

**マルチコアプロセッサシステム上で、
並列プログラムの高速・低消費電力動作を可能とする
世界初のソフトウェア標準(OSCAR API ver. 2.0)
を開発・無料公開**

学校法人早稲田大学

説明者・連絡先

理工学術院 基幹理工学部 情報理工学科 教授

グリーンコンピューティングシステム研究機構

アドバンスマルチコアプロセッサ研究所 所長

笠原博徳

<http://www.kasahara.cs.waseda.ac.jp/>

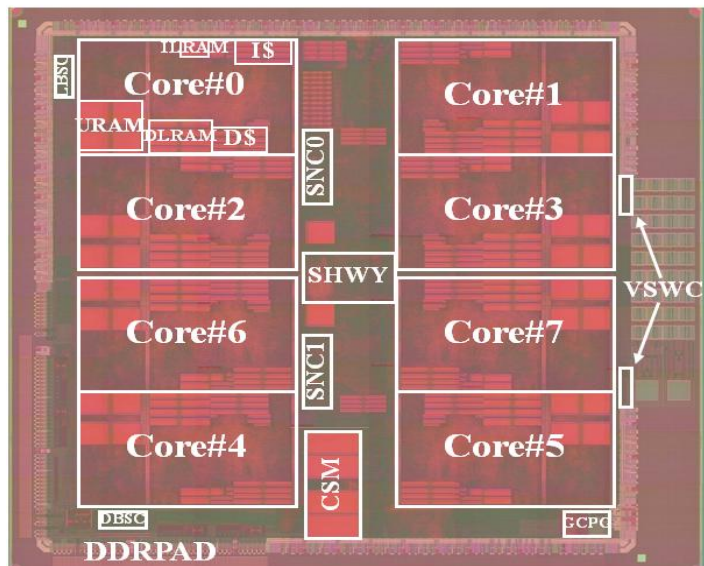
kasahara@waseda.jp

Tel. 03-3203-4485, FAX 03-3203-4523

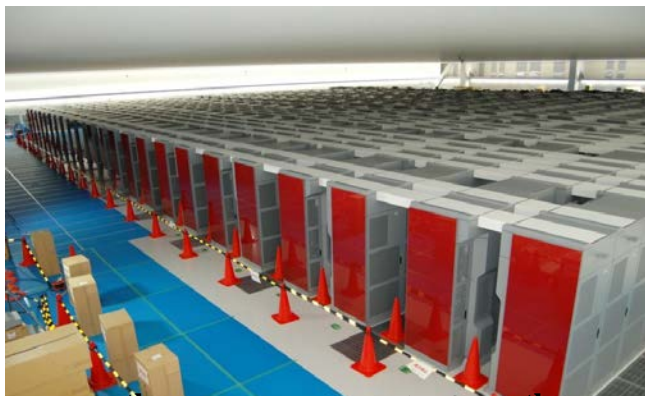
2012年 4月25日, グリーンコンピューティングシステム研究開発センターにて

Multi/Many-core Everywhere

Multi-core from embedded to supercomputers



OSCAR Type Multi-core Chip by Renesas in METI/NEDO Multicore for Real-time Consumer Electronics Project (Leader: Prof.Kasahara)



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%)

➤ 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), Tileria 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

Earth Simulator:**40TFLOPS**, 2002, 5120 vector proc.

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 8score128GFLOPS) -T (64 core,192GFLOPS:2011)

RIKEN Fujitsu “K” 10PFLOPS(8score 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

グリーン・コンピューティング・システム研究開発センター 概要

2011年4月13日竣工, 2011年5月13日開所(記念シンポジウム)

経済産業省「2009年度産業技術研究開発施設整備費補助金」
先端イノベーション拠点整備事業

<目標>

太陽電池で駆動可能で
冷却ファンが不要な
超低消費電力・高性能マルチコア/
メニーコアプロセッサ*のハードウェア,
ソフトウェア, 応用技術の研究開発

*1チップ上に多数のプロセッサコアを
集積する次世代マルチコアプロセッサ

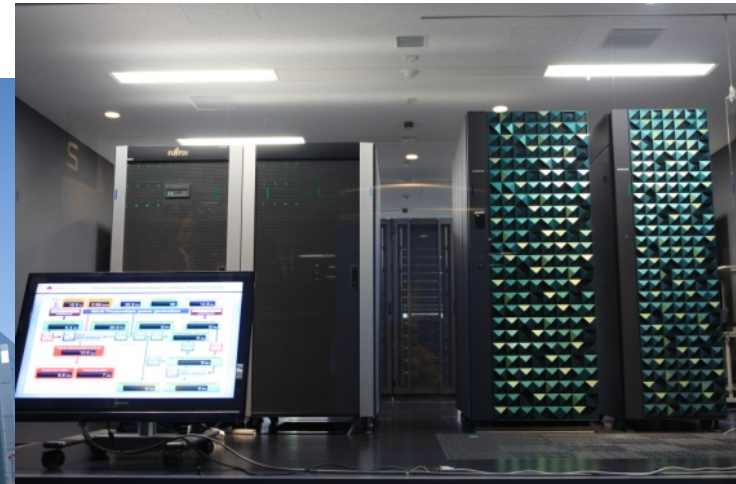
<産学連携>

日立, 富士通, ルネサス, NEC, トヨタ,
デンソー, オリンパス,
三菱電機(重粒子線ガン治療) 等

<波及効果>

超低消費電力メニーコア

- CO₂排出量削減
- サーバ国際競争力強化
- 我が国の産業利益を支える
情報家電, 自動車等の高付加価値化

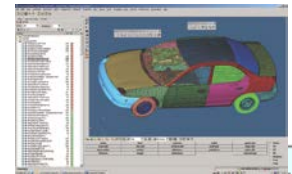


2011グッドデザイン賞受賞

命を守る

産官学連携研究開発・実用化(波及効果)

環境を守る



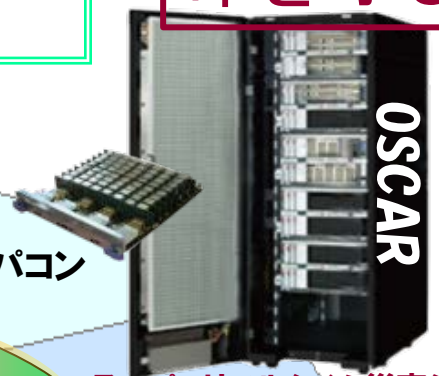
ロボット

車載(カーナビ・
統合制御・インフラ協調)

環境を守る

早稲田大学

グリーンスパコン



OSCAR

超低消費電力・メニーコアシステム技術

OSCAR API, OSCAR並列化コンパイラ

スーパーリアルタイム災害シミュレーション(地殻変動,津波)
耐震性解析, 火山噴火溶岩流



OSCAR
Many-core
Chip

グリーンクラウドサーバ

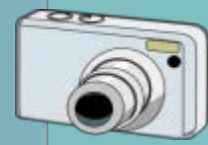
情報家電

ネットTV/DVD



カメラ

カムコーダ



情報家電

産業競争力を守る

モバイル系メニーコア

スマートフォン

携帯電話



太陽電池駆動・充電

ネットブック



産業界

クールデスクトップサーバ



放医研パンフレットより

医用画像処理等専用サーバ
(医療:重粒子線照射計画, 脳梗塞)
自動車・航空機設計サーバ
太陽電池駆動
クールサーバ

スパコン・サーバ



OSCAR

市場規模: 情報家電・携帯電話系 数10兆円

市場規模: スパコン・サーバ系 1兆円

OSCAR API v2.0

ホモジニアスマルチコア、ヘテロジニアスマルチコア、メニーコア及び共有メモリ型マルチプロセッササーバ用の低消費電力高性能並列プログラムを記述するためのソフトウェア標準

- NEDOプロジェクトで開発し、2008年11月14日に公開したホモジニアスマルチコア用OSCAR API ver.1.0を発展させた、22個のC及びFortran言語用のコンパイラ指示文の集合体(他に2つのヒント指示文有り)
 - 仕様が小さくシンプルで分かり易く、低コストで実装可能。
- 電力制御指示文により、CあるいはFortranプログラム中から、各プロセッサコア、メモリ、コネクションネットワークの動作周波数、電圧、電源遮断も行えるため、チップ消費電力の大幅削減可能
- 自動車等の組込システムで要求されるハードリアルタイム処理も行えるように、タイマー管理機能、ローカルメモリ、DMAC(データ転送ハードウェア)制御指示文も用意
- OSCAR API ver.2.0で記述された並列プログラムは、早稲田大学OSCAR並列化コンパイラにより、逐次プログラムから自動生成することができ、マルチコアで問題となるソフトウェアの開発コスト・開発期間を大幅削減可能
- APIとOSCARコンパイラを同時に用いることにより、スマートフォン、次世代カメラ、ゲーム機等の情報家電用マルチコア、自動車エンジン制御用ハードリアルタイム用マルチコア、医療用自然冷却(無冷却ファン)静音衛生的マルチコア、低消費電力クラウドサーバ用マルチコア、スーパーコンピュータ用ヘテロジニアスマルチコアあるいはメニーコアなどの低消費電力並列プログラムを、短期間・低コストで作成可能
- ユーザによる、並列プログラムの手動作成も可能
- API仕様は、2012年4月25日より<http://www.kasahara.cs.waseda.ac.jp/> で公開、無料ダウンロード可
- 国内12社+3大学で開発
 - イーソル(株)、オリンパス(株)、ガイオ・テクノロジー(株)、キャッツ(株)、(株)デンソー、(株)東芝、東邦大学、名古屋大学、日本電気(株)、(株)日立製作所、(株)富士通研究所、三菱電機(株)、ルネサスエレクトロニクス(株)、(株)ルネサスソリューションズ、早稲田大学(五十音順)

第1期マルチコア・アーキテクチャ・API検討委員会(2005-2008)

NEDO「リアルタイム情報家電用マルチコア技術の研究開発」

OSCAR API Version 1.0

- 委員長 笠原 博徳(早稲田大学)
- 副委員長 内山 邦男(株式会社日立製作所)
- 枝廣 正人(日本電気株式会社、オブザーバとして)
- 木村 啓二(早稲田大学)
- 佐藤 真琴(株式会社日立製作所)
- 須賀 敦浩(株式会社富士通研究所)
- 高橋 宏政(株式会社富士通研究所)
- 十山 圭介(株式会社日立製作所)
- 長谷川 淳(株式会社ルネサステクノロジ)
- 前田 誠司(株式会社東芝)
- 前田 昌樹(パナソニック株式会社)

第2期メニーコア・アーキテクチャ・API検討委員会(2009-2010)

NEDO「情報家電用ヘテロジニアス・マルチコア技術の研究開発」・

「メニーコア・プロセッサ技術の先導研究」

OSCAR API ヘテロジニアス及びメニーコア拡張検討

- 委員長 笠原 博徳(早稲田大学)
- 副委員長 内山 邦男(株式会社日立製作所)
- 枝廣 正人(日本電気株式会社)
- 木村 啓二(早稲田大学)
- 佐藤 真琴(株式会社日立製作所)
- 高橋 宏政(富士通株式会社)
- 長谷川 淳(株式会社ルネサステクノロジ)
- 前田 誠司(株式会社東芝)

オブザーバ

- 石坂 一久(日本電気株式会社)
- 神谷 幸男(富士通株式会社)
- 斎藤 靖彦(株式会社ルネサステクノロジ)
- 十山 圭介(株式会社日立製作所)

第3期メニーコア・アーキテクチャ・API検討委員会(2011-)

OSCAR API Version 2.0 策定

- 委員長 笠原 博徳(早稲田大学)
- 副委員長 内山 邦男(株式会社日立製作所)
- 浅野 昌尚(ガイオ・テクノロジー株式会社)
- 枝廣 正人(名古屋大学)
- 木村 啓二(早稲田大学)
- 久村 孝寛(日本電気株式会社)
- 権藤 正樹(イーソル株式会社)
- 佐藤 真琴(株式会社ルネサスソリューションズ)
- 徳吉 隆宏(株式会社東芝)
- 津上 浩伸(三菱電機株式会社)
- 中野 恵一(オリンパス株式会社)
- 中平 直司(株式会社 富士通研究所)
- 長谷川 淳(株式会社ルネサステクノロジ)
- 藤本 洋(キャッツ株式会社)
- 森 裕司(株式会社デンソー)
- 吉田 明正(東邦大学)

オブザーバ

- 新居 俊亮(三菱スペース・ソフトウェア株式会社)
- 神谷 幸男(富士通株式会社)
- 高村 守幸(富士通株式会社)
- 宮森 高(株式会社東芝 セミコンダクター社)

OSCAR API v2.0の指示文一覧

Fortran & C 用の22個の指示文 + 2ヒント指示文

▶ 並列実行API

- ▶ **parallel sections (*)**
- ▶ **flush (*)**
- ▶ **critical (*)**
- ▶ **execution**

▶ メモリ配置API

- ▶ **threadprivate (*)**
- ▶ **distributedshared**
- ▶ **onchipshared**

▶ 同期API

- ▶ **groupbarrier**

▶ 電力制御API

- ▶ **fvcontrol**
- ▶ **get_fvstatus**

▶ タイマーAPI

- ▶ **get_current_time**

▶ データ転送API

- ▶ **dma_transfer**
- ▶ **dma_contiguous_parameter**
- ▶ **dma_stride_parameter**
- ▶ **dma_flag_check**
- ▶ **dma_flag_send**

▶ アクセラレータAPI

- ▶ **accelerator_task_entry**

▶ キャッシュ制御API

- ▶ **cache_writeback**
- ▶ **cache_selfinvalidate**
- ▶ **complete_memop**
- ▶ **noncacheable**
- ▶ **aligncache**

