EEL 6763 PCA - Spring 2022

Department of Electrical and Computer Engineering

Lab 5 - 3/31/2022

Mohit Palliyil Sathyaseelan, Vishnu V

Part 1: Profiling CMT-bone-BE

(a) Use gprof to profile the CMT-bone-BE application and paste the obtained output from gprof in your report for the following command line arguments:

Case 1:

Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self us/call	total us/call	name
34.70	0.17	0.17	96000	1.77	1.77	operation_conv
20.41	0.27	0.10	96000	1.04	1.04	operation_ds
18.37	0.36	0.09	96000	0.94	0.94	operation_dr
10.20	0.41	0.05	96000	0.52	0.52	operation_dt
10.20	0.46	0.05	96000	0.52	0.52	operation_sum
4.08	0.48	0.02	96000	0.21	0.21	operation_rk
2.04	0.49	0.01	900	11.11	11.11	new_extracted_faces
0.00	0.49	0.00	288000	0.00	0.00	zero_ternix
0.00	0.49	0.00	1800	0.00	0.00	delete_vector
0.00	0.49	0.00	1800	0.00	0.00	new_vector
0.00	0.49	0.00	900	0.00	0.00	new_empty_faces
0.00	0.49	0.00	128	0.00	0.00	delete_element
0.00	0.49	0.00	64	0.00	0.00	new_random_element
0.00	0.49	0.00	64	0.00	0.00	new_zero_element
0.00	0.49	0.00	18	0.00	0.00	delete_ternix
0.00	0.49	0.00	9	0.00	0.00	new_random_ternix
0.00	0.49	0.00	9	0.00	0.00	new_zero_ternix
0.00	0.49	0.00	1	0.00	0.00	delete_matrix
0.00	0.49	0.00	1	0.00	0.00	new_random_matrix
0.00	0.49	0.00	1	0.00	0.00	setup_parameters

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 2.04% of 0.49 seconds

index	% time	self	children	called	name
					<spontaneous>
[1]	100.0	0.00	0.49		main [1]
		0.17	0.00	96000/96000	operation_conv [2]
		0.10	0.00	96000/96000	operation_ds [3]
		0.09	0.00	96000/96000	operation_dr [4]
		0.05	0.00	96000/96000	operation_dt [5]

```
             0.05    0.00   96000/96000       operation_sum [6]
             0.02    0.00   96000/96000       operation_rk [7]
             0.01    0.00     900/900         new_extracted_faces [8]
             0.00    0.00    1800/1800        delete_vector [10]
             0.00    0.00     900/1800        new_vector [11]
             0.00    0.00     900/900         new_empty_faces [12]
             0.00    0.00     128/128         delete_element [13]
             0.00    0.00      64/64          new_zero_element [15]
             0.00    0.00      64/64          new_random_element [14]
             0.00    0.00      18/18          delete_ternix [16]
             0.00    0.00       9/9           new_random_ternix [17]
             0.00    0.00       9/9           new_zero_ternix [18]
             0.00    0.00       1/1           setup_parameters [21]
             0.00    0.00       1/1           new_random_matrix [20]
             0.00    0.00       1/1           delete_matrix [19]
-----------------------------------------------
             0.17    0.00   96000/96000       main [1]
[2]     34.7    0.17    0.00   96000         operation_conv [2]
-----------------------------------------------
             0.10    0.00   96000/96000       main [1]
[3]     20.4    0.10    0.00   96000         operation_ds [3]
             0.00    0.00   96000/288000      zero_ternix [9]
-----------------------------------------------
             0.09    0.00   96000/96000       main [1]
[4]     18.4    0.09    0.00   96000         operation_dr [4]
             0.00    0.00   96000/288000      zero_ternix [9]
-----------------------------------------------
             0.05    0.00   96000/96000       main [1]
[5]     10.2    0.05    0.00   96000         operation_dt [5]
             0.00    0.00   96000/288000      zero_ternix [9]
-----------------------------------------------
             0.05    0.00   96000/96000       main [1]
[6]     10.2    0.05    0.00   96000         operation_sum [6]
-----------------------------------------------
             0.02    0.00   96000/96000       main [1]
[7]      4.1    0.02    0.00   96000         operation_rk [7]
-----------------------------------------------
             0.01    0.00     900/900         main [1]
[8]      2.0    0.01    0.00     900         new_extracted_faces [8]
             0.00    0.00     900/1800        new_vector [11]
-----------------------------------------------
             0.00    0.00   96000/288000      operation_dr [4]
             0.00    0.00   96000/288000      operation_ds [3]
             0.00    0.00   96000/288000      operation_dt [5]
```

```
[9]     0.0    0.00    0.00  288000         zero_ternix [9]
-----------------------------------------------
             0.00    0.00    1800/1800       main [1]
[10]    0.0    0.00    0.00    1800         delete_vector [10]
-----------------------------------------------
             0.00    0.00     900/1800       main [1]
             0.00    0.00     900/1800       new_extracted_faces [8]
[11]    0.0    0.00    0.00    1800         new_vector [11]
-----------------------------------------------
             0.00    0.00     900/900        main [1]
[12]    0.0    0.00    0.00     900         new_empty_faces [12]
-----------------------------------------------
             0.00    0.00     128/128        main [1]
[13]    0.0    0.00    0.00     128         delete_element [13]
-----------------------------------------------
             0.00    0.00      64/64         main [1]
[14]    0.0    0.00    0.00      64         new_random_element [14]
-----------------------------------------------
             0.00    0.00      64/64         main [1]
[15]    0.0    0.00    0.00      64         new_zero_element [15]
-----------------------------------------------
             0.00    0.00      18/18         main [1]
[16]    0.0    0.00    0.00      18         delete_ternix [16]
-----------------------------------------------
             0.00    0.00       9/9          main [1]
[17]    0.0    0.00    0.00       9         new_random_ternix [17]
-----------------------------------------------
             0.00    0.00       9/9          main [1]
[18]    0.0    0.00    0.00       9         new_zero_ternix [18]
-----------------------------------------------
             0.00    0.00       1/1          main [1]
[19]    0.0    0.00    0.00       1         delete_matrix [19]
-----------------------------------------------
             0.00    0.00       1/1          main [1]
[20]    0.0    0.00    0.00       1         new_random_matrix [20]
-----------------------------------------------
             0.00    0.00       1/1          main [1]
[21]    0.0    0.00    0.00       1         setup_parameters [21]
-----------------------------------------------
```

