

# CS 267: Homework 3

Richard Barnes, Danny Broberg, Jiayuan Chen

April 1, 2016

## 1 Introduction

This report describes our implementation of Unified Parallel C (UPC) for de novo genome DNA assembly. UPC is an extension of the C language, outfitted for a Partitioned Global Address Space (PGAS) programming model. This model has the advantage of a shared address space for all the processors, but each variable is physically associated with a specific processor.

Our paper is organized as follows:

1. description of the computational resources
2. a section that describes the general serial algorithm approach
3. a section that describes the general extension to UPC
4. evaluation of the code's performance

## 2 Machine Description

For our codes we use NERSC's Edison machine. The machine has 5,576 compute nodes. Each node has 64GB DDR3 1866 MHz RAM and two sockets, each of which is populated with a 12-core Intel "Ivy Bridge" processor running at 2.4 GHz. Each core has one or two user threads, a 256 bit vector unit, and is nominally capable of 19.2 Gflops. Notably, for our purposes, each core has its own L1 and L2 cache. The L1 cache has 64 KB (32 KB instruction cache, 32 KB data) and the L2 cache has 256 KB. The 12 cores collectively share a 30 MB L3 cache. The caches have a bandwidth of 100, 40, and 23 Gbyte/s, respectively.<sup>1</sup> Both the L1 and L2 caches are 8-way and have a 64 byte line size.<sup>2</sup>

## 3 Serial Approach

Before coding up the particle dynamics in parallel, we describe the approach for shotgun denova genome assembly as written in serial.

---

<sup>1</sup><http://www.nersc.gov/users/computational-systems/edison/configuration/>

<sup>2</sup><http://www.7-cpu.com/cpu/IvyBridge.html>

## **4    UPC approach**

Our extension to the UPC language follows the general outline of the serial code...

## **5    Performance**

## **6    Summary**