◆ ヒント指示文

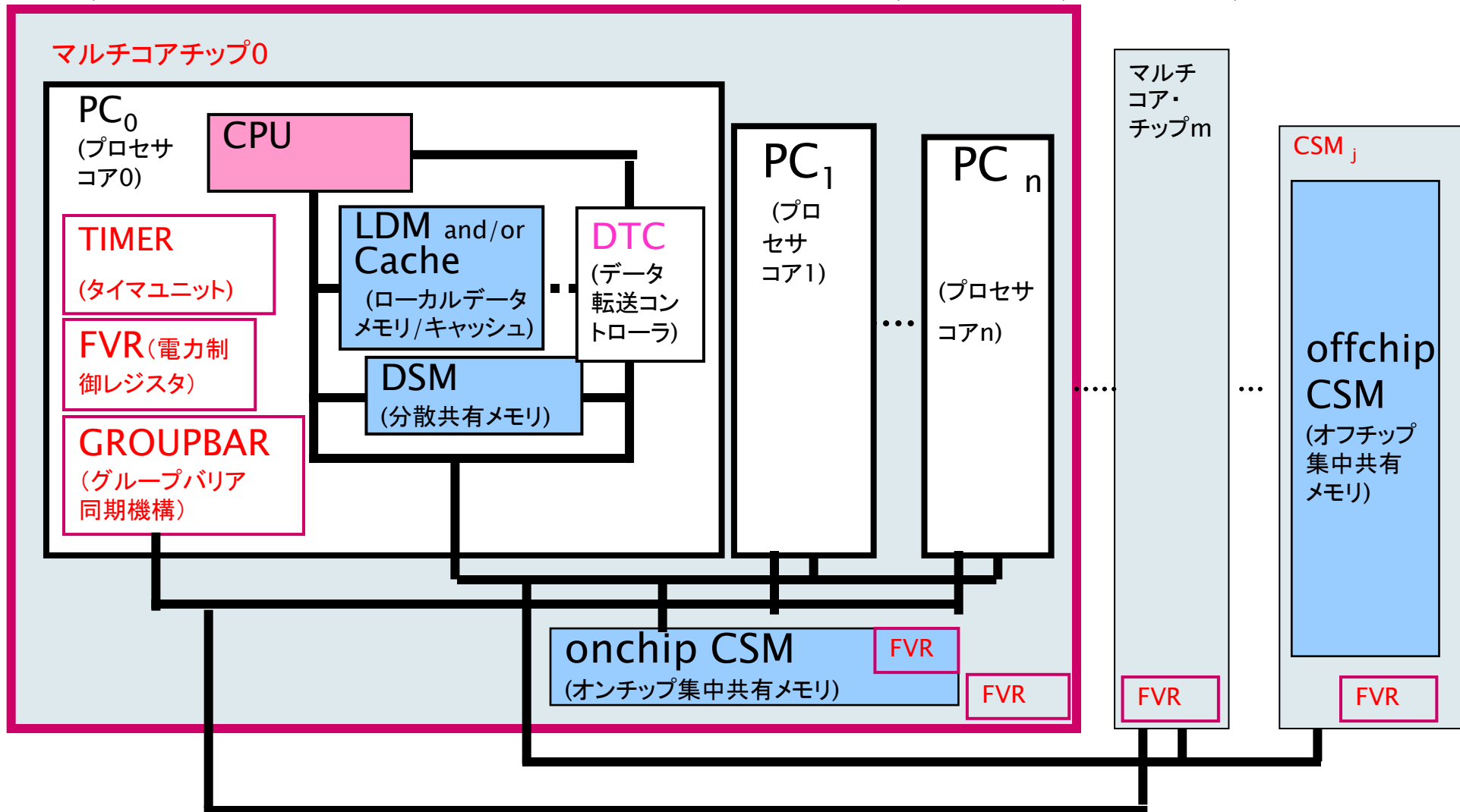
- ▶ **accelerator_task**
- ▶ **oscar_comment**

(*) **OpenMP指示文**

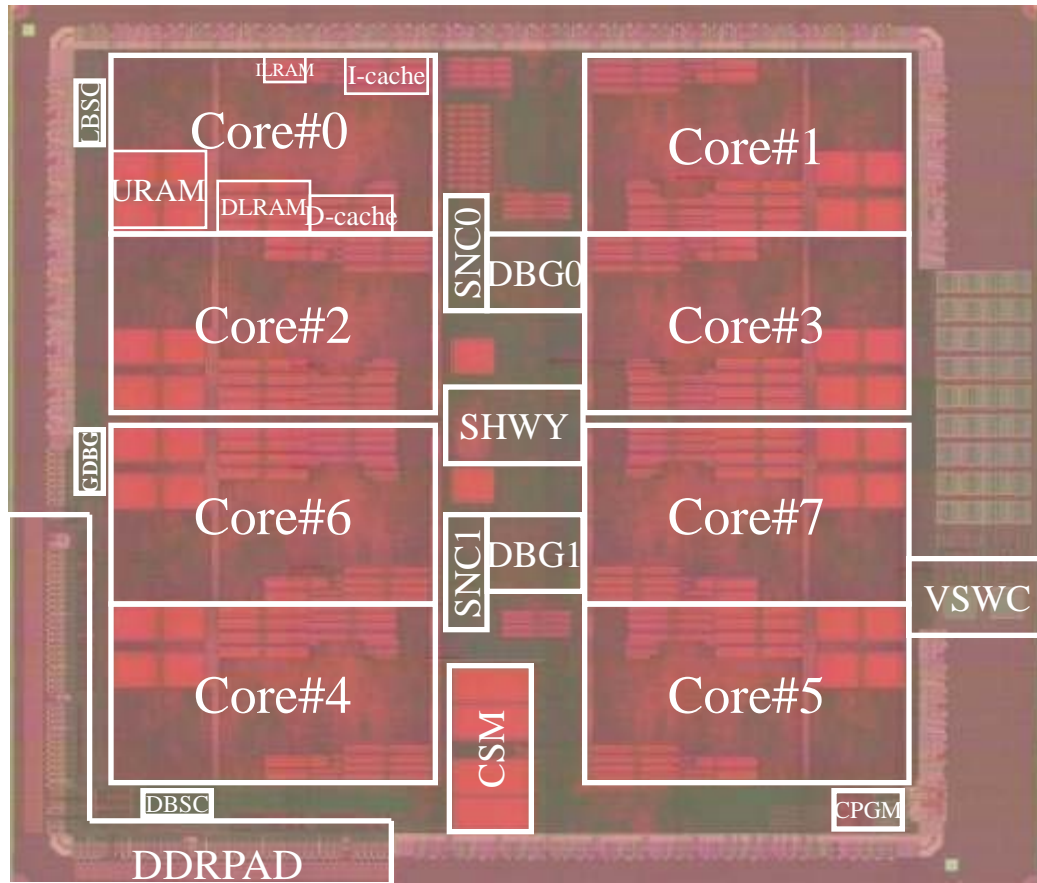
OSCAR: Optimally Scheduled Advanced Multiprocessor

OSCAR API Version 1.0が対象とする ホモジニアス用OSCARメモリアーキテクチャ

(SMP, 共有メモリ+ローカルメモリ/分散共有メモリ, DMAC, 電力制御, バリア同期等)



早稲田OSCARコンパイラ協調型アーキテクチャ ホモジニアスマルチコアRP2 SH4A8コア搭載



8コア集積マルチコアLSIチップ写真

プロセス	90nm CMOS, 8層メタル, 3種Vth
チップサイズ	104.8mm ² (10.61mm x 9.88mm)
電源電圧	1.0V-1.4V(コア), 1.8/3.3V(I/O)
動作周波数	600MHz
CPU性能	8640 MIPS (Dhrystone 2.1)
FPU性能	33.6 GFLOPS
低電力制御	<ul style="list-style-type: none"> ・CPU毎に独立した周波数変更 ・CPUコアのクロックを停止するスリープモード ・CPUコアの一部のクロックを停止するがキャッシュコヒーレンシ維持可能なライトスリープモード ・CPUコアの電源供給を停止するフル電源遮断モード ・URAM以外のCPUコアの電源供給を停止するレジューム電源遮断モード

ISSCC08発表: ISSCC08 論文番号4.5, M.ITO, et al., “An 8640 MIPS SoC with Independent Power-off Control of 8 CPUs and 8 RAMs by an Automatic Parallelizing Compiler”

OSCAR APIのヘテロジニアスマルチコアへの拡張

特定処理の高速かつ
低消費電力な処理への要求

- スマートフォン
- カメラ
- クラウドサーバ
- スパコン

ヘテロジニアスなコア構成の
マルチコア

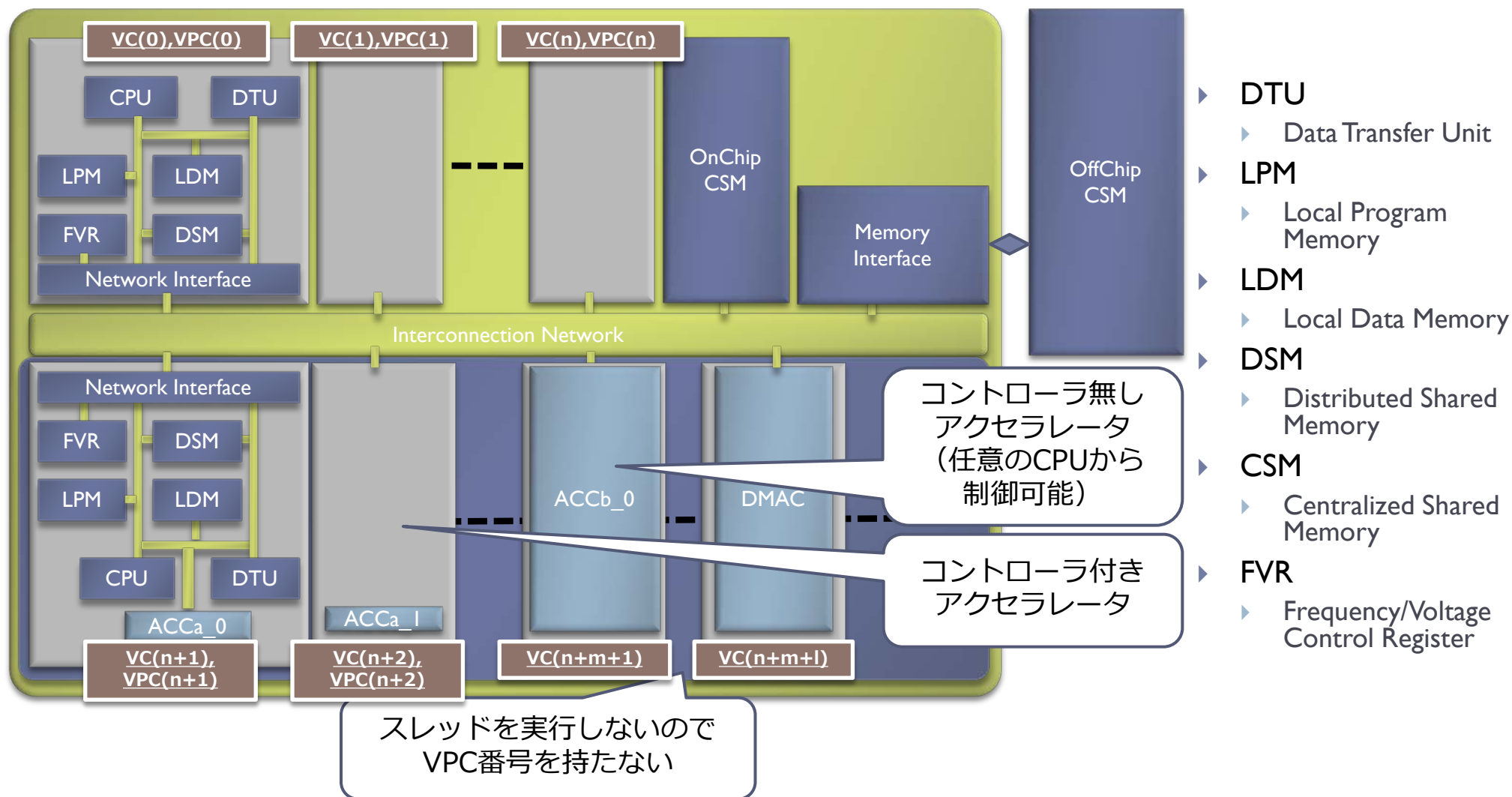
- 汎用コア+アクセラレータ
 - リコンフィギャラブルコア(DRP)、
グラフィクスアクセラレータ(GPU)等
- 電力あたりの処理能力の向上
- プログラム開発が難しい
- アクセラレータとのデータ転送遅い

