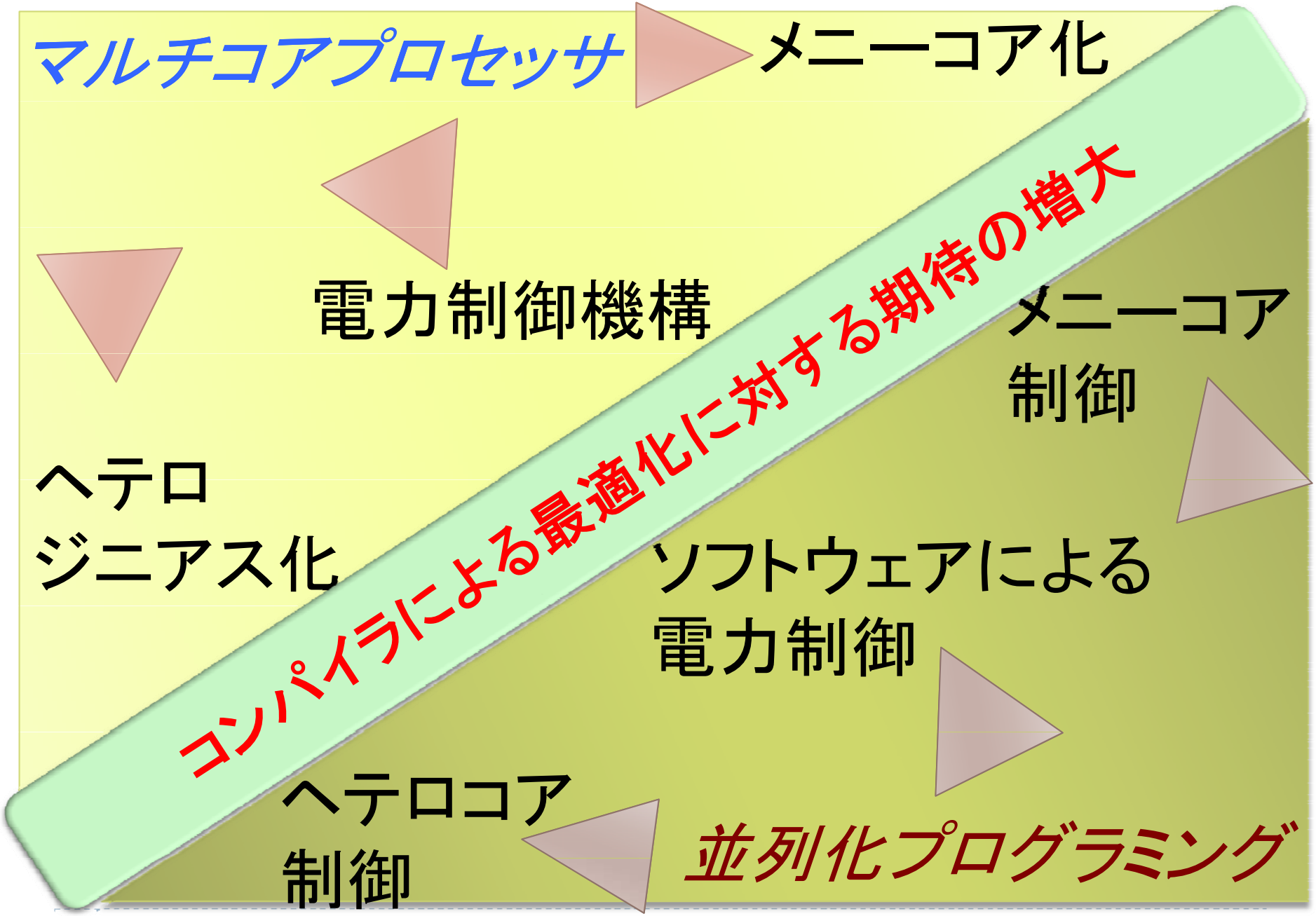
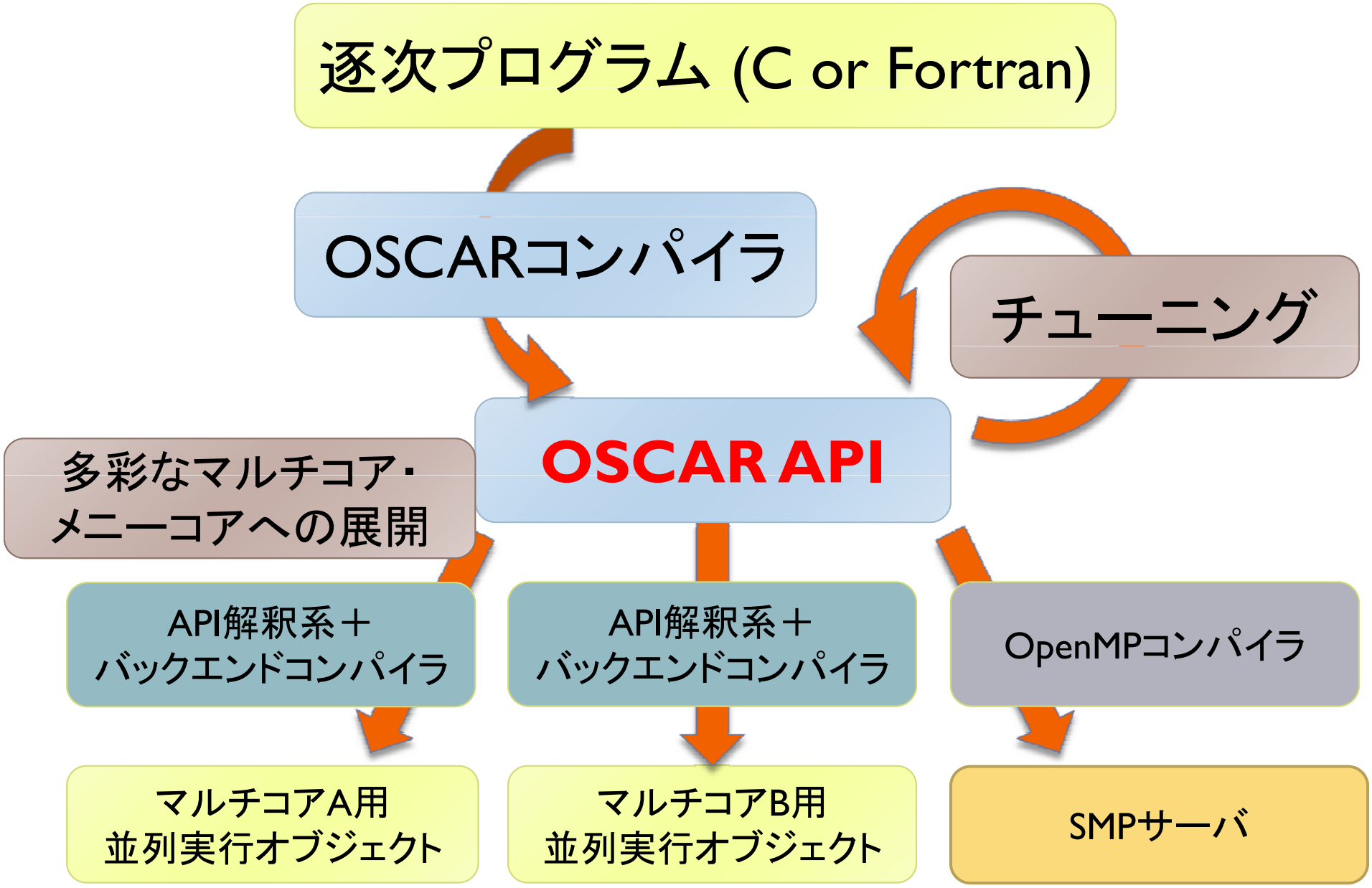
低消費電力メニーコア・コンパイラとOSCAR APIのメニーコアキャッシュ及びヘテロ拡張

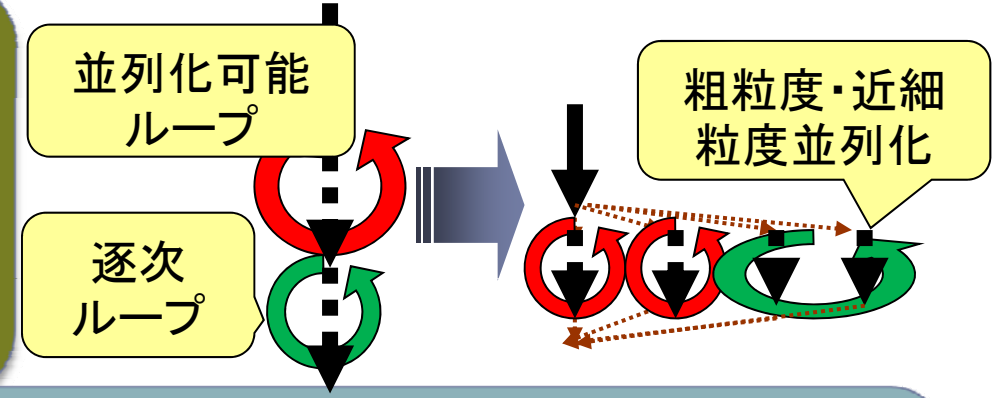
早稲田大学 理工学術院
木村啓二

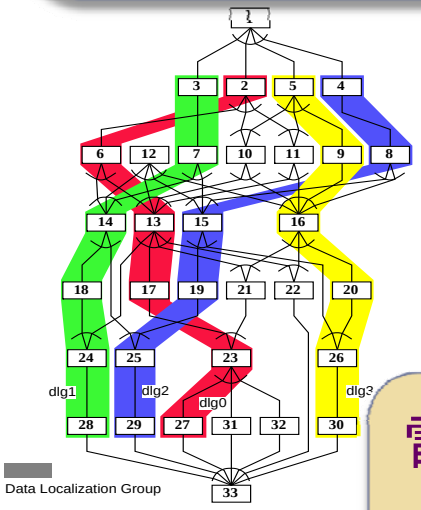

マルチコアプロセッサ
メニーコア化
電力制御機構
メニーコア
制御
ヘテロ
ジニアス化
コンパイラによる最適化に対する期待の増大
ソフトウェアによる
電力制御
ヘテロコア
制御
並列化プログラミング


