

Manual

- The purpose of this document is to enable any user to compile and execute the source code easily.
- It provides step by step information of how to execute the programs.

1. CPU:

- a. This code is written in c language.
- b. This benchmark is implemented in c language. The steps that need to be followed to run program is:
 - i. Navigate to cpu folder.
 - ii. Execute following command
 1. **bash run1.sh** (If you have don't have any restrictions on submitting the number of jobs than this command will execute all the operations)
 2. If you have restrictions on number of jobs as 10, than use this,
 - a. bash run1.sh
 - b. bash run2.sh
 3. This will compile and also executed by submitting the Slurm jobs.
 4. To view the output, navigate to "**cpu_SP_1thread.out.dat**" file inside the **output** folder of cpu.
 - iii. For **Linpack** benchmark,
 1. Execute:
 - a. **bash runLinpack1.sh** – It will execute the linpack benchmark for number of threads as 1,2 and 4 on different compute nodes.
 - b. This will generate an slurm output file where the program output can be viewed.

2. Memory:

- a. This code is written in c language.
- b. To run the program,
 - i. Navigate to memory folder.
 - ii. Execute the following command:
 1. **bash run1.sh** (If you have don't have any restrictions on submitting the number of jobs than this command will execute all the operations)
 2. If you have restrictions on number of jobs as 10, then use this,
 - a. bash run1.sh
 - b. bash run2.sh
 - c. bash run3.sh

3. This will compile and gets executed, by submitting the Slurm jobs.
 4. To view the output, navigate to “**memory-RWR-1-1thread.out.dat**” file inside the **output** folder of memory.
 5. In case of **latency**, the MyRAMBenchValue, TheoreticalValue and MyRAMBenchEfficiency which are present inside the file will reflect MyRamBenchLatency, TheoreticalLatency and MyRamBenchLatencyEfficiency respectively.
- iii. For Running **pmbw** benchmark, there are two ways:
1. Using Bash script.
 - a. Navigate to ‘cs553-pa1/bm/ pmbwBenchmark’ folder
 - b. Execute following command : **bash run1.sh**
 - c. This will generate all the required slurm output files and look for values of
 - i. Bandwidth as ‘Theoretical Throughput’.
 - ii. Rate as ‘latency’
 2. **Login to the compute / interactive node** and execute the below commands:
 - a. `srun -n 1 -p interactive --pty /bin/bash`
 - b. `srun -n 1 -p compute --pty /bin/bash`
 - c. after that execute all the scripts given in mem1.slurm to mem9.slurm.
3. **Disk:**
- a. This code is written in c language.
 - b. To run the program follow the below instructions:
 - i. Navigate to disk folder.
 - ii. If you have no restrictions on number of slurm jobs you can submit, just execute
bash run1.sh
 - iii. Or else execute:
 1. `bash run1.sh` to `run6.sh`
 - iv. This script will compile the c program and also run the executables produced.
 - v. This will run all the input files and the output will be generated inside the output folder which contains ‘**disk-RR-1-1thread.out.dat**’ file.
 - c. To run the disk standard benchmark - **IOZONE**:
 - i. Login to the compute or interactive node using the below command:
 1. `srun -n 1 -p interactive --pty /bin/bash`
 2. `srun -n 1 -p compute --pty /bin/bash`
 - ii. Run the following command and note the output by changing number of threads and block sizes.
 1. `iozone -R -i 0 -i 1 -s 10g -r 1M -t 1 -u 1 -I 1 -O -I -F /tmp/text.dat`

iii. The out file will display the throughput.

4. **Network:**

a. The network code is written in JAVA to ensure that we can use inbuilt libraries provided by java for TCP and UDP.

b. To run the network code use the below instructions:

i. Navigate to network folder.

ii. If there are no restrictions on number of Slurm jobs that you can submit, then just execute:

For TCP:

bash exec1.sh

For UDP

Bash udp1.sh

iii. If there are restrictions, then run all exec1.sh to exec3.sh for TCP and udp1.sh to udp3.sh for UDP.

iv. These scripts will compile the java codes and run the executables.

v. The output will be generated inside the network folder inside '**network-TCP-1-1thread.out.dat**' file.

c. To execute Network standard – **IPERF**:

i. There are two ways for this:

1. Navigate to bm/iperf folder and execute:

a. Sbatch run1.slurm

2. The output will be shown in out file of slurm.

ii. Or use below method:

1. Login to the compute or interactive node using the below command:

a. srun -n 1 -p interactive --pty /bin/bash

b. srun -n 1 -p compute --pty /bin/bash

2. Run the server command here:

a. iperf3 -s

3. Note down the hostname where server was executed and exit from there.

4. Login to another computer node and run the client command

a. iperf3 -c \$(hostname) -P 8 -u

iii. For latency:

1. Start server on one node by login into compute node.

2. Run ping hostname.