6.172 Project 2.1 Writeup

Albert Wang

October 12, 2011

1 1

1.1 1.1

```
cilkview: running with 20 workers Sort time: 1.032 seconds

    cilkview: running with 19 workers Sort time: 1.027 seconds

    cilkview: running with 18 workers Sort time: 1.033 seconds

    cilkview: running with 17 workers Sort time: 1.069 seconds

    cilkview: running with 16 workers Sort time: 1.055 seconds

    cilkview: running with 15 workers Sort time: 1.032 seconds

    cilkview: running with 14 workers Sort time: 1.024 seconds

    cilkview: running with 13 workers Sort time: 1.029 seconds
```

cilkview: running with 12 workers Sort time: 1.023 seconds

cilkview: running with 11 workers Sort time: 1.089 seconds

cilkview: running with 10 workers Sort time: 1.166 seconds

cilkview: running with 9 workers Sort time: 1.259 seconds

cilkview: running with 8 workers Sort time: 1.387 seconds

cilkview: running with 7 workers Sort time: 1.541 seconds

cilkview: running with 6 workers Sort time: 1.745 seconds

cilkview: running with 5 workers Sort time: 2.049 seconds

cilkview: running with 4 workers Sort time: 2.463 seconds

cilkview: running with 3 workers Sort time: 3.199 seconds

cilkview: running with 2 workers Sort time: 4.656 seconds

cilkview: running with 1 workers Sort time: 9.059 seconds

When running with more workers than processors, the times may increase, as extra time is being used to spawn threads.

1.2 1.2

The $sample_qsort$ recursive calls are overlapping with each other, which means that they could modify each other's data

1.3 1.3

Yes, cilkscreen found the race condition. Error rate = 916/1000.

2 2

2.1 2.1

For 8 processors: Actual Speedup: 7.57686084142 times faster, Theoretical Speedup: 6.26 - 8.00 times faster

```
cilkview: running with 1 workers

matrix2 time: 0ms

matrix256 time: 47ms

matrix1024 time: 9365ms

Test succeeded


cilkview: running with 12 workers

matrix2 time: 0ms

matrix256 time: 7ms

matrix1024 time: 932ms

Test succeeded
```

2.2 2.2

For 8 processors: Middle Loop Parallelization Speedup: 7.71439873418 Outer, Middle loop Parallelization Speedup 7.70332278481 Just outer loop parallel produces the best results because not as many threads are being spawned.

2.3 2.3

2.4 2.4

```
mm_internal(C11, A11, B11, n, mid);

mm_internal(C21, A21, B11, n, mid);

mm_internal(C12, A11, B12, n, mid);

mm_internal(C22, A21, B12, n, mid);

mm_internal(C11, A12, B21, n, mid);

mm_internal(C21, A22, B21, n, mid);

mm_internal(C12, A12, B22, n, mid);

mm_internal(C22, A22, B22, n, mid);
```

2.5 2.5

Can divide 8 recursive calls into two sets of 4 calls. Calls that change the same submatrix of C must be done separately. Passes cilkscreen. cilkview:

```
Whole Program Statistics
1) Parallelism Profile
   Work :                                        245,301,142,331 instructions
   Span :                                        188,099,983 instructions
   Burdened span :                               363,019,983 instructions
   Parallelism :                                 1304.10
   Burdened parallelism :                        675.72
   Number of spawns/syncs:                       1,246,607,152
   Average instructions / strand :               65
   Strands along span :                          21,537
   Average instructions / strand on span :       8,733
   Total number of atomic instructions :         1,246,607,173
   Frame count :                                 2,493,214,304
2) Speedup Estimate
   2 processors:         1.90 - 2.00
   4 processors:         3.80 - 4.00
   8 processors:         7.60 - 8.00

   16 processors:        15.20 - 16.00

   32 processors:        29.68 - 32.00
```

```
cilkview: running with 12 workers

matrix2 time: 0ms

matrix256 time: 103ms

matrix1024 time: 4912ms
```

About 1.9 times faster than nested single processor

2.6 2.6

Much faster.

```
cilkview: running with 12 workers

matrix2 time: 0ms

matrix256 time: 9ms

matrix1024 time: 295ms
```

About 16.7 times faster than 12 worker without looping small matrices.