逐次プログラム (C or Fortran)
OSCARコンパイラ
チューニング
OSCAR API
多彩なマルチコア・
メニーコアへの展開
API解釈系＋
バックエンドコンパイラ
API解釈系＋
バックエンドコンパイラ
OpenMPコンパイラ
マルチコアA用
並列実行オブジェクト
マルチコアB用
並列実行オブジェクト
SMPサーバ

OSCAR自動並列化コンパイラ

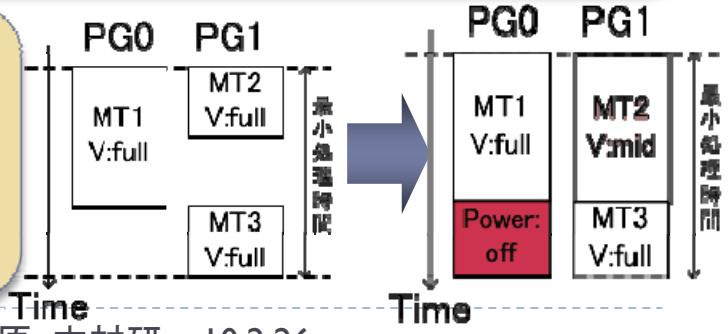
マルチグレイン並列処理
- 階層的かつ広域的な並列化
  - 粗粒度タスク並列性
  - ループ並列性
  - 近細粒度並列性





データローカリティ最適化
- キャッシュあるいはローカルメモリサイズを考慮したタスクの分割
- データ再利用可能なスケジューリング

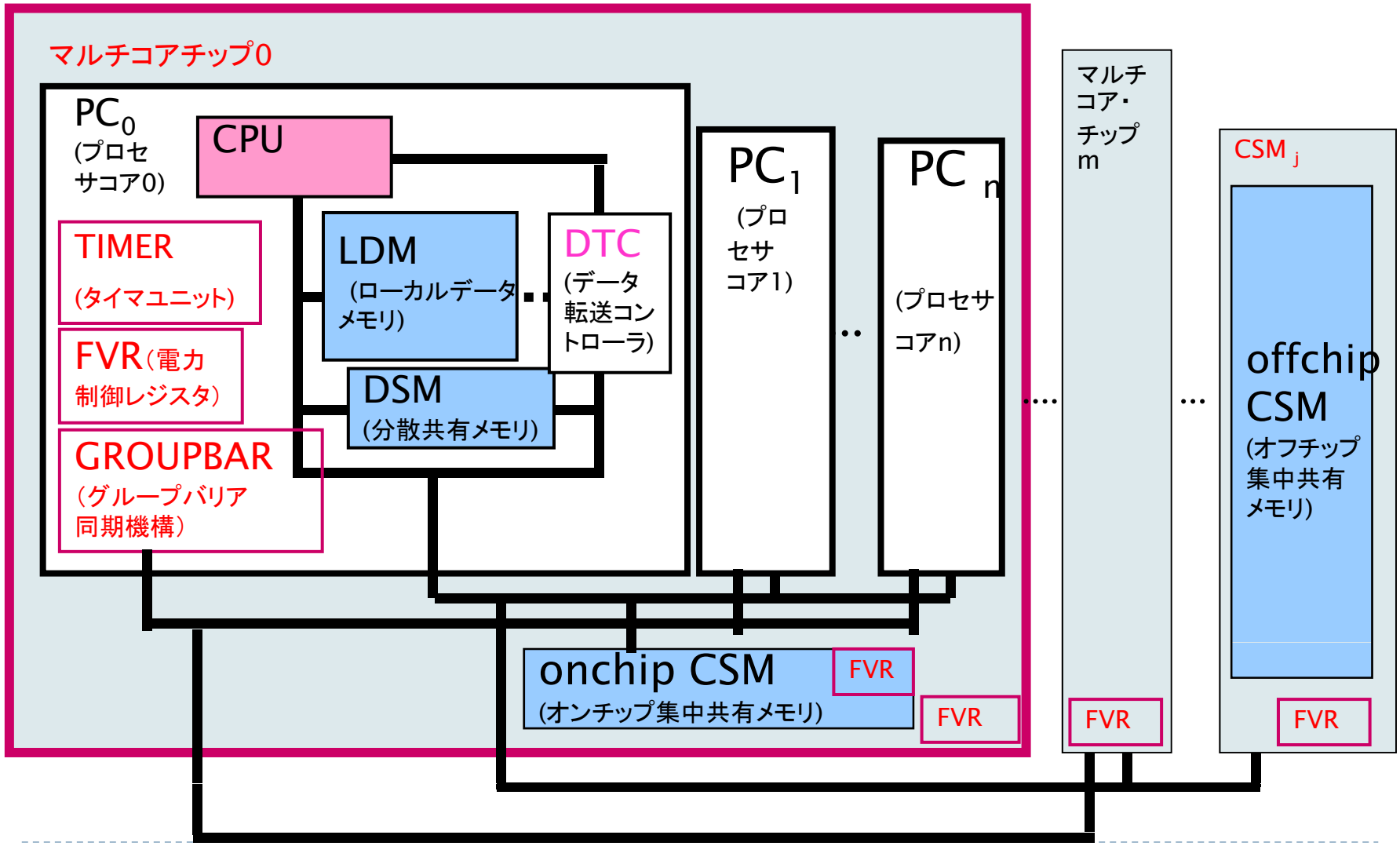
電力最適化
- CPU空き時間の周波数・電源を細かく制御



OSCAR API v1.0
（2008年11月に発表）

- 主にリアルタイム情報家電機器が対象
  - 様々なメモリアーキテクチャ
    - SMP, ローカルメモリ, 分散共有メモリ, ...
- 産官学連携
  - 日立、ルネサス、富士通、東芝、パナソニック、NEC
  - 経済産業省・NEDOのサポート
- OpenMPのサブセットがベース
  - 共有メモリモデルの、サーバ機等で非常によく使われる並列化API
  - OpenMPコンパイラでコンパイル可能
  - CとFortran双方の指示文を提供
- 6つのカテゴリ
  - 並列実行
  - メモリ配置
  - データ転送
  - 電力制御
  - タイマ
  - 同期

OSCARメモリアーキテクチャ



OSCAR API v1.0の指示文

- 並列実行API
  - parallel sections (*)
  - flush (*)
  - critical (*)
  - execution
- メモリ配置API
  - threadprivate (*)
  - distributedshared
  - onchipshared
- 同期API
  - groupbarrier
- データ転送API
  - dma_transfer
  - dma_contiguous_parameter
  - dma_stride_parameter
  - dma_flag_check
  - dma_flag_send
- 電力制御API
  - fvcontrol
  - get_fvstatus
- タイマーAPI
  - get_current_time

(* OpenMPからの指示文)

OSCAR APIによるOSCARコンパイラのコード生成イメージ

```
int flag1_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(flag1_1)
    { main_VC4(); }
…
  }
  return 0;
}
```

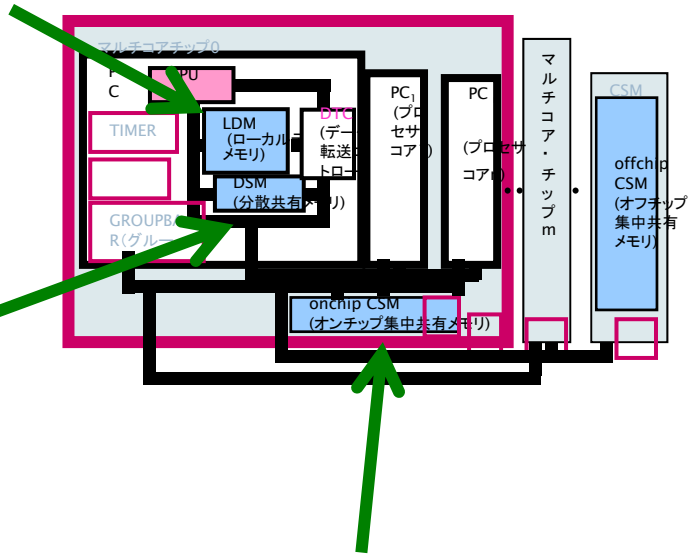
```
void
main_VC0() {
  MT1_1
  for Core0
#pragma omp flush
  flag1_1 = myver;
#pragma omp flush
  MT1_2
  for Core0
  …
}
```

```
void
main_VC1() {
  MT1_1
  for Core1
  MT1_2
  for Core1
  …
}
```

```
void
main_VC4() {
  /* waiting for
      signal from VC0 */
  do {
#pragma omp flush
  } while (flag1_1
          != myver);
  MT1_3
  for Core4
  …
}
```