OSCAR APIの
ヘテロジニアスマルチコアへの拡張

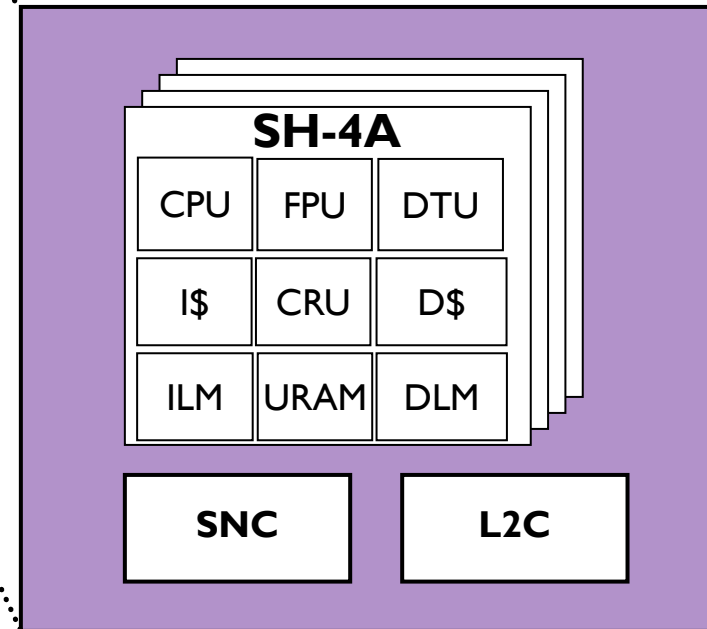
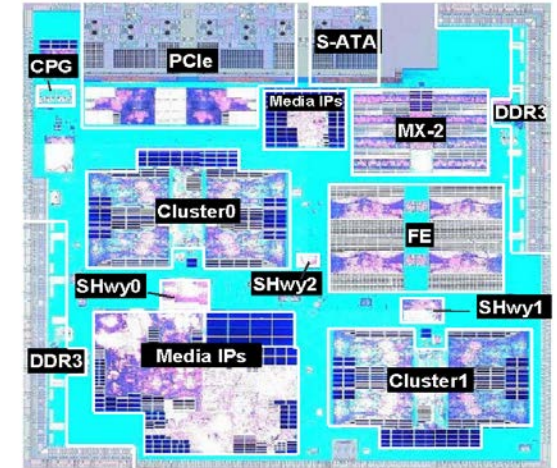
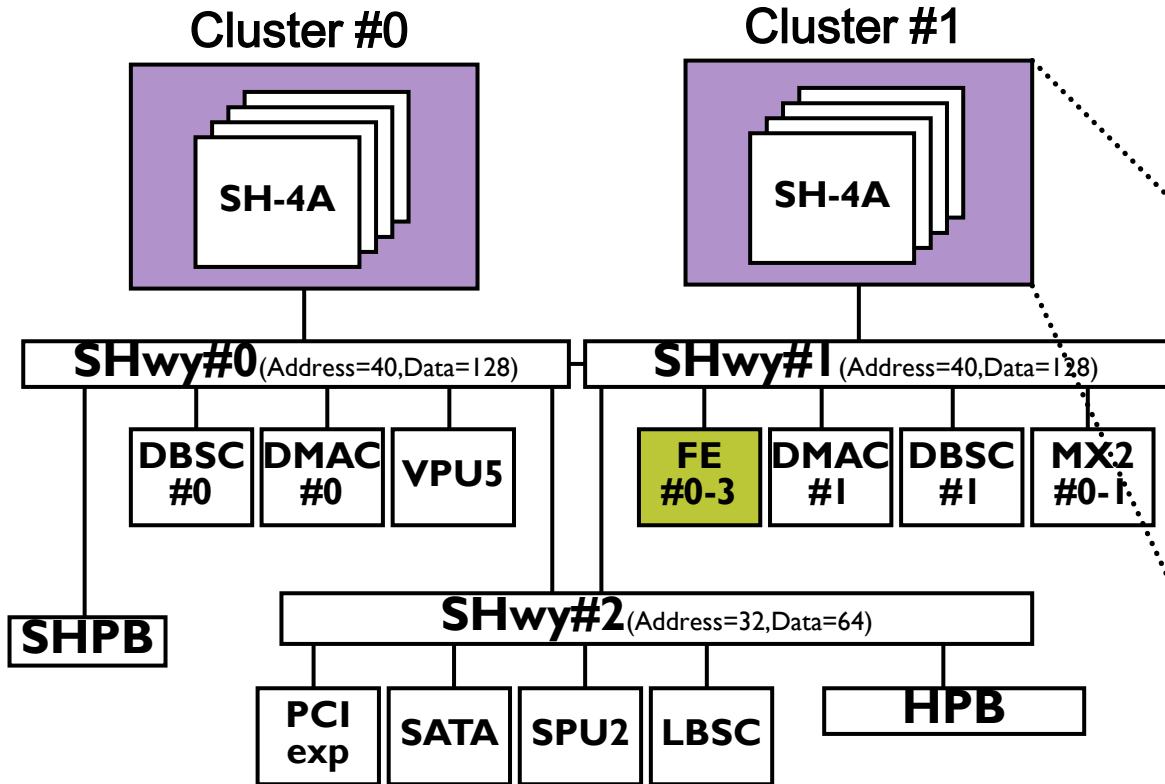
- アクセラレータコアの制御

OSCAR API Version 2.0で並列プログラムの作成が可能な ヘテロジニアス&ホモジニアスマルチコアアーキテクチャ

汎用コア+アクセラレータ (ベクター/SIMD/GPU等), メニーコア (ノンコヒーレント可)



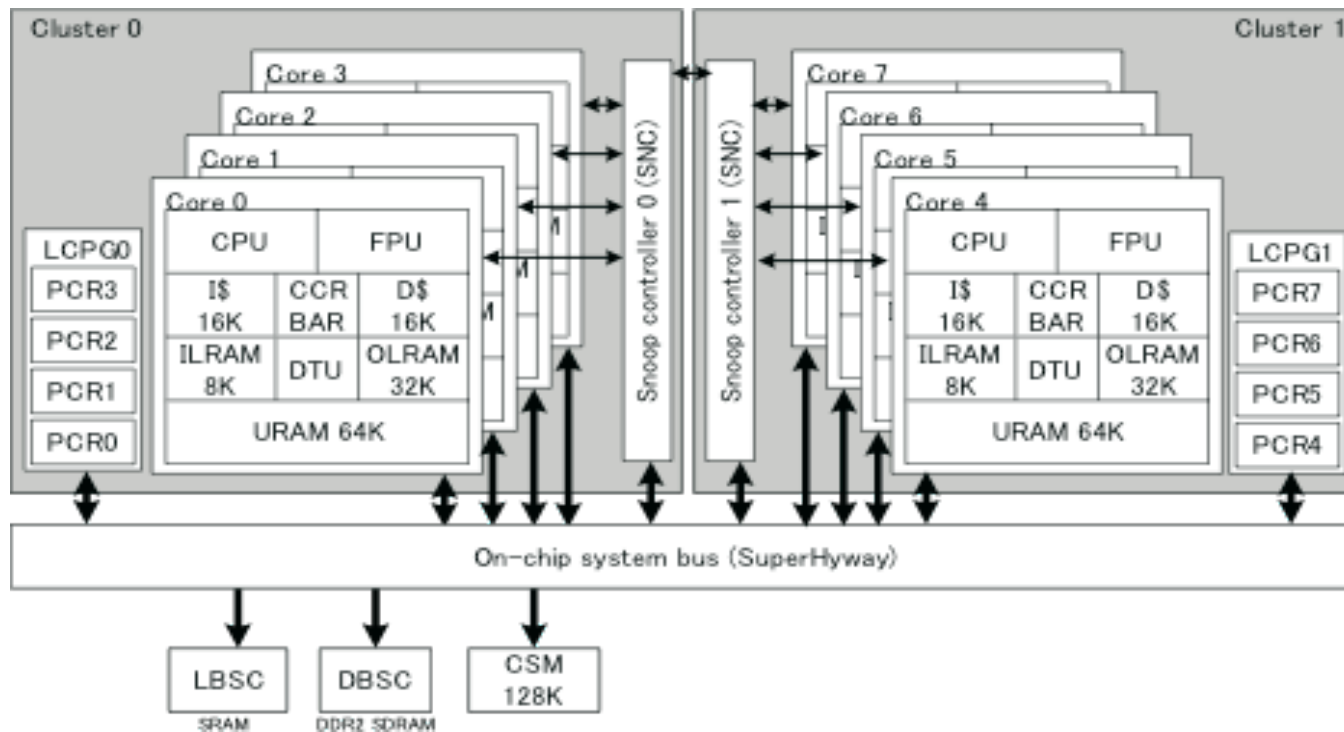
15コアヘテロジニアスマルチコアRP-X



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

ルネサステクノロジ・日立・東工大・早稲田により開発

情報家電マルチコアRP2を用いた コンパイラによるコヒーレンス制御



Process Technology	90nm, 8-layer, triple-Vth, CMOS
Chip Size	104.8mm ² (10.61mm x 9.88mm)
CPU Core Size	6.6mm ² (3.36mm x 1.96mm)
Supply Voltage	1.0V-1.4V (internal), 1.8/3.3V (I/O)
Clock frequency	600MHz, 300MHz, 150MHz, 75MHz
Power Domains	17 (8 CPUs, 8 URAMs, common)

M. Ito, et al., "An 8640 MIPS SoC with Independent Power-off Control of 8 CPU and 8 RAMS by an Automatic Parallelizing Compiler", ISSCC2008

クラスタ間ではハードウェアはコヒーレンスを維持しない

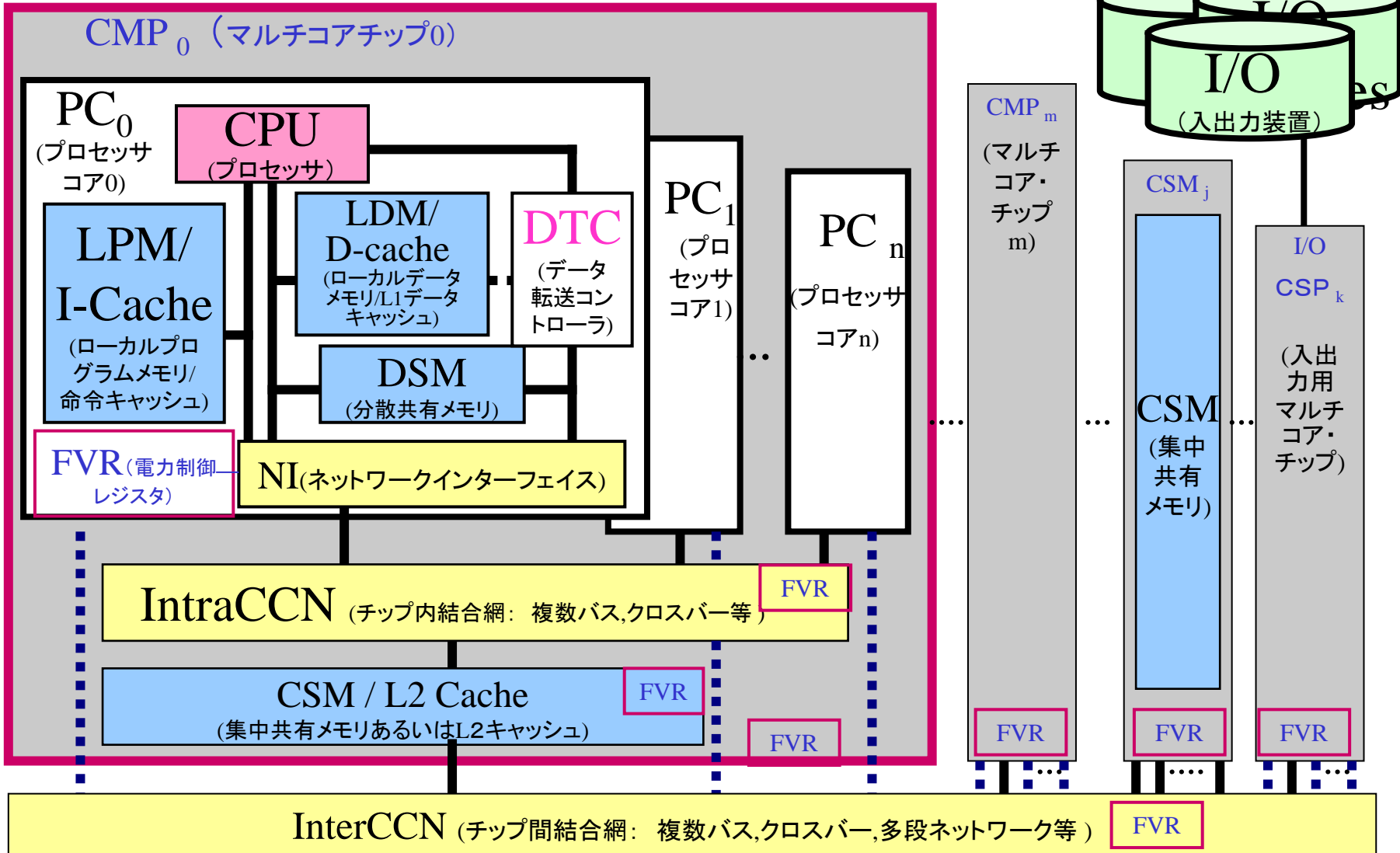
➡ ソフトウェアによるコヒーレンス制御が必要

OSCAR API標準的マルチコアシステム

OSCARメモリアーキテクチャ

(**O**ptically **S**cheduled **A**dvanced **M**ultiprocessor)

- ローカルメモリ
- 分散共有メモリ
- 集中共有メモリ(チップ内・外)
- DTC(高性能DMAC)
- 電力制御用FVR



(ホモジニアス/ヘテロジニアス)マルチコア・メニーコア用プログラム開発

API: Application Programming Interface

逐次C or Fortran
アプリケーションプログラム
(情報家電,自動車,医療,科学技術計算)

ホモジニアス

ヘテロ

ユーザによる
手動並列化

アクセラータ用コンパイラ
あるいはユーザ指示

- アクセラレータあるいはユーザが,アクセラレータで実行可能プログラム部分あるいはライブラリとその実行時間等をヒント指示文として逐次プログラム中に追加

早稲田大学OSCAR
自動並列化コンパイラ

- プログラム全域からマルチグレイン並列性を抽出(粗粒度タスク並列含む)
- データ配置の最適化(分散共有・ローカルメモリ利用の自動最適化)
- DMAを用いたデータ転送
- 周波数/電圧/電源制御低消費電力化

