OSCAR Compiler for Automatic Parallelization and Power Reduction for Multicores and Manycores

Hironori Kasahara
Professor, Dept. of Computer Science & Engineering
Director, Advanced Multicore Processor Research Institute
Waseda University, Tokyo, Japan
IEEE Computer Society Board of Governors
URL: http://www.kasahara.cs.waseda.ac.jp/
Green Computing Systems R&D Center
Waseda University
Supported by METI (Mar. 2011 Completion)

<R & D Target>
Hardware, Software, Application for Super Low-Power Manycore Processors
- More than 64 cores
- Natural air cooling (No fan)
  Cool, Compact, Clear, Quiet
- Operational by Solar Panel

<Industry, Government, Academia>
Hitachi, Fujitsu, NEC, Renesas, Olympus, Toyota, Denso, Mitsubishi, etc

<Ripple Effect>
- Low CO₂ (Carbon Dioxide) Emissions
- Creation Value Added Products
  - Consumer Electronics, Automobiles, Servers

Hitachi SR16000:
  Power7 128coreSMP
Fujitsu M9000
  SPARC VII 256 core SMP

Beside Subway Waseda Station,
Near Waseda Univ. Main Campus
Research, development and practical utilization through industry-government-academia partnerships (spillover effect)

Environment

Industry

Lives

Cancer Treatment
Carbon Ion Radiotherapy
Generated Solar Power and Server Power Consumption on April 2, 2012.4.2 (Clear) in Green Computing Center
Super Low Power Web Server Using Embedded Multicore Processor RPX

1W with 8 SH4A processor cores
Multi/Many-core Everywhere

Multi-core from embedded to supercomputers

- **Consumer Electronics (Embedded)**
  - Mobile Phone, Game, TV, Car Navigation, Camera,
    - IBM/ Sony/ Toshiba Cell, Fujitsu FR1000,
    - Panasonic Uniphier, NEC/ARM MPCore/MP211/NaviEngine,
    - Renesas 4 core RP1, 8 core RP2, 15core Hetero RP-X,
    - Plurality HAL 64(Marvell), Tilera Tile64/ -Gx100(->1000cores),
    - DARPA UHPC (2017: 80GFLOPS/W)

- **PCs, Servers**
  - Intel Quad Xeon, Core 2 Quad, Montvale, Nehalem(8cores),
    - Larrabee(32cores), SCC(48cores), Night Corner(50 core+:22nm),
    - AMD Quad Core Opteron (8, 12 cores)

- **WSs, Deskside & Highend Servers**
  - IBM(Power4,5,6,7), Sun (SparcT1,T2), Fujitsu SPARC64fx8

- **Supercomputers**
    - BG/Q (A2:16cores) Water Cooled20PFLOPS, 3-4MW (2011-12),
    - BlueWaters(HPCS) Power7, 10 PFLOP+(2011.07),
    - Tianhe-1A (4.7PFLOPS,6coreX5670+ Nvidia Tesla M2050),
    - Godson-3B (1GHz40W 8core128GFLOPS) -T (64 core,192GFLOPS:2011)
    - RIKEN Fujitsu “K” 10PFLOPS(8core SPARC64VIIIfx, 128GGFLOPS)

High quality application software, Productivity, Cost performance, Low power consumption are important

Ex, Mobile phones, Games

Compiler cooperated multi-core processors are promising to realize the above futures

---

The 37th (Nov. 20,2011) & 38th (Nov.14,2011) Top 500 No.1,
Riken Fujitsu “K” 705,024 cores
Peak 11.28 PFLOPS, (88,128procs)
LINPACK 10.510 PFLOPS (93.2%)
OSCAR Parallelizing Compiler

To improve **effective performance, cost-performance and software productivity and reduce power**

**Multigrain Parallelization**

coarse-grain parallelism among loops and subroutines, near fine grain parallelism among statements in addition to loop parallelism

**Data Localization**

Automatic data management for distributed shared memory, cache and local memory

**Data Transfer Overlapping**

Data transfer overlapping using Data Transfer Controllers (DMAs)

**Power Reduction**

Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.
Generation of coarse grain tasks

