

Final Project

Andres Calderon - SID:861243796

December 10, 2015

1 Data

2 Code

Full code and other materials are available at [?].

2.1 bfe.cu

```
1  #include <stdio.h>
2  #include <thrust/sort.h>
3  #include <thrust/functional.h>
4  #include <thrust/device_vector.h>
5  #include <thrust/host_vector.h>
6  #include "bfe.h"
7
8  int main(){
9      const int TIMESTAMP = 1;
10     const int EPSILON = 2000;
11
12     FILE *in;
13     FILE *out;
14     in = fopen("oldenburg.csv", "r");
15     out = fopen("output.csv", "w");
16     fprintf(out, "oid;time;lat;lon;grid_id\n");
17     char line[1024];
18     int n = 0;
19     long grid_id;
20     int oid; short time;
21     int lat; int lon;
22     int max_lat = INT_MIN; int min_lat = INT_MAX;
23     int max_lon = INT_MIN; int min_lon = INT_MAX;
24     int M = 0;
25     int N = 0;
26     while (fgets(line, 1024, in)){
27         atoi(strtok(line, ";"));
28         if(atoi(strtok(NULL, ";\n")) != TIMESTAMP) continue;
29         lat = atoi(strtok(NULL, ";\n"));
30         if(lat > max_lat) max_lat = lat;
31         if(lat < min_lat) min_lat = lat;
32         lon = atoi(strtok(NULL, ";\n"));
33         if(lon > max_lon) max_lon = lon;
34         if(lon < min_lon) min_lon = lon;
35         n++;
36     }
37     int x[n];
38     int y[n];
39     int g[n];
40     int i[n];
```

Figure 1: Data points.

```

41 printf("Min and max latitude:\t(%d, %d)\n", min_lat, max_lat);
42 printf("Min and max longitude:\t(%d, %d)\n", min_lon, max_lon);
43 M = (max_lat - min_lat) / EPSILON + 1;
44 N = (max_lon - min_lon) / EPSILON + 1;
45 rewind(in);
46 int j = 0;
47 while (fgets(line, 1024, in)){
48     oid = atoi(strtok(line, ";"));
49     time = atoi(strtok(NULL, ";\n"));
50     if(time != TIMESTAMP) continue;
51     lat = atoi(strtok(NULL, ";\n"));
52     lon = atoi(strtok(NULL, ";\n"));
53     g[j] = M * ((N - 1) - ((lon - min_lon) / EPSILON)) + ((lat - min_lat) / EPSILON);
54     x[j] = lat;
55     y[j] = lon;
56     i[j] = j;
57     j++;
58     //fprintf(out, "%d;%hi;%d;%d;%li\n", oid, time, lat, lon, grid_id);
59 }
60 printf("Number of points:\t%d\n", n);
61 printf("M x N : %d x %d\n", M, N);
62 //int r = createGrid("grid.shp", EPSILON, min_lat, max_lat, min_lon, max_lon);
63
64 thrust::device_vector<int> d_x(x, x + n);
65 thrust::device_vector<int> d_y(y, y + n);
66 thrust::device_vector<int> d_g(g, g + n);
67 thrust::device_vector<int> d_i(i, i + n);
68 thrust::sort_by_key(d_g.begin(), d_g.end(), d_i.begin());
69 thrust::gather(d_i.begin(), d_i.end(), d_x.begin(), d_x.begin());
70 thrust::gather(d_i.begin(), d_i.end(), d_y.begin(), d_y.begin());
71
72 for(j = 0; j < n; j++)
73     std::cout << g[j] << "-" << i[j] << "(" << x[j] << "," << y[j] << ")";
74 std::cout << std::endl;
75 std::cout << std::endl;
76 thrust::copy(d_g.begin(), d_g.end(), g);
77 thrust::copy(d_i.begin(), d_i.end(), i);
78 thrust::copy(d_x.begin(), d_x.end(), x);
79 thrust::copy(d_y.begin(), d_y.end(), y);
80 for(j = 0; j < n; j++)
81     std::cout << g[j] << "-" << i[j] << "(" << x[j] << "," << y[j] << ")";
82 std::cout << std::endl;
83 std::cout << std::endl;
84 //thrust::copy(d_x.begin(), d_x.end(), std::ostream_iterator<int>(std::cout, ","));
85 //std::cout << std::endl;
86 return 0;
87 }