ホモジニアス/ヘテロジニアス マルチコア及びメニーコア用OSCAR API

タスク生成・変数のメモリ割当て・キャッシュ一貫性制御・
データ転送・同期・電力制御を指示する指示文(コメント文)集

並列化API
C or Fプログラム

Proc0
指示文
用プログラム

Proc1
指示文
用プログラム

⋮

アクセラレータA
用プログラム

アクセラレータB
用プログラム

⋮

ホモジニアス用低電力
並列マシンコード生成

API 解釈系	既存逐次コ ンパイラ
------------	---------------

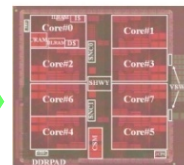
ヘテロジニアス用低電力
並列マシンコード生成

API 解釈系 (早稲田大 学より無料 配布)	既存逐次コ ンパイラ アクセラレー タ用コンパイ ラ/ライブラリ
-------------------------------------	--

⋮

サーバ用コード生成

市販OpenMP
コンパイラ



ホモジニアス
マルチコア/
メニーコアチップ
(共有メモリサーバ)



ヘテロジニアス
マルチ
コアチップ



共有メモリマルチ
プロセッササーバ

逐次コンパイラで各社チップ用並列コード作成可能

OSCAR APIとコンパイラによる並列化・電力制御

プロセッサ高速化における3大技術課題の解消

1. 半導体集積度向上(使用可能トランジスタ数増大)に対する速度向上率の鈍化

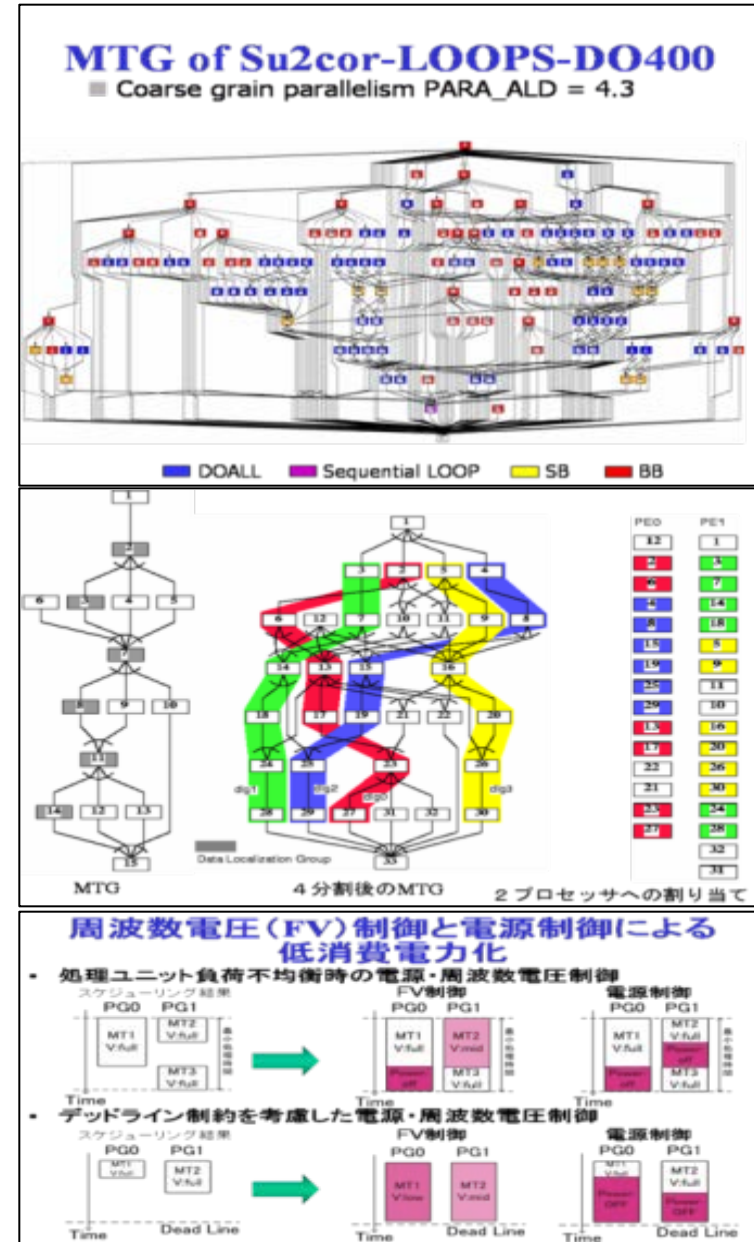
- 粗粒度タスク並列化,ループ並列化,近細粒度並列化によりプログラム全域の並列性を利用するマルチグレイン並列化機能により,従来の命令レベル並列性より大きな並列性を抽出し,複数マルチコアで速度向上

2. メモリウォール問題

- コンパイラによるローカルメモリへのデータ分割配置,DMAコントローラによるタスク実行とオーバーラップしたデータ転送によりメモリアクセス・データ転送オーバーヘッド最小化

3. 消費電力増大による速度向上の鈍化

- コンパイラによる低消費電力制御機能を用いたアプリケーション内でのきめ細かい周波数・電圧制御・電源遮断により消費電力低減

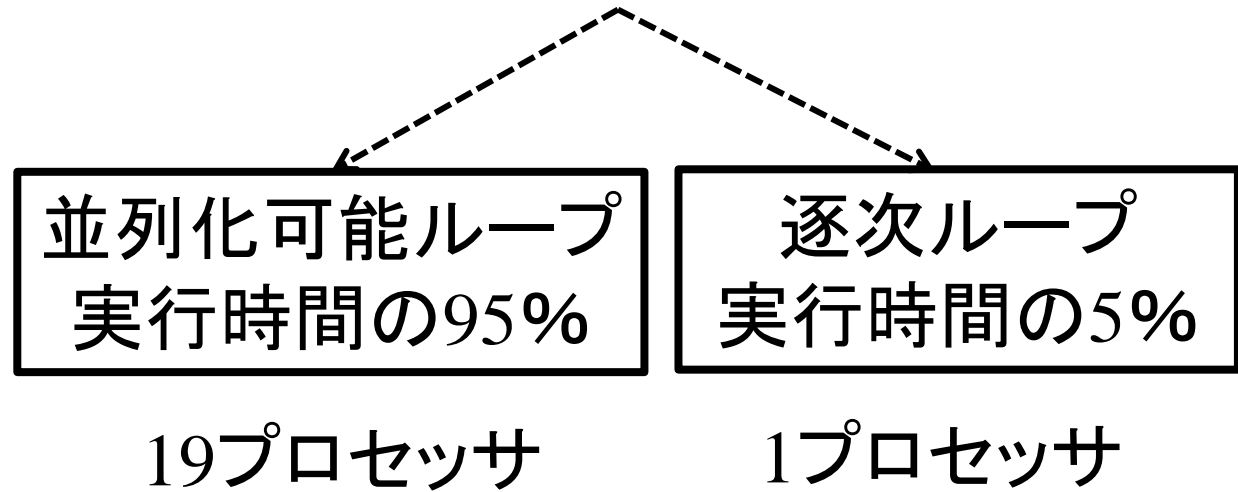
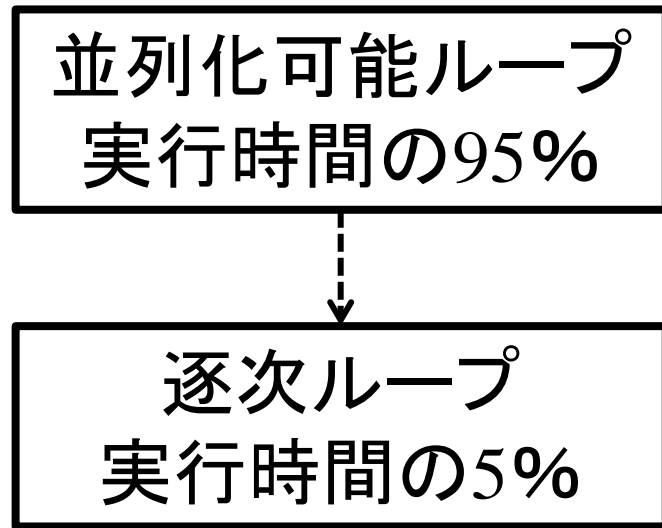