メモリ配置

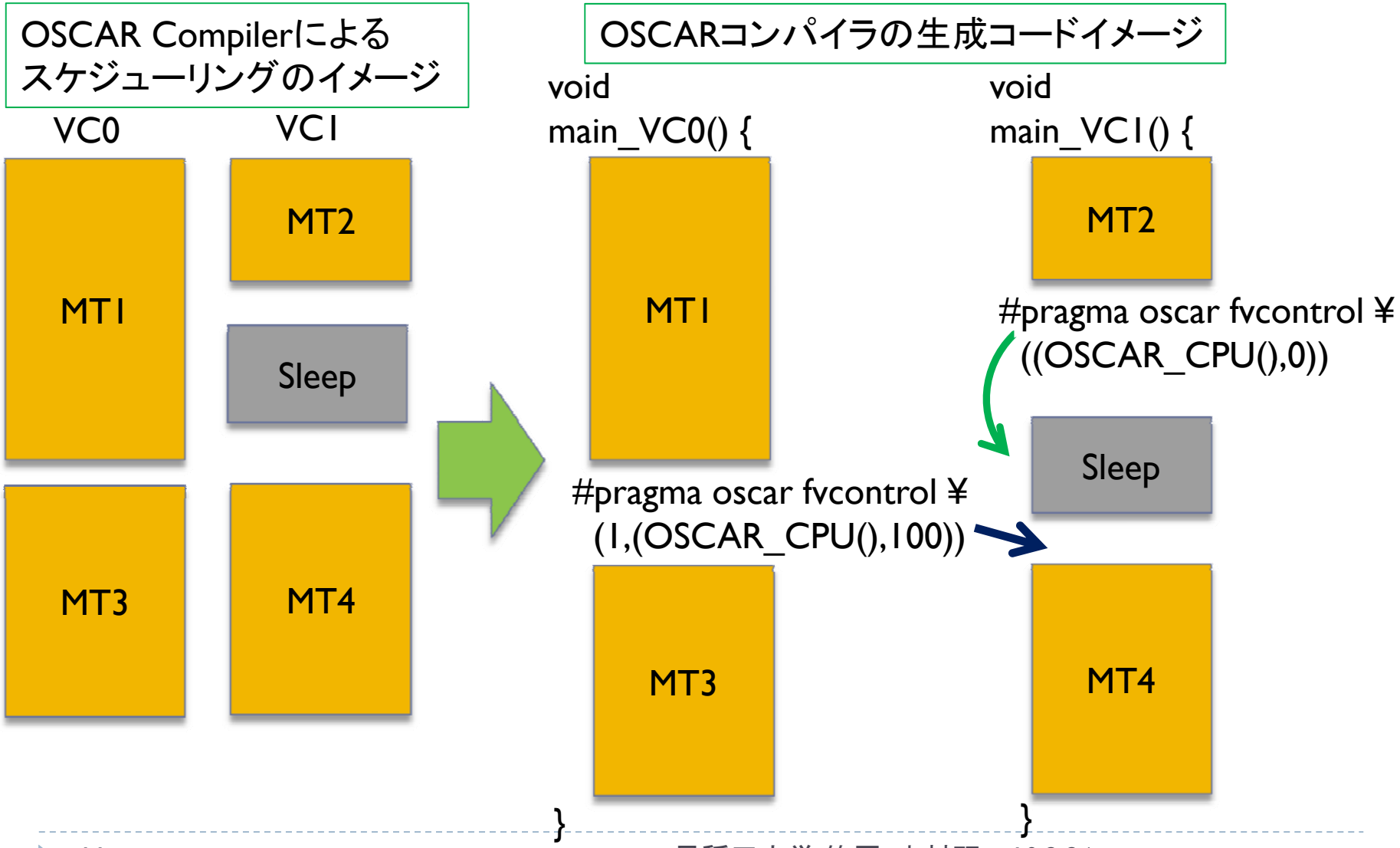
- ローカルデータメモリ(LDM)に変数を配置
  - #pragma omp threadprivate (C)
  - !$omp threadprivate (Fortran)
  - OpenMPの指示文に対する拡張
- 分散共有メモリ(DSM)に変数を配置
  - #pragma oscar distributedshared (C)
  - !$oscar distributedshared (Fortran)
- オンチップ集中共有メモリ(onchipCSM)に変数を配置
  - #pragma oscar onchipshared (C)
  - !$oscar onchipshared (Fortran)