Case 1 Discussion:
The program spent most of the time in the following function (operation_conv), (operation_dr), (operation_dt), and (operation_sum). This function (zero_ternix) was called by the program 288000 times. This is considered the biggest number of calls in the program. Those functions operation_conv), (operation_dr), (operation_dt), and (operation_sum) were called 96000 times by the program. The average time spent in the function (operation_dr) was the highest then (operation_conv) then (operation_dt)

Case 2:

Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self us/call	total us/call	name
36.50	10.22	10.22	192000	53.23	53.23	operation_conv
19.18	15.59	5.37	192000	27.97	28.39	operation_dr
17.82	20.58	4.99	192000	25.99	26.41	operation_ds
13.97	24.49	3.91	192000	20.37	20.78	operation_dt
6.11	26.20	1.71	192000	8.91	8.91	operation_sum
4.86	27.56	1.36	192000	7.08	7.08	operation_rk
0.86	27.80	0.24	576000	0.42	0.42	zero_ternix
0.39	27.91	0.11	900	122.23	122.23	new_extracted_faces
0.18	27.96	0.05				main
0.14	28.00	0.04				_intel_fast_memset
0.00	28.00	0.00	1800	0.00	0.00	delete_vector
0.00	28.00	0.00	1800	0.00	0.00	new_vector
0.00	28.00	0.00	900	0.00	0.00	new_empty_faces
0.00	28.00	0.00	256	0.00	0.00	delete_element
0.00	28.00	0.00	128	0.00	0.00	new_random_element
0.00	28.00	0.00	128	0.00	0.00	new_zero_element
0.00	28.00	0.00	18	0.00	0.00	delete_ternix
0.00	28.00	0.00	9	0.00	0.00	new_random_ternix
0.00	28.00	0.00	9	0.00	0.00	new_zero_ternix
0.00	28.00	0.00	1	0.00	0.00	delete_matrix
0.00	28.00	0.00	1	0.00	0.00	new_random_matrix
0.00	28.00	0.00	1	0.00	0.00	setup_parameters