粗粒度タスク並列

ループ並列の限界を越えるために

従来のループ並列化

粗粒度タスク並列化



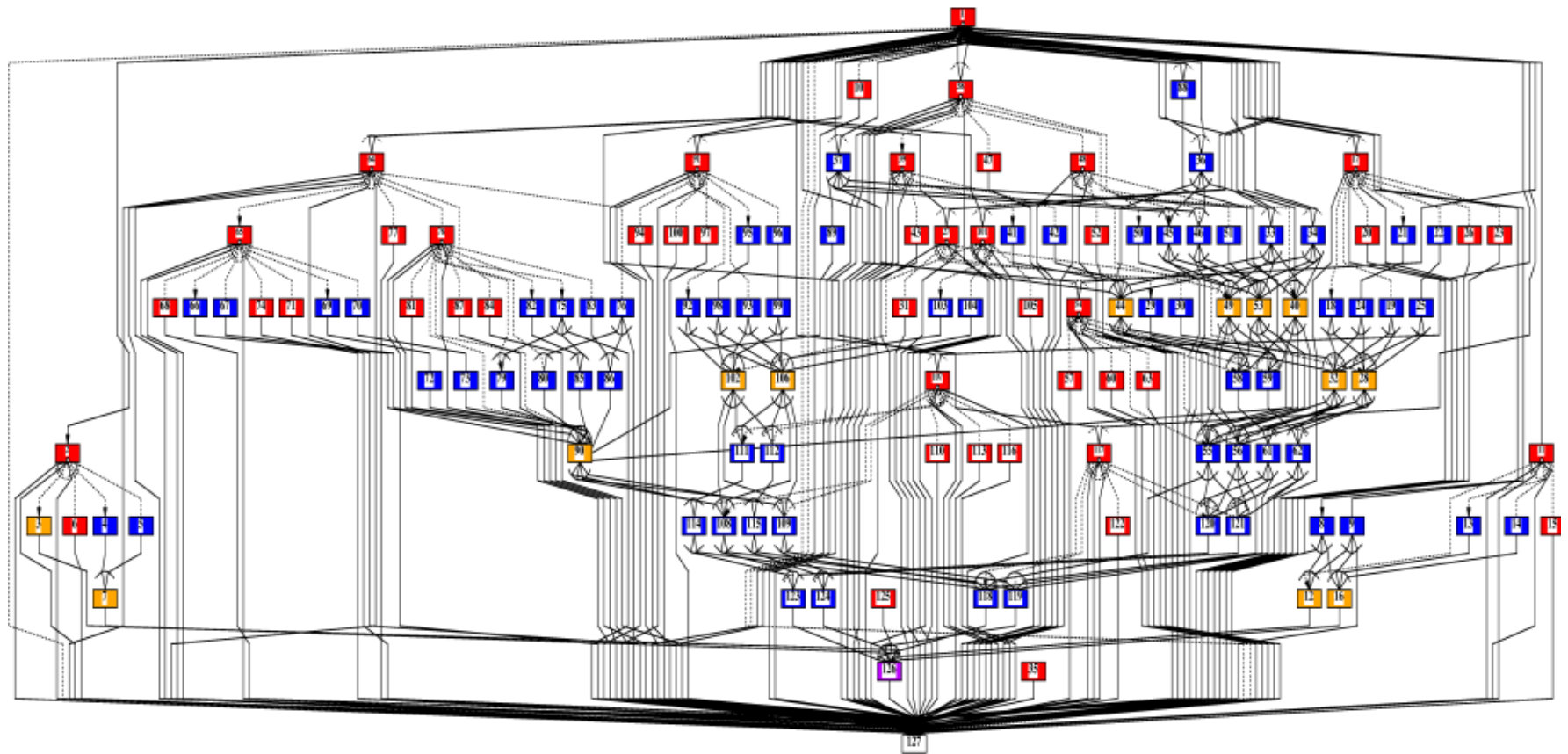
20プロセッサで並列処理

$$\text{速度向上率} = 100 / (95 / 20 + 5) \\ = 100 / (4.75 + 5) = 10.26 \text{ 倍}$$

$$\text{速度向上率} = 100 / \max(95 / 19, 5) \\ = 100 / 5 = 20 \text{ 倍}$$

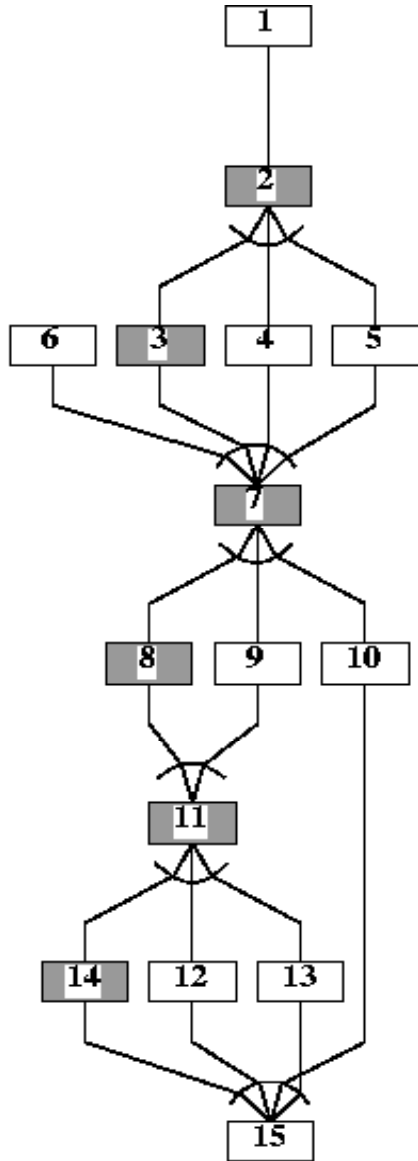
MTG of Su2cor-LOOPS-DO400

- Coarse grain parallelism $\text{PARA_ALD} = 4.3$

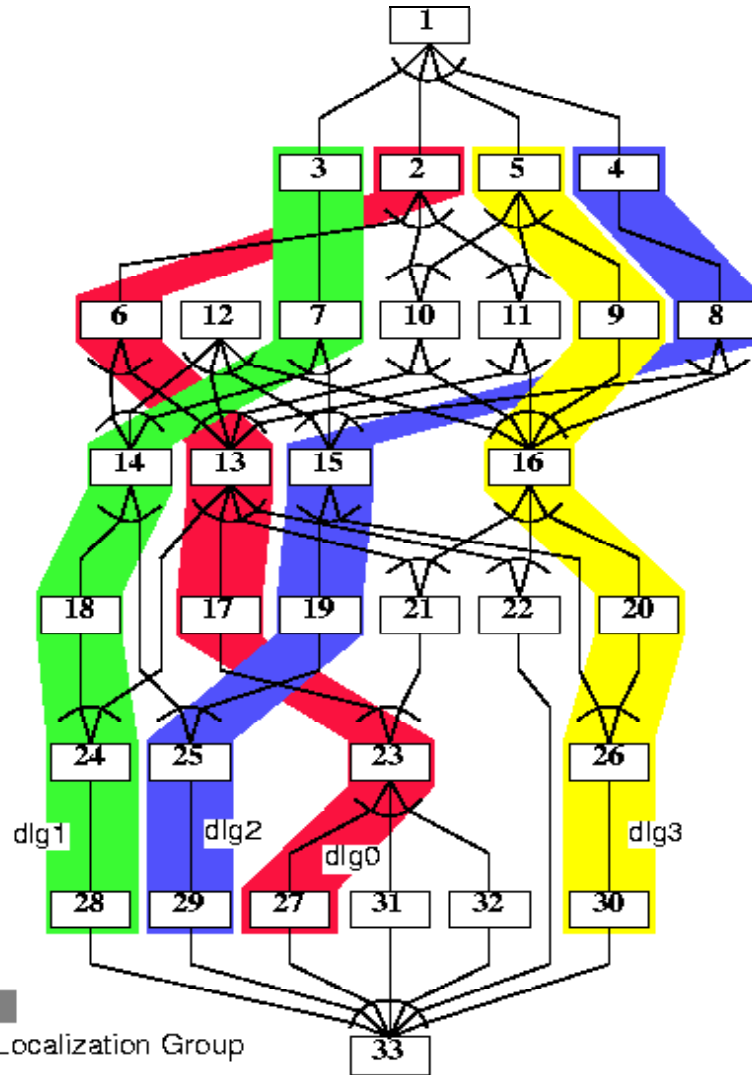


■ DOALL ■ Sequential LOOP ■ SB ■ BB

Data Localization



MTG



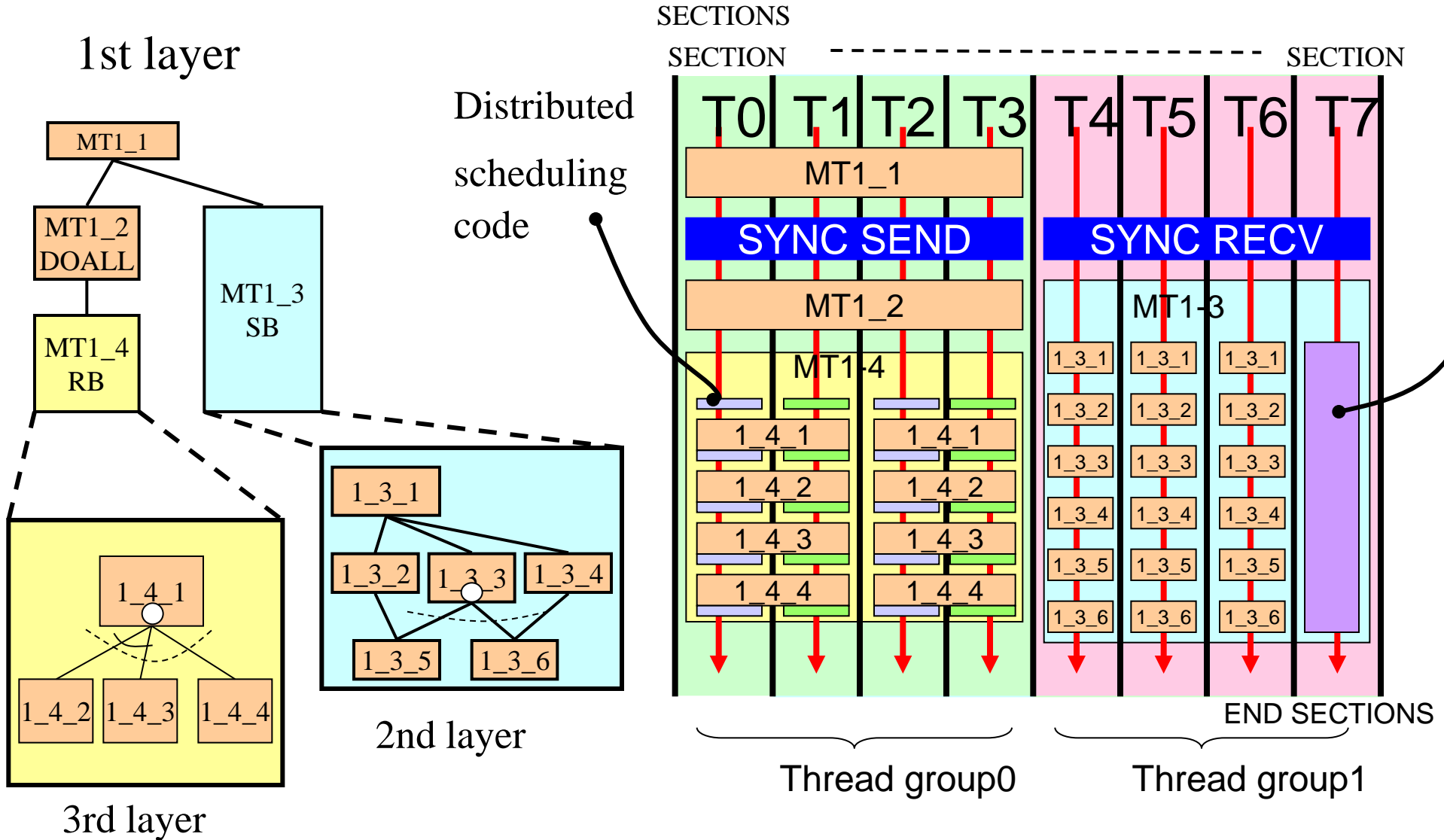
MTG after Division

PE0	PE1
12	1
2	3
6	7
4	14
8	18
15	5
19	9
25	11
29	10
13	16
17	20
22	26
21	30
23	24
27	28
	32
	31

A schedule for two processors

OSCARコンパイラが自動生成するマルチコアAPI (OpenMP互換)利用、階層マルチグレイン 並列プログラムのイメージ

Centralized
scheduling
code



OSCAR APIによる並列プログラム生成イメージ

(スレッド生成、ローカルメモリへの変数配置、プログラム間同期)

```
int flag_l_1;
int myver;
#pragma omp ¥
    threadprivate(myver)
int main() {
#pragma omp parallel sections
{
#pragma omp section
{ main_VC0();}
#pragma omp section
{ main_VC1();}
...
#pragma omp section
#pragma oscar ¥
    distributedshared(flag_l_1)
    { main_VC4();}
...
}
return 0;
}
```

void main_VC0() {
 MTI_1
 for Core0
#pragma omp flush
flag_l_1 = myver;
#pragma omp flush
 MTI_2
 for Core0
...
}

void main_VC1() {
 MTI_1
 for Core1
 MTI_2
 for Core1
...
}

void main_VC4() {
 /* waiting for
 signal from VC0 */
 do {
#pragma omp flush
 } while (flag_l_1
 != myver);
 MTI_3
 for Core4
...
}

メモリ配置指示文 (ハードリアルタイム処理&高速同期)

▶ ローカルデータメモリ(LDM)に変数を配置

- ▶ #pragma omp threadprivate (C)

- ▶ !\$omp threadprivate (Fortran)

- ▶ OpenMPの指示文に対する拡張

▶ 分散共有メモリ(DSM)に変数を配置

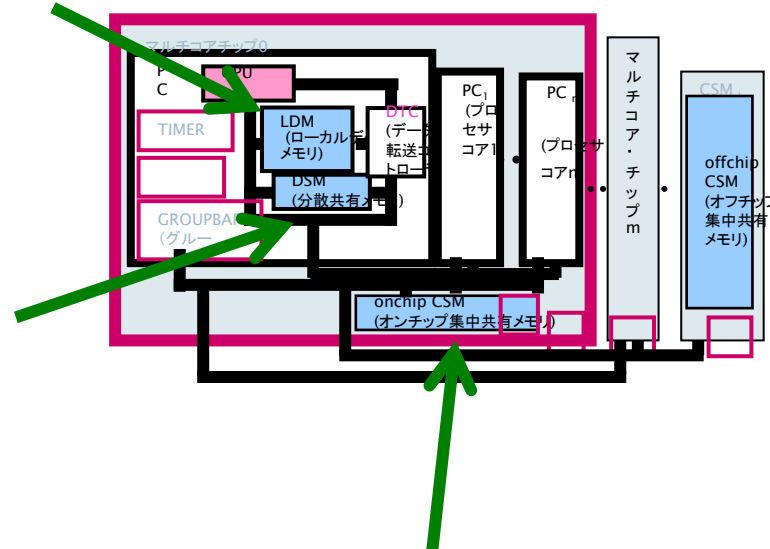
- ▶ #pragma oscar distributedshared (C)

- ▶ !\$oscar distributedshared (Fortran)

▶ オンチップ集中共有メモリ(onchipCSM)に変数を配置

- ▶ #pragma oscar onchipshared (C)

- ▶ !\$oscar onchipshared (Fortran)



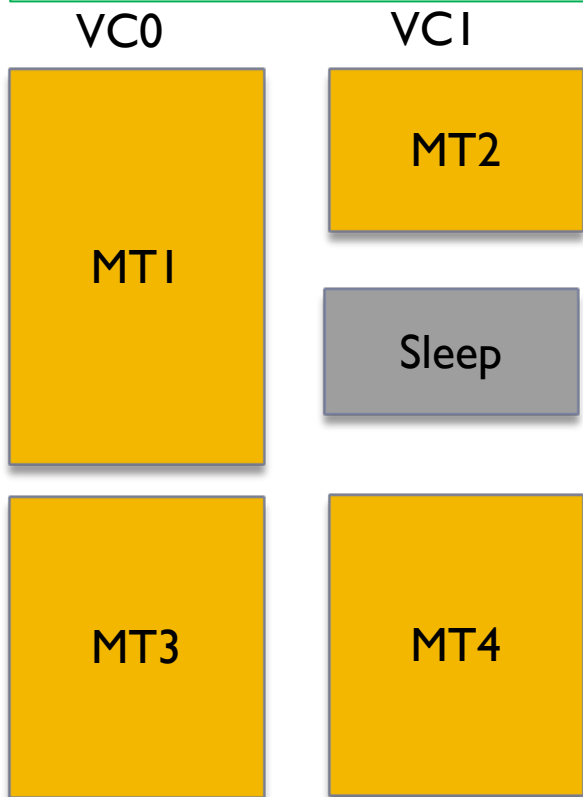
データ転送指示文

(ローカルメモリ・分散共有メモリ間等のオーバーラッピングデータ転送)

- ▶ **データ転送コントローラによる転送リストの指定**
 - ▶ #pragma oscar dmatransfer (C)
 - ▶ !\$oscar dma_transfer (Fortran)
- ▶ **連続領域のデータ転送**
 - ▶ #pragma oscar dma_contiguous_parameter (C)
 - ▶ !\$oscar dma_contiguous_parameter (Fortran)
- ▶ **ストライド転送**
 - ▶ #pragma oscar dma_stride_parameter
 - ▶ !\$oscar dma_stride_parameter
 - ▶ scatter/gather転送も可能
- ▶ **データ転送間の同期**
 - ▶ #pragma oscar dma_flag_check
 - ▶ !\$oscar dma_flag_check

OSCAR APIによる電力制御のイメージ

OSCAR Compilerによる
スケジューリングのイメージ



OSCARコンパイラの生成コードイメージ

```
void  
main_VC0() {
```



```
#pragma oscar fvcontrol ¥  
(1,(OSCAR_CPU(),100))
```

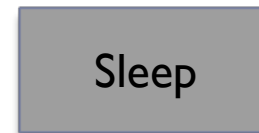


```
}
```

```
void  
main_VC1() {
```