2.7 2.7

3 3

3.1 3.1

Cilkview results:

```
cilkview: running with 12 workers

nbodies time: 6668ms

done.

cilkview: running with 11 workers

nbodies time: 6371ms

done.

cilkview: running with 10 workers

nbodies time: 6342ms

done.

cilkview: running with 9 workers

nbodies time: 6170ms

done.

cilkview: running with 8 workers
```

```
nbodies time: 5771ms

done.

cilkview: running with 7 workers

nbodies time: 5847ms

done.

cilkview: running with 6 workers

nbodies time: 6011ms

done.

cilkview: running with 5 workers

nbodies time: 6066ms

done.

cilkview: running with 4 workers

nbodies time: 5784ms

done.

cilkview: running with 3 workers

nbodies time: 6214ms

done.

cilkview: running with 2 workers

nbodies time: 5930ms
```

```
done.

cilkview: running with 1 workers

nbodies time: 8869ms

done.
```

1.3 times speedup

3.2 3.2

```
cilkview: running with 12 workers

nbodies time: 10506ms

done.

cilkview: running with 11 workers

nbodies time: 9710ms

done.

cilkview: running with 10 workers

nbodies time: 9365ms

done.

cilkview: running with 9 workers

nbodies time: 9211ms

done.
```

```
cilkview: running with 8 workers

nbodies time: 9508ms

done.

cilkview: running with 7 workers

nbodies time: 9647ms

done.

cilkview: running with 6 workers

nbodies time: 9977ms

done.

cilkview: running with 5 workers

nbodies time: 9804ms

done.

cilkview: running with 4 workers

nbodies time: 10426ms

done.

cilkview: running with 3 workers

nbodies time: 12504ms

done.

cilkview: running with 2 workers
```

```
nbodies time: 12680ms

done.

cilkview: running with 1 workers

nbodies time: 23158ms

done.
```

2.2 times speedup

3.3 3.3

```
Cilkview Scalability Analyzer V2.0.0, Build 1924


nbodies Statistics - Elapsed time: 7.789 seconds

1) Parallelism Profile

   Work :                                       3,876,860,183 instructions

   Span :                                       5,092,369 instructions

   Burdened span :                              23,488,877 instructions

   Parallelism :                                761.31

   Burdened parallelism :                       165.05

   Number of spawns/syncs:                      63,920

   Average instructions / strand :              20,217
   Strands along span :                          760
   Average instructions / strand on span :       6,700
   Total number of atomic instructions :         51,200,160
   Frame count :                                 1,151,280
2) Speedup Estimate
   2 processors:         1.90 - 2.00
   4 processors:         3.80 - 4.00
   8 processors:         7.46 - 8.00
   16 processors:        13.86 - 16.00
   32 processors:        24.26 - 32.00

Whole Program Statistics
1) Parallelism Profile
   Work :                                        36,552,872,516 instructions
   Span :                                        1,706,963,702 instructions
   Burdened span :                               1,872,529,867 instructions
   Parallelism :                                 21.41
   Burdened parallelism :                        19.52
   Number of spawns/syncs:                       575,280
   Average instructions / strand :               21,179

   Strands along span :                          13,681

   Average instructions / strand on span :       124,768

   Total number of atomic instructions :         460,801,440

   Frame count :                                 1,151,280

2) Speedup Estimate

   2 processors:         1.84 - 2.00

   4 processors:         3.17 - 4.00

   8 processors:         4.97 - 8.00

   16 processors:        6.94 - 16.00

   32 processors:        8.65 - 21.41
```

3.4 3.4

Yes, was faster than parallel nbodies_loops. Races come from the symmetric updating of forces (i.e. changing the force for j), which didn't show up before because it wasn't symmetric.

