Computer Architecture II
Lecture 2007-11-20

Parallelization on Chip Multiprocessors

Virtutech Simics, Introduction

Dan Ekblom
dan.ekblom@virtutech.com
Optimizing codes for CMPs:

Temporally blocked
Gauss-Seidel smoothers
SMP vs CMP

SMP

<table>
<thead>
<tr>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2</td>
</tr>
<tr>
<td>L1</td>
</tr>
<tr>
<td>CPU</td>
</tr>
</tbody>
</table>

Interconnect

CMP

<table>
<thead>
<tr>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2</td>
</tr>
<tr>
<td>L1</td>
</tr>
<tr>
<td>CPU</td>
</tr>
</tbody>
</table>

CMPs:

😊 Communication is cheap
😊 Cache/thread is small
Equation system: \( Ax = b \)

\[
x_i^{(k)} = b_i - \sum_{j<i} a_{ij} x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k-1)}
\]

Gauss-Seidel iterative method
Natural order Gauss-Seidel

- = sweep path
- = previous
- = current
- = future
- = data dependence
1 = iteration number
- = cacheline layout
Natural order Gauss-Seidel

- sweep path
- previous
- current
- future
- data dependence
- iteration number
- cacheline layout
Natural order Gauss-Seidel

- sweep path
- = previous
- = current
- = future
- = data dependence
- 1 = iteration number
- = cacheline layout
Natural order Gauss-Seidel

if (convergence_test)
  <done>
else
  <iterate again>
Natural order Gauss-Seidel

- Data dependencies
- 1 step per sweep

= sweep path
○ = iteration number
● = current
= data dependence
= cacheline layout
Red-black Gauss-Seidel, step 0.5

- = sweep path
- = iteration number
- = current
- = data dependence
Red-black Gauss-Seidel, step 0.5

1 = iteration number

= sweep path

= current

= data dependence
Red-black Gauss-Seidel, step 0.5

= sweep path

= iteration number

= current

= data dependence
Red-black Gauss-Seidel, step 0.5

- = iteration number
- = current
- = sweep path
- = data dependence
Red-black Gauss-Seidel, step 1.0

- iteration number
- current
- sweep path
- data dependence

Red-black Gauss-Seidel step 1.0:

1. 1 1 1 1
2. 1 1 1 1
3. 1 1 1 1
4. 1 1 1 1
5. 1 1 1 1
6. 1 1 1 1
7. 1 1 1 1
8. 1 1 1 1
9. 1 1 1 1
10. 1 1 1 1
11. 1 1 1 1
12. 1 1 1 1
13. 1 1 1 1
14. 1 1 1 1
15. 1 1 1 1
16. 1 1 1 1
17. 1 1 1 1
18. 1 1 1 1
19. 1 1 1 1
20. 1 1 1 1
Red-black Gauss-Seidel, step 1.0

- iteration number
- current
- data dependence

= sweep path

1 = iteration number

1 = current

= data dependence
Red-black Gauss-Seidel, step 1.0

- No dependencies between red and black points
- 0.5 steps per sweep
Parallel Red-black

IN PARALLEL {
    Update all blacks
    <barrier>
    Update all reds
    <barrier>
}
Gauss-Seidel behavior

Red-Black Gauss-Seidel

إجراءات كاسح: - البيانات مُحمَّلة مرتين في كل دورة

مباشرة لسيديل - البيانات مُحمَّلة مرتين في كل دورة

交流合作 -خفض

Natural order Gauss-Seidel

إجراءات كاسح: - البيانات مُحمَّلة مرتين في كل دورة

交流合作 - difficul to parallelize

Temporally blocked natural Gauss-Seidel

إجراءات كاسح: - البيانات مُحمَّلة مرتين في كل دورة

交流合作 - communication intense
Gauss Seidel temporally blocked

- = execution path
- = current
- = data dependence
- = iteration number
- = cacheline layout
- = active region

Iteration numbers: 1, 2, 3, 4

Current iteration: 4
Gauss Seidel temporally blocked

- current
- execution path
- data dependence
- iteration number
- cacheline layout
- active region
Gauss Seidel temporally blocked

- execution path
- current
- data dependence
- iteration number
- cacheline layout
- active region

iteration numbers: 1, 2, 3, 4

Gauss Seidel temporally blocked

Gauss Seidel temporally blocked
Gauss Seidel temporally blocked

= execution path

● = current

= data dependence

1 = iteration number

= cacheline layout

= active region
Gauss Seidel temporally blocked

= execution path

= current

= data dependence

= iteration number

= cacheline layout

= active region
Gauss Seidel temporally blocked

- execution path
- current
- data dependence
- iteration number
- cacheline layout
- active region
Gauss Seidel temporally blocked

