OSCAR Automatic Parallelization and Power Reduction Compiler for Homogeneous and Heterogeneous Multicores

Hironori Kasahara

Professor, Dept. of Computer Science & Engineering
Director, Advanced Multicore Processor Research Institute
Waseda University, Tokyo, Japan
IEEE Computer Society Multicore STC Chair
IEEE Computer Society 2015 Election
President Candidate

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

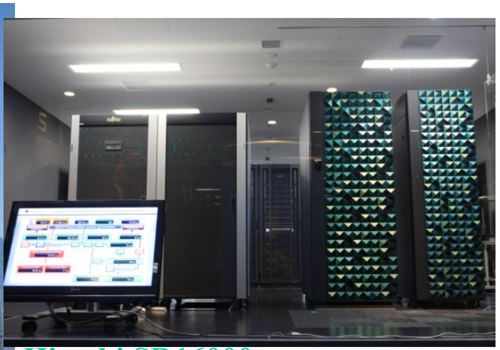
- More than 64 cores
- Natural air cooling (No fan)
  Cool, Compact, Clear, Quiet
- Operational by Solar Panel

<Industry, Government, Academia>

Hitachi, Fujitsu, NEC, Renesas, Olympus, Toyota, Denso, Mitsubishi, OSCAR Tech.

<Ripple Effect>

- Low CO_2 (Carbon Dioxide) Emissions
- Creation Value Added Products
  - Consumer Electronics, Automobiles, Servers

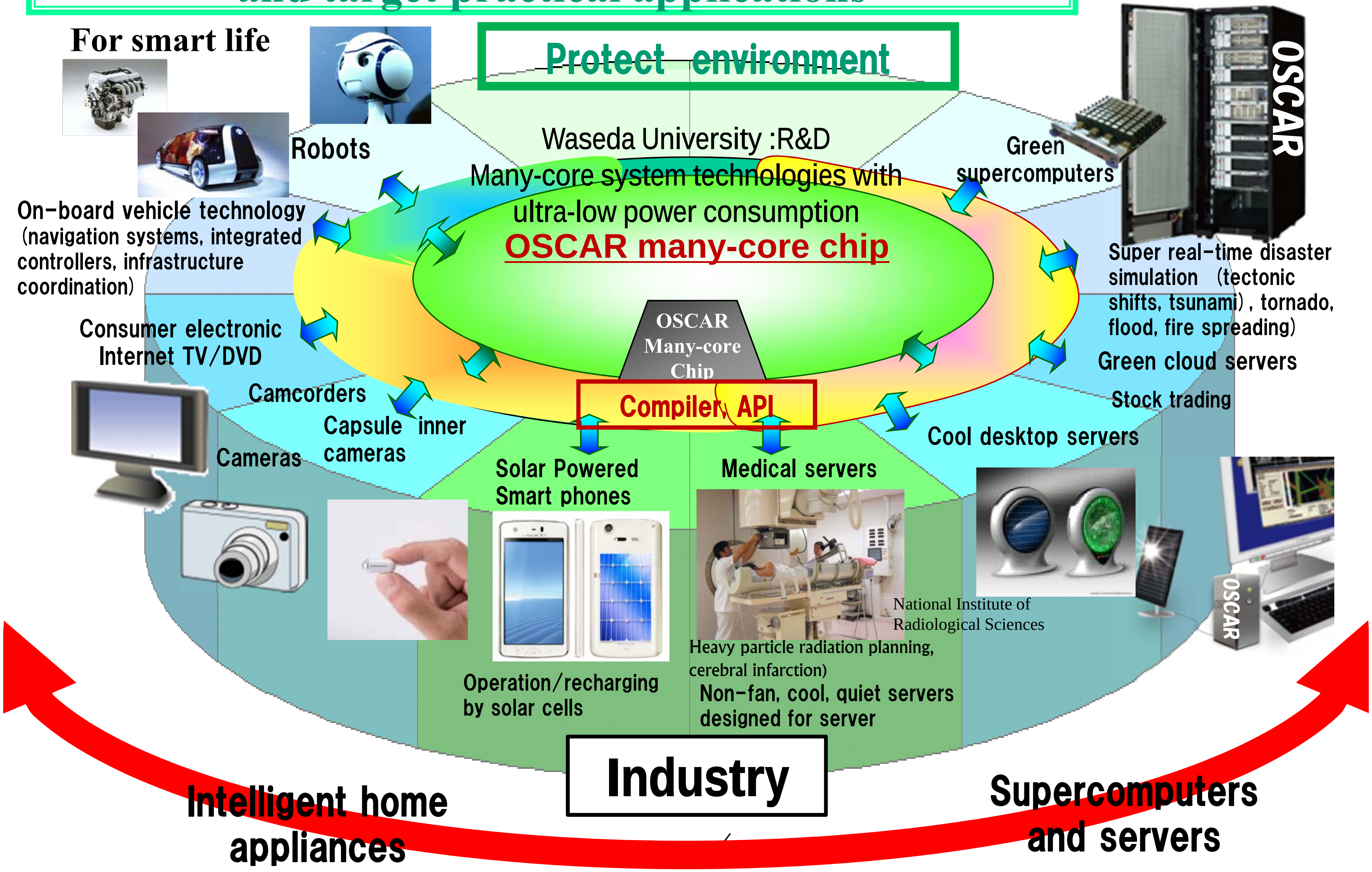


Hitachi SR16000:
Power7 128coreSMP
Fujitsu M9000
SPARC VII 256 core SMP

Beside Subway Waseda Station,
Near Waseda Univ. Main Campus

Industry-government-academia collaboration in R&D and target practical applications

Protect lives



For smart life

Robots

On-board vehicle technology (navigation systems, integrated controllers, infrastructure coordination)

Consumer electronic Internet TV/DVD

Camcorders

Capsule inner cameras

Cameras

Protect environment

Waseda University :R&D

Many-core system technologies with ultra-low power consumption

OSCAR many-core chip

OSCAR Many-core Chip

Compiler, API

Solar Powered Smart phones

Operation/recharging by solar cells

Medical servers

National Institute of Radiological Sciences

Heavy particle radiation planning, cerebral infarction)

Non-fan, cool, quiet servers designed for server

Green supercomputers

OSCAR

Super real-time disaster simulation (tectonic shifts, tsunami), tornado, flood, fire spreading)

Green cloud servers

Stock trading

Cool desktop servers

Industry

Intelligent home appliances

Supercomputers and servers

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

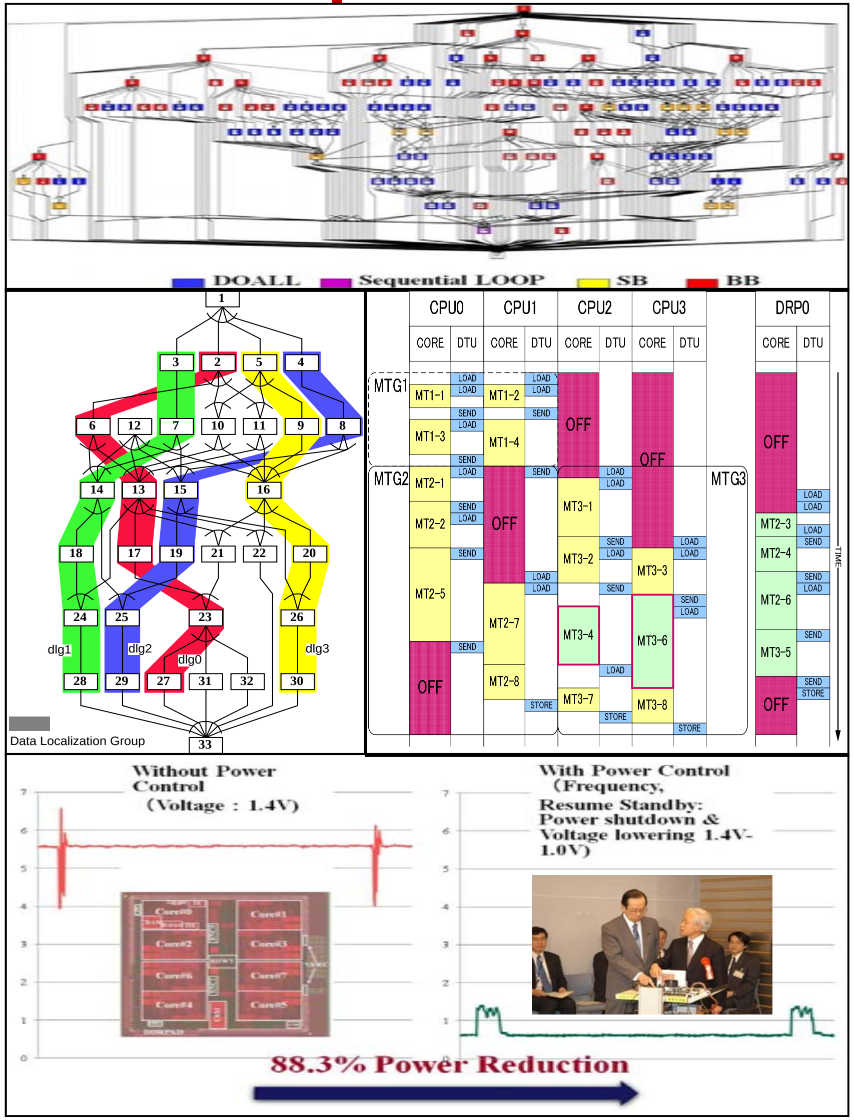
Automatic data management for distributed shared memory, cache and local memory

Data Transfer Overlapping

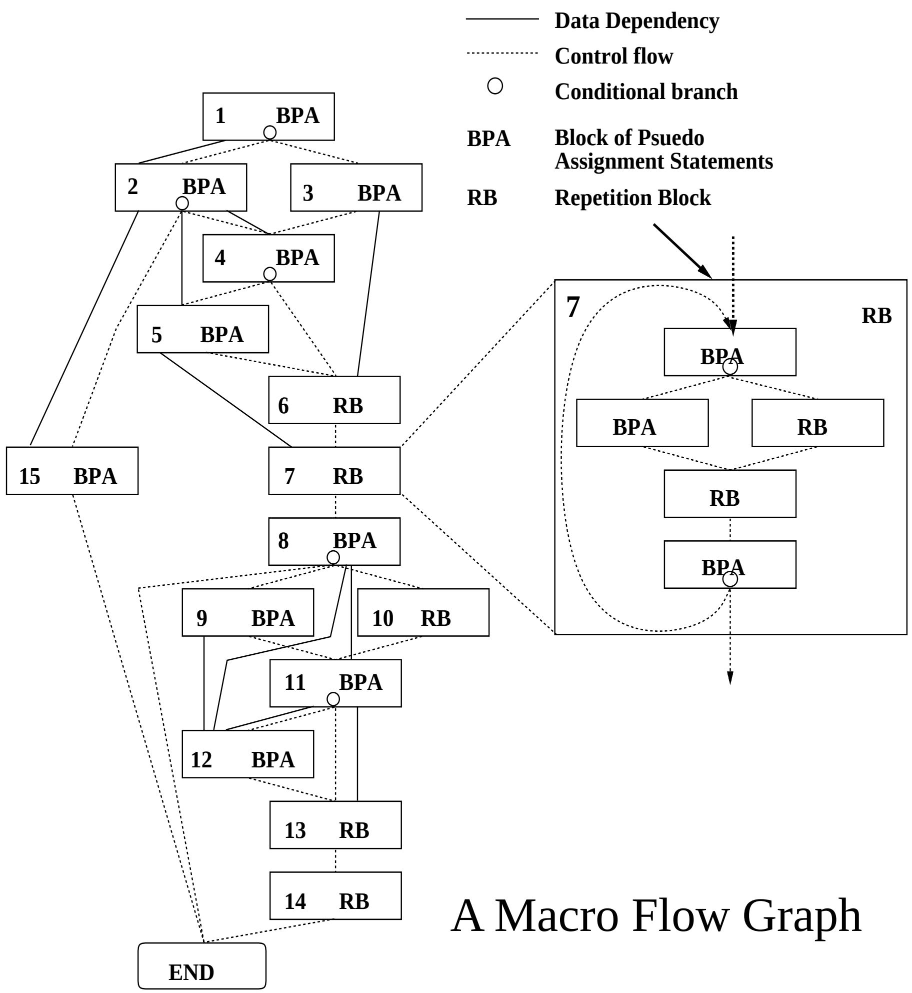
Data transfer overlapping using Data Transfer Controllers (DMAs)

Power Reduction

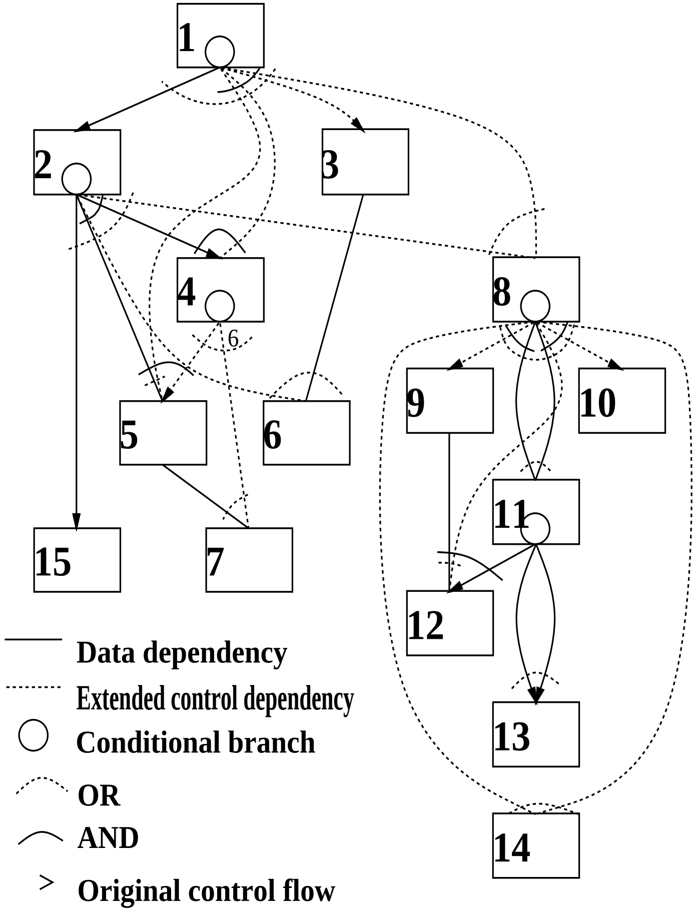
Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.



