

CSC5760 Programming Assignment 2

Rob Gillen

CSC5760 Programming Assignment 2

Pre-Requisites

Building this program requires the following to be installed on the platform:

- cmake version 3.6 or greater

Building the Code

Compiling the applications in this assignment requires the use of cmake. From the current directory, create a build directory, move into it, and then call cmake using the `CMakeLists.txt` file in the parent directory. Then, run `make` and the executable will be built into the directory. Instructions/commands are as follows:

```
$ mkdir build
$ cd build
$ cmake ..
$ make
```

This will result in two executables being built: `MatVec` and `MatVecMPI`.

Running the Code

In general, running each of the executables passing the `--help` argument should cause them to display their usage information, after which, running the application is as simple as providing the appropriate values. While not required, `MatVecMPI` is designed to be run using `mpiexec`.

Single-Threaded (sequential) Solution

The baseline for this project is provided via the `MatVec` program which is a sequential implementation of the application. While there are a handful of parameters that can be provided, the test data below was generated only providing the size parameter. This parameter defines the size of each side of the source matrix as well as the size of the source vector.

Example calls for each test:

```
$ ./MatVec --size 1024
$ ./MatVec --size 4096
$ ./MatVec --size 8192
$ ./MatVec --size 16384
```

	n=1024	n=4096	n=8192	n=16384
Answer	21,460,970	343,842,766	1,375,567,846	5,502,369,770
Run 1	7.775068e-03	4.253793e-02	1.708658e-01	7.046120e-01
Run 2	3.530025e-03	4.258704e-02	1.740370e-01	7.043641e-01
Run 3	7.662058e-03	4.253078e-02	1.756070e-01	7.032230e-01
Run 4	2.699137e-03	4.417491e-02	1.742520e-01	7.046010e-01
Run 5	2.681017e-03	4.275179e-02	1.762471e-01	7.047019e-01
Run 6	2.665997e-03	4.541397e-02	1.753070e-01	7.083530e-01
Run 7	7.786989e-03	4.433990e-02	1.761260e-01	7.011559e-01
Run 8	2.676964e-03	4.249620e-02	1.749589e-01	7.071111e-01
Run 9	3.082037e-03	4.365396e-02	1.743429e-01	7.044780e-01
Run 10	1.086497e-02	4.268694e-02	1.747971e-01	7.052510e-01
Average	5.1424262e-03	4.331734E-02	1.746541E-01	7.047851E-01

Methodology

As can be observed from the table above, each setting was run 10 times in order to flatten the effects of transient activities of the test platform on the resulting data. Results should be viewed for their trends/relative values rather than absolute measurements.

Example output from one of the runs:

```
[2017-11-26 14:04:31.243275] [0x00007fb897a67740] [info] Matrix-Vector Multiplication (Serial)
[2017-11-26 14:04:31.243320] [0x00007fb897a67740] [info] Cores available: 8
[2017-11-26 14:04:31.243426] [0x00007fb897a67740] [info] Using Seed Value: 0
[2017-11-26 14:04:31.243447] [0x00007fb897a67740] [info] Using Mod Value: 10
[2017-11-26 14:04:31.243459] [0x00007fb897a67740] [info] Using Size Value: 16384
[2017-11-26 14:04:31.243478] [0x00007fb897a67740] [info] Creating Matrix...
[2017-11-26 14:04:32.155438] [0x00007fb897a67740] [info] Creating Vector...
[2017-11-26 14:04:32.155512] [0x00007fb897a67740] [info] Performing Multiplication...
[2017-11-26 14:04:32.859932] [0x00007fb897a67740] [info] Summing Results...
[2017-11-26 14:04:32.859995] [0x00007fb897a67740] [info] Displaying Results:
[2017-11-26 14:04:32.860001] [0x00007fb897a67740] [info] Summed Result: 5502369770
[2017-11-26 14:04:32.860004] [0x00007fb897a67740] [info] Time Elapsed...
Elapsed time = 7.044780e-01 seconds
```

Multi-Threaded (MPI/Parallel) Solution

The comparison version of this project is provided via the **MatVecMPI** program which is a MPI implementation of the **MatVec** application. The parameters for both applications are the same, however the **MatVecMPI** application is intended to be run via **mpiexec** (see the examples below).