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.04% of 28.00 seconds

```
index % time    self  children    called     name
                                  <spontaneous>
[1]     99.9    0.05   27.91                 main [1]
               10.22    0.00  192000/192000      operation_conv [2]
                5.37    0.08  192000/192000      operation_dr [3]
                4.99    0.08  192000/192000      operation_ds [4]
                3.91    0.08  192000/192000      operation_dt [5]
                1.71    0.00  192000/192000      operation_sum [6]
                1.36    0.00  192000/192000      operation_rk [7]
                0.11    0.00     900/900         new_extracted_faces [9]
                0.00    0.00    1800/1800        delete_vector [11]
                0.00    0.00     900/1800        new_vector [12]
                0.00    0.00     900/900         new_empty_faces [13]
                0.00    0.00     256/256         delete_element [14]
                0.00    0.00     128/128         new_zero_element [16]
                0.00    0.00     128/128         new_random_element [15]
                0.00    0.00      18/18          delete_ternix [17]
                0.00    0.00       9/9           new_random_ternix [18]
                0.00    0.00       9/9           new_zero_ternix [19]
                0.00    0.00       1/1           setup_parameters [22]
                0.00    0.00       1/1           new_random_matrix [21]
                0.00    0.00       1/1           delete_matrix [20]
-----------------------------------------------
               10.22    0.00  192000/192000      main [1]
[2]     36.5   10.22    0.00  192000         operation_conv [2]
-----------------------------------------------
                5.37    0.08  192000/192000      main [1]
[3]     19.5    5.37    0.08  192000         operation_dr [3]
                0.08    0.00  192000/576000      zero_ternix [8]
-----------------------------------------------
                4.99    0.08  192000/192000      main [1]
[4]     18.1    4.99    0.08  192000         operation_ds [4]
                0.08    0.00  192000/576000      zero_ternix [8]
-----------------------------------------------
                3.91    0.08  192000/192000      main [1]
[5]     14.2    3.91    0.08  192000         operation_dt [5]
                0.08    0.00  192000/576000      zero_ternix [8]
-----------------------------------------------
                1.71    0.00  192000/192000      main [1]
[6]      6.1    1.71    0.00  192000         operation_sum [6]
-----------------------------------------------
                1.36    0.00  192000/192000      main [1]
[7]      4.9    1.36    0.00  192000         operation_rk [7]
```

```
-----------------------------------------------
             0.08    0.00  192000/576000      operation_dr [3]
             0.08    0.00  192000/576000      operation_ds [4]
             0.08    0.00  192000/576000      operation_dt [5]
[8]     0.9    0.24    0.00  576000         zero_ternix [8]
-----------------------------------------------
             0.11    0.00     900/900         main [1]
[9]     0.4    0.11    0.00     900         new_extracted_faces [9]
             0.00    0.00     900/1800        new_vector [12]
-----------------------------------------------
                                    <spontaneous>
[10]    0.1    0.04    0.00                 _intel_fast_memset [10]
-----------------------------------------------
             0.00    0.00    1800/1800        main [1]
[11]    0.0    0.00    0.00    1800         delete_vector [11]
-----------------------------------------------
             0.00    0.00     900/1800        main [1]
             0.00    0.00     900/1800        new_extracted_faces [9]
[12]    0.0    0.00    0.00    1800         new_vector [12]
-----------------------------------------------
             0.00    0.00     900/900         main [1]
[13]    0.0    0.00    0.00     900         new_empty_faces [13]
-----------------------------------------------
             0.00    0.00     256/256         main [1]
[14]    0.0    0.00    0.00     256         delete_element [14]
-----------------------------------------------
             0.00    0.00     128/128         main [1]
[15]    0.0    0.00    0.00     128         new_random_element [15]
-----------------------------------------------
             0.00    0.00     128/128         main [1]
[16]    0.0    0.00    0.00     128         new_zero_element [16]
-----------------------------------------------
             0.00    0.00      18/18          main [1]
[17]    0.0    0.00    0.00      18         delete_ternix [17]
-----------------------------------------------
             0.00    0.00       9/9           main [1]
[18]    0.0    0.00    0.00       9         new_random_ternix [18]
-----------------------------------------------
             0.00    0.00       9/9           main [1]
[19]    0.0    0.00    0.00       9         new_zero_ternix [19]
-----------------------------------------------
             0.00    0.00       1/1           main [1]
[20]    0.0    0.00    0.00       1         delete_matrix [20]
-----------------------------------------------
```

```
                0.00    0.00       1/1           main [1]
[21]     0.0    0.00    0.00       1         new_random_matrix [21]
-----------------------------------------------
                0.00    0.00       1/1           main [1]
[22]     0.0    0.00    0.00       1         setup_parameters [22]
-----------------------------------------------


Index by function name

  [10] _intel_fast_memset     [15] new_random_element      [4] operation_ds
  [14] delete_element         [21] new_random_matrix       [5] operation_dt
  [20] delete_matrix          [18] new_random_ternix       [7] operation_rk
  [17] delete_ternix          [12] new_vector              [6] operation_sum
  [11] delete_vector          [16] new_zero_element       [22] setup_parameters
   [1] main                   [19] new_zero_ternix         [8] zero_ternix
  [13] new_empty_faces         [2] operation_conv
   [9] new_extracted_faces     [3] operation_dr
```