Earliest Executable Condition Analysis for Coarse Grain Tasks (Macro-tasks)



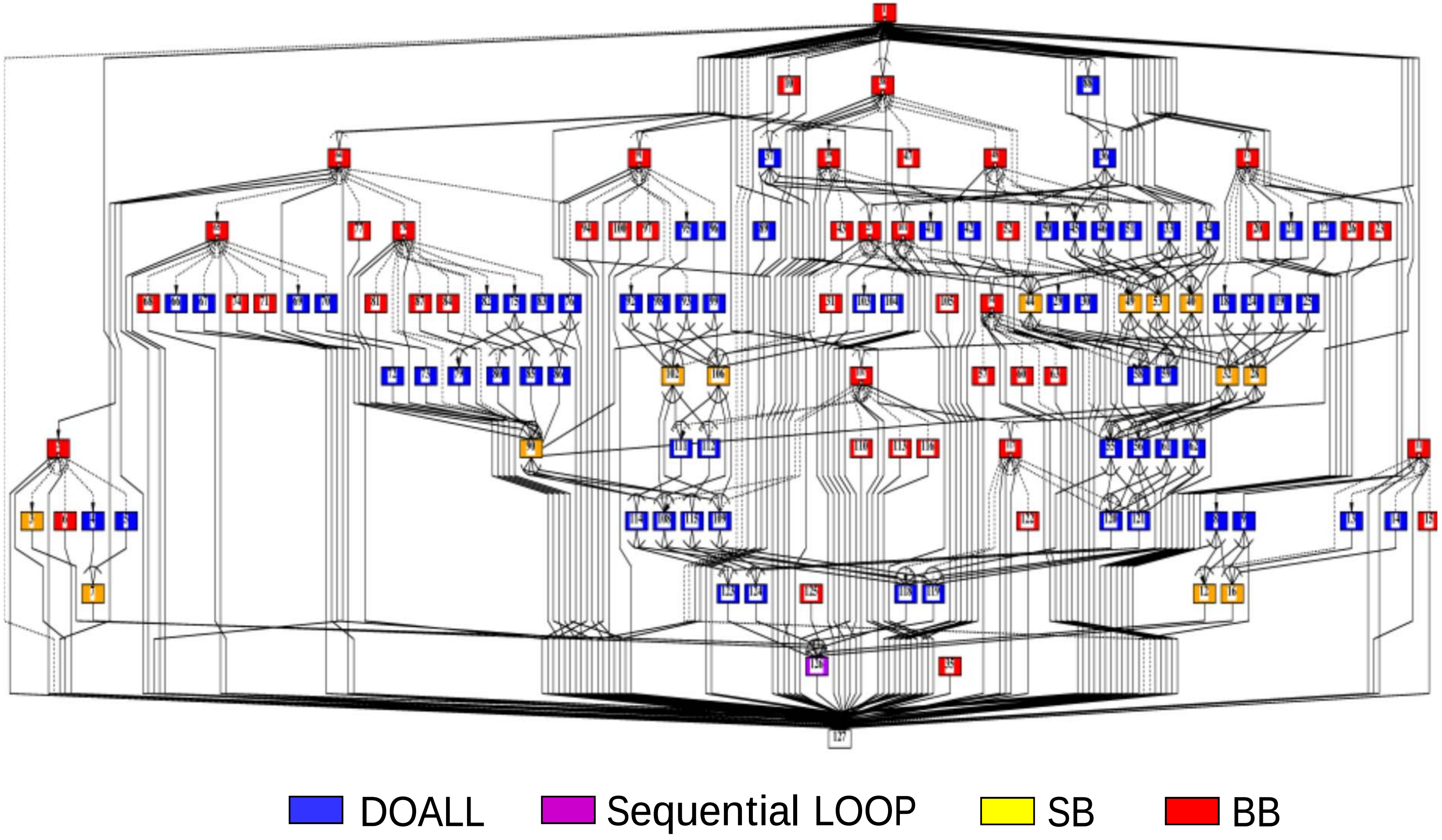
A Macro Flow Graph



A Macro Task Graph

MTG of Su2cor-LOOPS-DO400

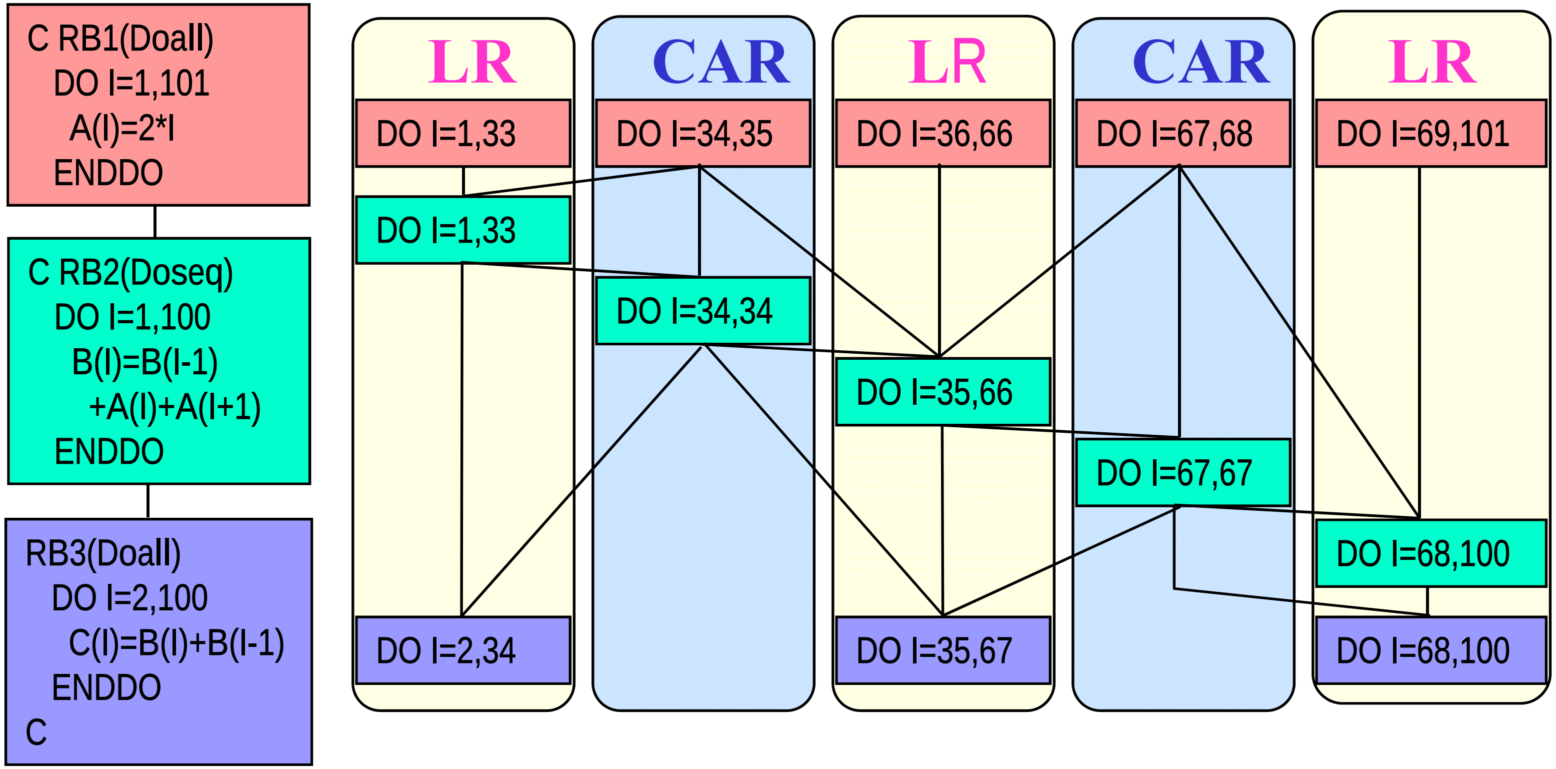
- Coarse grain parallelism PARA_ALD = 4.3



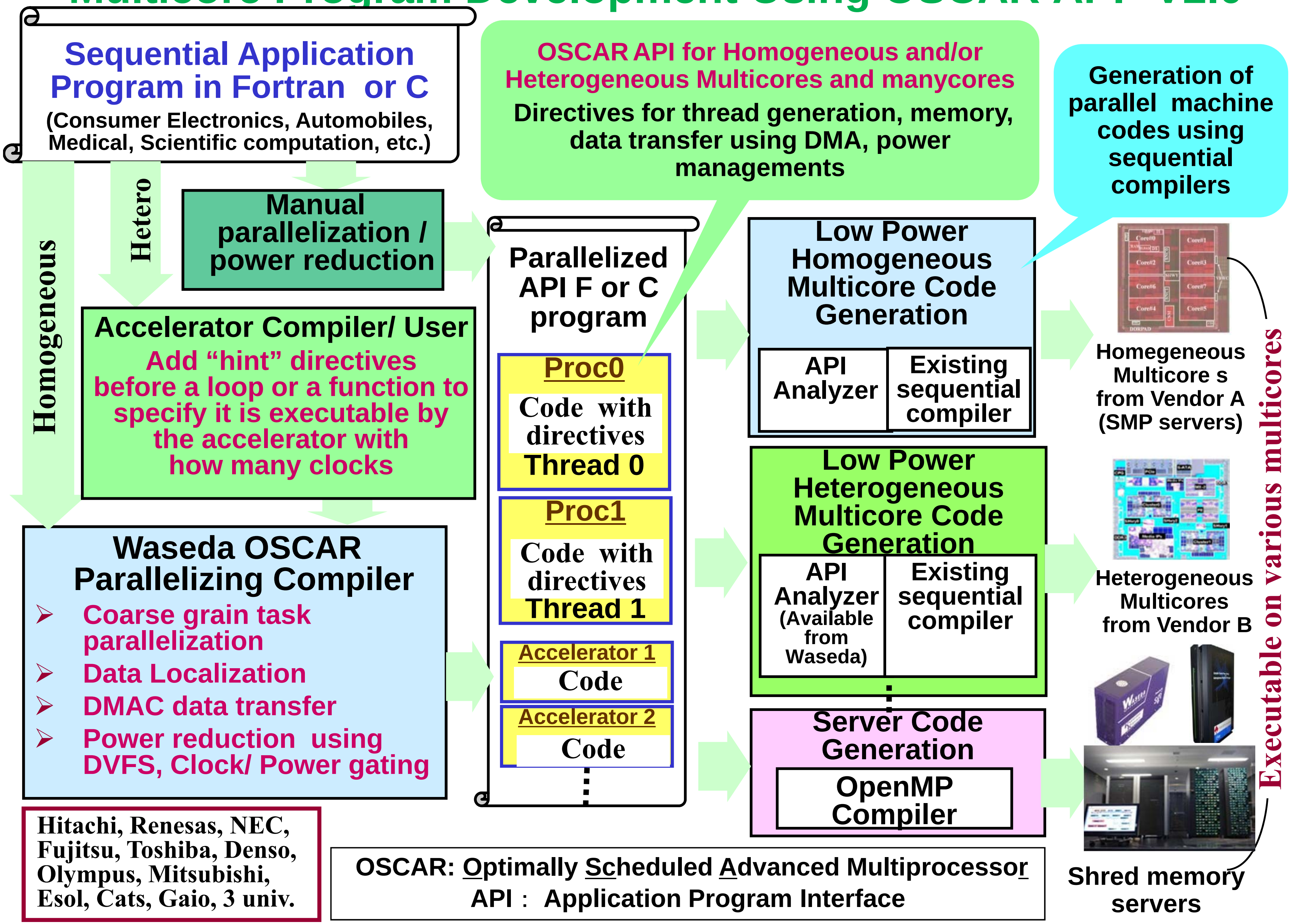
🟦 DOALL 🟪 Sequential LOOP 🟨 SB 🟥 BB

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**



Multicore Program Development Using OSCAR API V2.0



Sequential Application Program in Fortran or C

(Consumer Electronics, Automobiles, Medical, Scientific computation, etc.)

Homogeneous

Hetero

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

Hitachi, Renesas, NEC, Fujitsu, Toshiba, Denso, Olympus, Mitsubishi, Esol, Cats, Gaio, 3 univ.

OSCAR API for Homogeneous and/or Heterogeneous Multicores and manycores

Directives for thread generation, memory, data transfer using DMA, power managements

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

Homegeneous Multicore s from Vendor A (SMP servers)

Heterogeneous Multicores from Vendor B

Shred memory servers

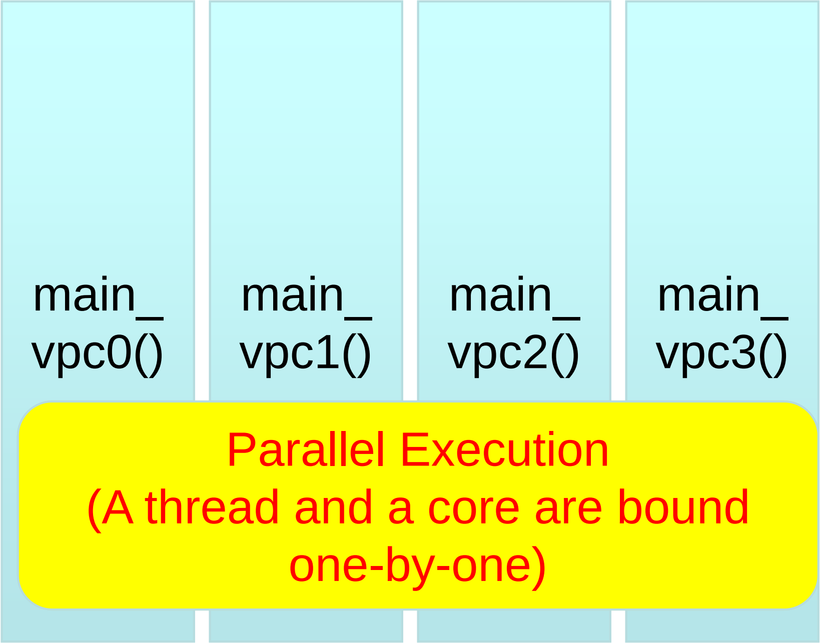
Executable on various multicores

OSCAR: Optimally Scheduled Advanced Multiprocessor

API : Application Program Interface

Thread Execution Model

```
#pragma omp parallel sections
    {
#pragma omp section
        main_vpc0();
#pragma omp section
        main_vpc1();
#pragma omp section
        main_vpc2();
#pragma omp section
        main_vpc3();
    }
```



VPC: Virtual Processor Core

Memory Mapping

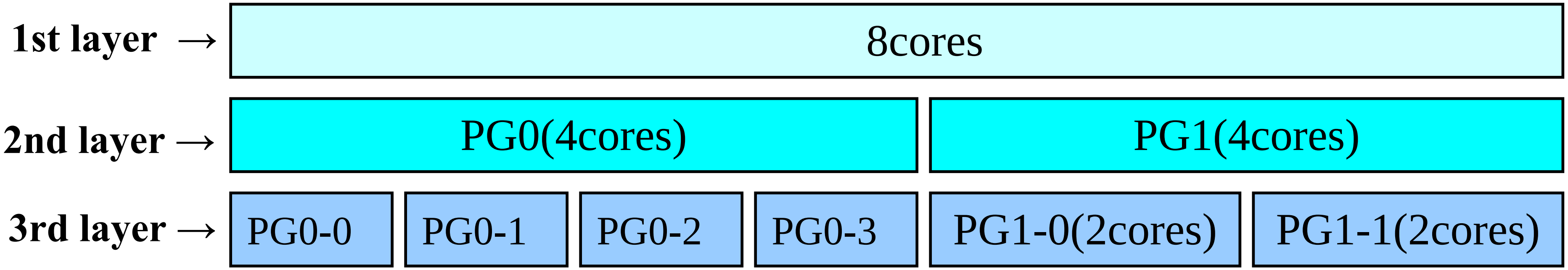
- Placing variables on an onchip centralized shared memory (onchipCSM)
  - #pragma oscar onchipshared (C)
  - !$oscar onchipshared (Fortran)
- Placing variables on a local data memory (LDM)
  - #pragma omp threadprivate (C)
  - !$omp threadprivate (Fortran)
  - This directive is an extension to OpenMP
- Placing variables on a distributed shared memory (DSM)
  - #pragma oscar distributedshared (C)
  - !$oscar distributedshared (Fortran)

Data Transfer

- Specifying data transfer lists
  - #pragma oscar dma_transfer (C)
  - !$oscar dma_transfer (Fortran)
  - Containing following parameter directives
- Specifying a contiguous data transfer
  - #pragma oscar dma_contiguous_parameter (C)
  - !$oscar dma_contiguous_parameter (Fortran)
- Specifying a stride data transfer
  - #pragma oscar dma_stride_parameter
  - !$oscar dma_stride_parameter
  - This can be used for scatter/gather data transfer
- Data transfer synchronization
  - #pragma oscsar dma_flag_check
  - !$oscar dma_flag_check

Hierarchical Barrier Synchronization

- Specifying **a hierarchical group barrier**
  - #pragma oscar group_barrier (C)
  - !$oscar group_barrier (Fortran)



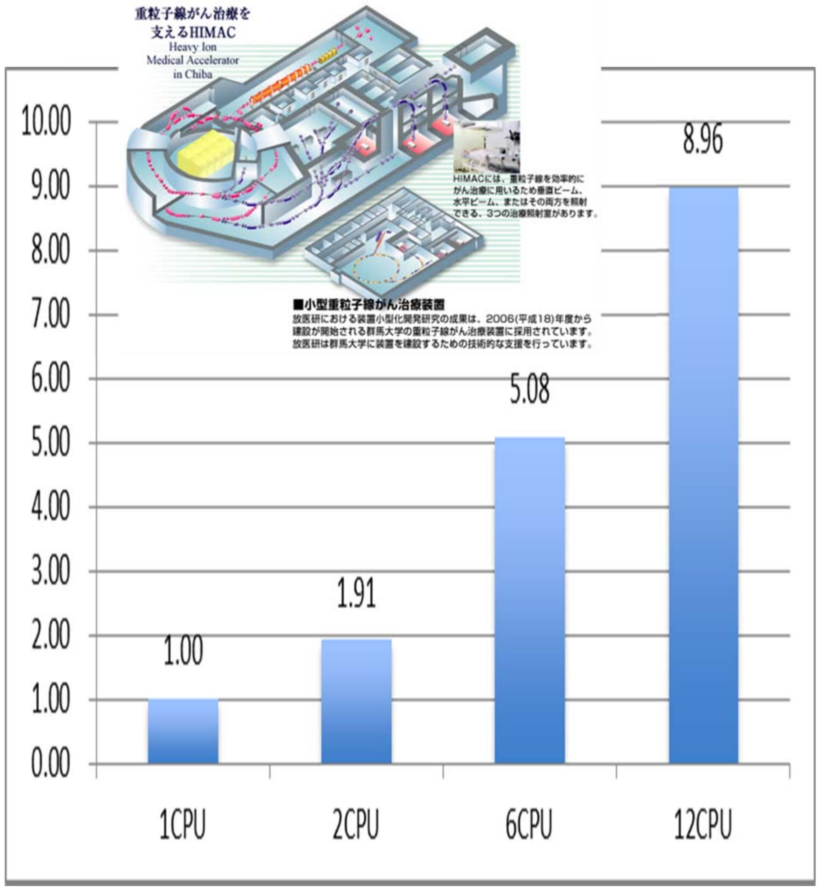
1st layer → 8cores

2nd layer → PG0(4cores) | PG1(4cores)

