Name : Sweta Rout
Unity ID : srout
Student ID : 200206715

Performance comparison: a table lists execution time, speedup (using mode 0 as the baseline) of each mode the excludes the data initialization and result verification.

I chose to implement merge sort.

Mode	Execution Time	Ratio to the base(mode 0)
0	27046002	1
1	20271504	0.75
2	28651328	1.05

What's the complexity of your algorithm?

Complexity of merge sort in openCL is **O(log(n)*log (stride)).** OpenCL sorts chunks (say, 256) and then does the merging. The entire thing is done for log(n) times.

Is the performance improvement as good as you expected? Why or why not?

Ideally, the algorithm should provide much better results. It should be improved by a factor of 2 at least. But my code uses local memory, so synchronization is an overhead and it degrades the performance of parallel sorting.

Comparing your performance with pthread

Name	Execution Time	Ratio to the base(pthread)
Pthread merge sorting	20271504	1
FPGA merge sorting	28651328	1.4

Performance should be better in openCL than compared to pthread. Because for 2 cores, pthread has two threads for each of the core. And, FPGA/openCL can have a large number of work units to work on a large number of elements in a parallel environment. Here, we are getting little slower performance because of overhead.

The list of students sharing the same board with you :

- Sweta Rout (srout@ncsu.edu)
- Cody Nesbitt (cznesbit@ncsu.edu)