- execution path
- current
- data dependence
- iteration number
- cacheline layout
- active region
Gauss Seidel temporally blocked

- execution path
- current
- data dependence
- iteration number
- cacheline layout
- active region
Gauss Seidel temporally blocked

= execution path

= current

= data dependence

= iteration number

= cacheline layout

= active region
StatCache, 3D, N=129

Cache miss ratio (percent)

Cache size

512 KB  1 MB  2 MB  4 MB  8 MB  16 MB  32 MB

RBGS 0.5  TBGS 1  TBGS 2  TBGS 4  TBGS 8  TBGS 16
### Sequential Execution time per step

(Sun Fire 15k, 1.5 MHz US4+, $=64kB/2MB/32MB)

<table>
<thead>
<tr>
<th></th>
<th>RBGS</th>
<th>TBGS</th>
<th>TBGS</th>
<th>TBGS</th>
<th>TBGS</th>
<th>TBGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>$N=129$</td>
<td>0.091</td>
<td>0.091</td>
<td>0.076</td>
<td>0.067</td>
<td>0.076</td>
<td>0.079</td>
</tr>
<tr>
<td>$WS=32MB$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N=257$</td>
<td>2.476</td>
<td>1.573</td>
<td>1.104</td>
<td>0.869</td>
<td>0.752</td>
<td>0.694</td>
</tr>
<tr>
<td>$WS=128MB$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N=513$</td>
<td>19.93</td>
<td>12.64</td>
<td>9.419</td>
<td>7.827</td>
<td>10.30</td>
<td>12.95</td>
</tr>
<tr>
<td>$WS=512MB$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Required data size for each active region

### TBGS

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>blocks</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>N=129</td>
<td>0.5 MB</td>
<td>0.76 MB</td>
<td>1.3 MB</td>
<td>2.3 MB</td>
<td>4.3 MB</td>
</tr>
<tr>
<td>WS=32MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=257</td>
<td>2.0 MB</td>
<td>3.0 MB</td>
<td>5.0 MB</td>
<td>9.1 MB</td>
<td>17 MB</td>
</tr>
<tr>
<td>WS=128MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=513</td>
<td>8.0 MB</td>
<td>12 MB</td>
<td>20 MB</td>
<td>36 MB</td>
<td>68 MB</td>
</tr>
<tr>
<td>WS=512MB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Parallel temp block natural G-S

Synchronization flags

- = execution path
- = current
- = data dependence
1 = iteration number
- = cacheline layout
- = active region
Parallel temp block natural G-S

- execution path
- current
- data dependence
- iteration number
- cacheline layout
- active region

Synchronization flags
Parallel temp block natural G-S

- thread 0
- thread 1
- thread 2

- execution path
- current
- data dependence
- iteration number
- cacheline layout
- active region

Synchronization flags

---

dan.ekblom@virtutech.com

Nov 20, 2007
Parallel temp block natural G-S

Synchronization flags

+ Low cache miss ratio
- A lot of communication
  - updating sync flags
  - exchanging data

= execution path
● = current
= data dependence
1 = iteration number
= cacheline layout
= active region

thread 0 thread 1 thread 2
Parallel temp block natural G-S

- current
- execution path
- data dependence
- iteration number
- cacheline layout
- active region

Synchronization flags

thread 0 | thread 1 | thread 2
---|---|---
1 | 1 | 0
2 | 2 | 0
3 | 3 | 0
Parallel G-S 3D
Parallel G-S 3D
Parallel G-S 3D

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>t1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Parallel G-S 3D
Parallel execution time, 3D, N=257

- **RBGS (RedBlack GS)**
- **TBGS (Temp. Blocked GS)**

Execution time per step (sec)

- threads=1
- threads=2
- threads=4
- threads=8

Values:
- 0.5
- 1
- 2
- 4
- 8
- 16
Performance ratio on 32-way Sun Fire 15k

(Sun Fire 15k, 1.5 MHz US4+, $=64kB/2MB/32MB)
Performance ratio, pure CMP, "SMP"

The performance ratio of TBGS 1 over RBGS 0.5 is shown in the graph. The performance ratio for pure CMP (blue squares) remains relatively constant as the number of threads increases. The performance ratio for "SMP" (pink triangles) also remains relatively constant but at a higher level compared to pure CMP.