3rd layer → PG0-0 | PG0-1 | PG0-2 | PG0-3 | PG1-0(2cores) | PG1-1(2cores)

Cancer Treatment
Carbon Ion Radiotherapy

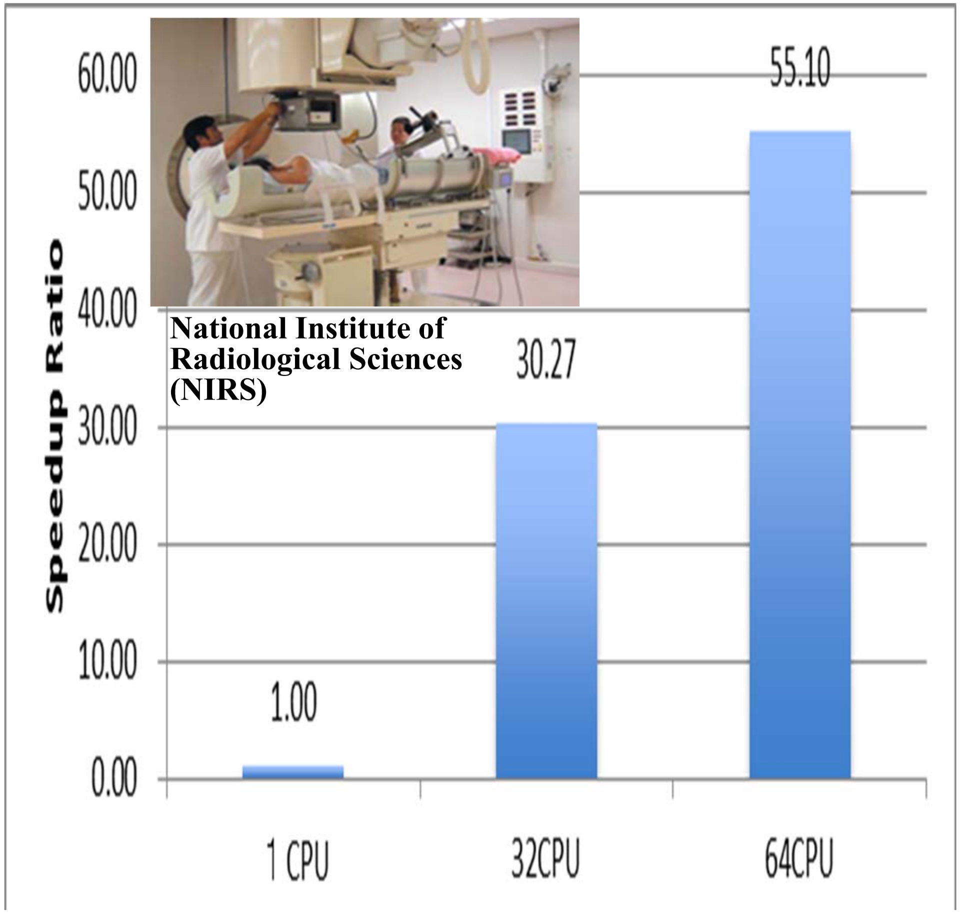
(Previous best was 2.5 times speedup on 16 processors with hand optimization)



HIMACには、重粒子線を効率的にがん治療に用いるため垂直ビーム、水平ビーム、またはその両方を照射できる、3つの治療照射室があります。

■小型重粒子線がん治療装置

放医研における装置小型化開発研究の成果は、2006(平成18)年度から建設が開始される群馬大学の重粒子線がん治療装置に採用されています。放医研は群馬大学に装置を建設するための技術的な支援を行っています。



National Institute of Radiological Sciences (NIRS)

8.9times speedup by 12 processors

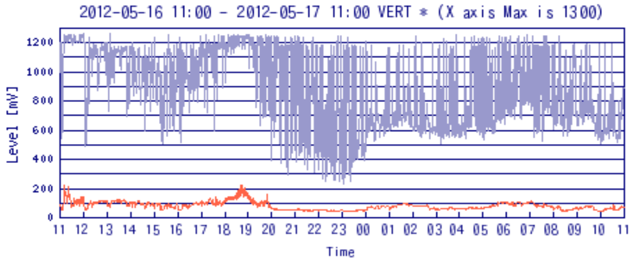
Intel Xeon X5670 2.93GHz 12 core SMP (Hitachi HA8000)

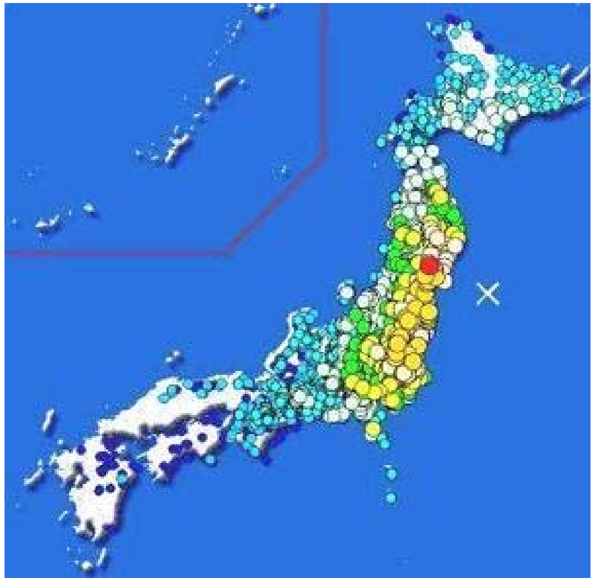
55 times speedup by 64 processors

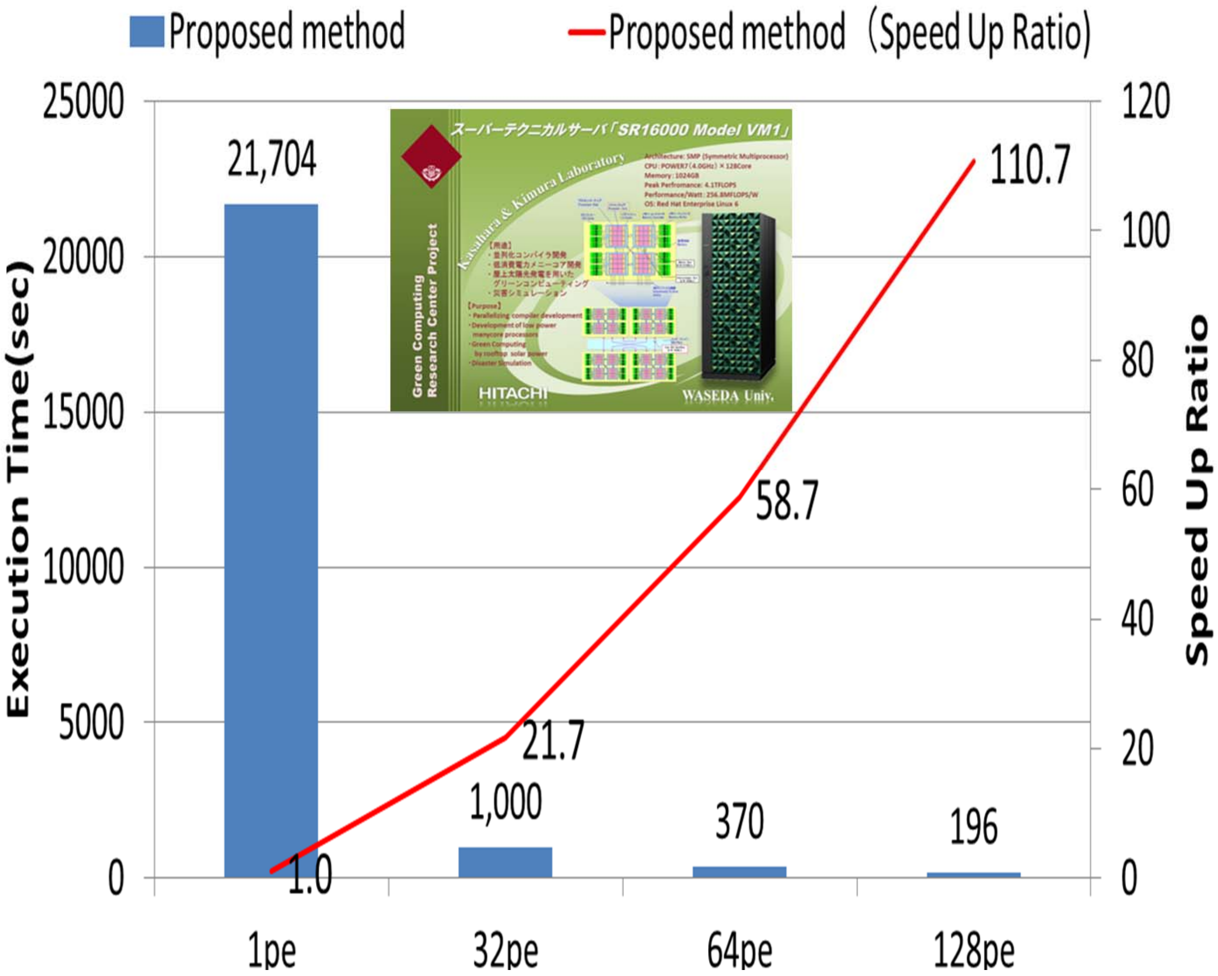
IBM Power 7 64 core SMP (Hitachi SR16000)

110 Times Speedup against the Sequential Processing for GMS Earthquake Wave Propagation Simulation on Hitachi SR16000

（Power7 Based 128 Core Linux SMP）





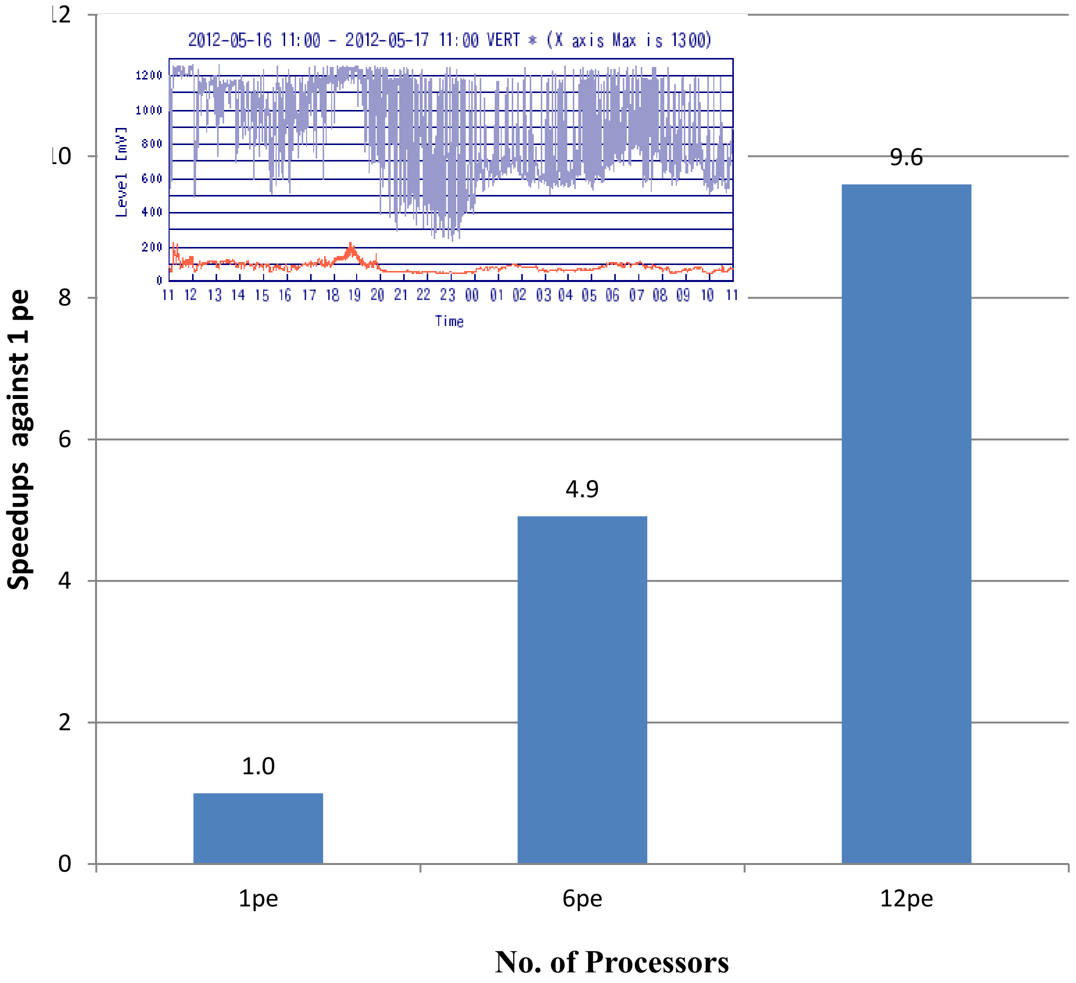


Proposed method
Proposed method（Speed Up Ratio）
Execution Time(sec)
Speed Up Ratio
25000
20000
15000
10000
5000
0
120
100
80
60
40
20
0
21,704
1,000
370
196
1.0
21.7
58.7
110.7
1pe
32pe
64pe
128pe
スーパーテクニカルサーバ「SR16000 Model VM1」
Kasahara & Kimura Laboratory
Architecture: SMP (Symmetric Multiprocessor)
CPU: POWER7（4.0GHz）× 128Core
Memory: 1024GB
Peak Perfromance: 4.1TFLOPS
Performance/Watt: 256.8MFLOPS/W
OS: Red Hat Enterprise Linux 6
【用途】
・並列化コンパイラ開発
・低消費電力メニーコア開発
・屋上太陽光発電を用いたグリーンコンピューティング
・災害シミュレーション
【Purpose】
・Parallelizing compiler development
・Development of low power manycore processors
・Green Computing by rooftop solar power
・Disaster Simulation
Green Computing Research Center Project
HITACHI
WASEDA Univ.

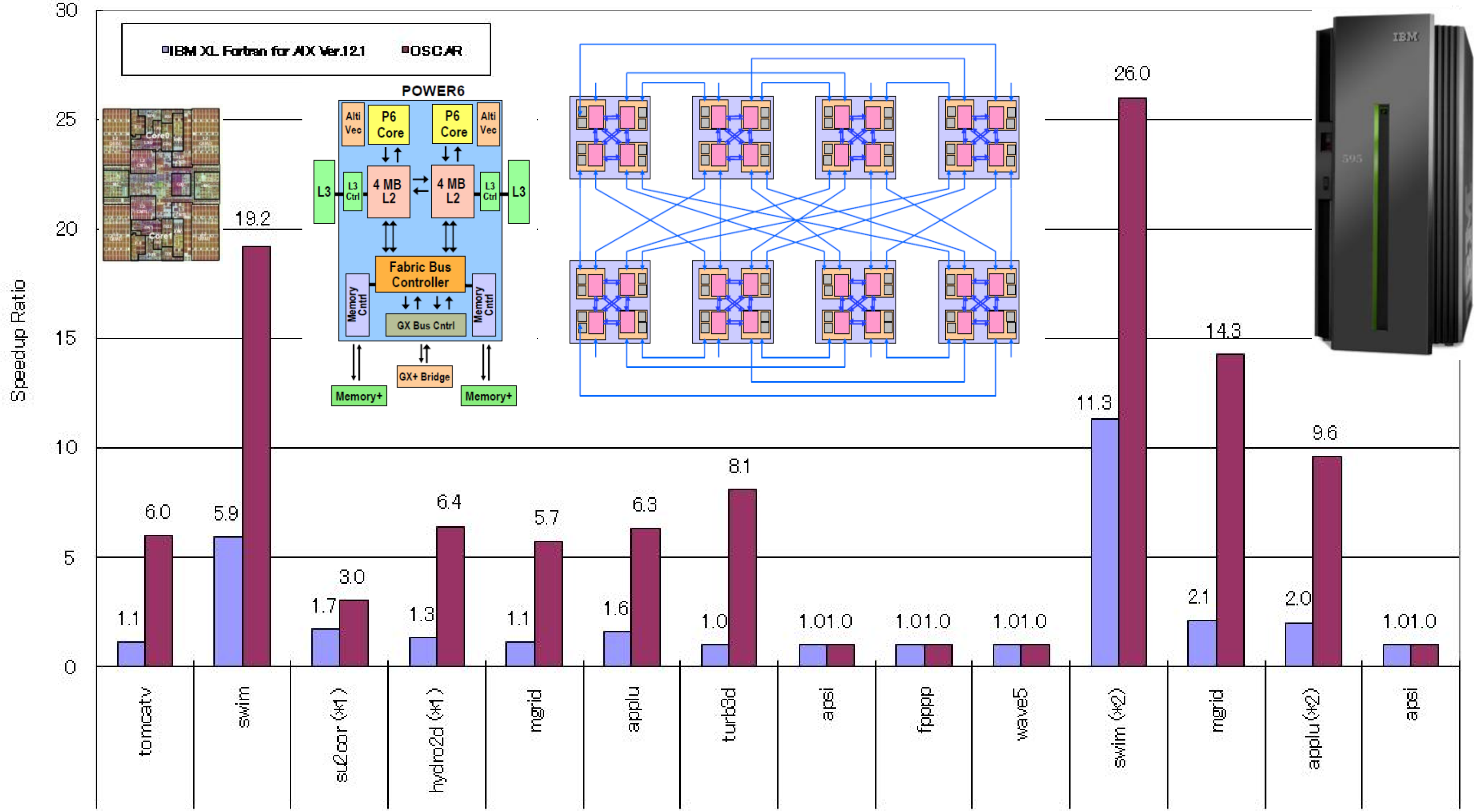
9.6 Times Speedup on 12 cores Power 8 against the Sequential Processing for GMS Earthquake Wave Propagation Simulation