- **Macro-tasks (MTs)**
  - Block of Pseudo Assignments (BPA): Basic Block (BB)
  - Repetition Block (RB): natural loop
  - Subroutine Block (SB): subroutine
Earliest Executable Condition Analysis for coarse grain tasks (Macro-tasks)

A Macro Flow Graph

A Macro Task Graph
Automatic processor assignment in su2cor

- Using 14 processors
  - Coarse grain parallelization within DO400 of subroutine LOOPS

\[ N_{PG}, N_{PE} = [PG,PE] \]
MTG of Su2cor-LOOPS-DO400

- Coarse grain parallelism PARA_ALD = 4.3
Data-Localization
Loop Aligned Decomposition

- Decompose multiple loop (Doall and Seq) into CARs and LRs considering inter-loop data dependence.
  - Most data in LR can be passed through LM.
  - LR: Localizable Region, CAR: Commonly Accessed Region
Data Localization

MTG

MTG after Division

Data Localization Group

dlg0
dlg1
dlg2
dlg3

A schedule for two processors
Data Layout for Removing Line Conflict Misses by Array Dimension Padding

Declaration part of arrays in spec95 swim

**before padding**

PARAMETER (N1=513, N2=513)

COMMON U(N1,N2), V(N1,N2), P(N1,N2),
*        UNEW(N1,N2), VNEW(N1,N2),
1        PNEW(N1,N2), UOLD(N1,N2),
*        VOLD(N1,N2), POLD(N1,N2),
2        CU(N1,N2), CV(N1,N2),
*        Z(N1,N2), H(N1,N2)

**after padding**

PARAMETER (N1=513, N2=544)

COMMON U(N1,N2), V(N1,N2), P(N1,N2),
*        UNEW(N1,N2), VNEW(N1,N2),
1        PNEW(N1,N2), UOLD(N1,N2),
*        VOLD(N1,N2), POLD(N1,N2),
2        CU(N1,N2), CV(N1,N2),
*        Z(N1,N2), H(N1,N2)

Box: Access range of DLG0
Power Reduction by Power Supply, Clock Frequency and Voltage Control by OSCAR Compiler

- Shortest execution time mode

- Realtime processing mode with dead line constraints
An Example of Machine Parameters for the Power Saving Scheme

- **Functions of the multiprocessor**
  - Frequency of each proc. is changed to several levels
  - Voltage is changed together with frequency
  - Each proc. can be powered on/off

<table>
<thead>
<tr>
<th>state</th>
<th>FULL</th>
<th>MID</th>
<th>LOW</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency</td>
<td>1</td>
<td>1/2</td>
<td>1/4</td>
<td>0</td>
</tr>
<tr>
<td>voltage</td>
<td>1</td>
<td>0.87</td>
<td>0.71</td>
<td>0</td>
</tr>
<tr>
<td>dynamic energy</td>
<td>1</td>
<td>3/4</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>static power</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- **State transition overhead**  (Example: not for RP2)

<table>
<thead>
<tr>
<th>state</th>
<th>FULL</th>
<th>MID</th>
<th>LOW</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL</td>
<td>0</td>
<td>40k</td>
<td>40k</td>
<td>80k</td>
</tr>
<tr>
<td>MID</td>
<td>40k</td>
<td>0</td>
<td>40k</td>
<td>80k</td>
</tr>
<tr>
<td>LOW</td>
<td>40k</td>
<td>40k</td>
<td>0</td>
<td>80k</td>
</tr>
<tr>
<td>OFF</td>
<td>80k</td>
<td>80k</td>
<td>80k</td>
<td>0</td>
</tr>
<tr>
<td>delay time [u.t.]</td>
<td>energy overhead [μJ]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL</td>
<td>0</td>
<td>20</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>MID</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>LOW</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>OFF</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>0</td>
</tr>
</tbody>
</table>
Power Reduction Scheduling

Fig. 6. V/F control of applu(4proc.)
An Image of Static Schedule for Heterogeneous Multi-core with Data Transfer Overlapping and Power Control
Generated Multigrain Parallelized Code
(The nested coarse grain task parallelization is realized by only OpenMP “section”, “Flush” and “Critical” directives.)
Multicore Program Development Using OSCAR API V2.0

**Sequential Application Program in Fortran or C**
(Consumer Electronics, Automobiles, Medical, Scientific computation, etc.)