Methodology

Consistent with the serial/baseline version of the tests, each setting was run 10 times in order to flatten the effects of transient activities of the test platform on the resulting data. Results should be viewed for their trends/relative values rather than absolute measurements.

Platform

These tests were performed/data was collected on an HP Zbook G4 with a Core i7 processor. This processor consists of 4 processors. As such, it should be expected that performance (relative to the sequential version) should improve for 4 (and possibly 8 due to hyperthreading) but will flatten or degrade for node counts beyond that.

Utilizing 4 processors:

```
$ mpiexec -n 4 MatVecMPI --size 1024
$ mpiexec -n 4 MatVecMPI --size 4096
$ mpiexec -n 4 MatVecMPI --size 8192
$ mpiexec -n 4 MatVecMPI --size 16384
```

	n=1024	n=4096	n=8192	n=16384
Answer	21,460,970	343,842,766	1,375,567,846	5,502,369,770
Run 1	4.791975e-03	2.620411e-02	1.052759e-01	2.941220e-01
Run 2	3.869057e-03	2.467608e-02	7.547903e-02	2.967138e-01
Run 3	3.401995e-03	1.971817e-02	7.501006e-02	2.983410e-01
Run 4	2.606153e-03	1.913810e-02	7.516885e-02	2.970049e-01
Run 5	4.166126e-03	1.976204e-02	7.532001e-02	2.961621e-01
Run 6	3.462076e-03	2.682590e-02	7.679510e-02	2.997420e-01
Run 7	4.349947e-03	2.467489e-02	7.609320e-02	2.967389e-01
Run 8	4.874945e-03	1.980996e-02	7.475901e-02	2.973878e-01
Run 9	3.540039e-03	1.984787e-02	7.537603e-02	3.027539e-01
Run 10	4.894018e-03	2.079105e-02	7.580805e-02	2.947688e-01
Average	3.995633E-03	2.214482E-02	7.850852E-02	2.973735E-01

Utilizing 8 processors:

```
$ mpiexec -n 8 MatVecMPI --size 1024
$ mpiexec -n 8 MatVecMPI --size 4096
$ mpiexec -n 8 MatVecMPI --size 8192
$ mpiexec -n 8 MatVecMPI --size 16384
```

	n=1024	n=4096	n=8192	n=16384
Answer	21,460,970	343,842,766	1,375,567,846	5,502,369,770
Run 1	1.296997e-03	1.770091e-02	9.710503e-02	3.826990e-01

	n=1024	n=4096	n=8192	n=16384
Run 2	1.352787e-03	1.978302e-02	7.955098e-02	4.224570e-01
Run 3	1.884937e-03	1.730704e-02	7.207799e-02	3.704011e-01
Run 4	1.830101e-03	1.781511e-02	7.723904e-02	4.503889e-01
Run 5	2.014160e-03	1.700497e-02	7.835484e-02	4.579749e-01
Run 6	1.316071e-03	1.862001e-02	8.709693e-02	5.267138e-01
Run 7	1.235008e-03	1.941800e-02	7.350707e-02	3.947990e-01
Run 8	1.357079e-03	1.805711e-02	7.504487e-02	3.819201e-01
Run 9	1.336813e-03	1.973701e-02	7.033896e-02	3.824360e-01
Run 10	1.257896e-03	1.828790e-02	8.003521e-02	3.856270e-01
Average	1.488185E-03	1.837311E-02	7.903509E-02	4.155417E-01

