Colorado State University

CS/ECE560: Foundations of Fine Grain Parallelism

Sanjay Rajopadhye
Colorado State University

Outline

- Class objectives, goals
- Why Fine grain parallelism?
- Equational Programming (intro)

Undergraduate vs Graduate

- Every problem is underspecified
- Questions are ill posed
  - Finding the right question is half the work
- Communicate: write, write, write
- Question everything

Colorado State University 3

CS/ECE 560: Plan

- Introductions:
  - Name
  - Year in school (e.g., MS in ECE, 2nd year)
  - Parallelism experience (SW/HW)
  - Interesting fact

Colorado State University 4

Administrative details

- Course website: http://www.cs.colostate.edu/~cs560
- Very little on RamCT/Canvas (only for grades)
- Course related email: cs560@cs.colostate.edu
- How to succeed
  - Check the schedule tab/page every day – frequent updates
  - Spend about 1-2 hrs every day outside class
    - General rule 4 credits = 8-12 hrs outside

Class Objectives

- Short term

  Become macho paralle programmer: write "heroically tuned" codes.
- Medium term

  Do it systematically: tuning for "accelerator of the day" e.g., Kepler k40 vs Xeon Phi: learn principles, not skills
- Long term

  Do it automatically: Learn the foundations of automatic compilation. Focus on a "regular subset" of programs
- Polyhedral Equational Model

Class Approach & Outline

- Big picture
  - Polyhedral Equations as programs: I'm loath to write C
- Equations vs (conventional) loop programs
- Equations-to-code (compiling equations)
  - Schedule
  - (processor) allocation
  - (memory) allocation
- But what about parallelism?

Class Approach & Outline …

8 assignments (basic + advanced) + term project

- Parallel program performance (2)
- Mathematical foundations: polyhedra, affine functions, and operations (2)
- Analysis: scheduling & allocation (2)
- Equational programming: Alpha/AlphaZ (1)
- Alpha analysis/transformation (1)
- Code generation/tiling (2)

Class grade breakdown

- Assignments (30%)
- Midterms in class+take home (15+20=35%)
- Final project (30% = 2+3+5+15+5)
  - Proposal
  - Advancement report
  - Final report
  - Quality of work
  - Final poster
- Participation/Discussion/Quizzes (5%)

The Polyhedral Model

- What are polyhedra?
- Why are they useful/important
- What is the polyhedral model?

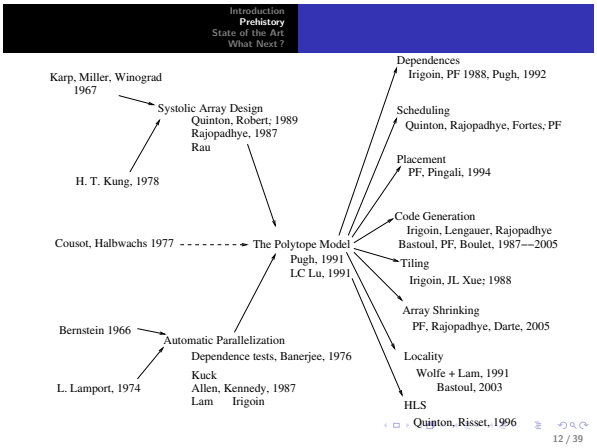
Model

- What is a model?
  - A mathematical/computational/mechanical/ … abstraction of some other (physical) entity
  - Objects in the model must "emulate" the "natural operations" of the modeled entities – semantics

Colorado State University 11

(his) story

From Feautrier's keynote at LCPC 2009



Colorado State University 12

"Real" vs "Abstract"

- Physical entity: programs/computations
- The Polyhedral Model is a "very high level" intermediate representation (IR) of "regular computations"
- Polyhedral equational model: real=abstract
- Amenable to:
  - Mathematical static analysis
  - Transformation within model: closure
  - Transformation outside model: (tiled) code generation

Outline

- ~~Class objectives, goals~~
- Why Fine grain parallelism?
- Equational Programming (intro)

Parallelism is (now) ubiquitous

- Parallel Programming is hard
  - "End of the free lunch" [Sut05]
  - Arrival of "manycores" signals the end of "La-Z-Boy Programming" [Pat06]

  Becoming a parallel programming expert will get you a good job

  But your skills may become obsolete – new machines, new languages, …

  Parallelism must return to La-Z-Boy programming

[Sut05] Herb Sutter. "The Free Lunch Is Over: A Fundamental Turn Toward Concurrency," in *Software. Dr. Dobb's Journal*, vol. 30, no. 3, 2005.
[Pat06] David Patterson, in keynote talk at the International Workshop on Languages and Compilers For Parallel Computers *LCPC 2006, New Orleans, LA*

Moore's Law and beyond

- Moore's law of density still lives on
- How to use the Si resources in the face of constraints
- Dark silicon – exponentially increasing fraction of the chip cannot be turned on
-

(One) proposed solution: accelerators

- Specialized (domain specific) circuits
- Energy efficiently do the computation
- Powered off otherwise

Outline

- ~~Class objectives, goals~~
- ~~Why Fine grain parallelism?~~
- Equational Programming (intro)

Tuning (GTX 280 class GPU)

- Many resources on the web (NVIDIA webinars)
- Coalescing (HW1a)
  - Challenge question: Achieve maximum bandwidth, with fewest threads-per-block
  - For a "strided-by-block" access pattern.
- Arithmetic peak: warps and "virtualization"
- Bank conflicts in shared memory

CUDA Tuning resources

- Oxford CUDA conf (CUDA webinar online)
- "Identifying Performance Limiters," Micikevicius NVIDIA/UCF (CUDA webinar)
- "Roofline for Fast Math" Sam Williams, LBL

Equational Programming

- Wiki page for Pascal's Triangle

  http://en.wikipedia.org/wiki/Pascal's_triangle

- … and also a non-standard way to compute Fibonacci numbers