- **Manual parallelization / power reduction**
- **Accelerator Compiler/ User**
  - Add “hint” directives before a loop or a function to specify it is executable by the accelerator with how many clocks

- **Waseda OSCAR Parallelizing Compiler**
  - Coarse grain task parallelization
  - Data Localization
  - DMAC data transfer
  - Power reduction using DVFS, Clock/ Power gating

- **Parallelized API F or C program**
  - Proc0
    - Code with directives
    - Thread 0
  - Proc1
    - Code with directives
    - Thread 1
    - Accelerator 1
      - Code
    - Accelerator 2
      - Code

- **Low Power Homogeneous Multicore Code Generation**
  - API Analyzer
  - Existing sequential compiler

- **Low Power Heterogeneous Multicore Code Generation**
  - API Analyzer (Available from Waseda)
  - Existing sequential compiler

- **Server Code Generation**
  - OpenMP Compiler

- **Generation of parallel machine codes using sequential compilers**

- **Executable on various multicores**

- **OSCAR: Optimally Scheduled Advanced Multiprocessor API**: Application Program Interface

- **Hitachi, Renesas, NEC, Fujitsu, Toshiba, Denso, Olympus, Mitsubishi, Esol, Cats, Gaio, 3 univ.**
Performance of OSCAR compiler on NEC NaviEngine(MPcore) with Linux in 2007

OSCAR compiler gave us 3.43 times speedup against 1 core for Fortran and 3.13 for C on ARM/NEC MPCore with 4 ARM 400MHz cores
2.9 Times Speed-up of AAC Encoding Compared with a Sequential Processing on 3 Core NaviEngine (ARM MPcore) with Realtime OS eT-Kernel Multi-Core Edition
Performance of OSCAR Compiler & API on 2 ARMv7-cores Qualcomm MSM8960 Android 4.0 for Smart Phones

1.81 times speedup by 2 cores on the average against 1 core
Performance of OSCAR Compiler on Intel 12 core SMP based on 6-core Xeon X5670

- OSCAR Compiler gives us 1.9 times speedup on the average against Intel Composer XE 2011
Performance of OSCAR Compiler on AMD 12-core SMP Based on Opteron 6174

- OSCAR Compiler gives us 2.2 times speedup on the average against Intel Composer XE 2011
Power Reduction of MPEG2 Decoding to 1/4 on 8 Core Homogeneous Multicore RP-2 by OSCAR Parallelizing Compiler

Without Power Control
(Voltage : 1.4V)

Avg. Power 5.73 [W]

With Power Control
(Frequency, Resume Standby: Power shutdown & Voltage lowering 1.4V-1.0V)

Avg. Power 1.52 [W]

73.5% Power Reduction
33 Times Speedup Using OS CAR Compiler and OSCAR API on RP-X (Optical Flow with a hand-tuned library)

CPU performs data transfers between SH and FE

<table>
<thead>
<tr>
<th>SH Count</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1SH</td>
<td>1</td>
</tr>
<tr>
<td>2SH</td>
<td>2.29</td>
</tr>
<tr>
<td>4SH</td>
<td>3.09</td>
</tr>
<tr>
<td>8SH</td>
<td>5.4</td>
</tr>
<tr>
<td>2SH+1FE</td>
<td>18.85</td>
</tr>
<tr>
<td>4SH+2FE</td>
<td>26.71</td>
</tr>
<tr>
<td>8SH+4FE</td>
<td>32.65</td>
</tr>
</tbody>
</table>

Y. Yuyama, et al., "A 45nm 3.3GOPS/W Heterogeneous Multi-Core SoC", ISSCC2010

111[fps]
8 Core RP2 Chip Block Diagram

LCPG: Local clock pulse generator
PCR: Power Control Register
CCN/BAR: Cache controller/Barrier Register
URAM: User RAM (Distributed Shared Memory)
Faster or Equal Processing Performance with Hardware Coherence Control on 8 core RP2 Multicore Precessor Having Hardware Coherent Mechanism Up-to 4 cores by OSCAR Compiler’s Software Coherence Control