Utilizing 16 processors:

```
$ mpiexec -n 16 MatVecMPI --size 1024
$ mpiexec -n 16 MatVecMPI --size 4096
$ mpiexec -n 16 MatVecMPI --size 8192
$ mpiexec -n 16 MatVecMPI --size 16384
```

	n=1024	n=4096	n=8192	n=16384
Answer	21,460,970	343,842,766	1,375,567,846	5,502,369,770
Run 1	3.668070e-03	3.942299e-02	9.492302e-02	3.886709e-01
Run 2	3.632092e-02	2.956414e-02	9.343815e-02	4.181468e-01
Run 3	4.865170e-03	3.959608e-02	9.364390e-02	4.262331e-01
Run 4	3.318071e-03	5.527616e-02	9.069109e-02	3.744950e-01
Run 5	2.375984e-02	4.132199e-02	1.170580e-01	4.029849e-01
Run 6	1.039505e-02	3.613806e-02	9.420800e-02	3.958762e-01
Run 7	1.078606e-02	2.775788e-02	8.911204e-02	3.986759e-01
Run 8	1.171899e-02	3.619099e-02	9.105706e-02	4.728169e-01
Run 9	7.834911e-03	3.936291e-02	9.680319e-02	4.194829e-01
Run 10	4.250050e-03	3.131604e-02	9.529305e-02	3.985529e-01
Average	1.169171E-02	3.759472E-02	9.562275E-02	4.095936E-01

Utilizing 32 processors:

```
$ mpiexec -n 32 MatVecMPI --size 1024
$ mpiexec -n 32 MatVecMPI --size 4096
$ mpiexec -n 32 MatVecMPI --size 8192
$ mpiexec -n 32 MatVecMPI --size 16384
```

	n=1024	n=4096	n=8192	n=16384
Answer	21,460,970	343,842,766	1,375,567,846	5,502,369,770
Run 1	4.726696e-02	4.016495e-02	1.238170e-01	4.492121e-01
Run 2	6.219316e-02	4.403996e-02	1.414070e-01	4.107640e-01
Run 3	3.169990e-02	4.787207e-02	1.105299e-01	3.979402e-01
Run 4	5.989075e-03	5.027103e-02	1.203640e-01	4.186370e-01
Run 5	7.196116e-02	3.658009e-02	1.288509e-01	3.909190e-01
Run 6	5.178213e-02	5.688119e-02	1.359520e-01	4.281290e-01
Run 7	5.483198e-02	4.332900e-02	1.311030e-01	4.589050e-01
Run 8	5.800962e-03	6.518984e-02	1.065428e-01	4.310610e-01

	n=1024	n=4096	n=8192	n=16384
Run 9	1.130319e-02	4.953098e-02	1.455572e-01	4.138210e-01
Run 10	3.220296e-02	4.215622e-02	1.232371e-01	4.305971e-01
Average	3.750315E-02	4.760153E-02	1.267361E-01	4.229985E-01

Utilizing 64 processors:

```
$ mpiexec -n 64 MatVecMPI --size 1024
$ mpiexec -n 64 MatVecMPI --size 4096
$ mpiexec -n 64 MatVecMPI --size 8192
$ mpiexec -n 64 MatVecMPI --size 16384
```

	n=1024	n=4096	n=8192	n=16384
Answer	21,460,970	343,842,766	1,375,567,846	5,502,369,770
Run 1	2.765393e-02	4.703498e-02	1.628301e-01	3.405440e-01
Run 2	1.354098e-02	6.280994e-02	1.102860e-01	3.664508e-01
Run 3	3.793907e-02	4.689598e-02	1.065559e-01	3.615880e-01
Run 4	3.970885e-02	3.945994e-02	1.414380e-01	3.310840e-01
Run 5	9.701705e-02	5.898190e-02	1.398430e-01	3.563850e-01
Run 6	5.897307e-02	1.090159e-01	1.314900e-01	3.894501e-01
Run 7	1.120930e-01	3.284907e-02	1.342032e-01	3.290040e-01
Run 8	1.539302e-02	5.396295e-02	1.339238e-01	3.209240e-01
Run 9	6.861186e-02	4.428101e-02	1.357501e-01	3.478329e-01
Run 10	7.886600e-02	3.568697e-02	1.308761e-01	3.613269e-01
Average	5.497968E-02	5.309786E-02	1.327196E-01	3.504590E-01

Discussion

As can be seen from the results...