Case 2 a Discussion:

The function (operation_conv) took the longest time. The functions (operation_dr), (operation_ds), and (operation_dt) took less time by the program compared to (operation_conv). The function (zero_ternix) was called 576000. This is considered the greatest number of calls in the program. Those functions (operation_conv), (operation_dr), (operation_ds), (operation_dt), and (operation_sum) were called 192000 times by the program. In this case the number of calls is higher than case 1. The average time spent in the function (operation_conv) was the highest then (operation_dr) then (operation_ds).

Case 2 b:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self us/call	total us/call	name
35.98	1.36	1.36	768000	1.77	1.77	operation_conv
16.40	1.98	0.62	768000	0.81	0.86	operation_dr
15.08	2.55	0.57	768000	0.74	0.79	operation_ds
14.55	3.10	0.55	768000	0.72	0.76	operation_dt
7.67	3.39	0.29	768000	0.38	0.38	operation_sum
5.56	3.60	0.21	768000	0.27	0.27	operation_rk
2.91	3.71	0.11	2304000	0.05	0.05	zero_ternix
1.06	3.75	0.04	7200	5.56	5.56	new_extracted_faces
0.79	3.78	0.03				main
0.00	3.78	0.00	14400	0.00	0.00	delete_vector
0.00	3.78	0.00	14400	0.00	0.00	new_vector
0.00	3.78	0.00	7200	0.00	0.00	new_empty_faces
0.00	3.78	0.00	1024	0.00	0.00	delete_element
0.00	3.78	0.00	512	0.00	0.00	new_random_element
0.00	3.78	0.00	512	0.00	0.00	new_zero_element
0.00	3.78	0.00	144	0.00	0.00	delete_ternix
0.00	3.78	0.00	72	0.00	0.00	new_random_ternix
0.00	3.78	0.00	72	0.00	0.00	new_zero_ternix
0.00	3.78	0.00	8	0.00	0.00	delete_matrix
0.00	3.78	0.00	8	0.00	0.00	new_random_matrix
0.00	3.78	0.00	8	0.00	0.00	setup_parameters
0.00	3.78	0.00	1	0.00	0.00	print_parameters