- IBM S812L
  - CPU:POWER8
    - 12 cores
    - Clock Frequency 3.026GHz
  - Memory:60GB
  - OS:Redhat Linux 7.1
  - Backend Fortran compiler: IBM xlf 15.1.1
  - Evaluation using medium size input data(12GB)

S812L(POWER8)



Performance of OSCAR Compiler on IBM p6 595 Power6 (4.2GHz) based 32-core SMP Server

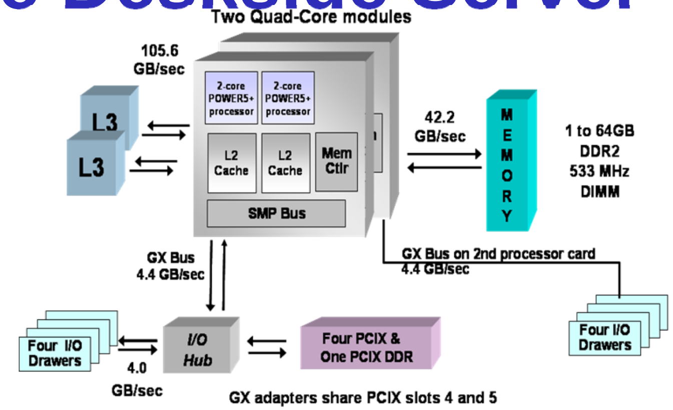


OpenMP codes generated by OSCAR compiler accelerate IBM XL Fortran for AIX Ver.12.1 about 3.3 times on the average

Compile Option:
(*1) Sequential: -O3 –qarch=pwr6, XLF: -O3 –qarch=pwr6 –qsmp=auto, OSCAR: -O3 –qarch=pwr6 –qsmp=noauto
(*2) Sequential: -O5 -q64 –qarch=pwr6, XLF: -O5 –q64 –qarch=pwr6 –qsmp=auto, OSCAR: -O5 –q64 –qarch=pwr6 –qsmp=noauto
(Others) Sequential: -O5 –qarch=pwr6, XLF: -O5 –qarch=pwr6 –qsmp=auto, OSCAR: -O5 –qarch=pwr6 –qsmp=noauto

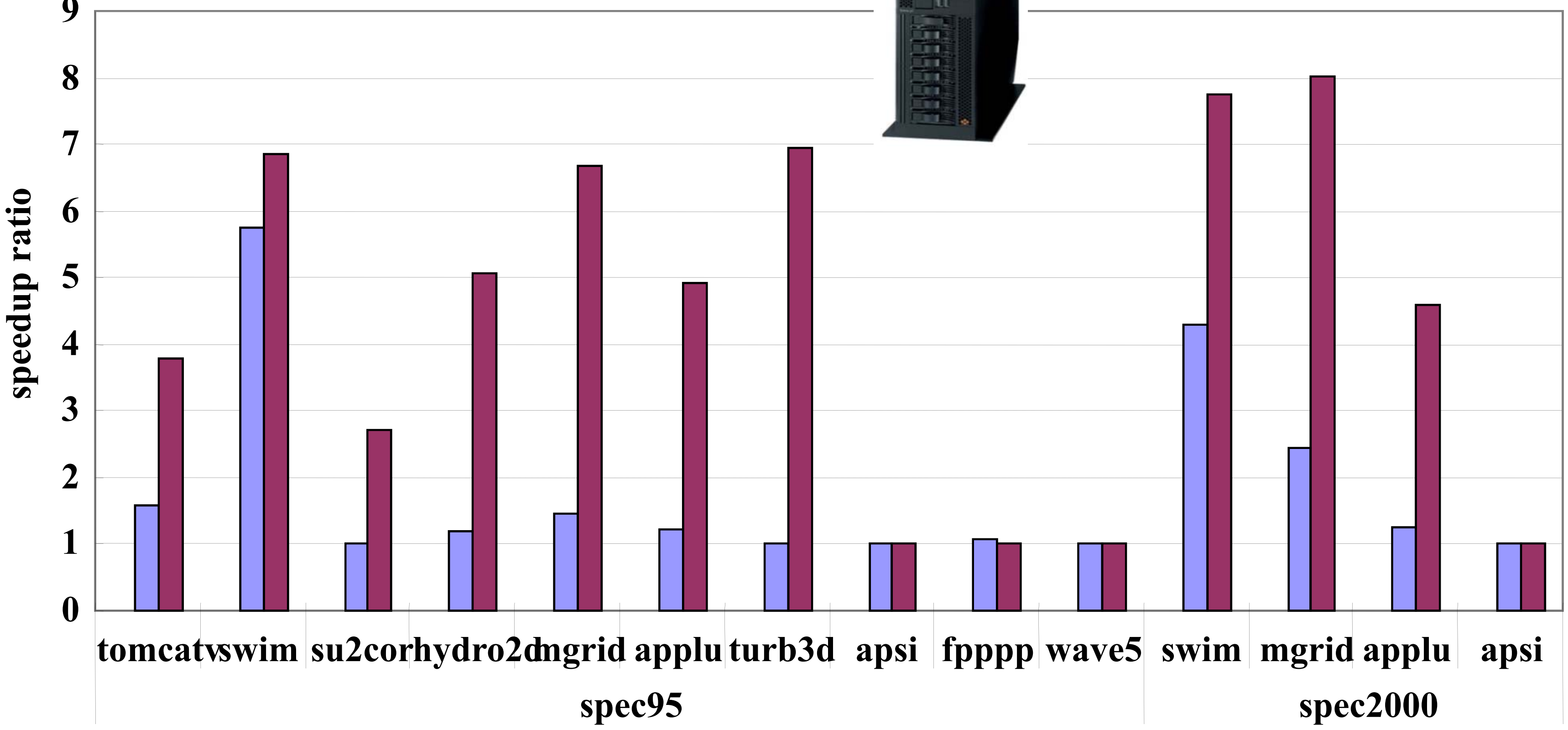
Performance of OSCAR Compiler for Fortran Programs on a IBM p550q 8core Deskside Server

- 2.7 times speedup against loop parallelizing compiler on 8 cores

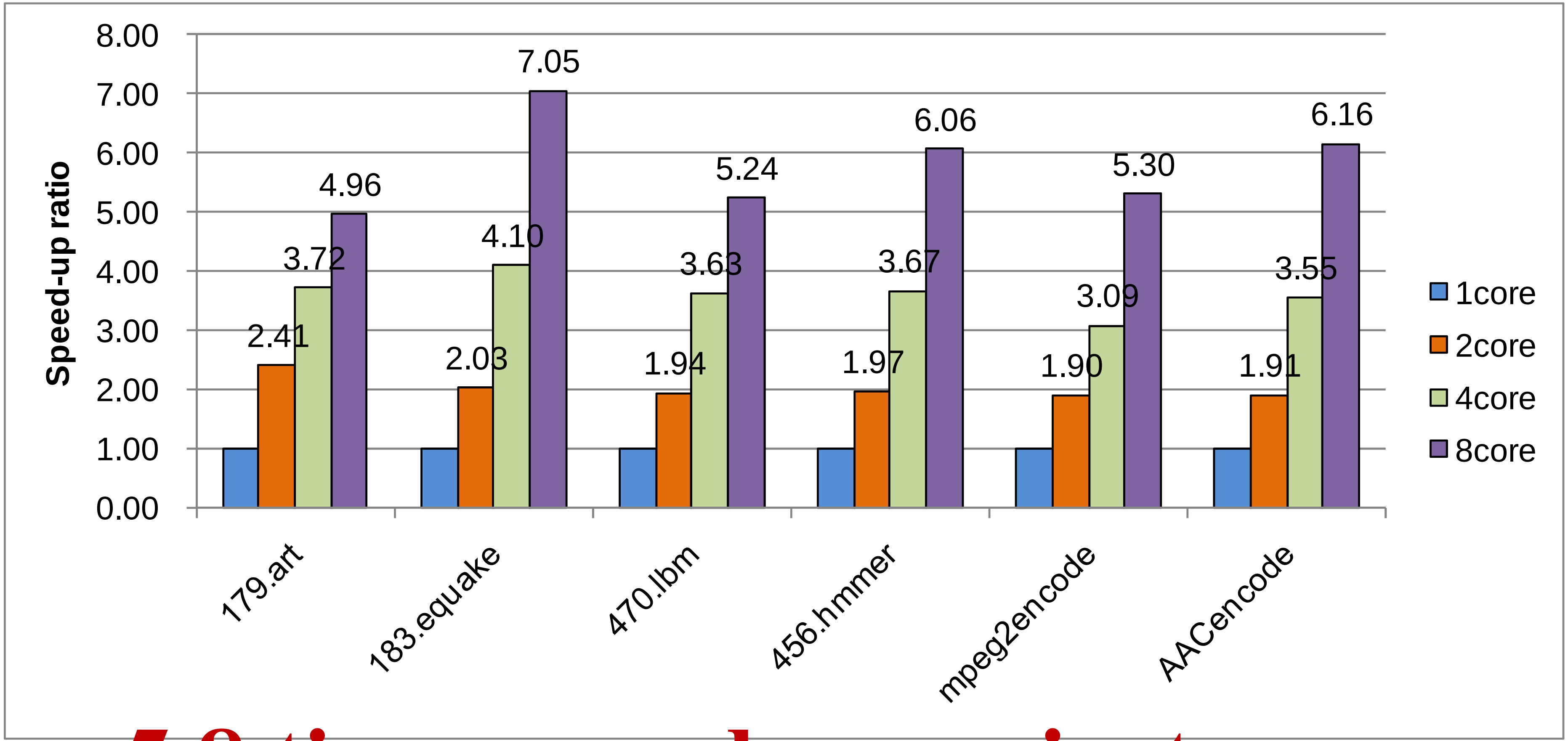


□ Loop parallelization

■ Multigrain parallelizition

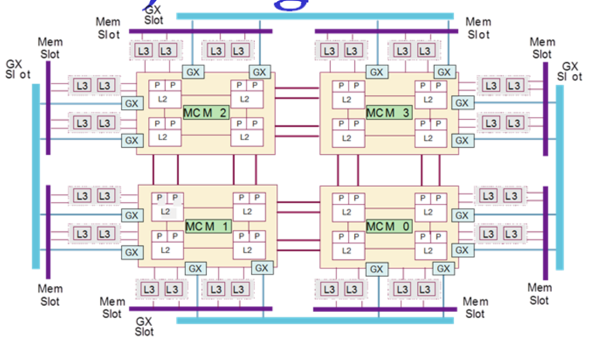


Performance for C programs on IBM p5 550Q



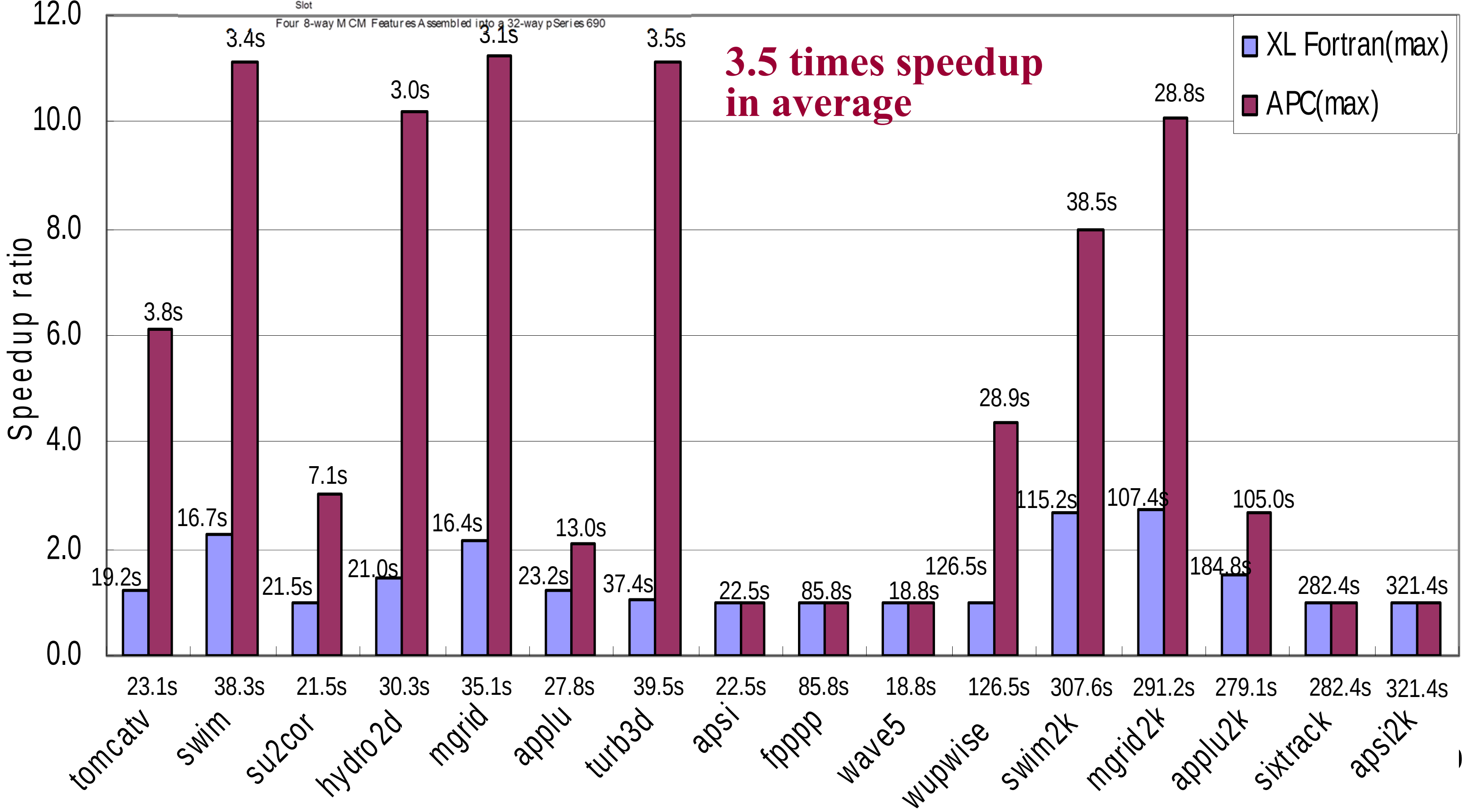
5.8 times speedup against one processor on average

Performance of OSCAR Compiler on IBM pSeries690 (Power 4) RegattaH 16 Processors High-end Server



