Exotic Methods in Parallel Computing
[Introduction]

Frank Feinbube, Fahad Khalid

Exotic Methods in Parallel Computing
Dr. Peter Tröger
Architectural Shift

- Memory Wall
- Hybrid Parallel Systems
- Multicore Systems
- Sequential Systems
- Power Wall & ILP Wall

Exotic Methods in Parallel Computing | FF 2012
Hybrid Computing

Flavors of massively multi-core systems:

- Radically new architectures under evaluation (Intel SCC)
- Accelerators accompany general purpose CPUs (Hybrid Systems)

Hybrid Systems

- **GPU Compute Devices**
  High Performance Computing
  (3 out of top 5 supercomputers are GPU-based!)
  - Business Servers, Home/Desktop Computers, Mobile and Embedded Systems

- **Special-Purpose Accelerators**
  - (de)compression, XML parsing, (en|de)cryption, regular expression matching
The beautiful new world of Hybrid Compute Environments
The beautiful new world of Hybrid Compute Environments

Hybrid Hardware is Everywhere

NVIDIA Tesla
NVIDIA ION
AMD Fusion
Intel Sandy Bridge
The beautiful new world of Hybrid Compute Environments

Huge Performance + Low Costs
The beautiful new world of Hybrid Compute Environments

Variety of Successful Applications

Medical, Finance, Military, Research, Video and Photo, Energy, ....
The beautiful new world of Hybrid Compute Environments

Standardized Programming Model

OpenCL
That would be too good to be for free!

Assumptions do not hold anymore.
Assumptions do not hold anymore

Assumption:
"Processors get faster"
"Software gets faster automatically"

FREE LUNCH IS OVER
Assumptions do not hold anymore

Assumption:
“Memory is Shared”

NUMA + Configurable Memory Hierarchies
Assumptions do not hold anymore

Assumption:
"Hardware copes with cache validity"

SOFTWARE MANAGED CACHES
Assumptions do not hold anymore

"Correctness can be seen in Code Reviews."

**RACE CONDITIONS + PRECISION HW-DEPENDENT**
Assumptions do not hold anymore

“Operating Systems:
* abstract hardware details away
* cope with: security, scheduling, distribution, …”

MANUAL ACCELERATOR MANAGEMENT
Assumptions do not hold anymore

Assumptions:

“Abstraction allows developers to focus on their sophistication”

“Application developers don’t need to understand the hardware”

**HYBRID PROGRAMMING MODELS REQUIRE A DEEP UNDERSTANDING OF THE HARDWARE**
The beautiful new world of Hybrid Compute Environments?

GETTING HUGE PERFORMANCE IS HARD

Huge Performance + Low Costs
The beautiful new world of Hybrid Compute Environments?

CURRENLY ONLY ACCURATE FOR GPUs
The beautiful new world of Hybrid Compute Environments?

Hybrid Hardware is Everywhere

NVIDIA Tesla
NVIDIA ION
AMD Fusion
Intel Sandy Bridge

Tianhe-1A
The beautiful new world of Hybrid Compute Environments?

Variety of Successful Applications

Medical, Finance, Military, Research, Video and Photo, Energy, ....
What can be done?
What does the future hold?
The beautiful new world of Hybrid Compute Environments

- Amazon Web Services
  - Hosted GPUs
  - GPU Cloud

- Peer 1 Hosting
  - Hosted GPUs
  - GPU Cloud
  - Hosted Reality Server

- Nimbix
  - Hosted GPUs
  - GPU on Demand

- Penguin Computing
  - Hosted GPUs
  - Hosted Reality Server

GPU Cloud Computing Service Providers
Intel Single-Chip Cloud Computer (SCC): a radically new architecture

- 24 tiles with two IA32 cores per tile
- A 24-router mesh network with 256GB bisection bandwidth
- 4 integrated DDR3 memory controllers
- Hardware support for message passing
- Power management:
  - 24 frequency islands, 6 voltage islands
- No hardware cache-coherency
SCC Tile Architecture

SCC has 24 dual-core tiles interconnected by mesh network.
Dynamically reconfigurable memory layout, no memory coherency protocols

- Original Pentium: Physical Address would be seen on address bus
- On SCC: Physical addresses mapped to **System addresses** via LookUp Table (LUT)

- No tile-local memory
  - Memory accesses via interconnection network
- 8-bit LUT index translates to:
  - 1-bit bypass
  - 8-bit route (msg tile)
  - 3-bit destination (mem ctrl)
  - 10-bit address extension
  - Up to 34 bit (16GB) accessible per memory controller
Memory Cubes, Memristors, ...