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.26% of 3.78 seconds

index	% time	self	children	called	name
					<spontaneous>
[1]	100.0	0.03	3.75		main [1]
		1.36	0.00	768000/768000	operation_conv [2]
		0.62	0.04	768000/768000	operation_dr [3]
		0.57	0.04	768000/768000	operation_ds [4]
		0.55	0.04	768000/768000	operation_dt [5]
		0.29	0.00	768000/768000	operation_sum [6]
		0.21	0.00	768000/768000	operation_rk [7]
		0.04	0.00	7200/7200	new_extracted_faces [9]
		0.00	0.00	14400/14400	delete_vector [10]

```
                0.00    0.00    7200/7200        new_empty_faces [12]
                0.00    0.00    7200/14400       new_vector [11]
                0.00    0.00    1024/1024        delete_element [13]
                0.00    0.00     512/512         new_random_element [14]
                0.00    0.00     512/512         new_zero_element [15]
                0.00    0.00     144/144         delete_ternix [16]
                0.00    0.00      72/72          new_zero_ternix [18]
                0.00    0.00      72/72          new_random_ternix [17]
                0.00    0.00       8/8           delete_matrix [19]
                0.00    0.00       8/8           new_random_matrix [20]
                0.00    0.00       8/8           setup_parameters [21]
                0.00    0.00       1/1           print_parameters [22]
-----------------------------------------------
                1.36    0.00  768000/768000      main [1]
[2]     36.0    1.36    0.00  768000         operation_conv [2]
-----------------------------------------------
                0.62    0.04  768000/768000      main [1]
[3]     17.4    0.62    0.04  768000         operation_dr [3]
                0.04    0.00  768000/2304000     zero_ternix [8]
-----------------------------------------------
                0.57    0.04  768000/768000      main [1]
[4]     16.0    0.57    0.04  768000         operation_ds [4]
                0.04    0.00  768000/2304000     zero_ternix [8]
-----------------------------------------------
                0.55    0.04  768000/768000      main [1]
[5]     15.5    0.55    0.04  768000         operation_dt [5]
                0.04    0.00  768000/2304000     zero_ternix [8]
-----------------------------------------------
                0.29    0.00  768000/768000      main [1]
[6]      7.7    0.29    0.00  768000         operation_sum [6]
-----------------------------------------------
                0.21    0.00  768000/768000      main [1]
[7]      5.6    0.21    0.00  768000         operation_rk [7]
-----------------------------------------------
                0.04    0.00  768000/2304000     operation_dr [3]
                0.04    0.00  768000/2304000     operation_ds [4]
                0.04    0.00  768000/2304000     operation_dt [5]
[8]      2.9    0.11    0.00 2304000         zero_ternix [8]
-----------------------------------------------
                0.04    0.00    7200/7200        main [1]
[9]      1.1    0.04    0.00    7200         new_extracted_faces [9]
                0.00    0.00    7200/14400       new_vector [11]
-----------------------------------------------
                0.00    0.00   14400/14400       main [1]
```

```
[10]     0.0    0.00    0.00   14400        delete_vector [10]
-----------------------------------------------
             0.00    0.00    7200/14400       main [1]
             0.00    0.00    7200/14400       new_extracted_faces [9]
[11]     0.0    0.00    0.00   14400        new_vector [11]
-----------------------------------------------
             0.00    0.00    7200/7200        main [1]
[12]     0.0    0.00    0.00    7200        new_empty_faces [12]
-----------------------------------------------
             0.00    0.00    1024/1024        main [1]
[13]     0.0    0.00    0.00    1024        delete_element [13]
-----------------------------------------------
             0.00    0.00     512/512         main [1]
[14]     0.0    0.00    0.00     512        new_random_element [14]
-----------------------------------------------
             0.00    0.00     512/512         main [1]
[15]     0.0    0.00    0.00     512        new_zero_element [15]
-----------------------------------------------
             0.00    0.00     144/144         main [1]
[16]     0.0    0.00    0.00     144        delete_ternix [16]
-----------------------------------------------
             0.00    0.00      72/72          main [1]
[17]     0.0    0.00    0.00      72        new_random_ternix [17]
-----------------------------------------------
             0.00    0.00      72/72          main [1]
[18]     0.0    0.00    0.00      72        new_zero_ternix [18]
-----------------------------------------------
             0.00    0.00       8/8           main [1]
[19]     0.0    0.00    0.00       8        delete_matrix [19]
-----------------------------------------------
             0.00    0.00       8/8           main [1]
[20]     0.0    0.00    0.00       8        new_random_matrix [20]
-----------------------------------------------
             0.00    0.00       8/8           main [1]
[21]     0.0    0.00    0.00       8        setup_parameters [21]
-----------------------------------------------
             0.00    0.00       1/1           main [1]
[22]     0.0    0.00    0.00       1        print_parameters [22]
-----------------------------------------------


  [13] delete_element          [20] new_random_matrix        [5] operation_dt
  [19] delete_matrix           [17] new_random_ternix        [7] operation_rk
  [16] delete_ternix           [11] new_vector               [6] operation_sum
```

