -

gcc -Wall -o MyNETBench-UDP MyNETBench-UDP .c -lpthread

After successful compilation, the code for execution on one terminal or node is -

./ **MyNETBench- UDP server <*.dat file name >**

and on other node or terminal

./ **MyNETBench- UDP client <*.dat file name > localhost**

(There will be total 4 command line arguments(including ./**MyNETBench-UDP**) for client and 3 for server.)

Using Makefile

Use "**make all**" command in terminal and it will remove all .o files and executable file **MyNETBench-UDP** . And then it will compile the code and create new MyNETBench-TCP executable file.

Note: If the makefile dose not help in getting proper output. Please try to remove MyNETBench- UDP executable and run the above compilation command.

SLURM Job command

The command for slurm job is -

sbatch <.slurm file name>

for example:

sbatch network-UDP-32000-1thread.slurm

The SLURM file and .sh files are different as these files will request 2 nodes for running server and client on each node. The code for SLURM and .sh files is same as the code given for TCP SLURM and .sh files.