Tera-scale software research:
Programming 10s-100s of cores

Jerry Bautista, PhD
Co-director, Tera-scale Computing Research
Intel Corporation
What is Tera-scale?

Teraflops of performance operating on Terabytes of data

Emerging Apps

3D & Video

Multimedia

Multi-core

Text

Single-core

Kilobytes

Megabytes

Gigabytes

Terabytes

Performance

Dataset Size

Intel Developer FORUM
Example emerging Application

Many emerging apps are parallel and “model-based”
Ex: Modeled body, modeled motion, modeled lighting

See Justin’s keynote on “Virtual World” Thursday
Tera-scale Computing: A New Drive for Parallel Programming

Growing need for new parallel programming tools and capabilities
Joint Hardware & Software R&D

New Application Ideas → Prototype Threaded Application → Prototype Multi-core Architecture → New Architecture Ideas

Prototype Threaded Application

New Application Ideas

Refine Software Designs

Examples: Media search, Physical Modeling

Examples: Cache hierarchy, Task scheduling

Cycle-accurate Simulators
kHz speeds
~hours to modify

FPGA Emulators
MHz speeds
Days to modify

Silicon Prototypes
GHz speeds
Months to modify

ReSEARCH TOOLS

See Emulator demo in tech showcase
See 80-core demo in tech showcase
The Tera-scale Computing Vision

Parallel Programming Tools & Techniques

Virtual Environments
Educational Simulation
Financial Modeling
Media Search & Manipulation
Web Mining Bots*

Model-Based Applications

Thread-Savvy Execution Environment

Stacked, Shared Memory

Scalable Multi-core Architectures

High Bandwidth I/O & Communications
Taking Parallel Programming Mainstream

Used for decades in HPC, parallel programming requires
- Special expertise, not easily automated
- Requires parallel languages/language extensions
- Must co-exist with legacy code

Parallel languages or parallel language extensions need to:
- **Extract** parallelism hiding within applications
- **Express** parallelism via programming constructs
- **Exploit** parallelism on multi-core platforms
Types of Parallelism

- **Task**: multiple independent activities, which may or may not share data

- **Data**: one or few tasks operating on a large amount of data
  - Will review Ct later in presentation
  - Subject of demo and class (TCRS0003)
# Common Parallel Programming Models

<table>
<thead>
<tr>
<th>Programming model</th>
<th>Example Application Today</th>
<th>Example Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Passing</td>
<td>HPC type apps on a cluster</td>
<td>MPI</td>
</tr>
<tr>
<td>Shared-memory or Thread</td>
<td>General Par Tasks, Database</td>
<td>OpenMP</td>
</tr>
<tr>
<td>Data Parallel</td>
<td>Mathematical simulations</td>
<td>MatLab</td>
</tr>
<tr>
<td>Streaming</td>
<td>Graphics shaders</td>
<td>$C_G$</td>
</tr>
</tbody>
</table>
Diverse Programming Environment

Historically, new languages often emerge and compete. Adoption can be slow and success hard to predict.

Lesson: No one language will be a silver bullet that solves the parallel programming challenges
Research to Help Enable Parallelism

• **STM (Software Transactional Memory)**
  - Shared memory model is dominant but problematic due to synchronization of locks in highly parallel environments

• **Ct (C for throughput computing)**
  - Relieves need to worry about threads in data parallel execution models providing parallel extensions to C and C++

• **Exo (Accelerator Exoskeleton)**
  - Bringing IA “look and feel “ of common development environments to parallel, *heterogeneous* environments.
Unlocking Parallelism in a Shared Memory Environment

Must carefully control how multiple threads access shared memory

Today we “lock” memory for one thread at a time.

- Other threads must wait, reducing multi-core benefit
- Locking code scales poorly, must re-do for more threads
- Can cause critical software deadlocks and errors
- We need to fix this...

Account locked during access

A $50
B $200
C $200

2003 Northeast blackout

Mars rover problem

Presented by Justin Rattner at IDF Day0, Spring 2006
STM: From Research to Reality

Ensures correct parallel memory access without locks

- Greater performance
- Easier to program
- Scales with hardware

Programmers can try it!