[10] delete_vector [15] new_zero_element [22] print_parameters
[1] main [18] new_zero_ternix [21] setup_parameters
[12] new_empty_faces [2] operation_conv [8] zero_ternix
[9] new_extracted_faces [3] operation_dr
[14] new_random_element [4] operation_ds

Part 2

Title : Parallel Sort of N keys in Integer Sort kernel (NAS parallel Benchmarks)

NAS NPB Kernel:

Over the years, the NPB kernels have been widely utilized for testing hardware-level strategies or compiler-level optimization. However, there are a set of parallel programming interfaces written over the C/C++ language that could be used in NPB.

IS(integer sort) Kernel:

IS performs an integer sort among a sparse set of numbers, which can be compared with particle-in-cell applications. By default, the sorting method is based on the bucket sorting approach. Accordingly, the number of keys for each bucket is determined, and the total count is distributed among each bucket. When completed, each bucket receives sorted numbers and points to the final accumulated sizes. Finally, the keys within each bucket are sorted, and a partial test is performed to verify the results. The steps are as follows:

1) Generation of the initial sequence of keys uniformly distributed in memory

2) Load all N keys into the memory system through the appropriate memory mapping

3) Computation of the sorting operation:
do i = 1 to max

a) Modify the sequence of keys:
Ki = i and Ki+max = (Bmax i)
b) Computes each key rank
c) For every iteration, performs the partial verification

4) Execution of the full verification to evaluate the sorting operation.

Code:

The C code can be accessed by searching the link provided. https://www.nas.nasa.gov/software/npb.html . The NPB 3.3.1 zip file was accessed for

Gprof. The application can be configured to use MPI + OpenMP, OpenMP and serial. Note the NAS benchmark uses classes as a dataset; Each class is unique to its size.

Class S: small for quick test purposes

Class W: workstation size (a 90's workstation; now likely too small)

Classes A, B, C: standard test problems; ~4X size increase going from one class to the next

Classes D, E, F: large test problems; ~16X size increase from each of the previous classes

We have tested for classes S,W,A,B. **For serial code**

Gprof : Class S

```
[mohit.palliyilsa@login2 bin]$ gprof is.S.1
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
100.01      0.01     0.01                             main
  0.00      0.01     0.00       11     0.00     0.00  rank

                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 99.99% of 0.01 seconds

index % time    self  children    called     name
                                             <spontaneous>
[1]           100.0    0.01       0.00                 main [1]
                0.00       0.00       11/11          rank [2]
-----------------------------------------------
                0.00       0.00       11/11          main [1]
[2] 0.0       0.00       0.00       11         rank [2]
-----------------------------------------------
```

Index by function name

[1] main [2] rank

Gprof : Class B

[mohit.palliyilsa@login6 bin]$ gprof is.B.1
Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self ms/call	total ms/call	name
48.12	5.24	5.24				main
46.65	10.32	5.08	11	461.85	461.85	rank
5.05	10.87	0.55				__intel_avx_rep_memcpy
0.18	10.89	0.02				__intel_avx_rep_memset

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.09% of 10.89 seconds

index	% time	self	children	called	name
					<spontaneous>
[1]	48.1	5.24	5.08		main [1]
		5.08	0.00	11/11	rank [2]

		5.08	0.00	11/11	main [1]
[2]	46.6	5.08	0.00	11	rank [2]

					<spontaneous>
[3]	5.1	0.55	0.00		__intel_avx_rep_memcpy [3]

					<spontaneous>
[4]	0.2	0.02	0.00		__intel_avx_rep_memset [4]

Index by function name

[3] __intel_avx_rep_memcpy [1] main
[4] __intel_avx_rep_memset [2] rank

Part 2 Discussion:

The maximum class size for a serial implementation is for class B. Class C,D,E,F requires more processors as the data size is huge.

As per the data provided, class S and B have different granularity time; this is due to the class size difference. It can be noted that the main function takes up 48.1% and rank function takes up 46.6% followed by 5.1% by memcpy and then .2% by memset.

It can be also noted that for higher class size, the memcpy and memset time stamp increases.Hence this will be a good project to test the parallel performance of OpenMP and MPI code.