Computational Logic in the Memory
Seminar Overview

Lecture type: Project seminar
Team size: 1-3 students

Recommended:
- experience with Java or C++
- Lecture: Parallel Programming Concepts
  http://www.dcl.hpi.uni-potsdam.de/teaching/parProg/

Vertiefungsgebiete (SO2010): ITSE, OSIS
Seminar Outline

Block 1: Introduction

- Introduction to project topics
- Parallel Programming Concepts Overview
- Multi-Agent Systems and Concurrency
- GPU Computing Hands-On
- Project Assignments, Discussion about success criteria
Seminar Outline

Block 2: Project Phase

- Time for the teams to work on the projects.
  - Consulting hours at seminar time

- Three presentations per team:
  - Startup presentation
    - problem domain, problem, possible approaches
  - Progress presentation
    - selected approaches, achievements and challenges
  - Final presentation
    - solution, evaluation
Berkeley Dwarfs

Dwarf Popularity: **Red** = Hot; **Blue** = Cold

<table>
<thead>
<tr>
<th></th>
<th>Embed</th>
<th>SPEC</th>
<th>DB</th>
<th>Games</th>
<th>ML</th>
<th>HPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Finite State Mach.</td>
<td><strong>Red</strong></td>
<td>Yellow</td>
<td><strong>Red</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Combinational</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Graph Traversal</td>
<td><strong>Red</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Structured Grid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Dense Matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Sparse Matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Spectral (FFT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Dynamic Prog</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>N-Body</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>MapReduce</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Backtrack/ B&amp;B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Graphical Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Unstructured Grid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Health | Image | Speech | Music | Browser
Possible Projects

<table>
<thead>
<tr>
<th>Linear Algebra</th>
<th>Statistics (PCA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast-Fourier-Transformations</td>
<td>NBody</td>
</tr>
<tr>
<td>Nqueens, Sudoku, Wator, ...</td>
<td>Molecular Modeling</td>
</tr>
<tr>
<td>Sorting Algorithms</td>
<td>Logic Simulation</td>
</tr>
<tr>
<td>Encryption</td>
<td>Compression</td>
</tr>
<tr>
<td>XML Parsing</td>
<td>Anti-Virus Engine</td>
</tr>
<tr>
<td>Regular Expression Parsing</td>
<td>Image Processing</td>
</tr>
<tr>
<td>Monte Carlo</td>
<td>Speech recognition</td>
</tr>
<tr>
<td>Fractals</td>
<td>Hidden Markov Models</td>
</tr>
<tr>
<td>Fluids</td>
<td>Graphs</td>
</tr>
<tr>
<td>Ray Tracing</td>
<td></td>
</tr>
</tbody>
</table>
Possible Projects Mine

NAS Parallel Benchmarks
SHOC
Rodinia
PARSEC
HPC Challenge
PLASMA
Paraboil Benchmark suite

Alpbench, Biobench, Parkbench, Mediabench, Minebench, Bioparallel

Pjbench
Possible Projects Mine

- **Statistical constraints on binary black hole inspiral**
- **SofwaremGPU: Using graphics processing units for...**
- **BMC Bioinformatics**
- **Acceleration of the Smith-Waterman Algorithm using...**
- **Graphic processors to speed-up simulations for the...**
- **Cmatch: Fast Exact String Matching on the GPU**
- **Quantitative Risk Analysis and Algorithmic Trading...**
- **Harvesting graphics power for MD simulations**
- **Graphic-Card Cluster for Astrophysics (GraCCA)**
- **Quantum Chemistry Two-Electron Integral Evolution**
- **Accelerating Statistical Static Timing Analysis**
- **Distributed Password Recovery**
- **Folding@home**

http://www.nvidia.com/object/tesla_testimonials.html
Special Focus: Performance Optimization