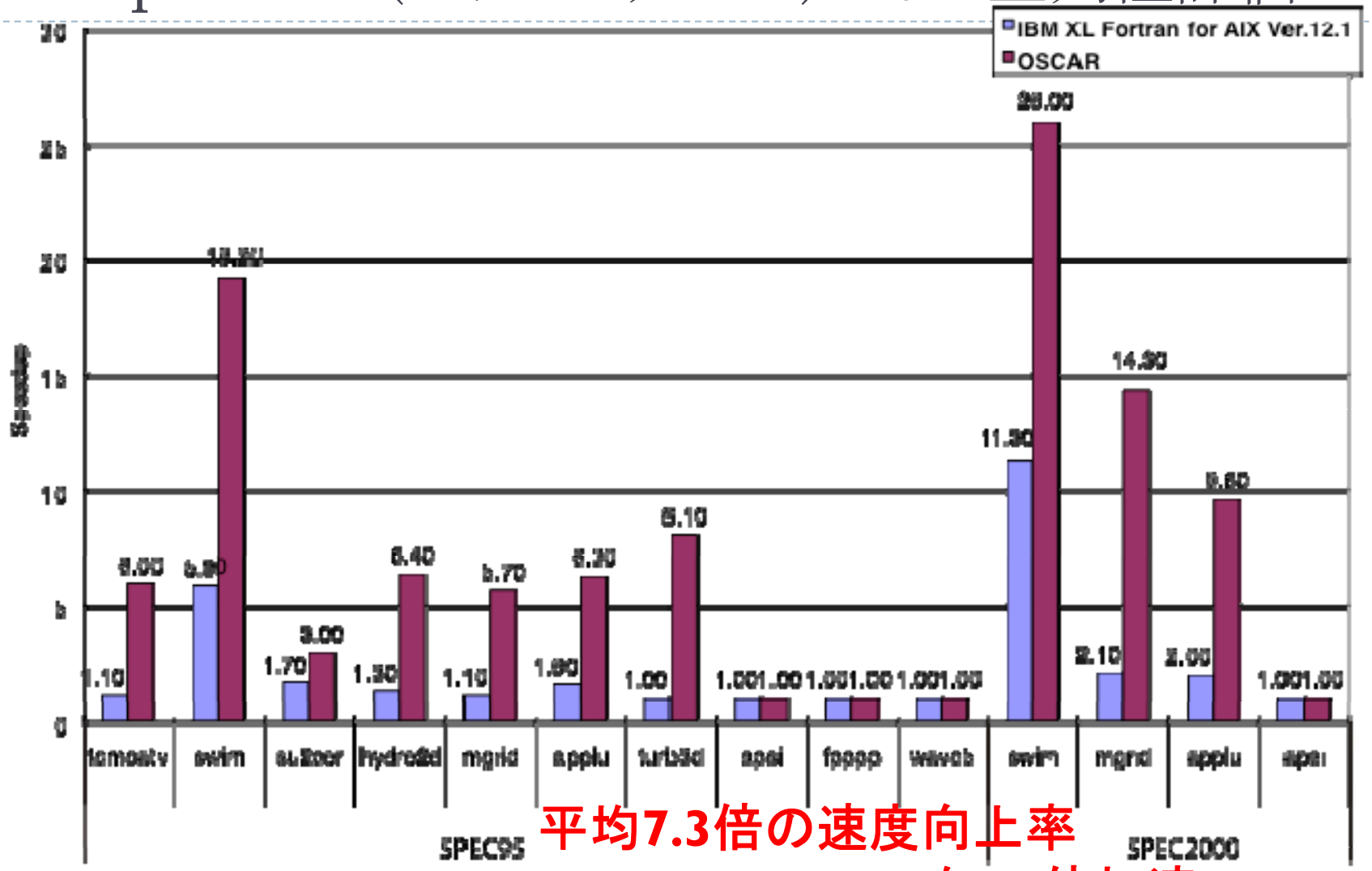
データ転送

- 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



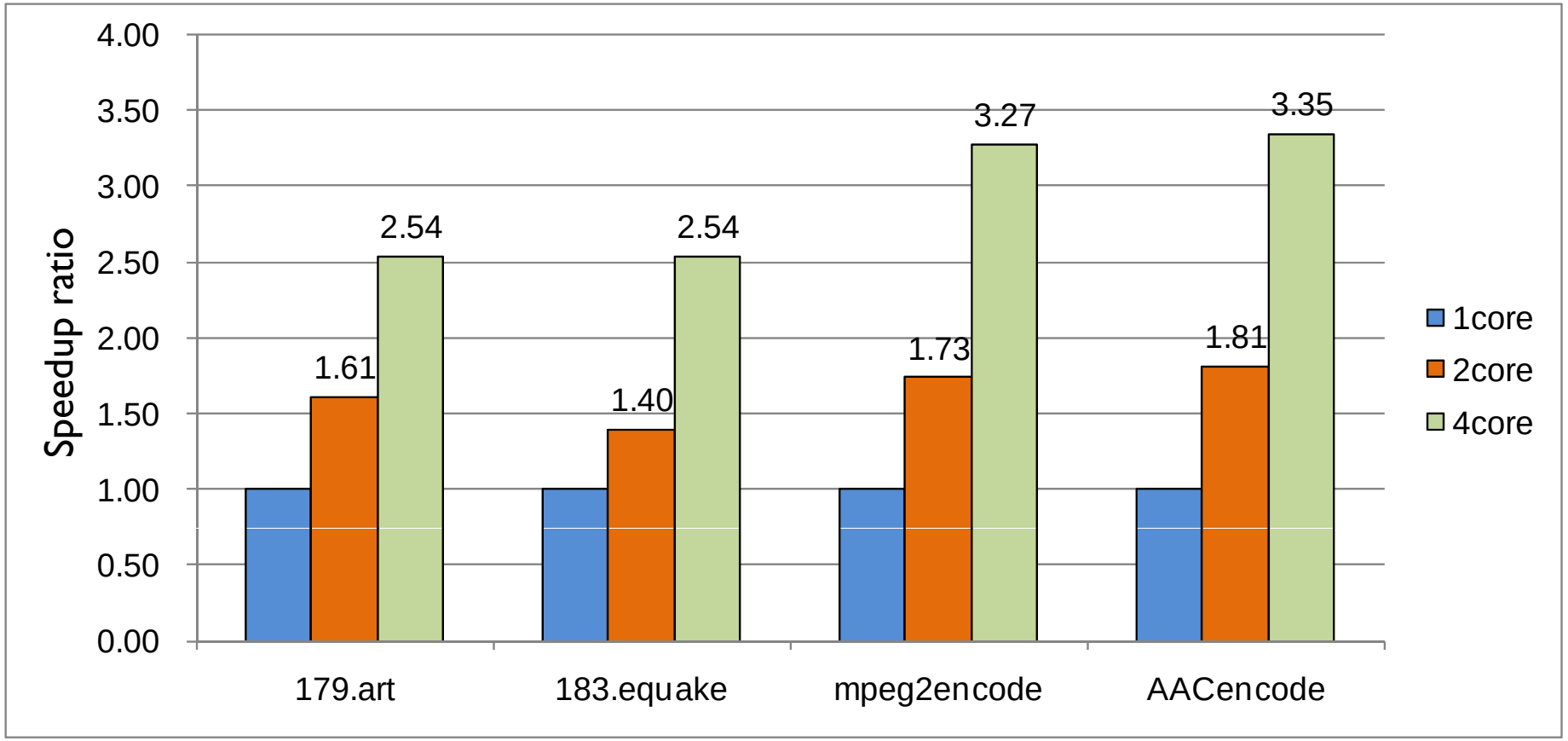
IBM p6 595（32コアサーバ）での並列性評価



平均7.3倍の速度向上率

IBM XL Fortranを3.3倍加速

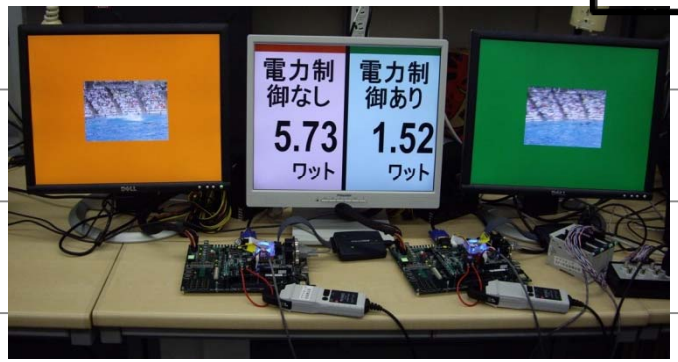
情報家電マルチコアRP2（8コア中4コア）の並列性能評価



平均2.9倍の速度向上率

OSCARコンパイラとOSCAR APIによるMPEG2デコーダの電力制御

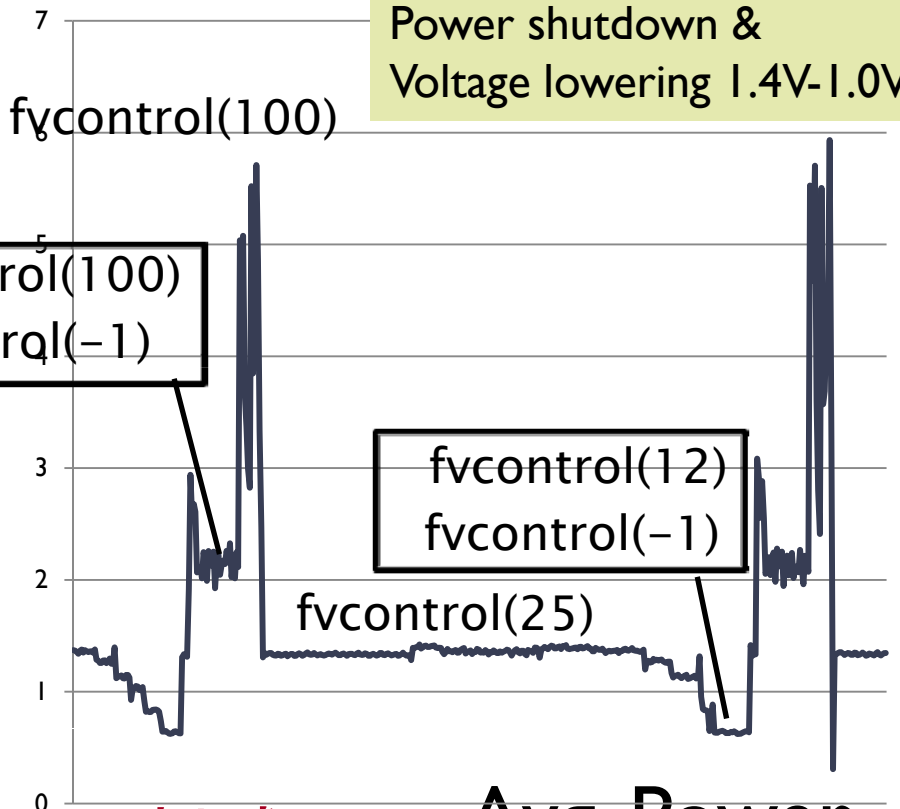
電力制御無し
(Voltage: 1.4V)



Avg. Power
5.41 [W]

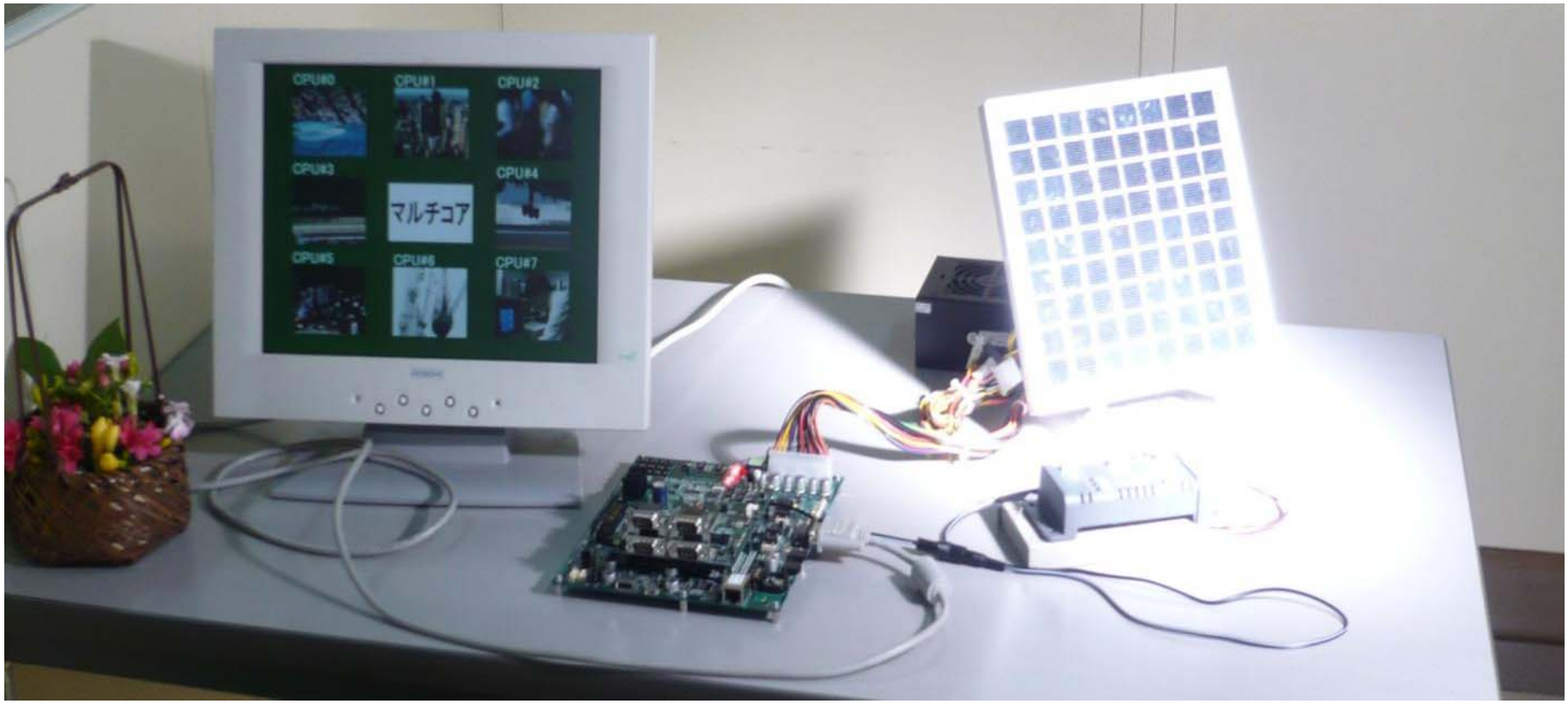
消費電力76.0%削減

電力制御有り
(Frequency,
Resume Standby:
Power shutdown &
Voltage lowering 1.4V-1.0V)