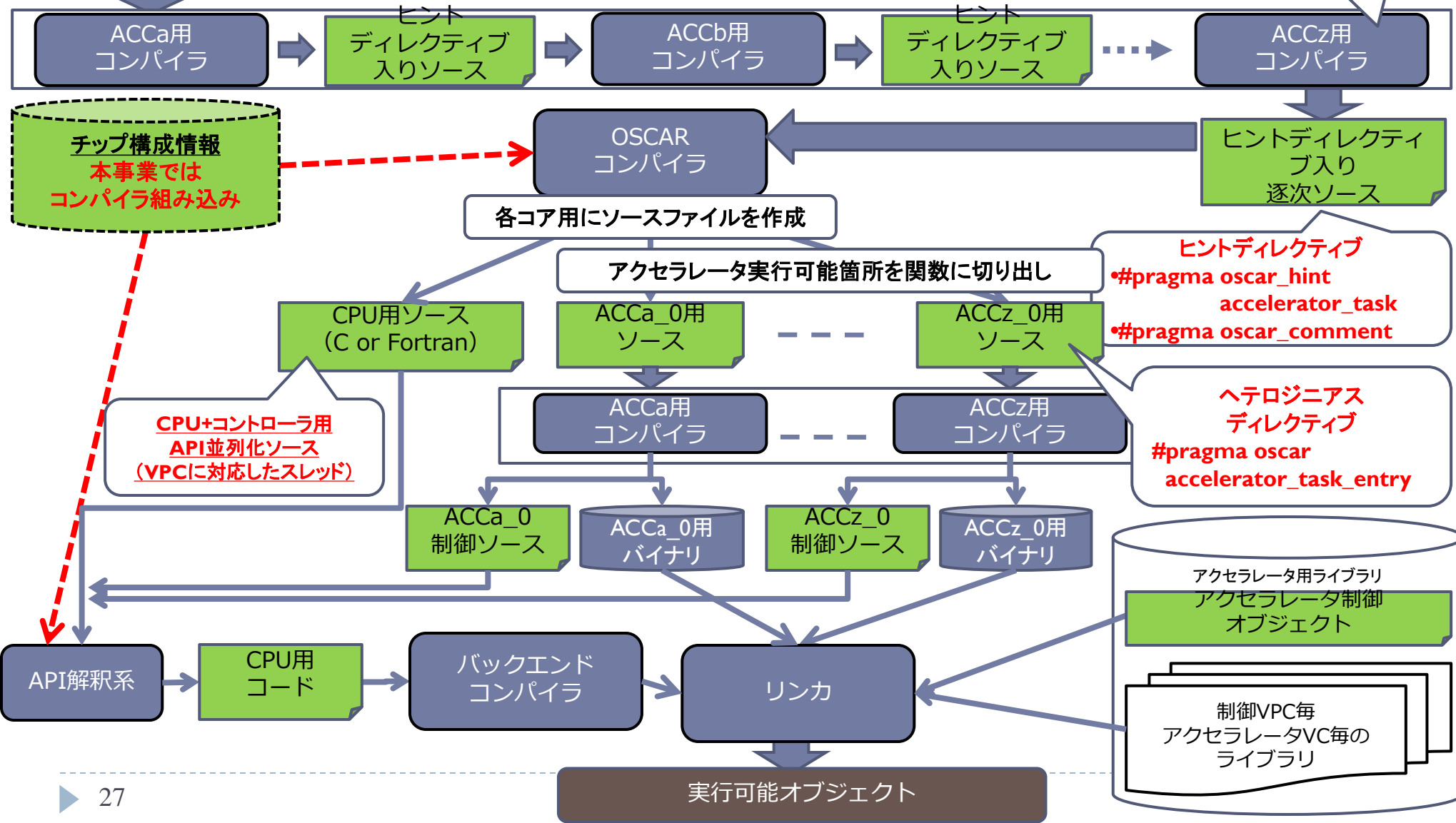
```
#pragma oscar fvcontrol ¥  
((OSCAR_CPU(),0))
```



```
}
```

ヘテロマルチコア向け コンパイルフロー全体像

OSCAR
コンパイラへの
ヒントディレクティブ
を生成する



関数呼び出し部分に対する指定方法と出力コード例

関数呼び出しにaccelerator_task
ヒントディレクティブが付加された入力ソース

- loop1, loop2, loop3はアクセラレータ
コンパイラでコンパイル可能
- func1はライブラリ関数

Sample.c

```
main() {  
  int a, b[10];  
  #pragma oscar_hint accelerator_task  
  (ACCa) cycle(1000) in(a, b[0:9])  
  out(b[0:9])  
  task_func();  
}  
task_func() {  
  for (...) {...} // loop1  
  for (...) {...} // loop2  
  #pragma oscar_comment "XXXXXX"  
  func1 (...) // func1  
  for (...) {...} // loop3  
}
```

関数呼び出し

汎用コア用
ソース

OSCAR
コンパイラ

アクセラレータ用
ソース

Sample.omp.c

```
main() {  
  #pragma omp parallel sections  
  {  
    #pragma omp section  
    {  
      /*VC0,VPC 0*/  
      ...  
      oscartask_CTRL0_task_func();  
      ...  
    }  
  }  
}
```

アクセラレータ用に
関数呼び出しの書き換え

ICPU+IACC用
並列化APIソース

Sample.VC1.c

```
#pragma oscar accelerator_task_entry  
controller(0) oscartask_CTRL0_task_func() {  
  for (...) {...} // loop1  
  for (...) {...} // loop2  
  #pragma oscar_comment "XXXXXX"  
  oscarlib_CTRL0_ACCEL1_func1 (...) //  
  func1  
  for (...) {...} // loop3  
}
```

関数本体の記述が
取り出される

ACCa_0用ソース

OSCAR APIのメニーコアキャッシュ制御への拡張

コヒーレントキャッシュ

- 現在のマルチコアの標準的なキャッシュ構成
- コア(キャッシュ)間データの整合性はハードウェアが維持

組込用には
ハードウェア重く
コスト高

メニーコアでは、ハードウェアが複雑でコア数に限界・コスト高・電力消費大

- ## ソフトウェア制御キャッシュ
- ハードウェアシンプル
 - 低コスト
 - ハードウェア制限なくコア数を増やせる:メニーコア

OSCAR APIにキャッシュ制御のための仕様を拡張

ノンコヒーレントキャッシュ用追加指示文

▶ キャッシュ操作指示文

- ▶ `cache_writback`
 - ▶ キャッシュ上のダーティラインの書き戻し
- ▶ `cache_selfinvalidate`
 - ▶ キャッシュラインの無効化

▶ メモリ配置指示文

- ▶ `noncacheable`
 - ▶ 変数をキャッシュされないメモリ領域に配置する
- ▶ `aligncache`
 - ▶ 変数の先頭をキャッシュラインの境界に配置調整する

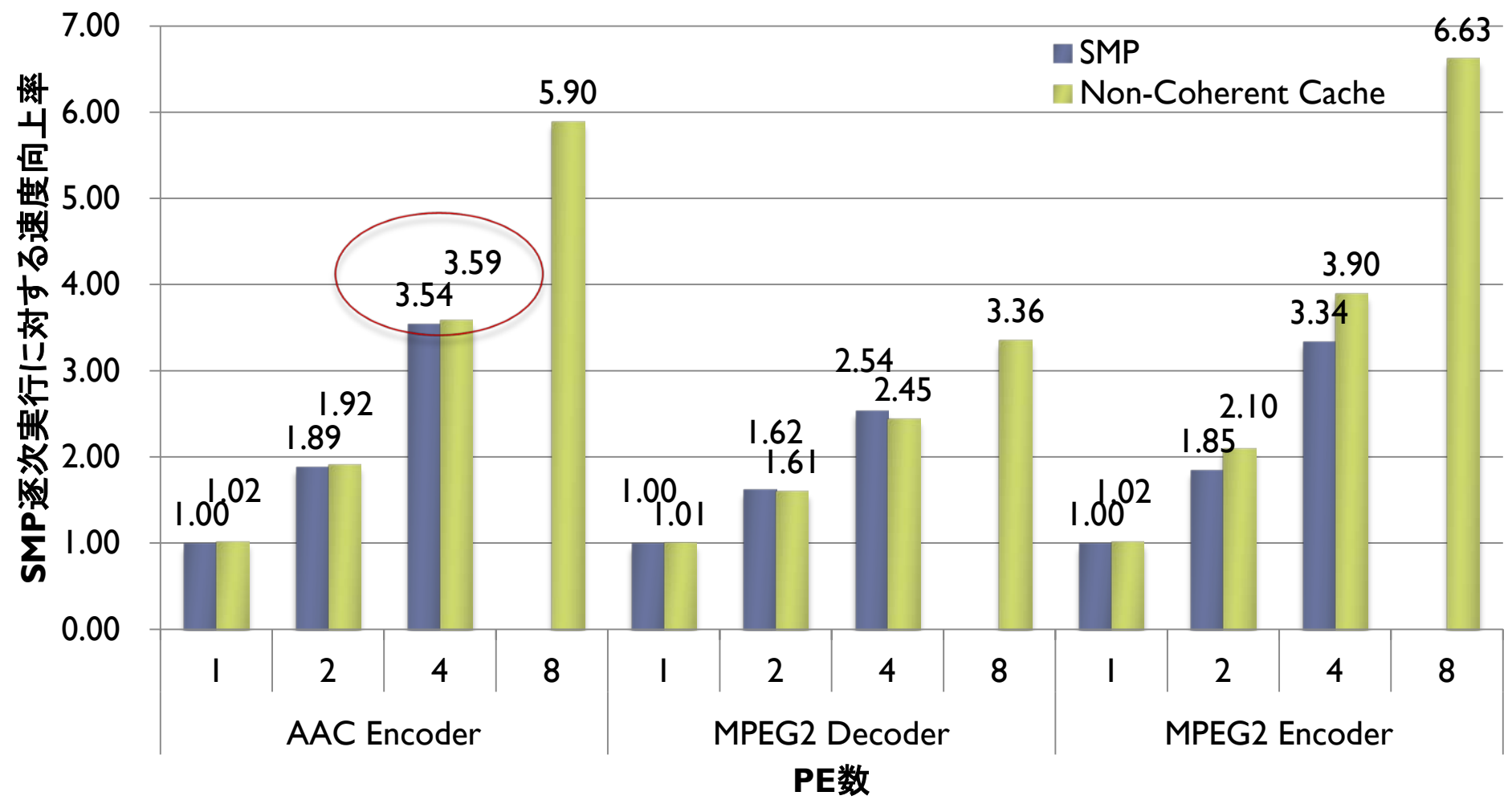
▶ メモリ操作順序保証指示文

- ▶ `complete_memop`
 - ▶ メモリ操作の完了



デモ

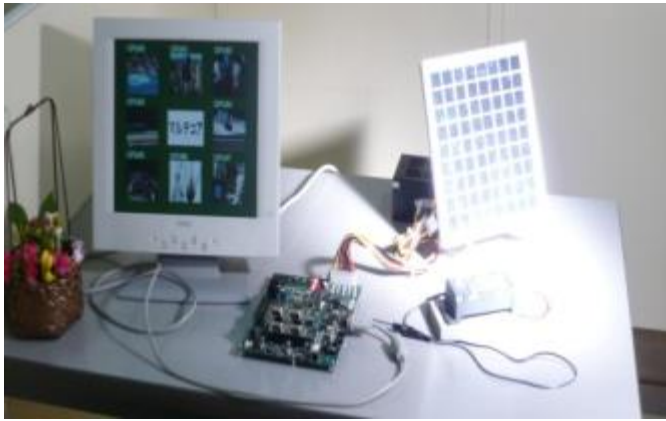
RP2マルチコア上で、OSCAR並列化コンパイラによる自動ソフトウェアコヒーレンス制御により、4コアAAC処理をハードウェアコヒーレンス制御と同等以上の速度向上



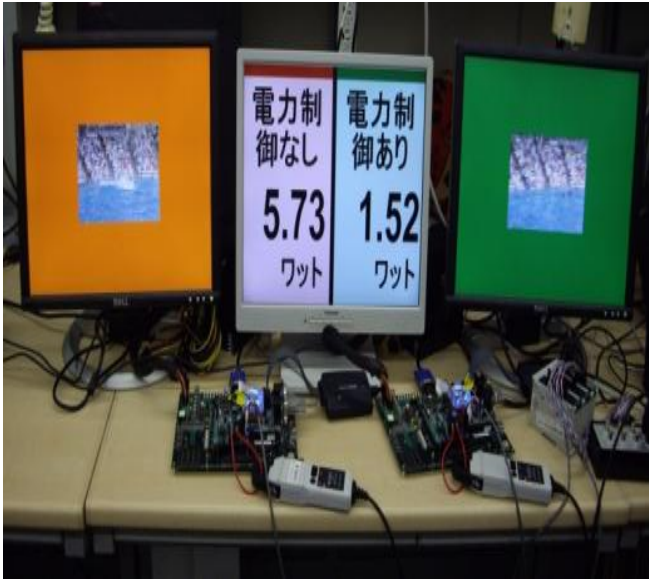
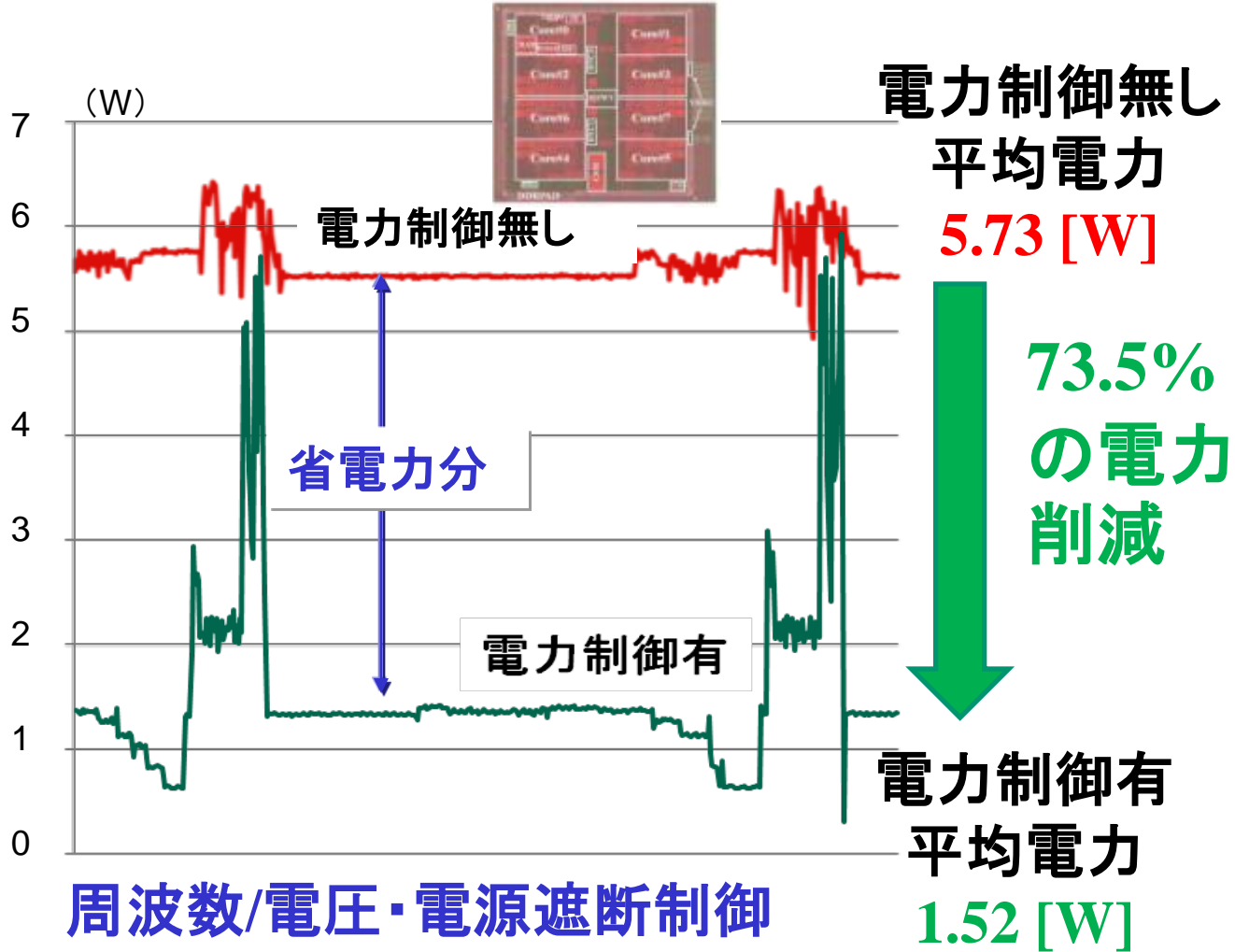
リアルタイムMPEG2デコードを、8コアホモジニアスマルチコアRP2上で、消費電力1/4に削減