```

2.1.1 kernel.cu

3 Output

4 Output

References

- [1] Andres Calderon. *GitHub Personal Repository*, 2015. <https://github.com/aocalderon/PhD/tree/master/Y1Q1/GPU/lab3>.

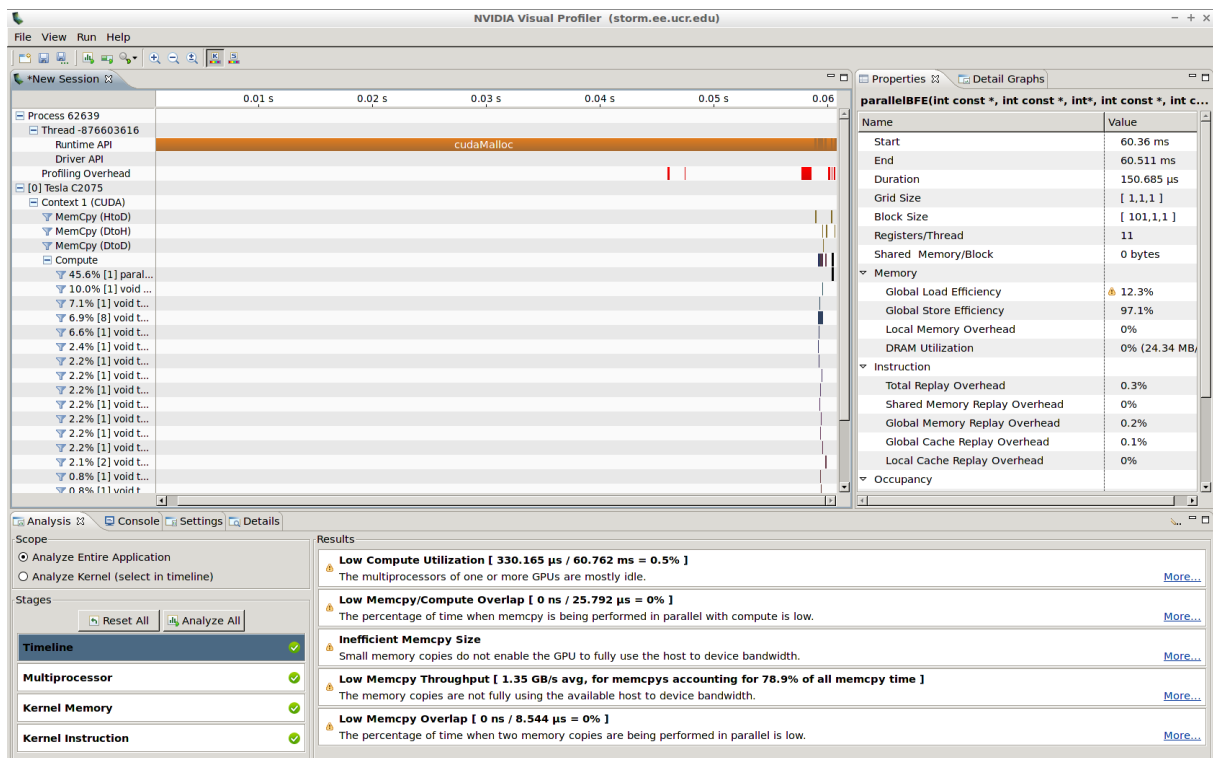


Figure 2: NVVP performance analysis for T1-E2K-M3.