3.5 3.5

```
Cilkview Scalability Analyzer V2.0.0, Build 1924
```

```
nbodies Statistics - Elapsed time: 6.943 seconds

1) Parallelism Profile

   Work :                                    3,225,295,516 instructions

   Span :                                    8,244,840 instructions

   Burdened span :                           24,973,893 instructions

   Parallelism :                             391.19

   Burdened parallelism :                    129.15

   Number of spawns/syncs:                   63,920

   Average instructions / strand :           16,819

   Strands along span :                      800

   Average instructions / strand on span :   10,306

   Total number of atomic instructions :     51,200,160

   Frame count :                             1,151,280

2) Speedup Estimate

   2 processors:         1.90 - 2.00

   4 processors:         3.80 - 4.00

   8 processors:         7.33 - 8.00

   16 processors:        13.36 - 16.00
   32 processors:        22.73 - 32.00


Whole Program Statistics

1) Parallelism Profile

   Work :                                       30,688,883,602 instructions

   Span :                                       1,735,328,515 instructions

   Burdened span :                              1,885,890,905 instructions

   Parallelism :                                17.68

   Burdened parallelism :                       16.27

   Number of spawns/syncs:                      575,280

   Average instructions / strand :              17,781

   Strands along span :                         14,401

   Average instructions / strand on span :      120,500

   Total number of atomic instructions :        460,801,440

   Frame count :                                1,151,280

2) Speedup Estimate

   2 processors:         1.81 - 2.00

   4 processors:         3.05 - 4.00

   8 processors:         4.62 - 8.00
   16 processors:        6.23 - 16.00

   32 processors:        7.55 - 17.68
```

3.6 3.6

A triangle itself is split into two triangles and a rectangle. Therefore, the rectangles will coarsen the data for it's parent triangles. Also, a triangle itself doesn't do much work, other than to spawn other triangles and rectangles.

3.7 3.7

Added cilk_spawn for the first lines of A, B, and C, and a sync at the end of each A/B/C block.

3.8 3.8

```
cilkview: running with 12 workers

nbodies time: 1112ms

done.

cilkview: running with 11 workers

nbodies time: 1168ms

done.
```

```
cilkview: running with 10 workers
nbodies time: 1247ms
done.
cilkview: running with 9 workers
nbodies time: 1346ms
done.
cilkview: running with 8 workers
nbodies time: 1479ms
done.
cilkview: running with 7 workers
nbodies time: 1650ms
done.
cilkview: running with 6 workers
nbodies time: 1881ms
done.
cilkview: running with 5 workers
nbodies time: 2215ms
done.
cilkview: running with 4 workers
```

```
nbodies time: 2682ms

done.

cilkview: running with 3 workers

nbodies time: 3557ms

done.

cilkview: running with 2 workers

nbodies time: 5230ms

done.

cilkview: running with 1 workers

nbodies time: 10421ms

done.

cilkview: generating scalability data

nbodies time: 65083ms

done.

   --- albertyw-12.stderr ---

Cilkview Scalability Analyzer V2.0.0, Build 1924



nbodies Statistics - Elapsed time: 7.220 seconds

1) Parallelism Profile
```

```
   Work :                                        3,315,985,410 instructions
   Span :                                        106,840,183 instructions
   Burdened span :                               216,840,183 instructions
   Parallelism :                                 31.04
   Burdened parallelism :                        15.29
   Number of spawns/syncs:                       116,000
   Average instructions / strand :               9,528
   Strands along span :                          5,360
   Average instructions / strand on span :       19,932
   Total number of atomic instructions :         51,252,240
   Frame count :                                 3,132,360
2) Speedup Estimate
   2 processors:         1.80 - 2.00
   4 processors:         3.00 - 4.00
   8 processors:         4.50 - 8.00
   16 processors:        6.00 - 16.00
   32 processors:        7.20 - 31.04


Whole Program Statistics
```

```
1) Parallelism Profile

   Work :                                        31,504,917,256 instructions

   Span :                                        2,622,692,230 instructions

   Burdened span :                               3,612,692,230 instructions

   Parallelism :                                 12.01

   Burdened parallelism :                        8.72

   Number of spawns/syncs:                       1,044,000

   Average instructions / strand :               10,059

   Strands along span :                          96,481

   Average instructions / strand on span :       27,183

   Total number of atomic instructions :         461,270,160

   Frame count :                                 3,132,360

2) Speedup Estimate

   2 processors:         1.67 - 2.00

   4 processors:         2.52 - 4.00

   8 processors:         3.38 - 8.00

   16 processors:        4.08 - 12.01

   32 processors:        4.54 - 12.01
```