Steps:

1. Load already uploaded modules

module load apps libs cports

1. Check available modules

module avail

1. Load modules required
   1. module load default-gcc-openmpi
   2. module load default-intel-openmpi
   3. module load libs mkl/64/11.0.074
   4. module load libs acml/64/gfortran/4.3.0
   5. module load opnempi/1.6.5-gnu4.8.2
2. List the modules loaded

module list

* + 1. cports
    2. gcc/4.8.2-gnu
    3. openmpi/1.6.5-gnu4.8.2
    4. libs
    5. mkl/64/11.0.074
    6. default-gcc-openmpi
    7. intel/12.1/composer\_xe\_2011\_sp1.9.293
    8. default-intel-openmpi
    9. acml/64/gfortran/4.3.0
    10. apps
    11. cports6
    12. intel/12.1/opnempi/64/1.6.5

1. Clean the older data
   1. make arch=lonsdale\_gcc493\_openmpi186\_acml430 clean arch all
   2. make arch=lonsdale\_gcc493\_openmpi186\_mkl11 clean arch all
   3. make arch=lonsdale\_intel12\_openmpi186\_acml430 clean arch all
   4. make arch=lonsdale\_intel12\_openmpi186\_mkl11 clean arch all
2. Create “Make.arch” files
   1. Make.lonsdale\_gcc493\_openmpi186\_acml430
   2. Make.lonsdale\_gcc493\_openmpi186\_mkl11
   3. Make.lonsdale\_intel12\_openmpi186\_acml430
   4. Make.lonsdale\_intel12\_openmpi186\_mkl11
3. Make the files
   1. make arch=lonsdale\_gcc493\_openmpi186\_acml430
   2. make arch=lonsdale\_gcc493\_openmpi186\_mkl11
   3. make arch=lonsdale\_intel12\_openmpi186\_acml430
   4. make arch=lonsdale\_intel12\_openmpi186\_mkl11
4. Check the .dat and xml files in the bin folder
   1. Four folders created inside bin folder
   2. Two files in each folder
      1. HPL.dat
      2. Xhpl
5. Test Run
   1. mpirun -np 8 xhpl
   2. Results:

T/V : Wall time / encoded variant.

N : The order of the coefficient matrix A.

NB : The partitioning blocking factor.

P : The number of process rows.

Q : The number of process columns.

Time : Time in seconds to solve the linear system.

Gflops : Rate of execution for solving the linear system.

1. Change parameters
   1. More sensitive parameters -> N, NB, P, Q
   2. N: The problem Size

Use 80% of available RAM

(8 cores x N x N) = 80% x 16GB = 12.8 x 230

* + - * N = 41448
  1. NB: The problem size

From a data distribution point of view, the smallest NB, the better the load balance. From a computation point of view, a too small value of NB may limit the computational performance by a large factor because almost no data reuse will occur in the highest level of the memory hierarchy. The number of messages will also increase. Efficient matrix-multiply routines are often internally blocked. The bottom line is that "good" block sizes are almost always in the [32...256] interval. The best values depend on the computation / communication performance ratio of your system

* 1. P x Q: The process grid

Lonsdale has 8 cores

* + - * P = 2
      * Q = 4

1. Shell scripts for running
   1. bashRunGccMKL.sh
   2. bashRunGccACML.sh
   3. bashRunIccMKL.sh
   4. bashRunIccACML.sh
2. Run for optimization
   1. Ran for various values of N, NB, P & Q
   2. Best results are shown in the below table
3. Store in output files
   1. sh bashRunGccMKL.sh >> output.txt
   2. sh bashRunGccACML.sh >> output.txt
   3. sh bashRunIccMKL.sh >> output.txt
   4. sh bashRunIccACML.sh >> output.txt
4. Relevant files/folders(created/edited) for the assignment:
   1. Makefiles edited for compilation

Make.lonsdale\_gcc493\_openmpi186\_acml430

Make.lonsdale\_gcc493\_openmpi186\_mkl11

Make.lonsdale\_intel12\_openmpi186\_acml430

Make.lonsdale\_intel12\_openmpi186\_mkl11

* 1. Will have the bash script for getting the results

bashRun.sh

* 1. Will have compilation results

bin/lonsdale\_gcc493\_openmpi186\_acml430/

bin/lonsdale\_gcc493\_openmpi186\_mkl11/

bin/lonsdale\_intel12\_openmpi186\_acml430/

bin/lonsdale\_intel12\_openmpi186\_mkl11/

* 1. Slurm files were created for each sbatch run

RESULTS:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ARCH name | lonsdale\_gcc493\_openmpi186\_acml430 | lonsdale\_gcc493\_openmpi186\_mkl11 | lonsdale\_intel12\_openmpi186\_acml430 | lonsdale\_intel12\_openmpi186\_mkl11 |
| Compiler Name | GCC | GCC | ICC | ICC |
| Compiler Name Version | 4.9.3 | 4.9.3 | 12.1 | 12.1 |
| BLAS Library Name | ACML | MKL | ACML | MKL |
| BLAS Library Name Version | 4.3.0 | 11 | 4.3.0 | 11 |
| OpenMPI Version | 1.8.6 | | | |
| T/V | WR00C2R2 | WR00L2R2 | WR00R2C4 | WR00L2L2 |
| N | 41000 | 41000 | 41472 | 41000 |
| NB | 102 | 88 | 88 | 88 |
| P | 2 | 2 | 2 | 2 |
| Q | 4 | 4 | 4 | 4 |
| Time | 826.44 | 809.94 | 796.27 | 809.67 |
| Gflops | 55.6 | 56.73 | 59.72 | 56.75 |
| File | bin/lonsdale\_gcc493\_openmpi186\_acml430/output.txt | bin/lonsdale\_gcc493\_openmpi186\_mkl11/output.txt | slurm-263190.out | slurm-263191.out |