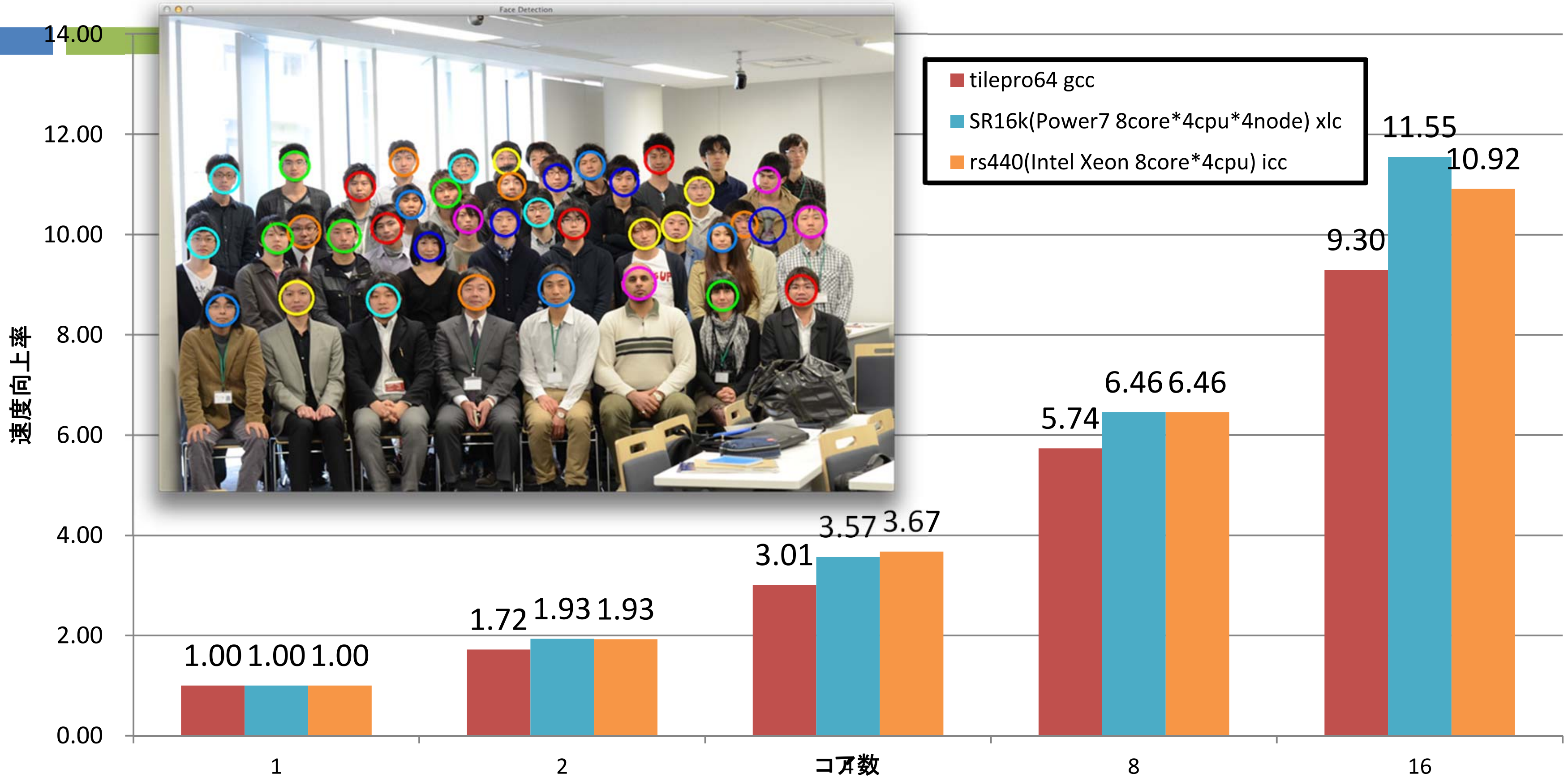
Four 8-way MCM Features Assembled into a 32-way pSeries 690

IBM XL Fortran for AIX Version 8.1



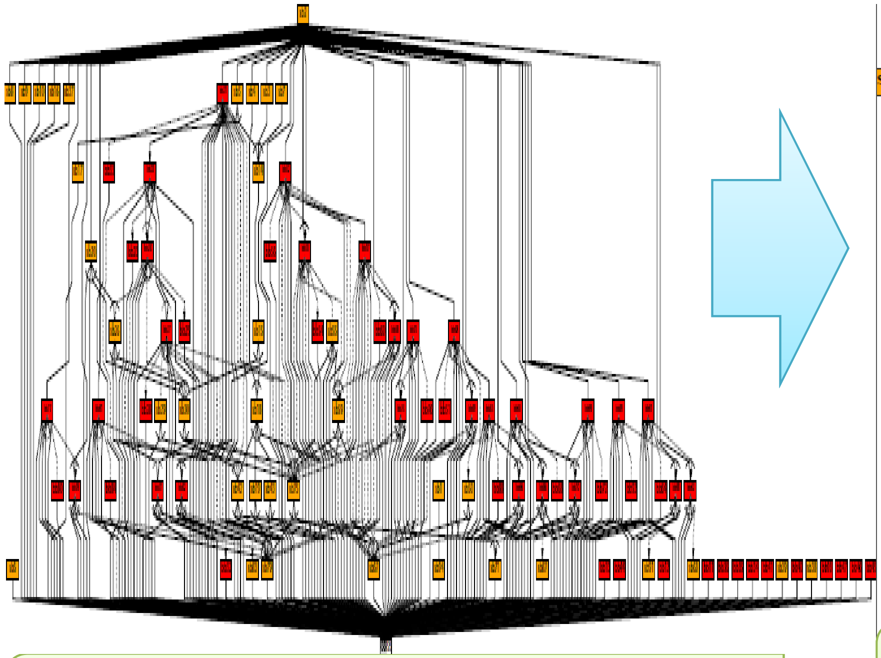
3.5 times speedup in average

Parallel Processing of Face Detection on Manycore, Highend and PC Server

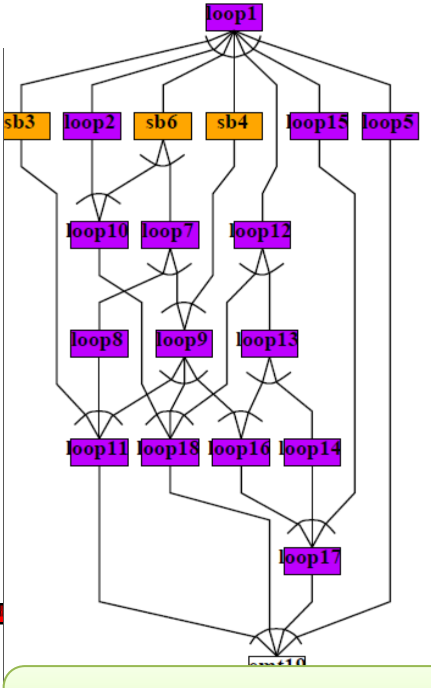


- OSCAR compiler gives us 11.55 times speedup for 16 cores against 1 core on SR16000 Power7 highend server.

Speedup with 2cores for Engine Crankshaft Handwritten Program on Renesas RPX Multi-core Processor

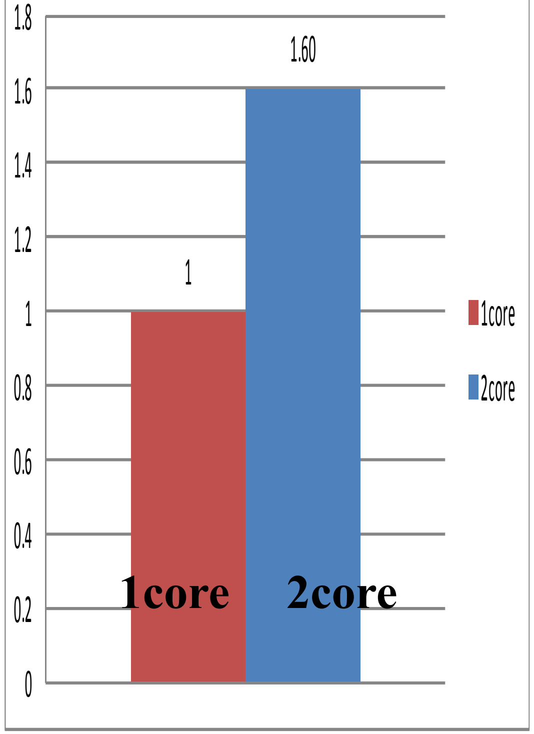


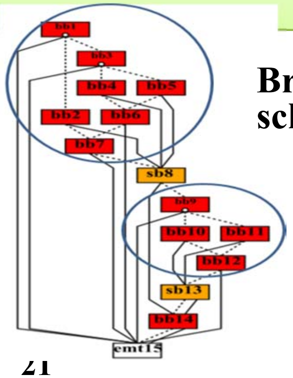
Macrotask graph with a lot of conditional branches



Macrotask graph after task fusion

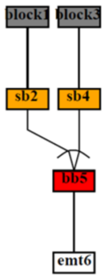
1.6 times Speed up by 2 cores against 1core



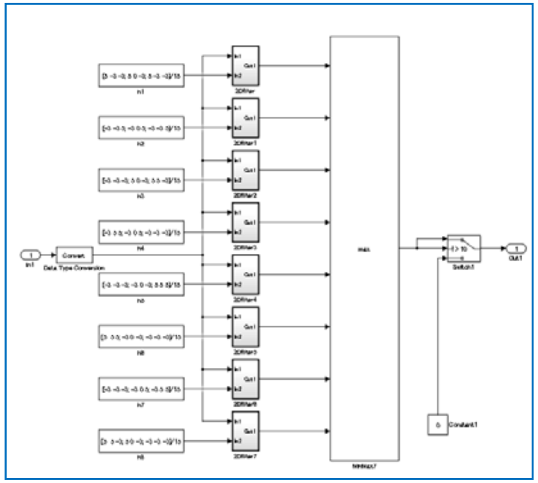


Branches are fused to macrotasks for static scheduling

Grain is too fine (us) for dynamic scheduling.

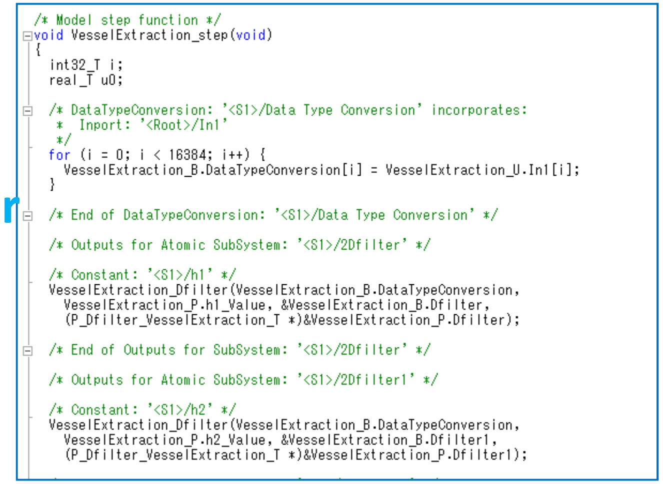


OSCAR Compile Flow for Simulink Applications



Simulink model

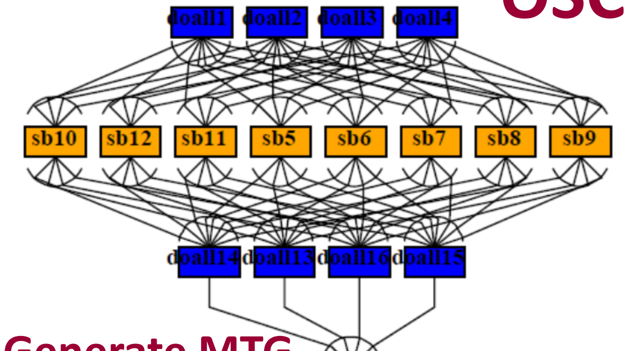
Generate C code using Embedded Coder



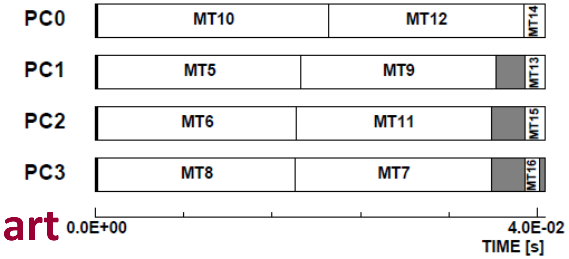
```
/* Model step function */
void VesselExtraction_step(void)
{
  int32_T i;
  real_T u0;

  /* DataTypeConversion: '<S1>/Data Type Conversion' incorporates:
   *  Inport: '<Root>/In1'
   */
  for (i = 0; i < 16384; i++) {
    VesselExtraction_B.DataTypeConversion[i] = VesselExtraction_U.In1[i];
  }

  /* End of DataTypeConversion: '<S1>/Data Type Conversion' */

  /* Outputs for Atomic SubSystem: '<S1>/2Dfilter' */

  /* Constant: '<S1>/h1' */
  VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
    VesselExtraction_P.h1_Value, &VesselExtraction_B.Dfilter,
    (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter);

  /* End of Outputs for SubSystem: '<S1>/2Dfilter' */

  /* Outputs for Atomic SubSystem: '<S1>/2Dfilter1' */

  /* Constant: '<S1>/h2' */
  VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
    VesselExtraction_P.h2_Value, &VesselExtraction_B.Dfilter1,
    (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter1);
```

C code

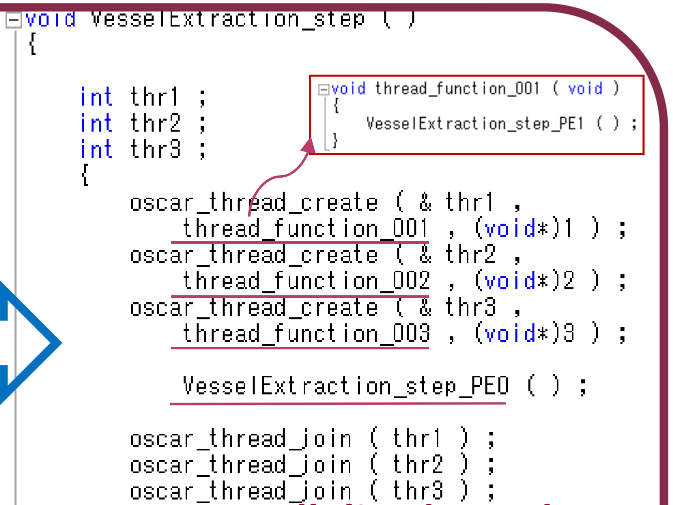
OSCAR Compiler



(1) Generate MTG → Parallelism



(2) Generate gantt chart → Scheduling in a multicore



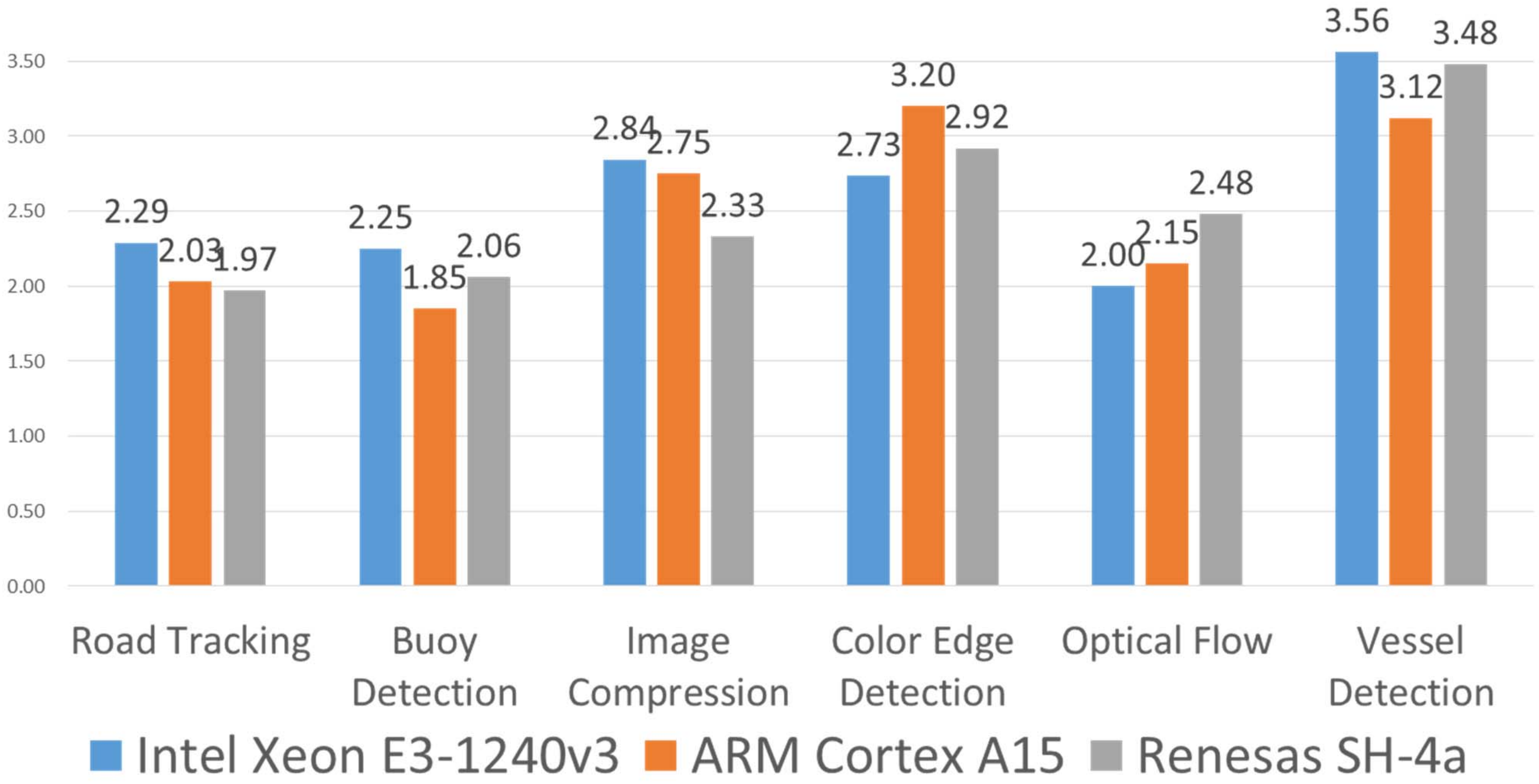
```
void VesselExtraction_step ( )
{
    int thr1 ;
    int thr2 ;
    int thr3 ;
    {
        oscar_thread_create ( & thr1 ,
            thread_function_001 , (void*)1 ) ;
        oscar_thread_create ( & thr2 ,
            thread_function_002 , (void*)2 ) ;
        oscar_thread_create ( & thr3 ,
            thread_function_003 , (void*)3 ) ;

            VesselExtraction_step_PE0 ( ) ;

        oscar_thread_join ( thr1 ) ;
        oscar_thread_join ( thr2 ) ;
        oscar_thread_join ( thr3 ) ;
    }
}
```

```
void thread_function_001 ( void )
{
    VesselExtraction_step_PE1 ( ) ;
}
```