Avg. Power
1.30 [W]

太陽電池駆動RP2



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

組込用にはやや重いハードウェア

スケーラビリティに欠けメニーコアには不向き

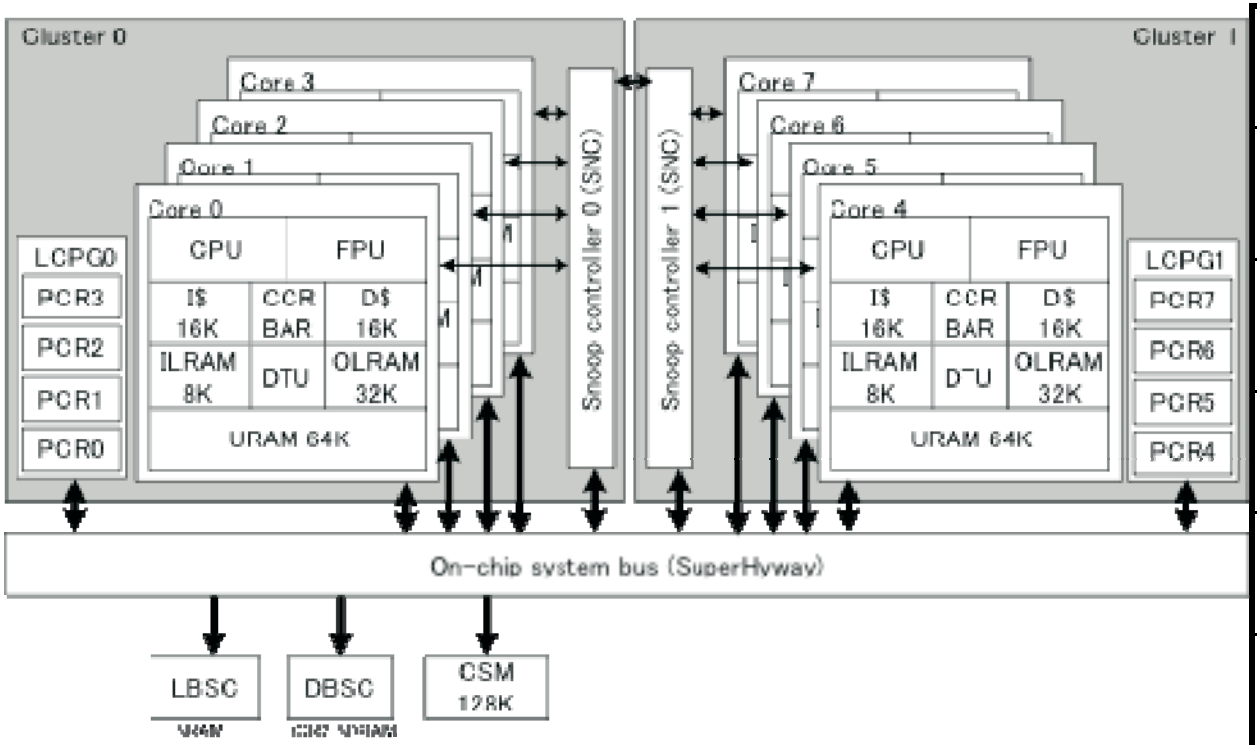
ノンコヒーレントキャッシュアーキテクチャ
- データ整合性はソフトウェアにより維持
- メニーコア化には必須

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_writeback
    - キャッシュ上のダーティーラインの書き戻し
  - 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

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



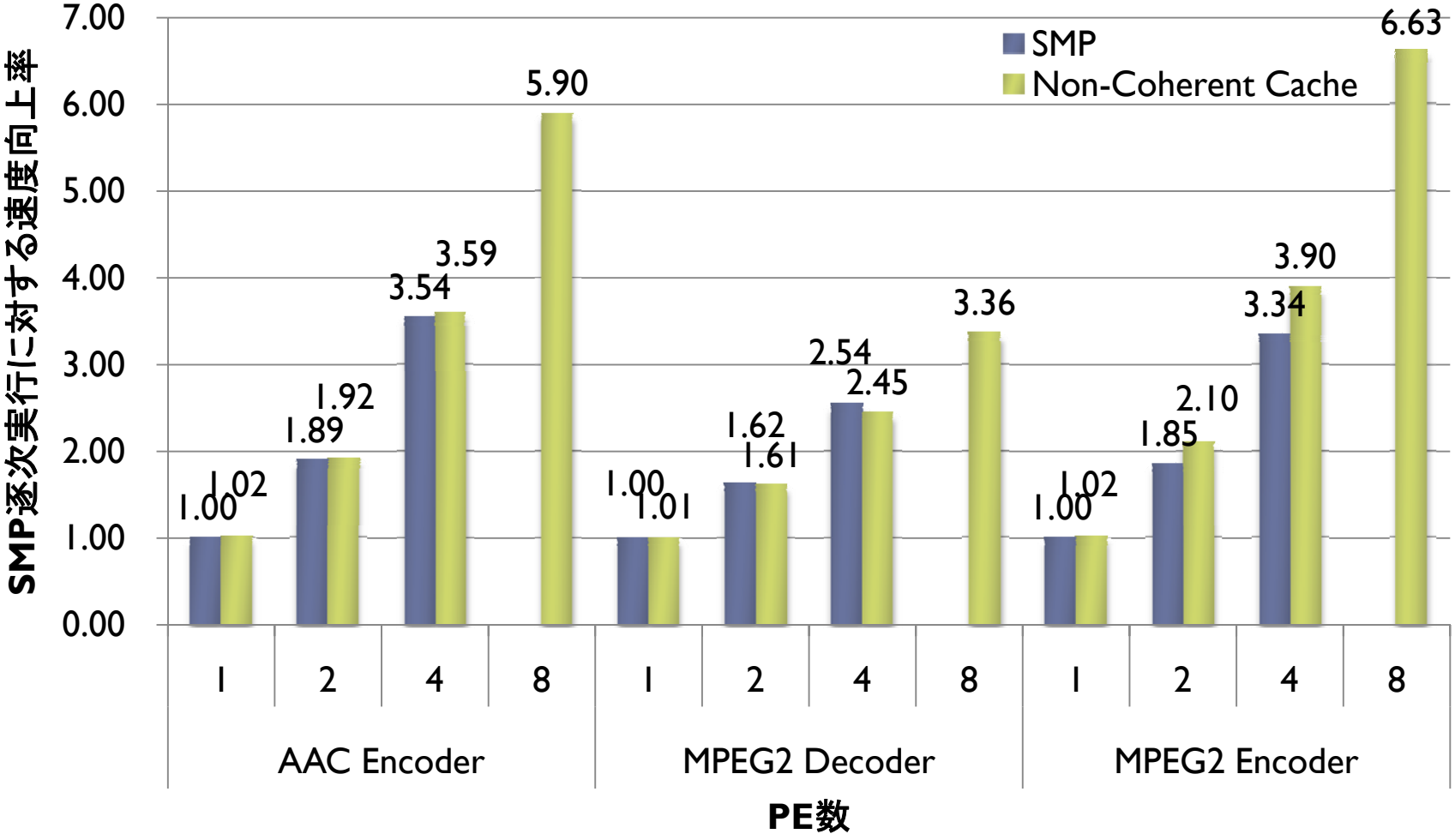
Process Technology	90nm, 8-layer, triple-Vth, CMOS
Chip Size	104.8mm^2 (10.61mm x 9.88mm)
CPU Core Size	6.6mm^2 (3.36mm x 1.96mm)
Supply Voltage	1.0V–1.4V (internal), 1.8/3.3V (I/O)
Clock frequency	600MHz, 300MHz, 150MHz, 75MHz
Power Domains	17 (8 CPUs, 8 URAMs, common)