デモ

NEDOプロジェクトで開発した低消費電力マルチコア(8コア)上でのマルチメディア処理

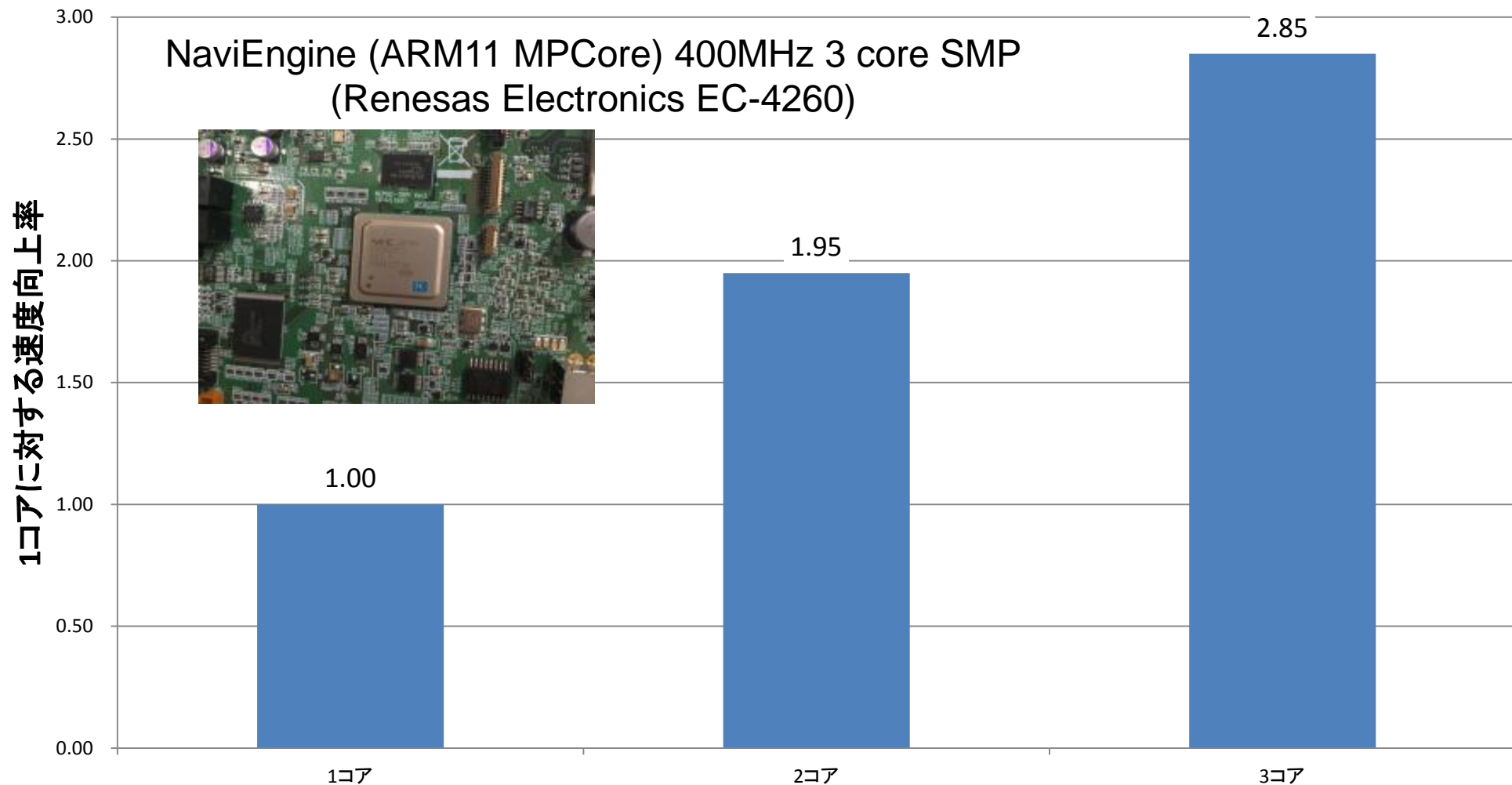


太陽電池で駆動可



デモ

3コア NaviEngine (リアルタイムOS eT-Kernel Multi-Core Edition) 上でAACエンコードを逐次に比べ2.9倍高速化



デモ

RPX上で、リアルタイムオプティカルフロー計算の消費電力を1/3に削減

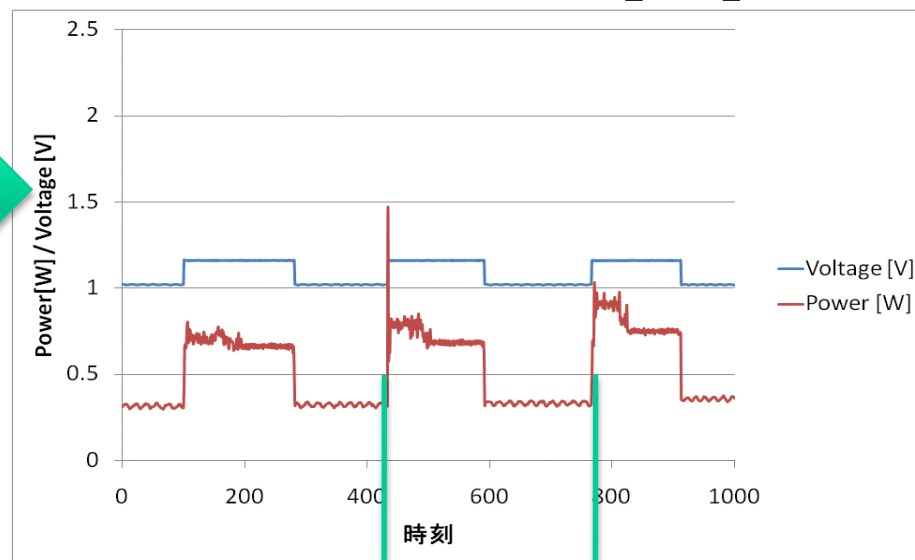
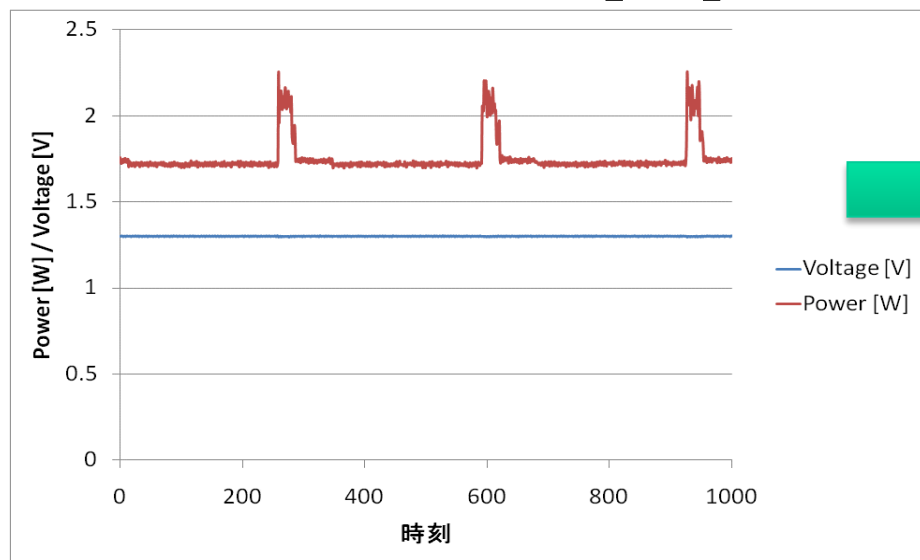
コンパイラ制御なし

コンパイラ制御適用

およそ70[%]の電力削減

平均1.76[W]

平均0.54[W]



1周期 : 33[ms]
→30[fps]

