CS267 Assignment 2

Hussain Al Salem Jason Poulos Yang You

February 20, 2016

1 Introduction

In this report, we describe several parallel implementations of a 2D particle simulator and report their performance. The goal is to parallelize code that runs in time $T = O(n)$ on a single processor to run in time T/p when using p processors by taking advantage of shared and distributed memory models. Specifically, we try three implementations: serial code that runs in $O(n)$ time; a MPI distributed memory implementation that runs in $O(n)$ time and $O(n/p)$ scaling; and a OpenMP shared memory implementation. For Part 2 of the assignment, we implement in GPU.

2 Serial Implementation

2.1 Data structures

Describe the serial structures used to achieve $O(n)$.

2.2 Results

3 OpenMP implementation

3.1 Synchronization

A description of the synchronization you used in the shared memory implementation.

Figure 1: A plot in log-log scale that shows that serial and parallel codes run in O(n) time.

Figure 2: A plot in log-linear scale that shows performance as a percent of peak performance for different numbers of processors.

Figure 3: $O(n)$ scaling of the OpenMP implementation.

3.2 Results

4 MPI Implementation

4.1 Communication between nodes

A description of the communication you used in the distributed memory implementation.

4.2 Results

5 Comparison of distributed and shared implementations

5.1 Where does the time go?

Consider breaking down the runtime into computation time, synchronization time and/or communication time. How do they scale with p?

5.2 Discussion

A description of the design choices that you tried and how did they affect the performance.

A discussion on whether it is possible to do better

A discussion on using pthreads, OpenMP and MPI.

6 GPU Implementation

A description of any synchronization needed A description of any GPU-specific optimizations you tried

Figure 4: Weak scaling of the OpenMP implementation.

Figure 5: Strong scaling of the OpenMP implementation.

Figure 6: $O(n)$ scaling of the MPI implementation.

6.1 Results

6.2 Discussion

A discussion on the strengths and weaknesses of CUDA and the current GPU architecture

Figure 7: Weak scaling of the MPI implementation.

Figure 8: Strong scaling of the MPI implementation.

Figure 9: Speedup plots that show how closely parallel codes approach the idealized p-times speedup.

Figure 10: A plot of the speedup of the GPU code versus the serial, openmp, mpi runs on the CPU of the node

Figure 11: A plot in log-log scale that shows the performance of your code versus the naive GPU code.