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シを維持しない

⇒ ソフトウェアによるコヒーレンシ制御が必要

RP2上でのコンパイラによる コヒーレント制御の初期評価



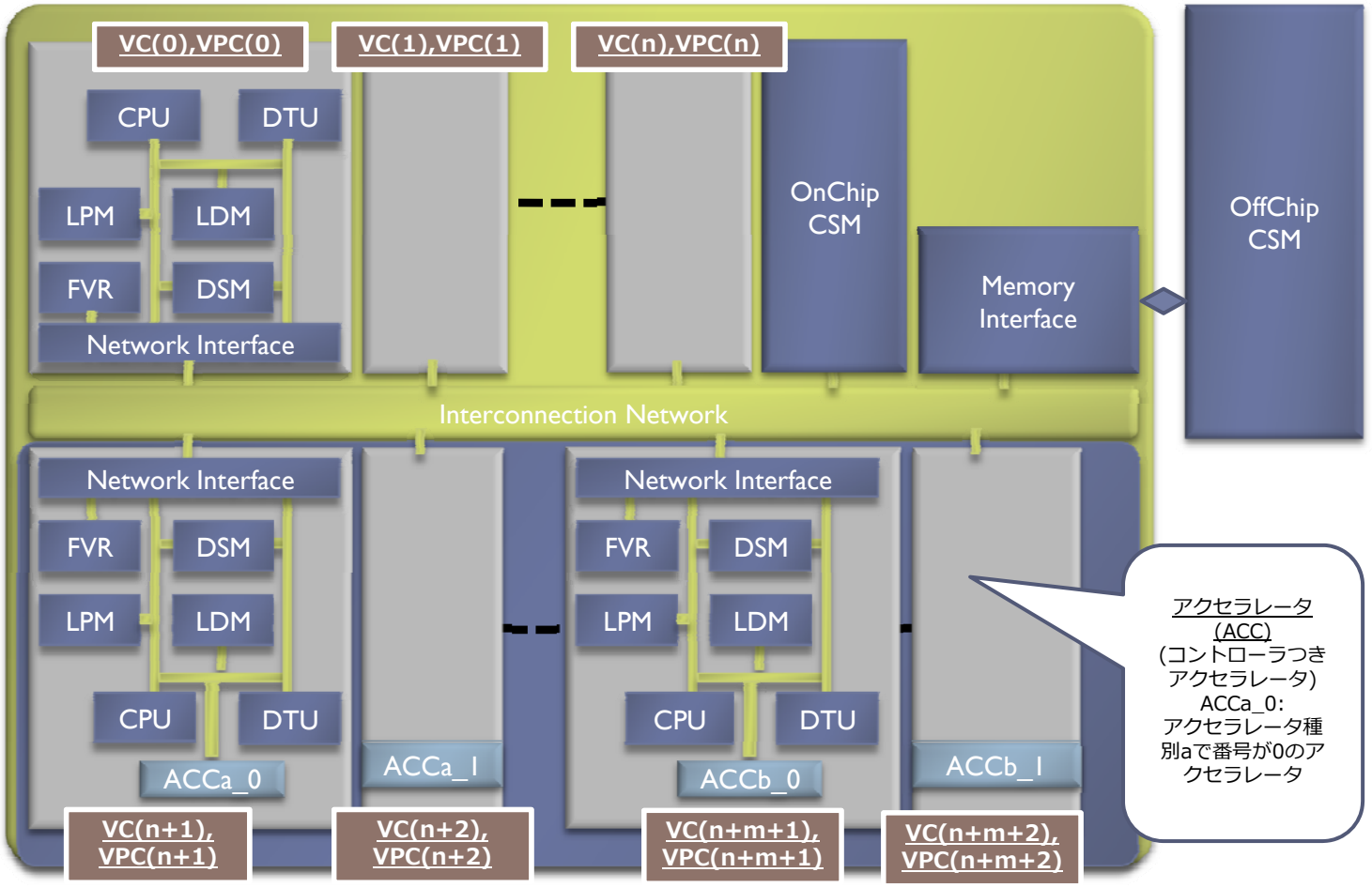
OSCAR APIのヘテロジニアスマルチコアへの拡張

特定処理の高速かつ低消費電力な処理への要求
- メディアコーデック
- 科学技術計算

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、グラフィクスアクセラレータ等
- 電力あたりの処理能力の向上

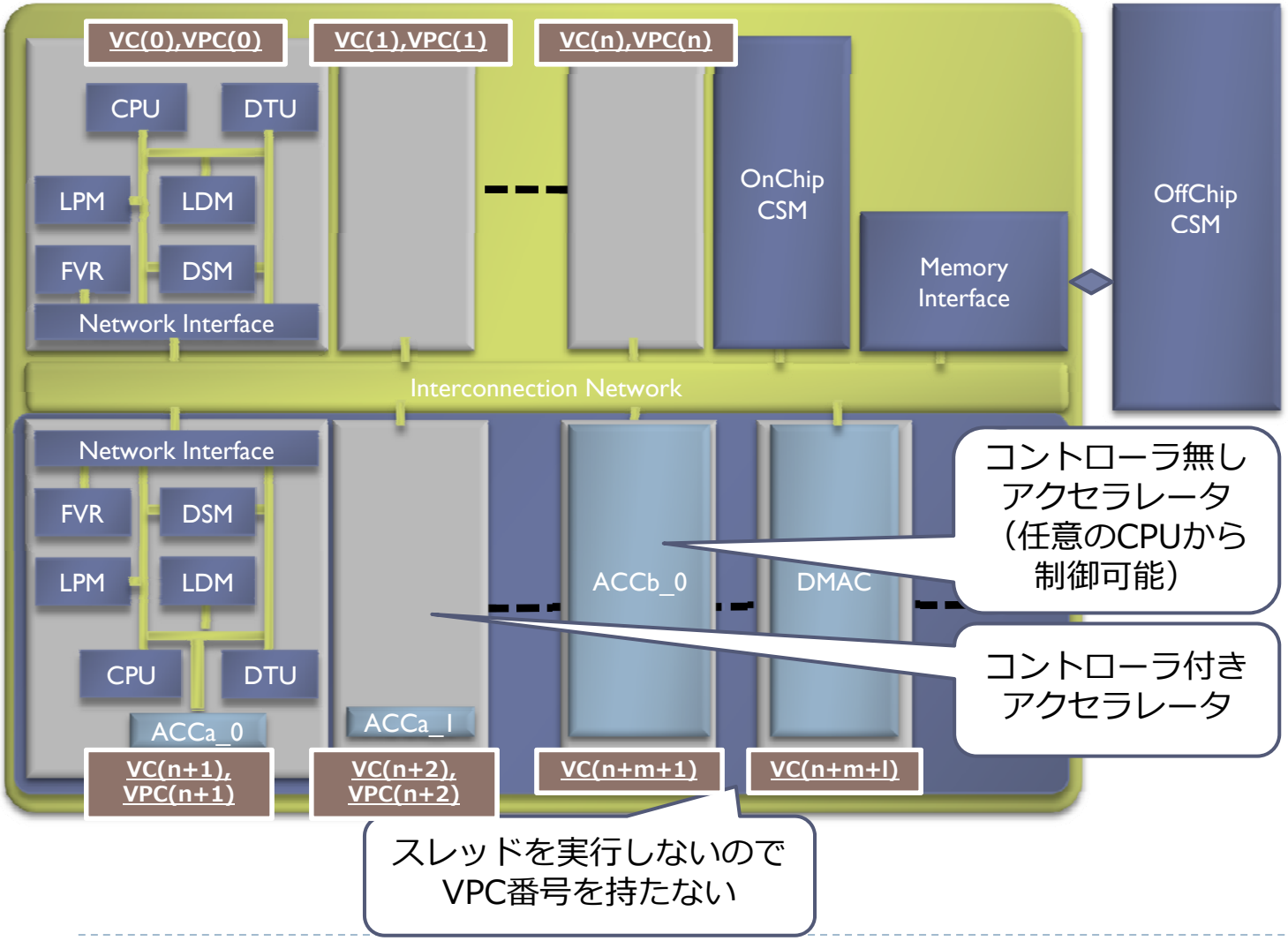
OSCAR APIのヘテロジニアスマルチコアへの拡張
- アクセラレータコアの制御