(3) Generate parallelized C code using the OSCAR API → Multiplatform execution (Intel, ARM and SH etc)

Speedups of MATLAB/Simulink Image Processing on Various 4core Multicores

(Intel Xeon, ARM Cortex A15 and Renesas SH4A)



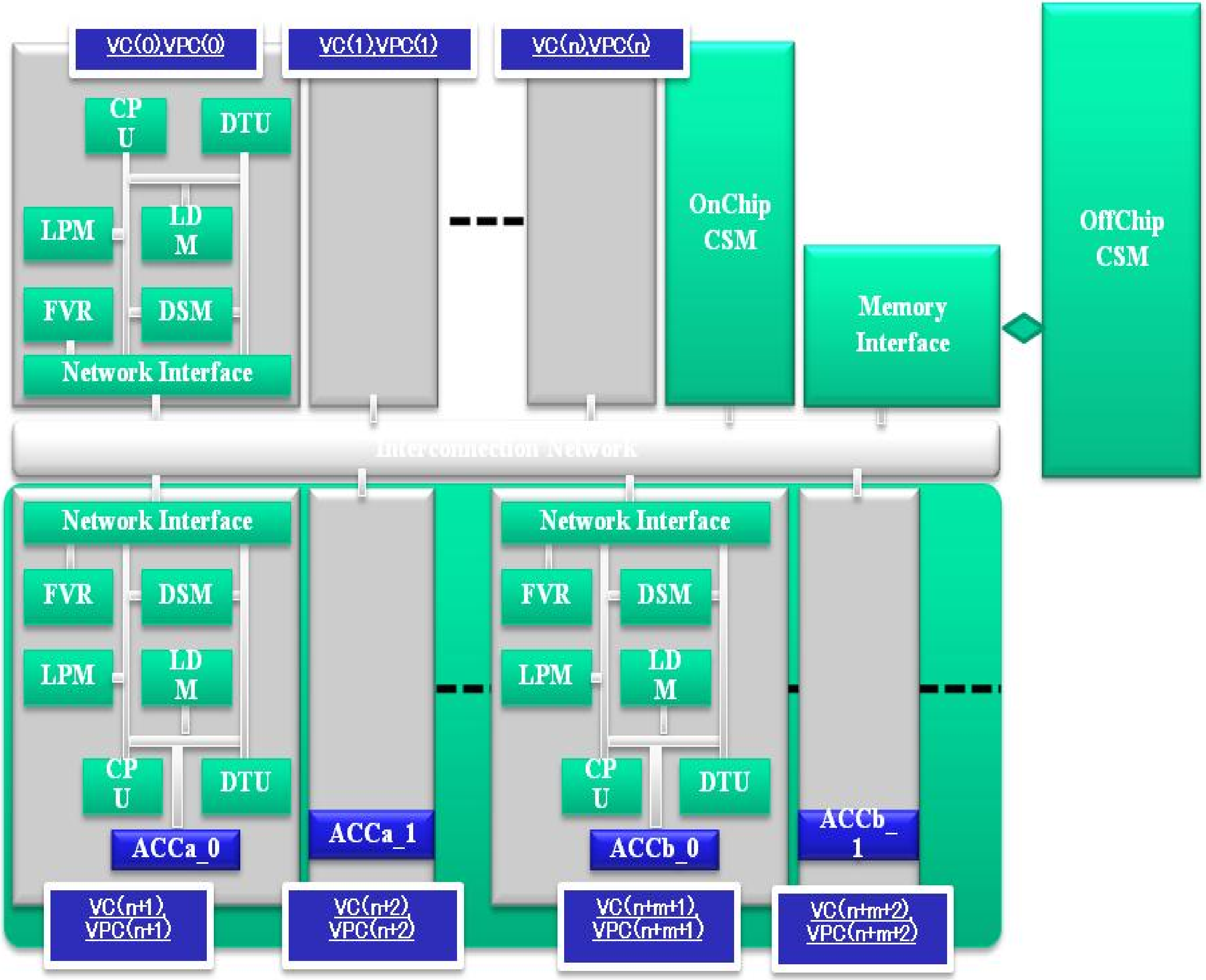
Road Tracking, Image Compression : http://www.mathworks.co.jp/jp/help/vision/examples

Buoy Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/44706-buoy-detection-using-simulink

Color Edge Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/28114-fast-edges-of-a-color-image--actual-color--not-converting-to-grayscale-/

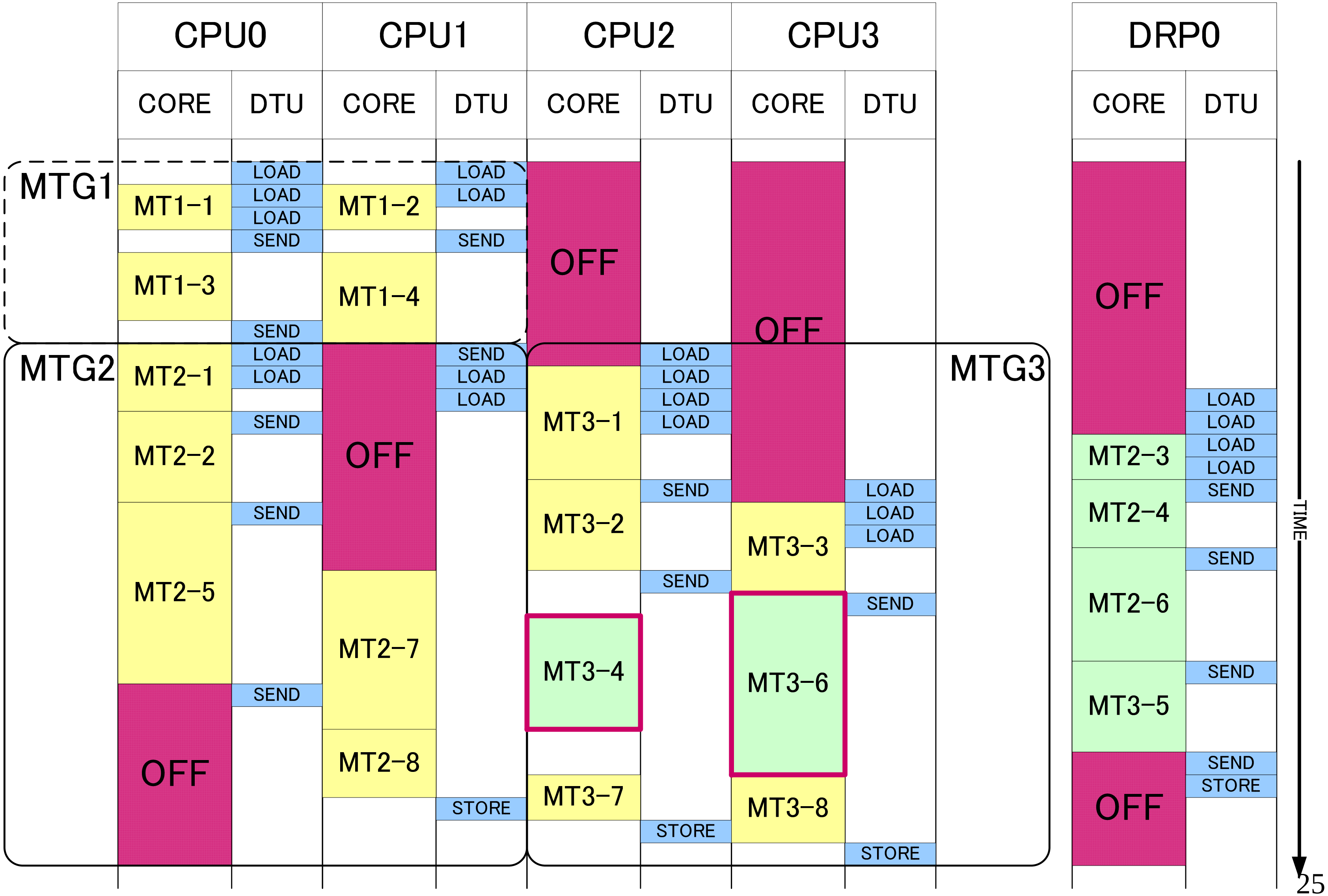
Vessel Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/24990-retinal-blood-vessel-extraction/

OSCAR Heterogeneous Multicore



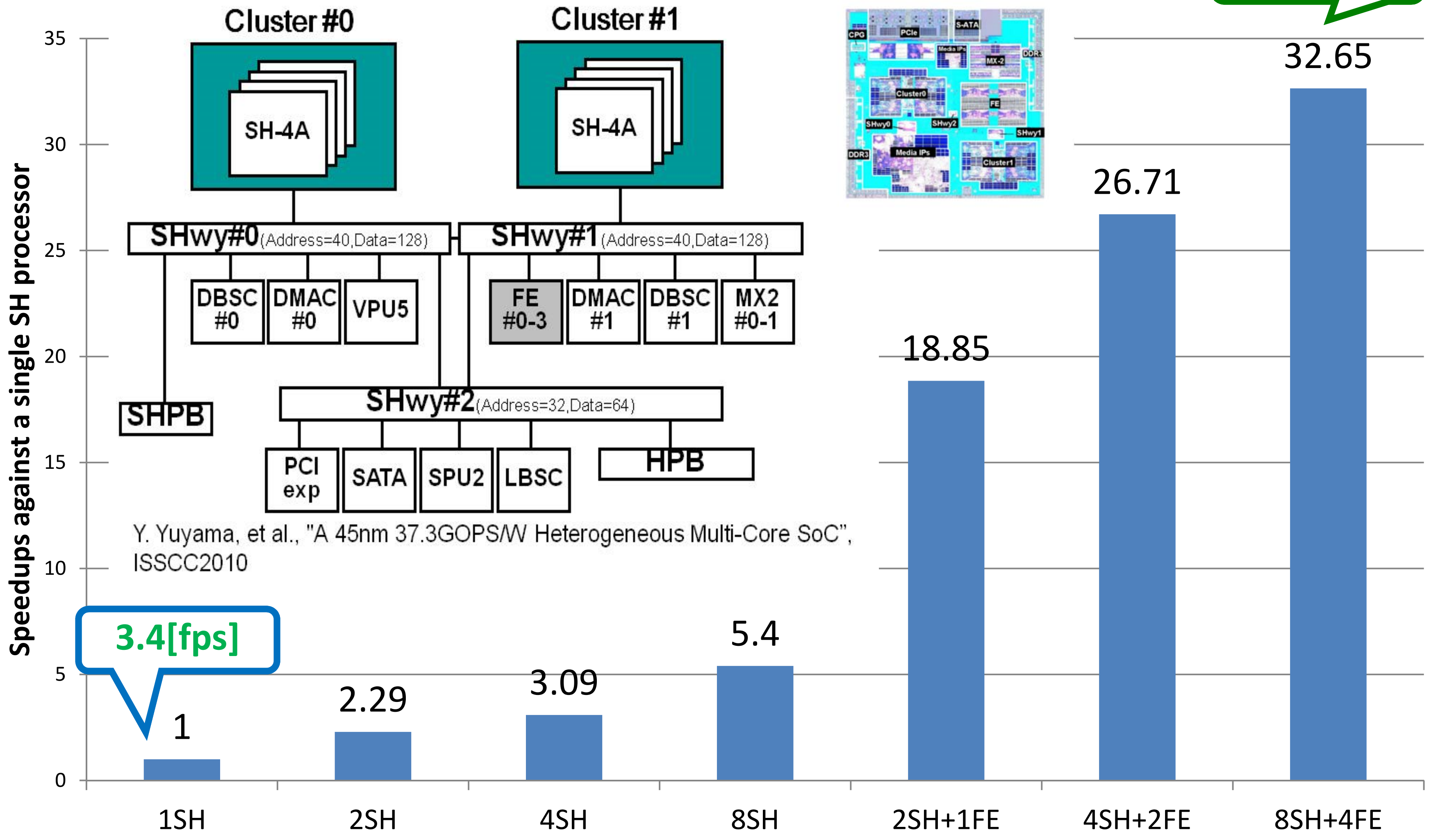
- DTU
  - Data Transfer Unit
- LPM
  - Local Program Memory
- LDM
  - Local Data Memory
- DSM
  - Distributed Shared Memory
- CSM
  - Centralized Shared Memory
- FVR
  - Frequency/Voltage Control Register

An Image of Static Schedule for Heterogeneous Multi-core with Data Transfer Overlapping and Power Control



33 Times Speedup Using
OSCAR Compiler and OSCAR API on RP-X

(Optical Flow with a hand-tuned library)



111[fps]

3.4[fps]

Y. Yuyama, et al., "A 45nm 37.3GOPS/W Heterogeneous Multi-Core SoC", ISSCC2010

Power Reduction in a real-time execution controlled by OSCAR Compiler and OSCAR API on RP-X (Optical Flow with a hand-tuned library)

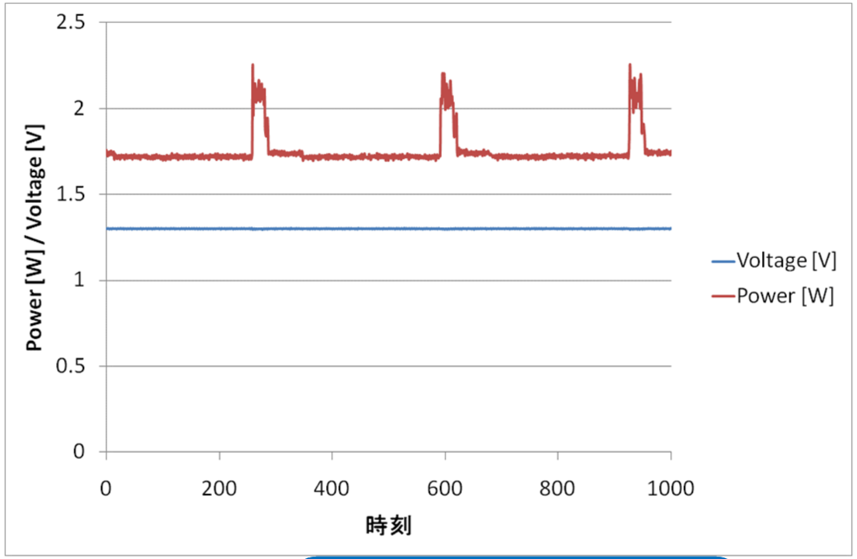
Without Power Reduction

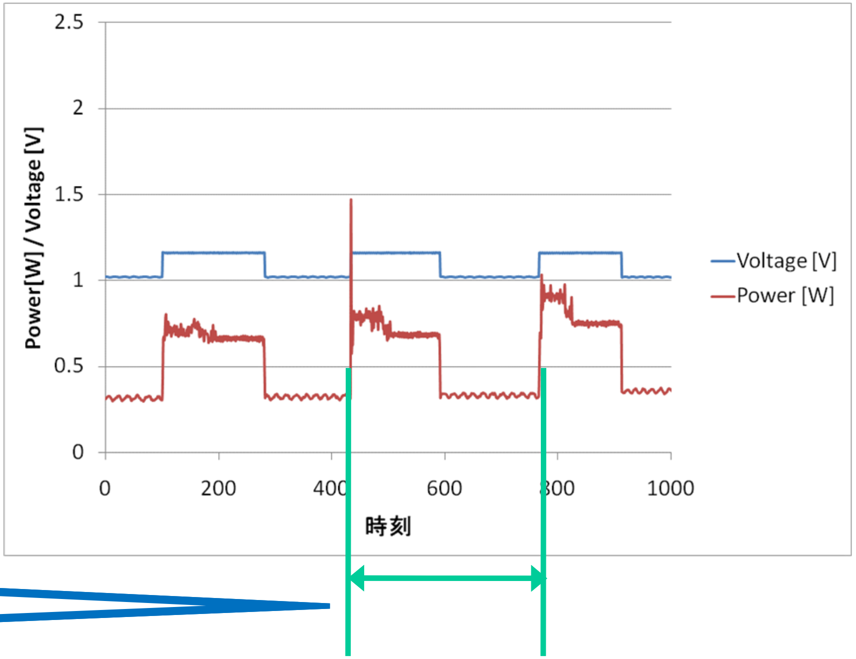
With Power Reduction by OSCAR Compiler

70% of power reduction

Average:1.76[W]