Thread count is on the x-axis, ranging from 0 to 30, and the performance ratio is on the y-axis, ranging from 0.0 to 2.5.
Performance ratio, UltraSPARC T1

- UltraSPARC T1 - TBGS 1
- UltraSPARC T1 - TBGS 2
- UltraSPARC T1 - TBGS 4

Threads

Performance ratio TBGS/RBGS

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

0.0 0.5 1.0 1.5 2.0 2.5
Gauss-Seidel inefficient alone…

G-S: Smoother in multigrid method
- Algorithm requires $O(N)$ arithmetic operations

**Multigrid V-cycle:**
1. Apply smoother $\gamma$ times
2. Compute defect
3. Restrict
4. Recursively call Multigrid V-cycle
5. Prolong
6. Correct solution
7. Apply smoother $\gamma$ times
Conclusions

Efficient Parallellization on Chip Multiprocessors:

- Communication is free on chip!
- Optimize for cache usage!

=>

Parallel code may have to be rewritten on CMPs for optimal performance!
Virtutech Simics
Introduction
Virtutech technology

- Full system simulation
  - Complete machines, networks, backplanes
  - System-level from the beginning
- Runs complete software stack
  - Firmware, device drivers, OS, hypervisor, etc…
- Very high performance
  - Typically 100s of MIPS
Simics is a System Simulator
Simics is a System Simulator
Simics is a System Simulator
Traditional Software Development

- Software development methodology creates production binary
- Production binary runs on the real hardware
Virtualized Software Development

- Same binary runs inside virtualized software development environment

The software can’t tell the difference

Virtutech Simics Virtual HW
Simics: Full-System Simulation

Identical build tools chain
The software can't tell the difference
Runs binaries from real target

Complete production software
Simulated hardware

User program
Server | DB | Middleware

Operating system
Drivers | Firmware

Hardware
CPU
RAM
ROM
Bus
PCI
I2C
Disk
Disk Ctrl
FLASH
Network
LCD
ASIC

The software can't tell the difference between simulated and real hardware.

Identical build tools chain
Runs binaries from real target

Identical build tools chain
The software can't tell the difference
Runs binaries from real target
Complete Virtualization

- **Host hardware**
- **Host operating system**
- **Simics**
  - **Simulated target hardware**
  - **Target operating system**
  - **User program**

**All software**: arbitrary & unmodified. Same as on a real system.

**Arbitrary**: Alpha, ARM, IA64, x86, x86-64, SparcV8, SparcV9, MIPS32, MIPS64, PPC-32, PPC-64, MSP430, TI C6400

- **PC (32- & 64-bit)** or Sparc
- **Linux, Solaris, Windows**

*Nov 20, 2007*
Simics features

- **Checkpointing**
  - Store current state; pick up and continue later
  - Position workload once, use many times
  - Spread a system state to multiple developers
  - Can checkpoint an entire network of machines

- **Determinism**
  - Same initial state gives same execution;
  - Repeat the same execution any number of times
  - Investigate a problem time after time
  - For multiprocessor systems & network systems
Simics features

- **Visibility (insight without intrusion)**
  - All state can be observed
  - All events can be traced and logged

- **Controllability**
  - Any part of machine or state can be changed
  - Fault injection

- **Virtual time**
  - Time is completely virtual
  - Global synchronization across all machines in a network
  - Global stop across all processors in a multiprocessor
Simics features

- **Configurability**
  - Any parameter of system can be changed

- **Sandboxing**
  - Simulated machine complete isolated
  - Allows investigating "nasty code"

- **Backwards debugging**
  - Roll back execution to previous state
  - Reverse breakpoints
  - Investigate details of program errors
What does it look like?

- **Target console**
  - Programs
    - Complete and unmodified
  - Virtual Target HW
  - Unmodified target operating system
  - Simics
  - Host operating system
  - Host hardware
Demo
Cache modeling in Simics

Basic model
1 instr = 1 cycle
No cache, perfect memory
100+ MIPS speed

Cache model
Compute instr = 1 cycle
Memory instr = cache time
Cache statistics & traces
1+ MIPS speed
Virtutech

- About 60 employees
- Head quarter in San Jose, California
- All engineering in Stockholm
- 30+ people in Stockholm
Virtutech looks for thesis students (exjobbare)!
http://www.virtutech.com/about/academia/masters-thesis-list.html

Example of projects:
- Performance evaluation of distributed simulation on the Ericsson CPPemu platform
- Simulating an Intel MacOS computer
- Translation and integration of external device models in Simics
- Identifying shared data structures for avoiding race conditions
- Tracing network traffic and a Linux TCP/IP stack
- Application environment testing in a simulator, creating methods and tools
- Optimize device models in Simics for better performance
- Study efficient simulation techniques of CPUs with exposed pipelines, e.g. DSPs