OSCARヘテロジニアス マルチコアアーキテクチャ



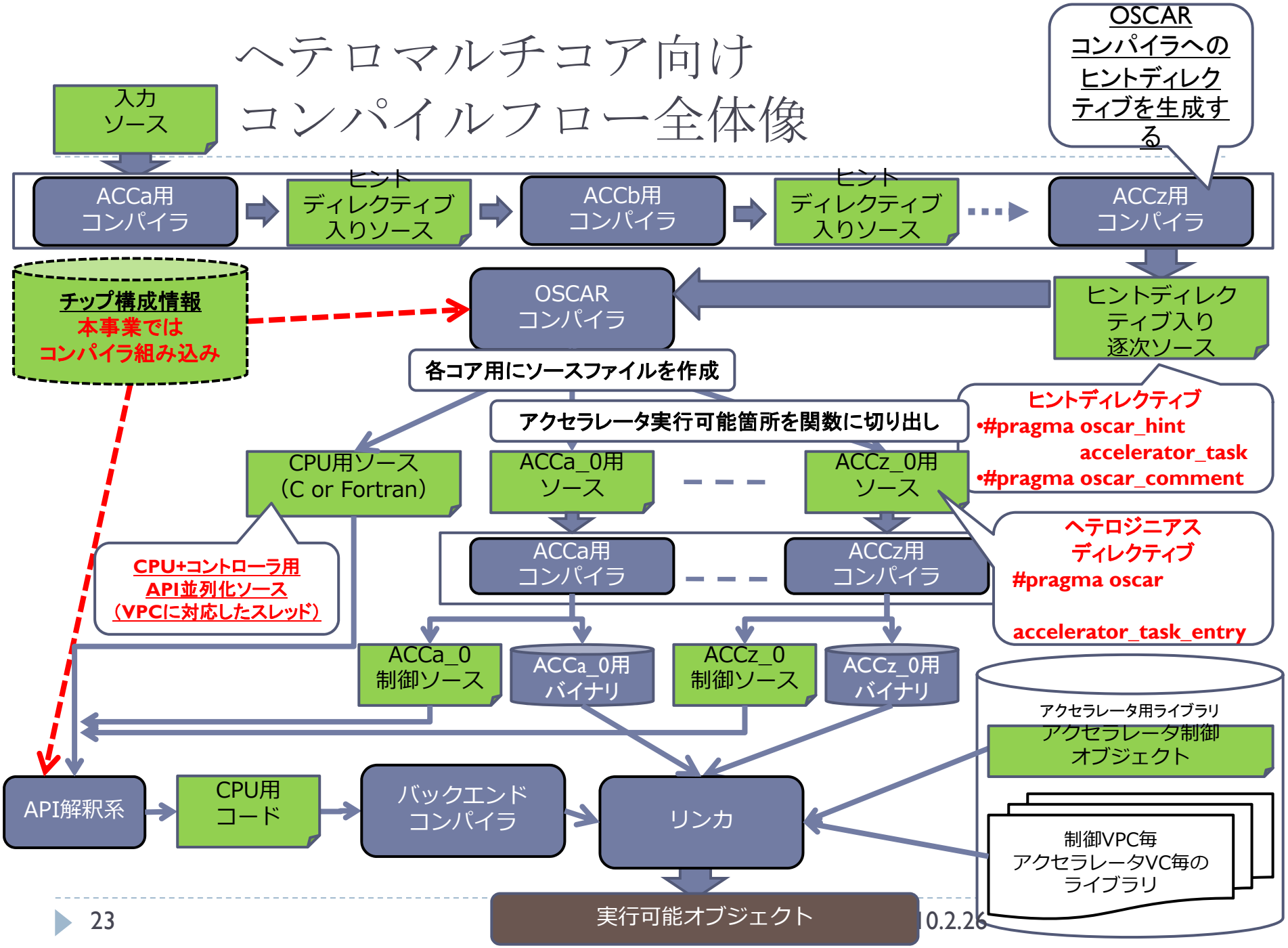
- DTU
  - Data Transfer Unit
- LPM
  - Local Program Memory
- LDM
  - Local Data Memory
- DSM
  - Distributed Shared Memory
- CSM
  - Centralized Shared Memory
- FVR
  - Frequency/Voltage Control Register

OSCAR API-Applicable
ヘテロジニアスマルチコアアーキテクチャ



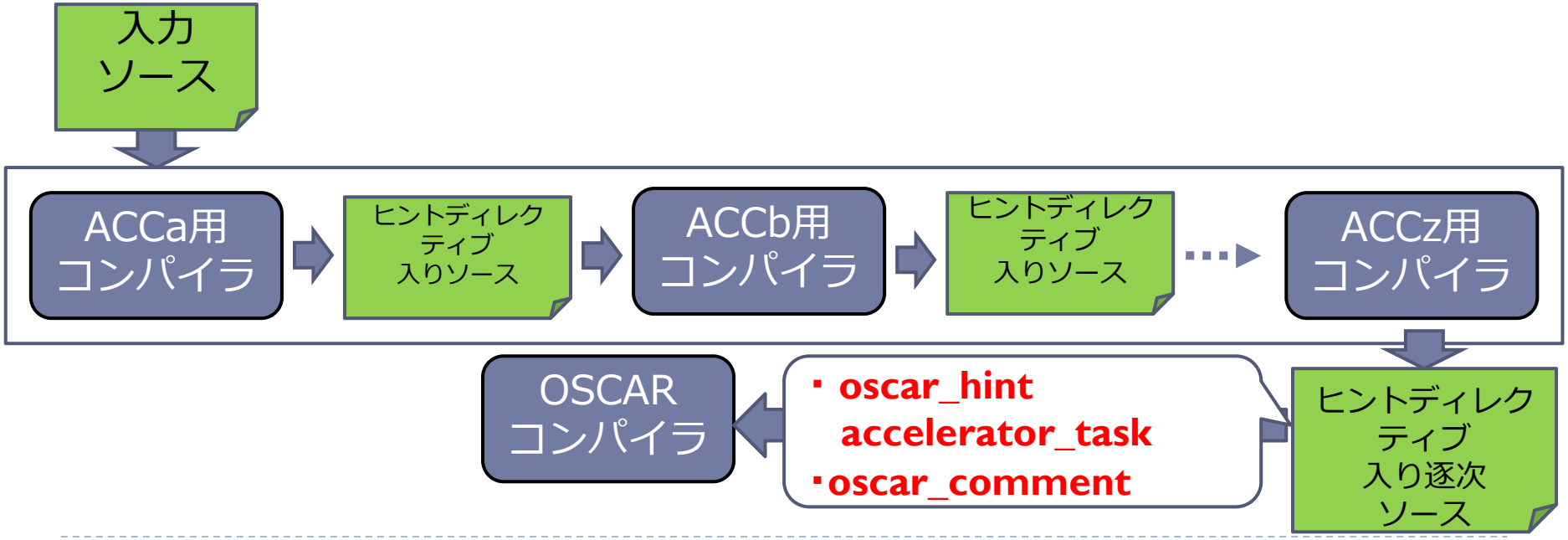
- DTU
  - Data Transfer Unit
- LPM
  - Local Program Memory
- LDM
  - Local Data Memory
- DSM
  - Distributed Shared Memory
- CSM
  - Centralized Shared Memory
- FVR
  - Frequency/Voltage Control Register

ヘテロマルチコア向け コンパイルフロー全体像



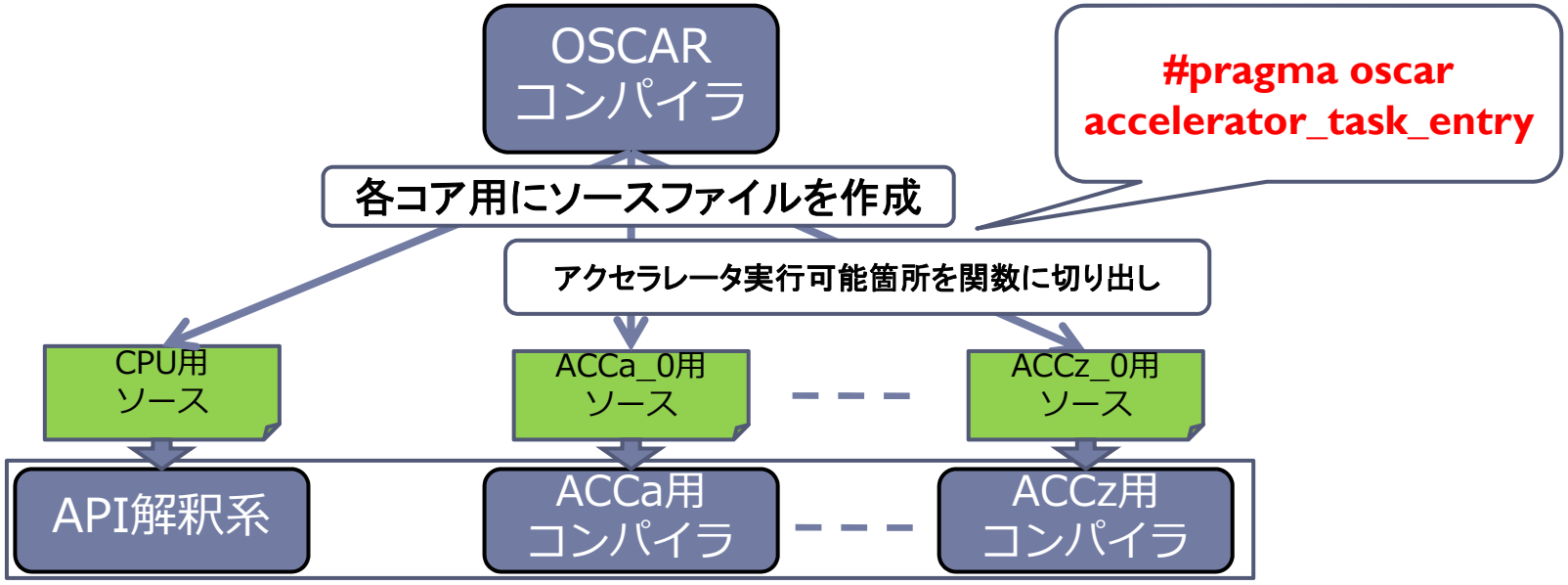
OSCARコンパイラ入力用ヒントディレクティブ
oscar_hint accelerator_task, oscar_comment

- アクセラレータコンパイラもしくはユーザーにより付加される
- oscar_hint accelerator_task:プログラム中の当該個所がアクセラレータで実行可能なことを示す
- oscar_comment:OSCARコンパイラ出力コードに含めるべきコメント文を指示する