<table>
<thead>
<tr>
<th>No. of processor cores</th>
<th>AAC Encoder</th>
<th>MPEG2 Decoder</th>
<th>MPEG2 Encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.02</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1.92</td>
<td>1.01</td>
<td>1.02</td>
</tr>
<tr>
<td>4</td>
<td>3.54</td>
<td>1.62</td>
<td>1.85</td>
</tr>
<tr>
<td>8</td>
<td>3.59</td>
<td>2.45</td>
<td>3.36</td>
</tr>
</tbody>
</table>

SMP
Non-Coherent Cache
92 Times Speedup against the Sequential Processing for GMS Earthquake Wave Propagation Simulation on Hitachi SR16000 (Power7 Based 128 Core Linux SMP)

- 0.08秒 102km × 72km × 30kmの領域の地震をシミュレーション
Cancer Treatment
Carbon Ion Radiotherapy
(Previous best was 2.5 times speedup on 16 processors with hand optimization)

8.9 times speedup by 12 processors
Intel Xeon X5670 2.93GHz 12 core SMP (Hitachi HA8000)

55 times speedup by 64 processors
IBM Power 7 64 core SMP (Hitachi SR16000)
Conclusions

- OSCAR compiler automatic parallelizes C or Fortran program using multigrain parallelization, data localization for cache and local memory with DMA data transfers and generates C or Fortran parallelized code with OSCAR API version 2.0.
- It supports shared memory homogeneous and heterogeneous multicores and manycores including non-coherent cache architectures.
- In addition to the automatic parallelization, automatic power control using DVFS and Clock and Power gating has been implemented for real-time processing and minimum execution time processing modes.
- The following performance has been attained on various multicores and servers:
  - 55 times speedup by 64 processors for Carbon Ion Radiotherapy Cancer treatment on IBM Power 7 64 core SMP (Hitachi SR16000)
  - 46 Times Speedup for GMS Earthquake Wave Propagation Simulation on Hitachi SR16000
  - Faster or Equal Processing Performance with Hardware Coherence Control on 8 core RP2 Multicore Precessor Having Hardware Coherent Mechanism Up-to 4 cores by OSCAR Compiler’s Software Coherence Control
  - 33 Times Speedup for Optical Flow on 8 SH4A and 4 DRP accelerators on RP-X heterogeneous multcore.
  - Power Reduction of MPEG2 Decoding to 1/4 on 8 Core Homogeneous Multicore RP-2.
  - 2.2 times speedup on the average against Intel Composer XE 2011 on AMD 12-core SMP against Intel Composer XE 2011 Based on Opteron 6174.
  - 1.9 times speedup on the average on Intel 12 core SMP based on 6-core Xeon X5670.
  - 2.9 Times Speed-up for AAC Encodeing on 3 Core NaviEngine (ARM MPcore) with Realtime OS eT-Kernel Multi-Core Edition.
OSCAR Multi-Core Architecture

CMP_0 (chip multiprocessor 0)

- CPU
- LPM/ I-Cache
- LDM/ D-cache
- DSM
- DTC
- Network Interface
- CSM / L2 Cache
- Intra-chip connection network (Multiple Buses, Crossbar, etc)

CMP_m

- I/O Devices

CSM_j

- I/O Devices

Inter-chip connection network (Crossbar, Buses, Multistage network, etc)

CSM: central shared mem.
LDM: local data mem.
DSM: distributed shared mem.
LPM: local program mem.
DTC: Data Transfer Controller
FVR: frequency / voltage control register
OSCAR Memory Space (Global and Local Address Space)

- **SYSTEM MEMORY SPACE**
  - UNDEFINED (Local Memory Area)
  - BROADCAST
  - NOT USE
  - CP (Control Processor)
  - LOCAL MEMORY SPACE
  - NOT USE
  - CONTROL
  - LOCAL PROGRAM MEMORY (LPM (Bank0))
  - LOCAL DATA MEMORY (LDM)
  - DISTRIBUTED SHARED MEMORY (DSM)
  - NOT USE
  - NOT USE

- **LOCAL MEMORY SPACE**
  - OSARC Memory Space (Global and Local Address Space)
  - PE0
  - PE1
  - PE15
  - CSM1, 2, 3 (Centralized Shared Memory)
  - NOT USE
  - CONTROL
  - SYSTEM
  - ACCESSING AREA
  - FFFFFFFF