Software Construction Laboratory

Week 6 Part 2

Lab 3

Mushi Zhou

Winter 2017

UCLA

Prof. Paul Eggert

Outline for Today

- OpenCL
- OpenMP
- Massive parallelism,
- Clusters, grids, clouds
- Some Help for Assignment 6

More on Parallelism

- Today, single thread is not enough for most of jobs
- The speed of CPU has reached its threshold due temperatures
- i.e. Exponentially increase in temperature for a small increase in frequency, cannot be faster
- The scale of problems are still increasing
- For example, we have exponentially increased size of data to process
- What can we do?
- Massively Parallel!

Parallelism in Production

- POSIX threads is just a standard library, it is still very limited for a verity of tasks
- Need easy ways to make different kinds of jobs to be parallel including image processing, data analysis, communications, etc
- What is been used in industry?
- OpenCL
- OpenMP

OpenCL

- Open Computing Language (OpenCL) is a framework of open standard for parallel programming of heterogeneous systems
- Framework -> a collection of software environment that includes support programs, compilers, code libraries, tool sets, and application programming interfaces (APIs)
- Heterogeneous systems: any systems that has a CPU/Processor
- OpenCL provides a standard interface for parallel computing using task-based and data-based parallelism
- Open standard maintained by the non-profit technology consortium Khronos Group

OpenCL

- Very widely used
- Very relevant
- Most general approach
- More than just a library or API
- A simple example way of using it is to use the API library provided in C++
- i.e. it can easily utilize all the available hardware resources on the running system to maximize parallel performance

OpenMP

- OpenMP (Open Multi-Processing) is an API that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran, on most platforms, processor architectures and operating systems, including Linux, OS X, and Windows
- It consists of a set of compiler directives, library routines, and environment variables that influence run-time behavior
- Managed by the nonprofit technology consortium *OpenMP ARB*
- OpenMP uses a portable, scalable model that gives programmers a simple and flexible interface for developing parallel applications

OpenMP

- Widely used
- Very relevant
- Less general than OpenCL, only an API
- More portable
- Less overhead

Massively Parallel

- We want not only multithreading our programs, but also running all our computing jobs in parallel
- This can be achieved by having multi-core CPUs, multiple computers, and/or distributed systems
- Massively Parallel -> use of a large number of processors or separate computers to perform a set of coordinated computations in parallel simultaneously
- In production:
- Grid Computing
- Cluster Computing
- Cloud Computing

Grid Computing

- Grid computing is the collection of computer resources from multiple locations to reach a common goal
- The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files
- A central control and many “worker” nodes
- Each node is set to perform a different task/application
- Grid computers also tend to be more heterogeneous and geographically dispersed

Cluster Computing

- A computer cluster consists of a set of loosely or tightly connected computers that work together so that, in many respects, they can be viewed as a single system
- Unlike grid computers, it has each node set to perform the same task(a portion), controlled and scheduled by software
- The components of a cluster are connected to each other through fast local area networks ("LAN"), with each node running its own instance of an operating system
- In most cases, all of the nodes use the same hardware and the same operating system
- Our SEASnet Computers are an example

Cloud Computing

- Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand
- A model for providing ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g. servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort
- The parallel computations are done remotely in cluster-style computing, the user has the virtual control of the hardware
- Amazon Elastic Compute Cloud (EC2) is an example

Assignment 6 Help

- This is not a difficult assignment to do, but the concept behind are very important!
- For the lab, you want to time only the sorting time, not including the time to create test data
- To create test data, single-precision float points are 4 bytes, make sure you “od” correct number of values
- Use –A n to not print number byte offset
- Use “tr” to transform space to ‘\n’
- Use “sed” to delete all empty lines

Assignment 6 Help

For HW

- Do something useful using multiple threads
- You are doing computation for all pixels (width * height)
- The easiest would be slicing the outer loop, which is the width, for multiple threads, and run each thread with all heights (inner loop)
- Define the inner loop to run in a new function
- Then you need to pass argument for each thread, what you can do is to define a struct that contains variables to each thread, because pthread only allow input function to take one argument (You need to malloc all arguments)
- Cast the argument to (void*) and cast back in each thread
- You only need to pass 5 arguments in the struct if you do this way
- No need to pass enum values, which are global

Assignment 6 Help

- You need to modify makefile to compile the library using pthread
- (-lpthread) option
- Use diff to compare your output to the given output
- The are graphs, you can view them using GIMP