Apply Fine-Grain Adaptive Multithreading to Irregular Applications

Guang R. Gao
Computer Arch. and Parallel System Laboratory
University of Delaware
(www.capsl.udel.edu)
Outline

• Introduction: Class-A and Class-B Applications
• Fine-Grain Multithreading – A Clarification
• EARTH and Beyond: A Case Study of Fine-Grain Multithreading (*EARTH-MANNA*)
• Examples of Irregular Applications
• Blue-G/L and Fine-Grain Multithreading
• Summary
The “Memory-Wall” Problem

Performance

“best performance”

“reasonable performance”

man x years

Effort + expert Knowledge

A1

A2

An
What is a **Class B** Application?

- Data dependence: *irregular/dynamic*
- Data access pattern: *irregular/dynamic*
- Control flow: *irregular/dynamic*
- Computation load evolution: *irregular/dynamic*
- Others?
Memory Wall: 4 Types of Latencies

- Memory access/communication latency
- Synchronization latency
- Task spanning/termination latency
- Task migration latency
- Others?
Research Layout

Future Programming Model and System Software

- Scientific Computation
- High-performance Bio-computing
- Other High-end Applications

Advanced Execution/Programming Model
- Percolation
- Location Consistency

Base Execution Model
- Fine-Grain Multithreading (e.g. EARTH, CARE)

Infrastructure and Tools
- System software
- Simulation/Emulation
- Analytical modeling

Bluegene L/C Architectures

PIM Based Architectures

Other Architectures

8/14/2002

Bluegene-08-2002
Coarse-Grain vs. Fine-Grain Multithreading

Coarse-Grain thread-
The family home model

Fine-Grain thread-
The “hotel” model
**EARTH: a Fine-Grain Multithreaded Execution Model**

Two Level of Fine-Grain Threads:
- threaded procedures
- fibers

- fiber within a frame
- Parallel function invocation
- A sync operation
- Invoke a threaded func

8/14/2002

Bluegene-08-2002
Fiber States Transition

- DORMANT
  - Fiber created
  - Fiber terminated
  - Synchronizations received
  - Fiber completed

- ENABLED

- ACTIVE
  - EU ready

- EU ready
The EARTH Virtual Machine Model

- EARTH node consists of an Execution Unit and a Synchronization Unit (SU)
- EU executes active threads
- SU handles synchronization and scheduling of threads, and communication
- Ready queue, event queue, and token queue
The **Threaded-C** Language – Defining the API for EARTH Virtual Machine

- Threaded C = ANSI C + extensions for multithreading
- Extensions include:
  - Threaded functions
  - Threaded synchronization
  - Support for global address space
  - Data transfer primitives
- Threaded-C is:
  - The “instruction set” of the EARTH PXM
  - A target language for high-level compilers
If n < 2
    DATA_RSYNC (1, result, done)
else
    {
      TOKEN (fib, n-1, & sum1, slot_1.1);
      TOKEN (fib, n-2, & sum2, slot_1.2);
    }
END_THREAD();

THREAD-1;
    DATA_RSYNC (sum1 + sum2, result, done);
END_THREAD();

END_FUNCTION

The Fibonacci Example in Threaded-C
Features of Fine-Grain Threaded Programming

- Thread formation
  - Thread length vs useful parallelism
  - Where to “cut”?  
- Split-phase synchronization and communication
- Parallel threaded function invocation
- Dynamic load balancing
- Other advanced features
The EARTH Operation Set

- The base operations
- Thread synchronization and scheduling ops
  SPAWN, SYNC
- Split-phase data & sync ops
  GET_SYNC, DATA_SYNC
- Threaded function invocation and load balancing ops
  INVOKE, TOKEN
EARTH-C and Threaded-C

• Design simple high-level extensions for C that allow programmers to write programs that will run efficiently on multi-threaded architectures. (EARTH-C)

• Develop compiler techniques to automatically translate programs written in EARTH-C to multi-threaded programs. (EARTH-C, Threaded-C)

• Determine if EARTH-C + compiler can compete with hand-coded Threaded-C programs.
An Evolutionary Path for EARTH

- Parallel machines
- PC-clusters
- ...

