Plan for Today

Announcements

– Presentation slots are filling up.
– Example research computations for projects will be available by the end of this week. Project proposal presentation (5 minutes) will be on Thursday Feb 25. Preliminary project proposals will be due Feb 18th.

LULESH Paper Discussion
Review of
Exploring Traditional and Emerging Parallel Programming Models Using a Proxy Application

Michelle Strout
February 2, 2016
What problem did the paper address?

**Big picture problem**
- Current parallel programming models do not result in programs that are performance portable.

**Typical Approach**
- Hand-tune programs per target architecture.
- Develop domain-specific libraries and/or languages.
- Develop new programming models.

**Specific Problem**
- Which new parallel programming models?
- DOE and DOD application developers
- heterogeneity and scale of emerging architectures
- performance portability, application applicability, and programmer productivity
Why should we care?

Software simulation is fundamental to the DOE and DOD.

**LULESH proxy application**
- Performs hydrodynamics simulation.
- Hydrodynamics “accounts for about 27% of data center utilization at DOD”
- Time is energy is money.

**Other studies did not have ...**
- An in depth study of a proxy application.
- Comparison against so many current and emerging programming models.
What is the approach used to solve this problem?

Comparison of three provided LULESH implementations with new implementations in ...
- Chapel
- CHARM++
- Liszt
- Loci

Productivity
- SLOC
- SLOC change for optimizations

Performance
- Single core performance
- Weak and strong scaling.

Ease of Optimization
- Programmer or compiler able to perform optimizations.
- Opts: Loop fusion, global allocation, data structure trans, vectorization, blocking, C-C overlap
Code Examples

Liszt, see online presentation

Loci, see paper

Chapel, see online presentation

CHARM++, see webpage example
How does the paper support the conclusions it reaches?

Some of their conclusions (most based directly on empirical results)

- Loci and Chapel programs are up to 80% smaller than MPI implementations. SLOC counts.
- Liszt is highly portable, but performance is not quite there yet.

Performance Experiments

- CUDA and Liszt GPU versions ran on one node of Intel Westmere CPUs and NVIDIA Tesla M2050.
- Intel Sandy cluster with icc and BG/Q system with xlc for non-GPU
- Strong and weak scaling results.
Future Research Questions

How should programming models be selected for such studies?

Is there a metric for cognitive load that new programming models introduce?

How much does tool support impact programmability?
Critique

Introduction

– No real hypothesis.
– Do back up their choice of programming models, but weakly.

Examples of in-depth analysis

– Specific reason for Chapel not matching MPI (1D linearization).
– Coloring thread-scheduling in Liszt was not best for LULESH.

Important Point about MPI

– “The MPI standard specifies the functionality of the routines and a library writer is then free to implement them as seen fit for a particular hardware.”
– One of the main reasons MPI is so popular.

Presentation Issues

– The Table II caption is not self contained.
– Analysis of Loci is overly friendly. (end of Section VI)
– Did not say how results over 10 runs were aggregated.
– Comparing graphs on different pages is a pain.
Relation to CSc 620

Example of what is expected from term projects

- Motivate the example program you use in your case study.
- Attempt to determine ahead of time evaluation metrics.
- Develop a hypothesis about how your three programming models might compare and cite previous work to back up your initial hypothesis.
- Carefully consider which three programming models to compare.
- You will be interacting with the scientists who wrote the initial code.
- Show code examples comparing how different programming constructs handle various aspects of the application.
- Aim for a publication, although this might take time after this semester.

Modifications to make this a term project

- Project example algorithm will be smaller than LULESH.
- One of the programming models you use in the comparison could be the original programming model for the example algorithm.
Next Time

Reading
- Productivity in Parallel Programming: A Decade of Progress: Looking at the design and benefits of X10 by Richards et al.

Class
- Example programming model presentation for X10.

Due
- Wednesday (tomorrow) there is a quiz due.