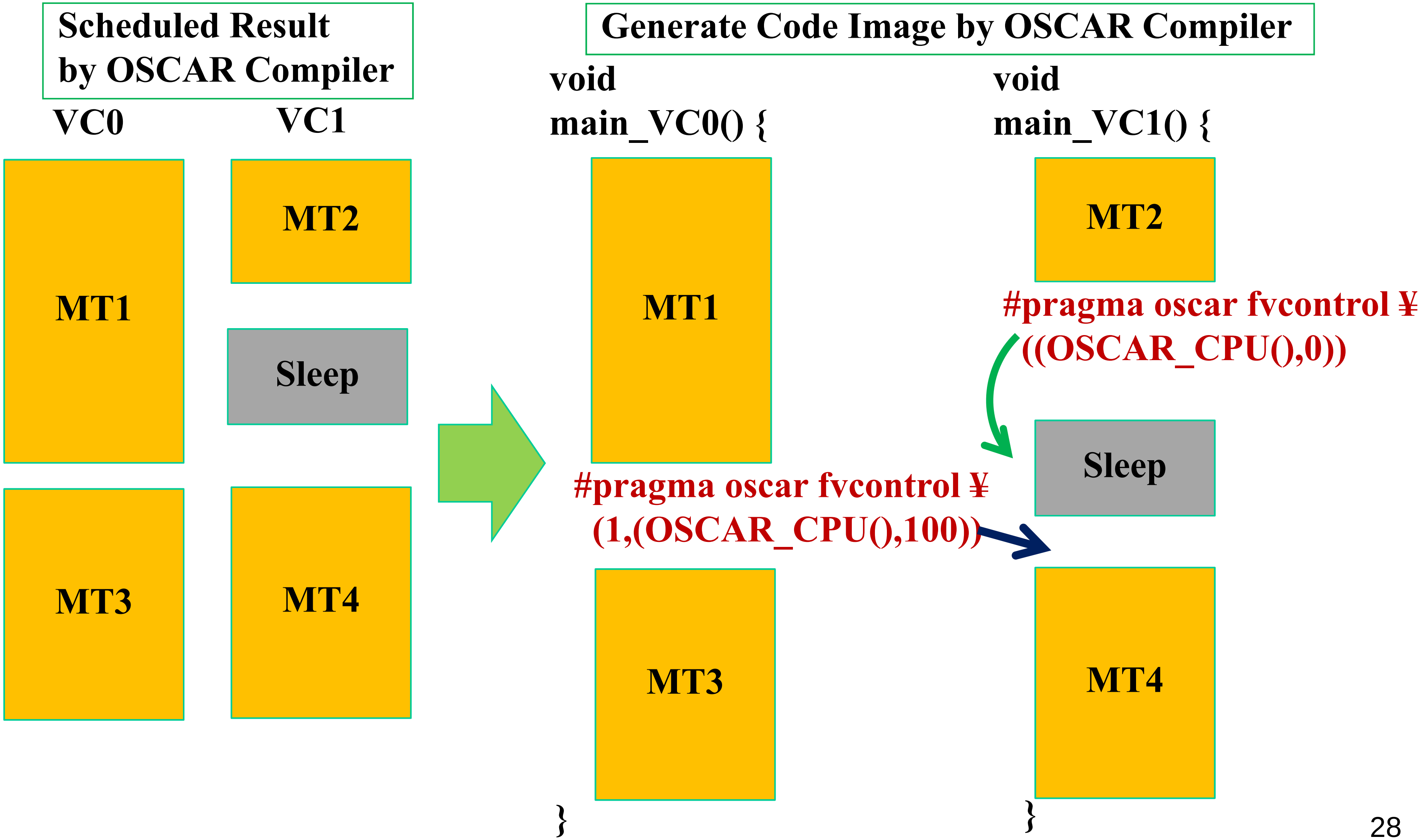
Average:0.54[W]





1cycle : 33[ms]
→30[fps]

Low-Power Optimization with OSCAR API



Scheduled Result
by OSCAR Compiler

VC0 VC1

MT1 MT2

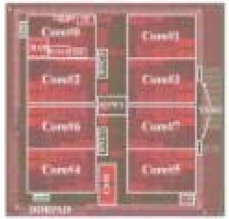
Sleep

MT3 MT4

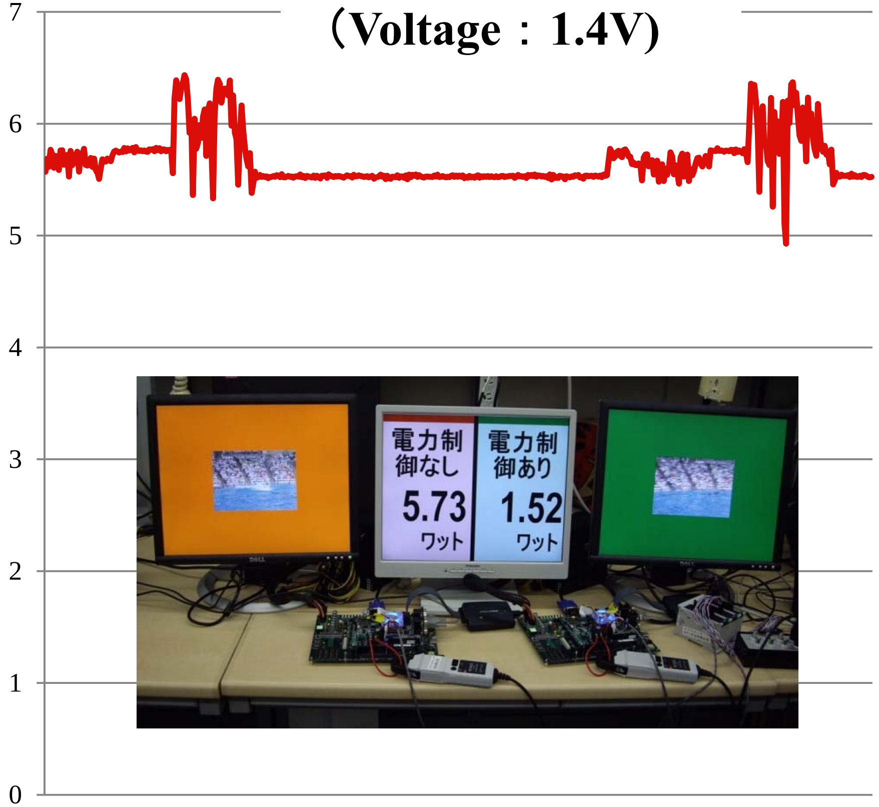
Generate Code Image by OSCAR Compiler

```
void
main_VC0() {
    MT1
#pragma oscar fvcontrol ¥
  (1,(OSCAR_CPU(),100))
    MT3
}

void
main_VC1() {
    MT2
#pragma oscar fvcontrol ¥
  ((OSCAR_CPU(),0))
    Sleep
    MT4
}
```

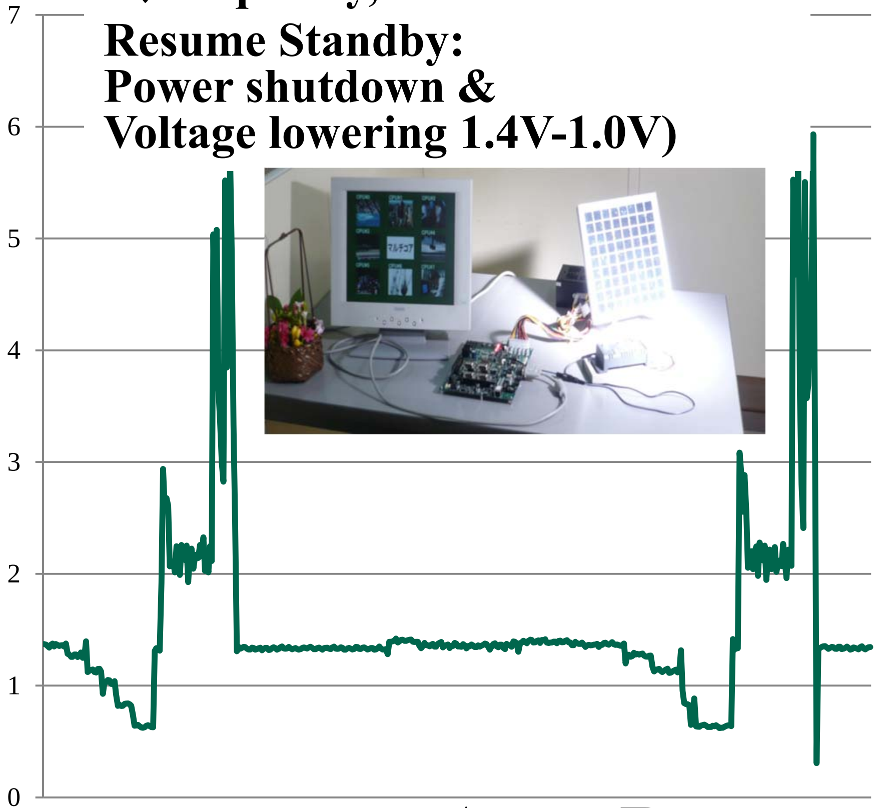
Power Reduction of MPEG2 Decoding to 1/4 on 8 Core Homogeneous Multicore RP-2 by OSCAR Parallelizing Compiler



MPEG2 Decoding with 8 CPU cores



Without Power Control （Voltage：1.4V)



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

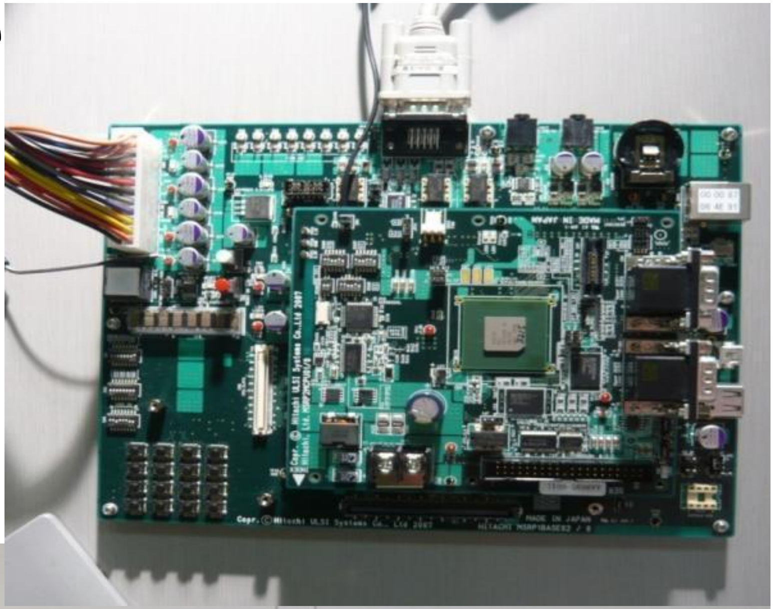
Avg. Power 5.73 [W]

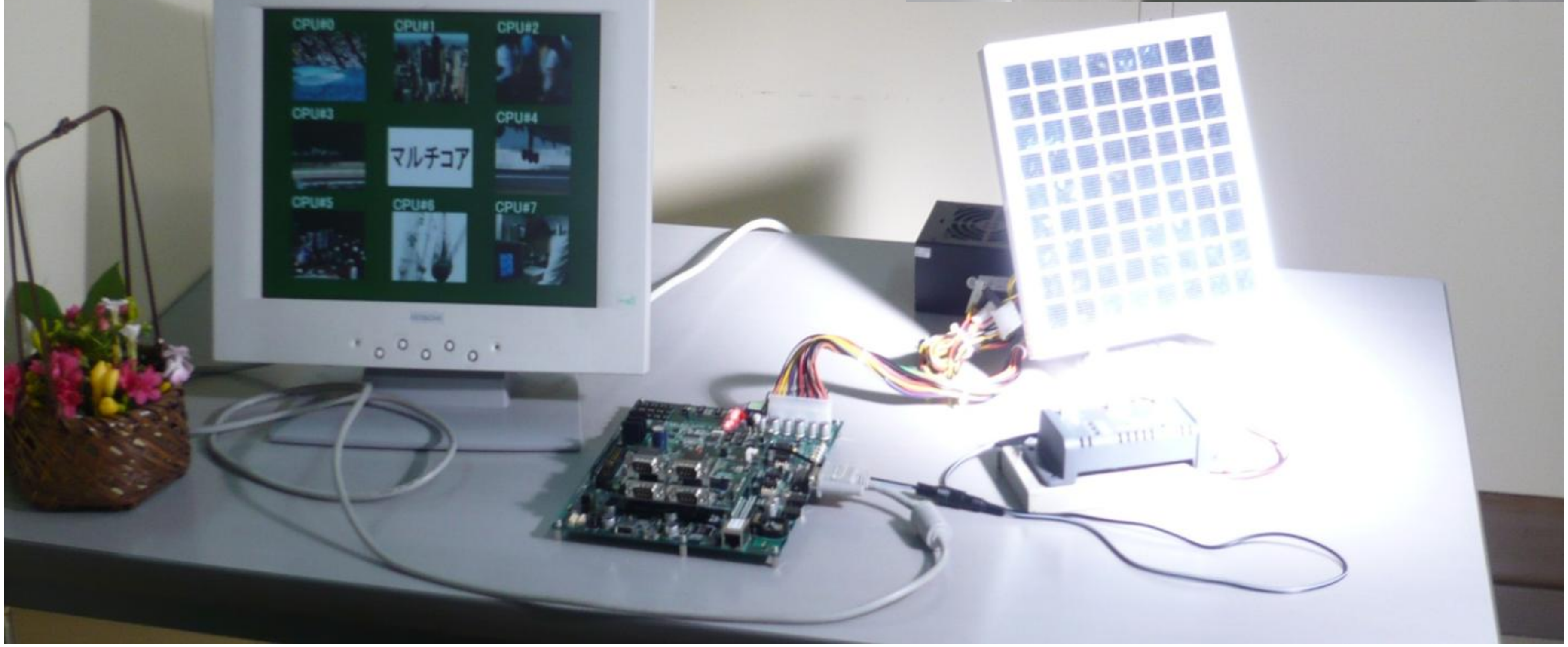
73.5% Power Reduction

Avg. Power 1.52 [W]

Low Power High Performance Multicore Computer with Solar Panel

- Clean Energy Autonomous
- Servers operational in deserts





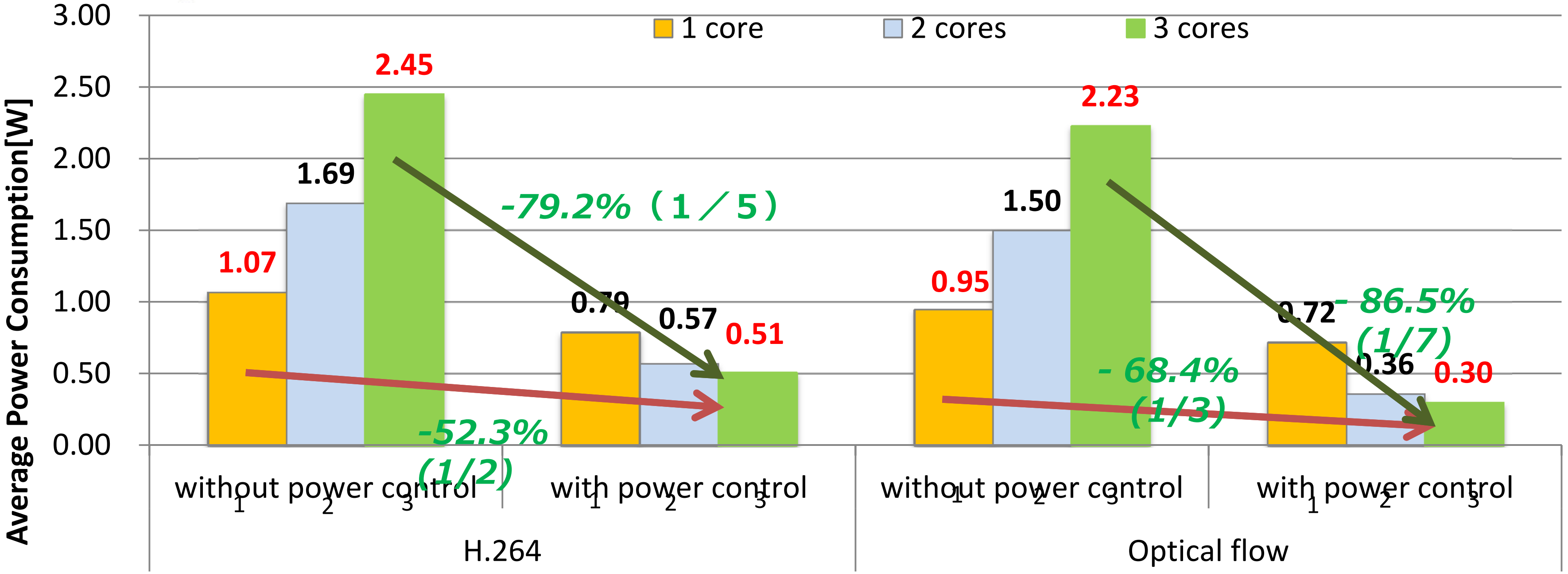
Power on 4 cores ARM CortexA9 with Android

http://www.youtube.com/channel/UCS43lNYEIkC8i_KIgFZYQBQ

H.264 decoder & Optical Flow (Using 3 cores)

ODROID X2

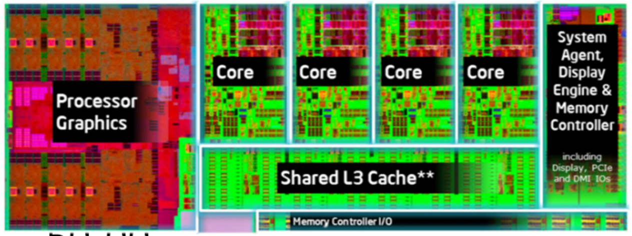
Samsung Exynos4412 Prime, ARM Cortex-A9 Quad core
1.7GHz～0.2GHz, used by Samsung's Galaxy S3



- On the same 3 cores, the power control reduced the power to 1/5～1/7 against no power control.
- The power control reduced the power to 1/2～1/3 compared with the ordinary sequential execution on 1 core without power control.