MANNA-dual/spn

SU-int

SU-ext

SEMi Simulation Platform (Theobald99)
The EARTH-MANNA Multiprocessor Testbed

- no “traditional OS
- EARTH runtime system management the CPs
- system calls are handled by host nodes
- for users: the entire CPs are viewed as a single compute engine
- asynchronous events and the “polling-watchdog”

The cache coherence between the two CPs affects the performance of the EARTH RTS
Performance Study Platform #1 – EARTH on Chiba City Cluster

- **DOE/ANL Chiba City Cluster**
- 256 dual-CPU Pentium III 500 MHz Computing Nodes with 512 MB of RAM and 9G of local disk.
- Switched Fast Ethernet for file service and management functionality connecting all computing nodes.

Whole Genome Comparison
ATGC (EARTH Threaded-C version) on Chiba City Cluster

<table>
<thead>
<tr>
<th>Number of Processors</th>
<th>Absolute Speedups</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Seq 4</td>
<td>10</td>
</tr>
<tr>
<td>Seq 8</td>
<td>20</td>
</tr>
<tr>
<td>Seq 16</td>
<td>30</td>
</tr>
<tr>
<td>Seq 32</td>
<td>40</td>
</tr>
<tr>
<td>Seq 64</td>
<td>50</td>
</tr>
<tr>
<td>Seq 90</td>
<td>60</td>
</tr>
</tbody>
</table>

EARTH Runtime Model

- Application
- Execution Module
- Receiver Module
- Sender Module
- Network
- Ready Queue
- Send Queue
- Token Queue
Open64/Kylin Compiler Infrastructure

CAPSL Extension/Plug-Ins
- Transformation
- Optimization
- Percolation
- Location Consistency

Architecture Model
- Uniprocessor Model
  - Cache
  - Pipeline
- Multiprocessor Model
  - Multithreading
  - Memory Model
  - Percolation Model

High-level loop transformations
High-level Multithreading Transformations

MultiThreaded Code Generation & Optimization

Kylin Code Optimizer
- Itanium
- IBM Cyclops
- Others
Performance Evaluation Through Multiple Methods

- Bottleneck Analysis (selected computation kernels)
- Simulation (e.g. SEMi and Its Extensions)
- Emulation (e.g. EARTH-Cluster style)
- Analytical Performane Model (e.g. Numarwarkar97, etc.)
- A Judiciary combination of the above
Important Benchmarks Studied

- Example 1: Gannon’s algorithm for parallel matrix multiply (Theobald99)
- Example 2: Adaptive unstructured grids (IPDPS99, Irregular99, Thulasiram00)
- Example 3: Wavelet computation (IPDPS99, Thulasiram00)
- Example 4: FFT computation (SPAA00, Thulasiram00)
- Example 5: Conjugate Gradient (CG) Code (EuroPar00, SC00)
- Example 6: Genome/Protein Sequence Comparison (Smith-Waterman/Needleman-Wunch Alg.) (PSB00, RECOMB01)
- Others (e.g. EBS benchmarks)
Performance of N-Queens(12)

- Achieved high absolute efficient under EARTH-MANNA: 117.8 fold speedup on a 120 node Machine!
- 1,637,099 tokens are generated!
- Average, 30+ tokens are maintained per processors
- Fine-grain two-level multithreading
Case Study: Conjugate Gradient

- Dominant part is repeated sparse matrix-vector multiply
- Sequential section insignificant
- Inner products and scaling must be efficient
Implementation in Threaded-C

Two independent copies of this structure
Global Reduction & Broadcast
CG: Observations

• CG mapped very efficiently under EARTH – achieve > 75% absolute efficiency on EARTH-MANNA

• Fine-grain multithreading and synchronization methods play a significant role in achieving this performance.
An Advanced Program Execution Model

- The *Percolation* Model: A *dynamic and adaptive* parallel execution model
EARTH on Bluegene/L – some thoughts

• EARTH appears to be an interesting addition to the current Bluegene/L programming models
• EARTH Virtual Machine should be able to be mapped efficiently onto the current Bluegene/L dual-processor node architecture (do not need the L1 coherence)
• Other research topics enabled by EARTH on BG//L
Acknowledgements

- NSF: the NGS program (D. Frederica)
- DOE: SciDAC P-Model Center (F. Johnson, PI: R. Lusk)
- Other funding sources
- Members of CAPSL
- Our collaborators