AVAILABLE STARTING TODAY
Whatif.intel.com
Intel® C++ STM Compiler Prototype Edition
Ct: Nested Data Parallel Programming

Ct adds *new parallel* data structures & operators to C/C++; But uses *existing & unmodified* C/C++ compilers

C/C++ Compiler

Ct-based Parallel Data Types

Scalable, Adaptive Performance

Tera-scale
Ct Motivation and Vision

- Make parallel programming easier now:
  - Extend *deterministic* parallel programming models
    - I.e. Data races not possible
  - Express complex behaviors through simple operators
  - Present a simple and predictable performance model

- Provide a forward-scaling programming model that maximizes ISV ROI for new code creation
  - “Future-proof” apps from increasing core count and inevitable ISA evolution

See today’s post by Anwar Ghuloum at blogs.intel.com/research for more info
Programming for Heterogeneous Cores

- Multi-core brings the opportunity to integrate fixed function accelerators with IA cores
- Implementation difficult and very HW specific

Accelerators today rely on custom drivers bound tightly to the OS.

The software layers between the app accelerator cause performance-limiting overhead
Programming Accelerators Today

**SW DEVELOPMENT**

1. Obtain custom kits from hw vendor
2. Learn accelerator tools, language
3. Compile accelerator program
4. Accelerate!
5. Compile host IA program
6. Host!

**SW EXECUTION**

1. MAIN MEMORY
   - Host loads accelerator program
   - Accelerate!

2. HOST
   - Abstraction Layer
   - OS
   - Driver
   - Send commands via driver interface
   - Accelerate!

3. DEVICE MEMORY
   - Abstraction Layer
   - OS
   - Driver
   - Results transferred from device memory
   - Accelerate!

4. RESULTS!

5. Intel Developer FORUM
Exo - Accelerator Exoskeleton Model

The exoskeleton makes accelerators appear like a part of the processor

**SW DEVELOPMENT**

- Use standard tools to design code
- Program accelerator(s) like IA extensions
- Compile single program

**SW EXECUTION**

- Application
- Operating System
- Multi-threading Runtime Library
- ISA with Accelerator Exoskeleton Extensions

Program

Intel Developer FORUM

18
Demonstrations

1. Video enhancement application (de-interlacing)

- Code for graphics in standard IDE
- Use Intel compiler to build a single "exe"
- Exoskeleton environment divides the work

2. Similar demo with a financial analytics application
Convergence of Many Parallel Apps

RMS Taxonomy
- Recognition
- Mining
- Synthesis

MODEL-BASED Computing

Security Biometrics
Cancer Detection
Media Indexing
Web search

Body Tracking
Interactive Virtual Worlds
Financial Predictions
Data Warehousing

Facial Animation
Ray Tracing

MATHEMATICAL MODELS & METHODS
- Vector Training
- Optimization Methods
- Monte Carlo
- Geometric Structures
- Classifiers
- Collision Detection

COMMON NUMERICAL METHODS & DATA STRUCTURES

Key architectural enhancements based on application and user needs

Intel Developer FORUM
Independent IDC Market Analysis

• Spending on computing continues to grow

• New usage models are emerging

• IDC working with Intel and others to understand this space
  - "IDC believes that new use cases for computing are emerging which will drive significant growth...Our research has shown that highly parallel applications and multi-core processors are key drivers enabling this emerging trend."
    - Matt Eastwood, Group Vice President, IDC Enterprise Platform Research

Full report to be published in October
Summary

• The key challenges to parallel programming
  - Programmability and scalability
  - Interoperability and integration of heterogeneous execution blocks

• Parallel programming is the key to unlocking the full potential of the Tera-scale platforms
  - Mainstream programmers enabled through broad set of development and optimization tools.
  - Technologies developed to extend well-known programming environments (Ct, Exo and STM)
  - Tools and technologies are already being deployed now:
    ▪ TBB – threadingbuildingblocks.org
    ▪ STM on Whatif.intel.com

• Model-based computing applications are compelling with market projections supporting potential for broad adoption
More information on Tera-scale...

Tera-scale Computing Research Chalk Talk

Chair: Jerry Bautista, Co-Director, Tera-scale Computing Research

TCRC001: Wed, Sept. 19, 4:40 - 5:30, Chalk Talk Room

Other Tera-scale Sessions (see IDF guide for full info)

- TCRS001 Energy Management Innovations for Future Multi-Core Processors
- TCRS002 Intelligent On-chip Interconnects: The 80-core Prototype and Beyond
- TCRS003 Data Parallel Programming for Tera-scale with Ct
- TCRS004 Modeling Reality: Ray Traced Graphics and other Apps of the Future
- TCRS005 Silicon Photonics: Enabling Terabit Data Pipes
- QATS003 Accelerator Exoskeleton: IA Look-n-Feel for Heterogeneous Cores

Tech & Research Pavilion Demos:

80-core Processor, Multi-core emulator, Ct, Log-based Architectures, Ray Tracing

Learn more at www.intel.com/go/terascale and blogs.intel.com/research