組込マルチコアRPX利用低消費電力Webサーバ

8コア動作時1W



早稲田大学 基幹理工学部 情報理工学科

笠原研究室

現在のサーバの消費電力

1.01 W

| Japanese | English |

研究室用



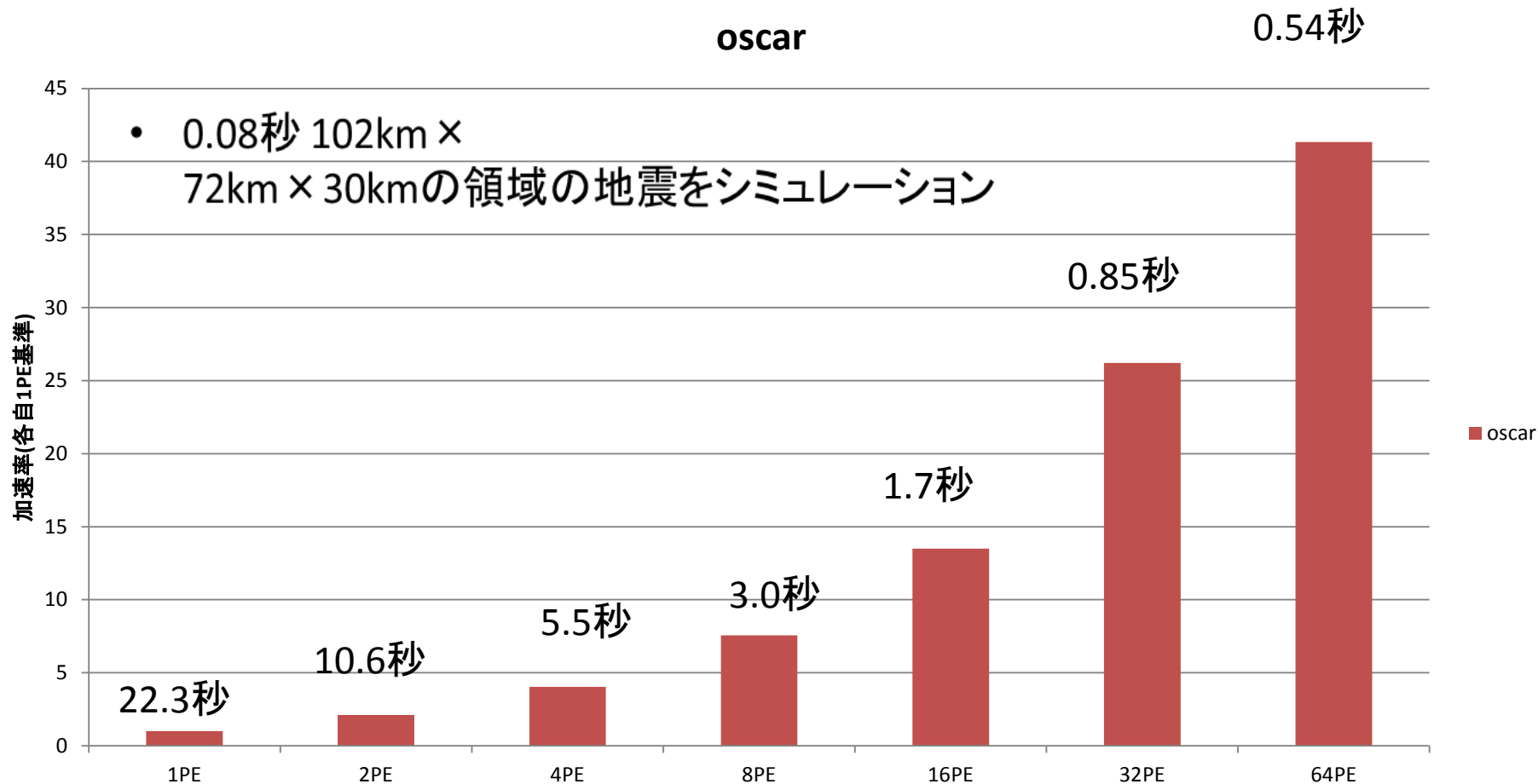
Contents

- ▶ 教授紹介
- ▶ 准教授紹介
- ▶ 発表論文
- ▶ メンバー
- ▶ 授業情報
- ▶ 所在地
 - ▶ 訪問者一覧
 - ▶ 関連サイト

News

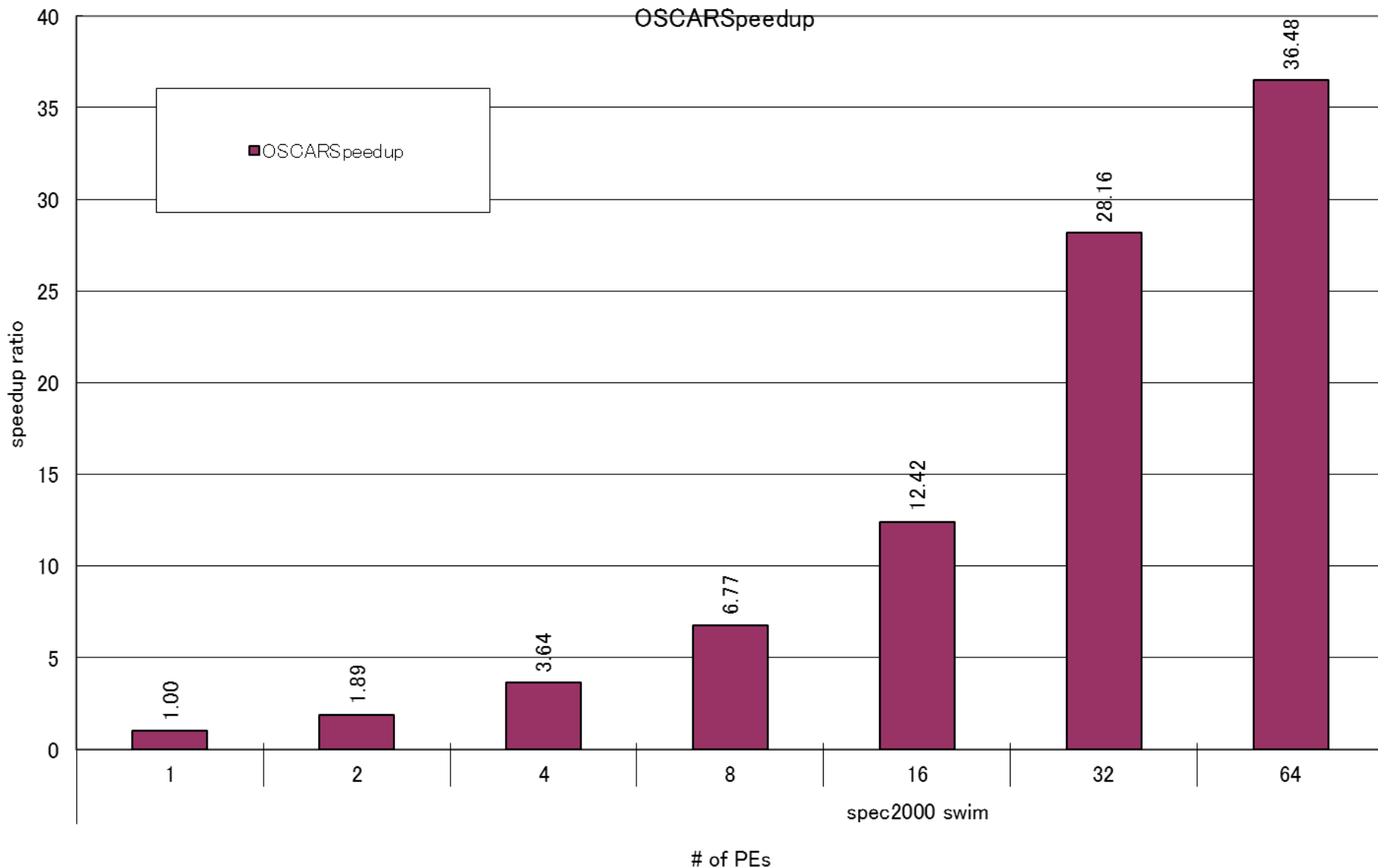
- 2011.10.07 笠原博徳教授がIEEE Computer Society 理事(2012-2014)に再選されました。ご支援ありがとうございました。
- 2011.9.6 25周年記念 LCPC2012 (グリーンコンピューティングシステム研究開発センターにて2012年9月11日から13日開催) の情報を掲載しました
- 2011.5.24 グリーン・コンピューティング・システム研究機構「低炭素社会を支えるグリーンITで日本の競争優位を確立する」が読売オンライン_研究力「WASEDA研究特区」とWASEDA研究特区-プロジェクト研究最前線 に掲載されました
- 2011.5.13 早稲田大学グリーンコンピューティングシステム研究開発センターでの産官学連携研究開始記念シンポジウム“未来を拓くグリーンコンピューティング”を開催いたします。
- 2011.5.10 早稲田大学グリーンコンピューティングシステム研究開発センターのパンフレット(施設紹介&教員・研究紹介)を掲載致します。
- 2011.3.3 早稲田大学グリーンコンピューティングシステム研究開発センターに日立SR16000 モデルVM1 (Power7ベース128コアSMP)システムが先行導入されました

日立SR16000 (Power7ベース64コアLinux SMP) 上でGMS地震波伝搬シミュレーションを 逐次に比べ41倍高速化



デモ

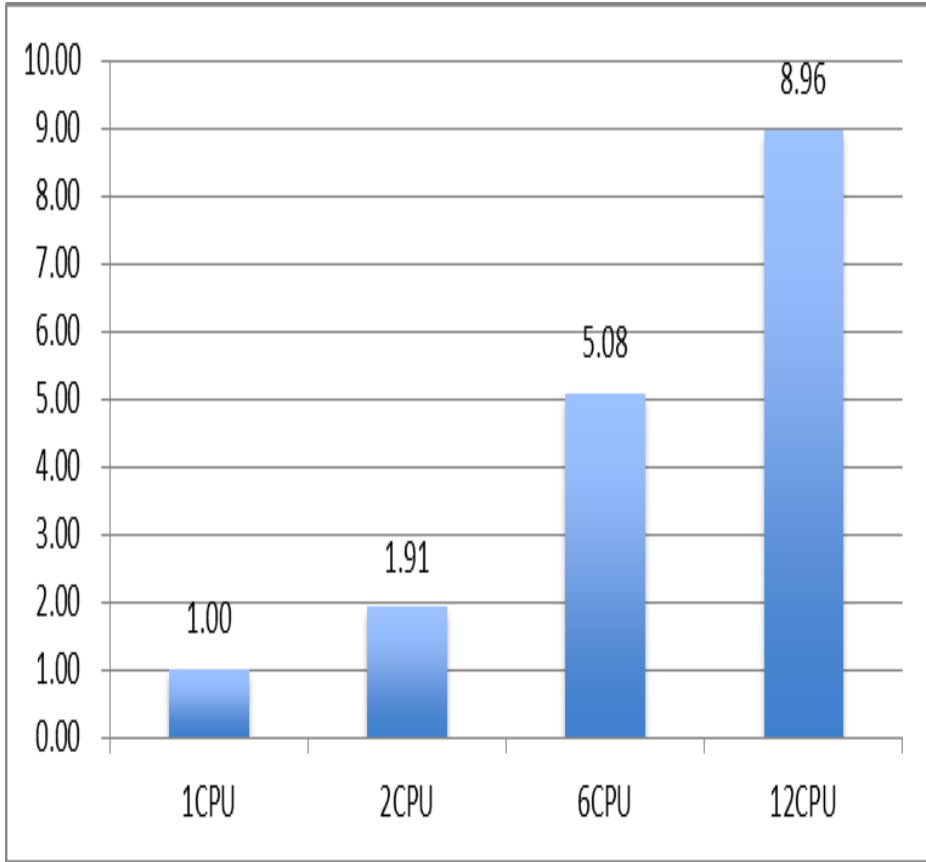
富士通 SPARC Enterprise M9000 (SPARC64 VIIベース64コア Solaris SMP)上で、SPEC2000 171.swimベンチマークプログラムを逐次に比べ36倍高速化



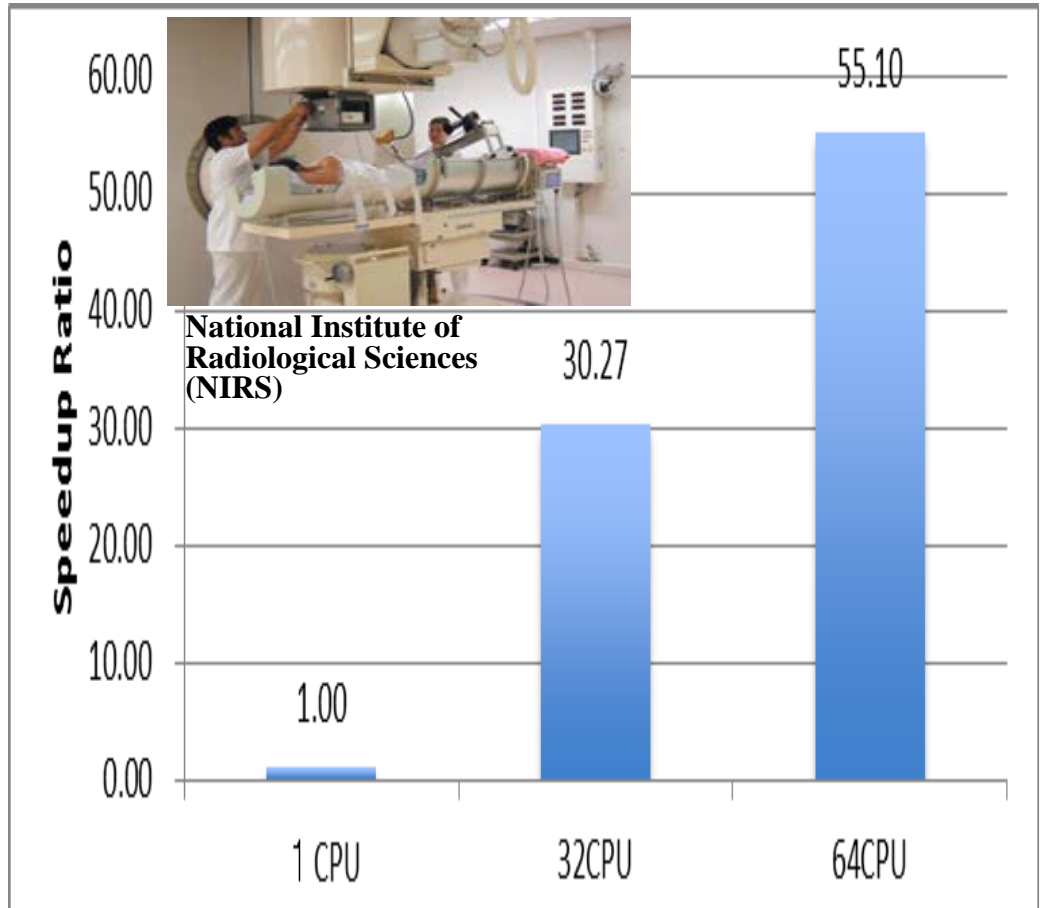
デモ

重粒子線ガン治療装置

線量計算自動並列化性能 (従来手動で16コアで2.5倍)



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)

まとめ

- 組み込み機器からスパコンまでに使われる各種のマルチコア(ホモジニアスマルチコア及びヘテロジニアスマルチコア)、メニーコア及び共有メモリ型マルチプロセッササーバ上で、並列プログラムを高速かつ低消費電力で動作させることができるソフトウェア標準(OSCAR API ver. 2.0)を開発
- 本APIを用いることにより、各社のマルチコアプロセッサ上で、情報家電(スマートフォン等)、自動車、医療、科学技術計算用の並列プログラムを短期間・低コストで開発可能。APIの詳細仕様は、2012年4月25日より
<http://www.kasahara.cs.waseda.ac.jp/> で公開、無料でダウンロード可
- OSCAR API ver. 2.0とOSCAR並列化コンパイラを用い、種々の実システム上で、下記のようなアプリケーションの自動並列化高速実行及び自動電力削減を実現

デモ

- ヘテロジニアスマルチコアRPX上でのオプティカルフロー計算において、8つのSH4Aプロセッサと4つのアクセラレータFEGA(動的再構成可能プロセッサ)利用時に逐次に比べ33倍高速化。
- RPX上で、リアルタイムオプティカルフロー計算の消費電力を1/3に削減。
- RP2マルチコア上で、OSCAR並列化コンパイラによる自動ソフトウェアコヒーレンス制御により、4コアAAC処理をハードウェアコヒーレンス制御と同等以上の速度向上。
- リアルタイムMPEG2デコードを、8コアホモジニアスマルチコアRP2上で、消費電力1/4に削減。
- 3コアNaviEngine(リアルタイムOS eT-Kernel)上でAACエンコードを逐次に比べ2.9倍高速化。
- SPARC Enterprise M9000 (SPARC64 VIIベース64コアSolaris SMP)上で、SPEC2000 171.swimベンチマークプログラムを逐次に比べ36倍高速化
- SR16000 (Power7ベース64コアLinux SMP) 上でGMS地震波伝搬シミュレーションを逐次に比べ41倍高速化。
- SR16000上で三菱電機重粒子線ガン治療装置用線量計算を、逐次に比較し55倍高速化。