Language: CUDA, OpenCL, X10, Java (Aparapi)

Course of Action

- Start Simple
- Identify Bottleneck / Optimization strategy
- Apply Performance Optimization

Please record for all versions for the final report:

- Data Access Patterns of the Algorithm
- Performance Optimization Steps
- Evaluation of Performance Improvements
Hybrid Computing for High-Level Programmers: The Hybrid Parallel Project

Access to Neighbors:
None, One, Two-Dimensional

Access to Neighbors: Simple vs. Ring

Strided Access: 1, 2, 4, 8

[Pattern], [Access], [Frequency], [Stride], [Neighbors], [Borders]

map

Private
Constant
Texture
Local
Global
Host
Cycles of War

- Multi-agent based strategy for dominating the universe 😊
- Game programming using *concurrent agents*, and perhaps some winning strategies developed in *Game Theory*

Multi-agent based Simulation (MABS)

- Mathematical models can get too complicated at times
- MABS gives us the possibility to utilize *emergence* to find possible solutions
Project ideas for Concurrent Multi-agent Systems ... contd.

Agent evolution

- Using *Genetic Algorithms/Genetic Programming* to simulate an agent’s DNA, and let the agent colony *evolve* to achieve some objective
- The objective does not *have* to be reproduction ... it can be any useful *objective function*

Artificial Neural Networks (ANN), Genetic Algorithms (GA) and GPUs

- Metric-based learning in ANNs
- *Machine Learning* meets evolution 😊
## Seminar Schedule: Block 1

<table>
<thead>
<tr>
<th>Schedule</th>
<th>Mo</th>
<th>Tu</th>
<th>We</th>
<th>Th</th>
<th>Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Agents by Fahad</strong></td>
<td>07.05.2012</td>
<td>08.05.2012</td>
<td>02.05.2012</td>
<td>03.05.2012</td>
<td>04.05.2012</td>
</tr>
<tr>
<td><strong>Agents by Fahad</strong> Wish for Projects</td>
<td>14.05.2012</td>
<td>15.05.2012</td>
<td>09.05.2012</td>
<td>10.05.2012</td>
<td>11.05.2012</td>
</tr>
<tr>
<td><strong>FREE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17.05.2012 Himmelfahrt</td>
</tr>
<tr>
<td><strong>GPUs by Frank</strong> Project Assignments Discussion about success criteria</td>
<td>21.05.2012</td>
<td>22.05.2012</td>
<td>16.05.2012</td>
<td>24.05.2012</td>
<td>18.05.2012</td>
</tr>
<tr>
<td>Schedule</td>
<td>Mo</td>
<td>Tu</td>
<td>We</td>
<td>Th</td>
<td>Fr</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Work on Projects</td>
<td>28.05.2012</td>
<td>29.05.2012</td>
<td>23.05.2012</td>
<td>25.05.2012</td>
<td></td>
</tr>
<tr>
<td>Consulting hours</td>
<td>Pfingstmontag</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Presentations</td>
<td>04.06.2012</td>
<td>05.06.2012</td>
<td>30.05.2012</td>
<td>31.05.2012</td>
<td>01.06.2012</td>
</tr>
<tr>
<td>Deadline for Enrollment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work on Projects</td>
<td>11.06.2012</td>
<td>12.06.2012</td>
<td>06.06.2012</td>
<td>07.06.2012</td>
<td>08.06.2012</td>
</tr>
<tr>
<td>Consulting hours</td>
<td>13.06.2012</td>
<td>14.06.2012</td>
<td>15.06.2012</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recapitulation / Snippets
Parallel Programming Concepts

Based on:

Lecture: Parallel Programming Concepts
http://www.dcl.hpi.uni-potsdam.de/teaching/parProg/