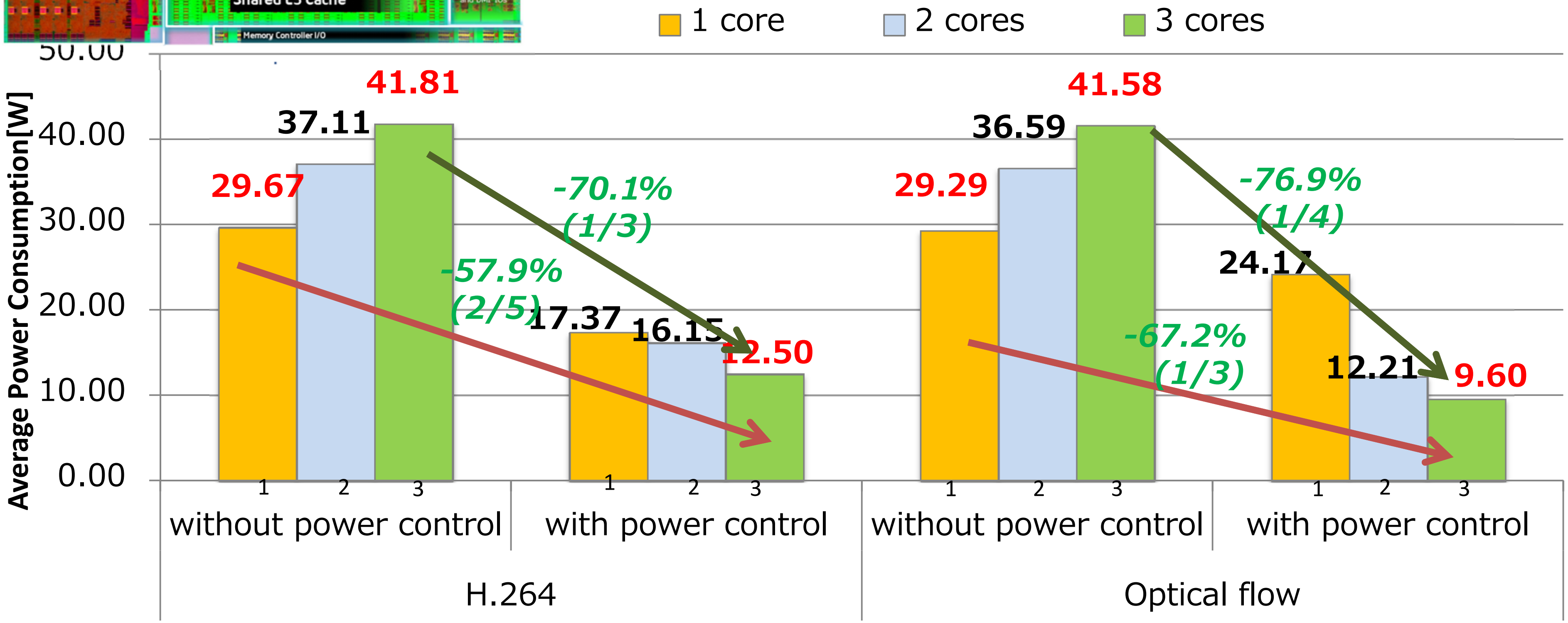
Power Reduction on Intel Haswell 3cores

H.264 decoder & Optical Flow



Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O

H81M-A, Intel Core i7 4770k
Quad core, 3.5GHz～0.8GHz



- ➢ **The power consumption was reduced to 1/3～1/4 by OSCAR power control against no power control on the same 3 processor cores.**
- ➢ **The power reduced to 2/5～1/3 by the compiler power control against 1core no power control.**

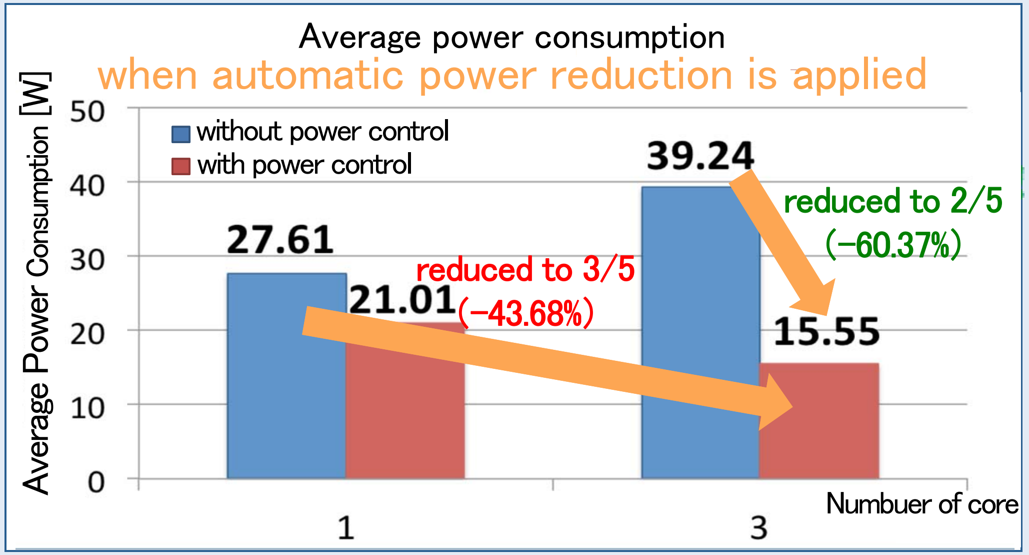
WASEDA UNIVERSITY

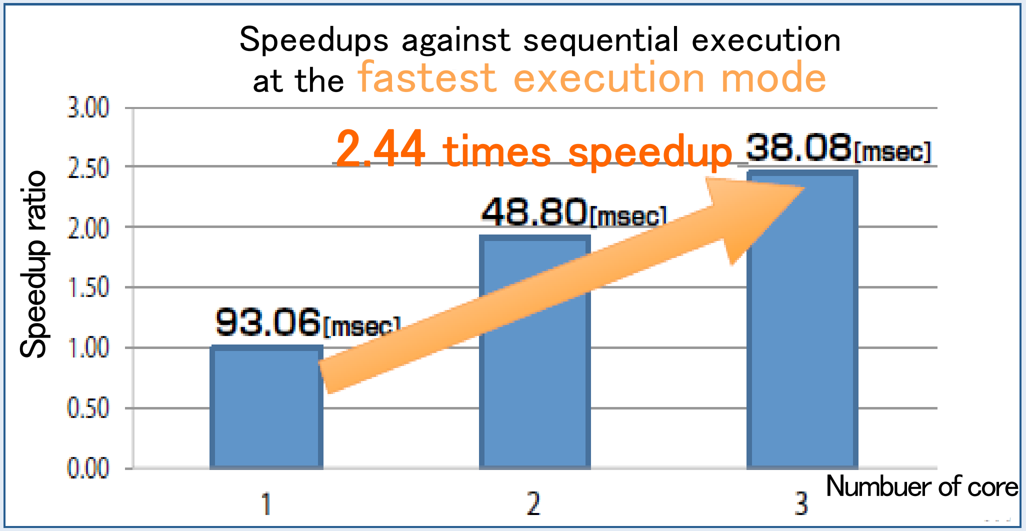
Automatic Power Reduction by OSCAR Compiler on Intel Haswell 4 Core Multicore

- Power Consumption for real-time face detection was reduced to 2/5 -

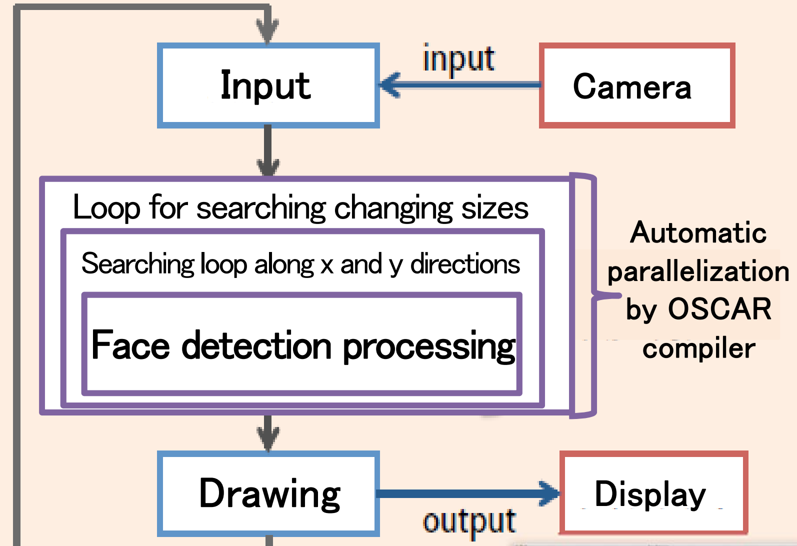
- OSCAR Compiler
- Intel Haswell
- Power Reduction

Parallel processing of face detection program on Intel Haswell 4cores





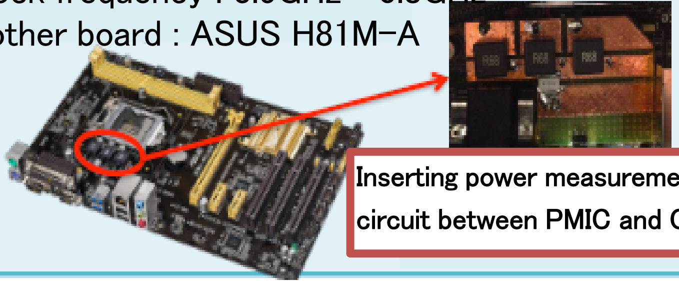
Parallelization flow of OpenCV face detection program



Input
input
Camera
Loop for searching changing sizes
Searching loop along x and y directions
Face detection processing
Automatic parallelization by OSCAR compiler
Drawing
output
Display
next frame

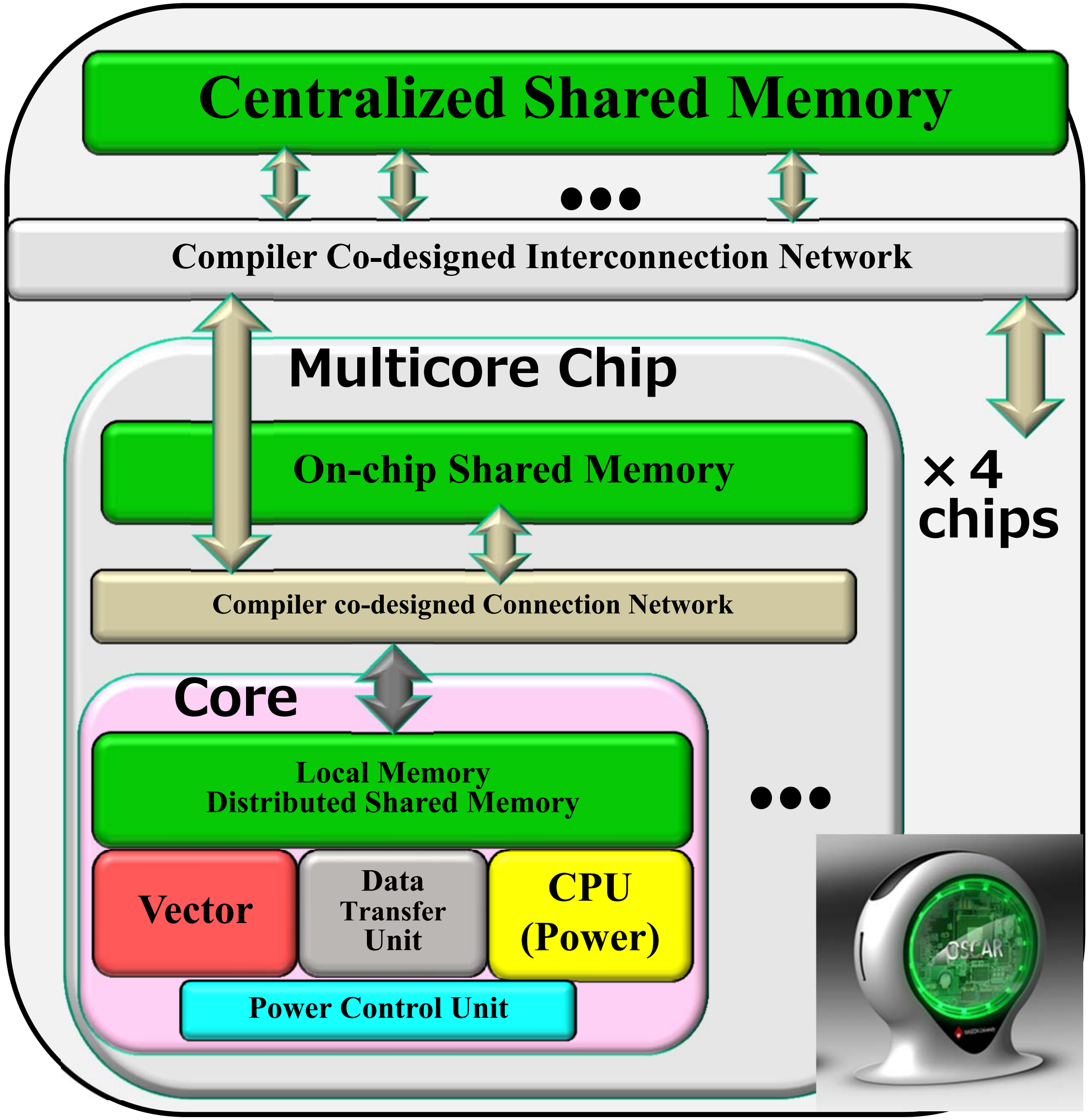
Power measurement on Intel Haswell board

CPU : Inel Core i7 4770K
Number of core : 4
Clock frequency : 3.5GHz～0.8GHz
Mother board : ASUS H81M-A



Inserting power measurement circuit between PMIC and CPU

OSCAR Vector Multicore with Power Processor Core and Compiler for Embedded to Severs with OSCAR Technology



Centralized Shared Memory

●●●

Compiler Co-designed Interconnection Network

Multicore Chip

On-chip Shared Memory

Compiler co-designed Connection Network

Core

Local Memory
Distributed Shared Memory

Vector

Data Transfer Unit

CPU (Power)

Power Control Unit

●●●

×4 chips

Target:

- Solar Powered with compiler power reduction.
- Fully automatic parallelization and vectorization including local memory management and data transfer.

Summary

- OSCAR Automatic Parallelizing and Power Reducing Compiler has succeeded speedup and/or power reduction of scientific applications including medical applications and natural disaster simulation, and real-time applications like Automobile Engine Control and MATLAB/Simulink, and media applications codec and face detection on various homogeneous and heterogeneous multicores.
- In automatic parallelization on Power Architectures:

  OSCAR Compiler has been developed on Power architectures like Power 4, 5, 5+, 6, 7 and 8.

  - for "Earthquake Wave Propagation Simulation", 110 times speedup on 128 cores of IBM Power 7 and 9.6 times speedup on 12 cores IBM Power8 against 1 core,
  - for "Cancer Treatment Using Carbon Ion", 55 times speedup on 64 cores of IBM Power 7 against 1 core
- In automatic power reduction, consumed powers were reduced to 1/2 or 1/3 using 3 cores on ARM Cortex A9 and Intel Haswell.
- We are planning to realize automatic power reduction using "On-Chip Regulator" on Power Processors with accelerators.