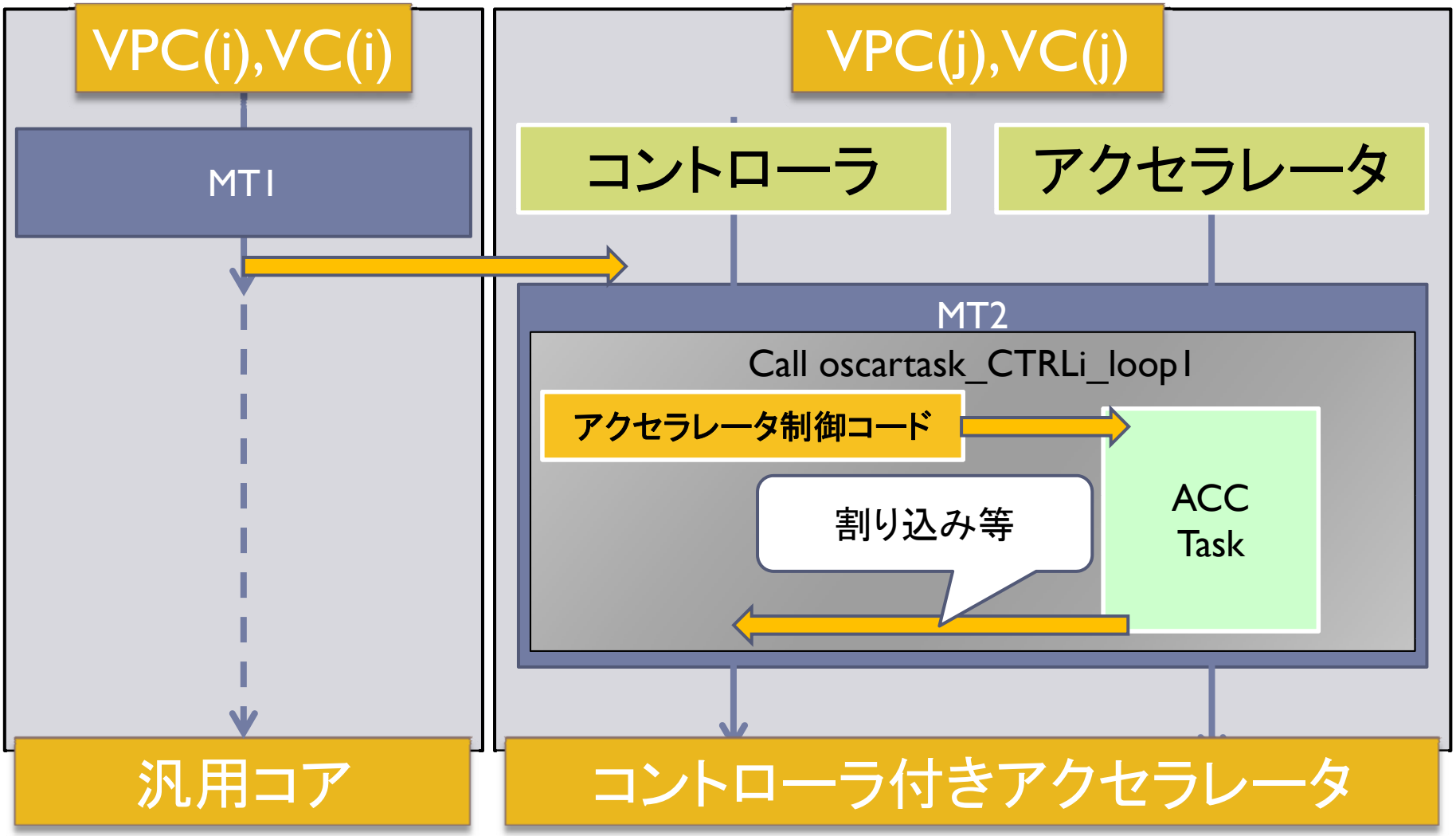


OSCARヘテロジニアスディレクティブ

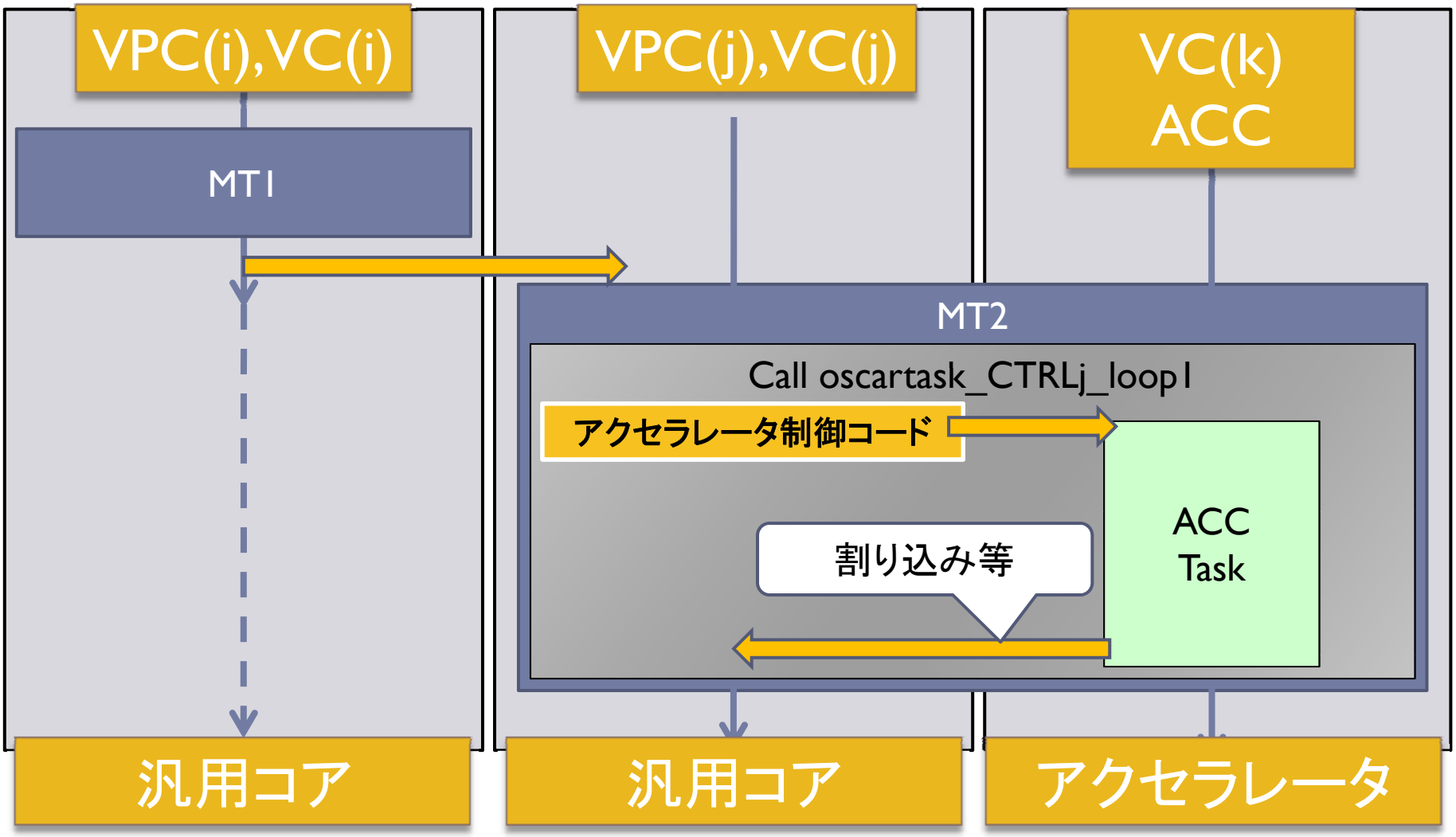
- OSCARコンパイラにより自動付加される
- 従来のホモジニアスAPIを拡張



本APIを用いた場合の実行モデル例 (コントローラ付きアクセラレータ)



本APIを用いた場合の実行モデル例 (コントローラなしアクセラレータ)



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 acccelerator_task
 (ACCa)  cycle(1000) in(a, b[0:9])
out(b[0:9])
  task_func();
}
task_func() {
  for (…) {…}// loop1
  for (…) {…}// loop2
#pragma oscar_comment "XXXXX"
  func1 (…) // func1
  for (…) {…}// loop3
}
```

関数呼び出し

OSCAR
コンパイラ

汎用コア用
ソース

Sample.omp.c

1CPU+1ACC用
並列化APIソース

```
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

アクセラレータ用
ソース

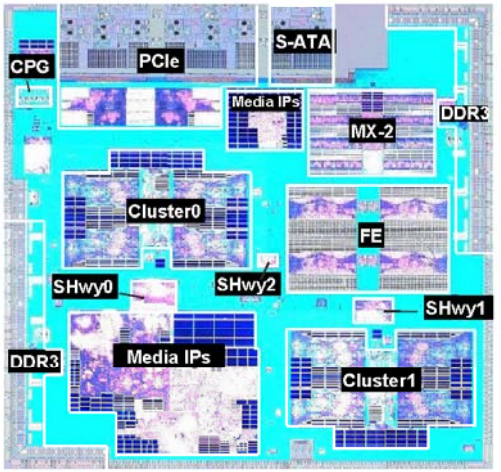
Sample.VC1.c

ACCa_0用ソース

```
#pragma oscar accelerator_task_entry
controller(0)
oscartask_CTRL0_task_func
oscartask_CTRL0_task_func() {
 for (…) {…}// loop1
  for (…) {…}// loop2
#pragma oscar_comment "XXXXX"
  oscarlib_CTRL0_ACCE
func1
  for (…) {…}// loop3
}
```

関数本体の記述が
取り出される

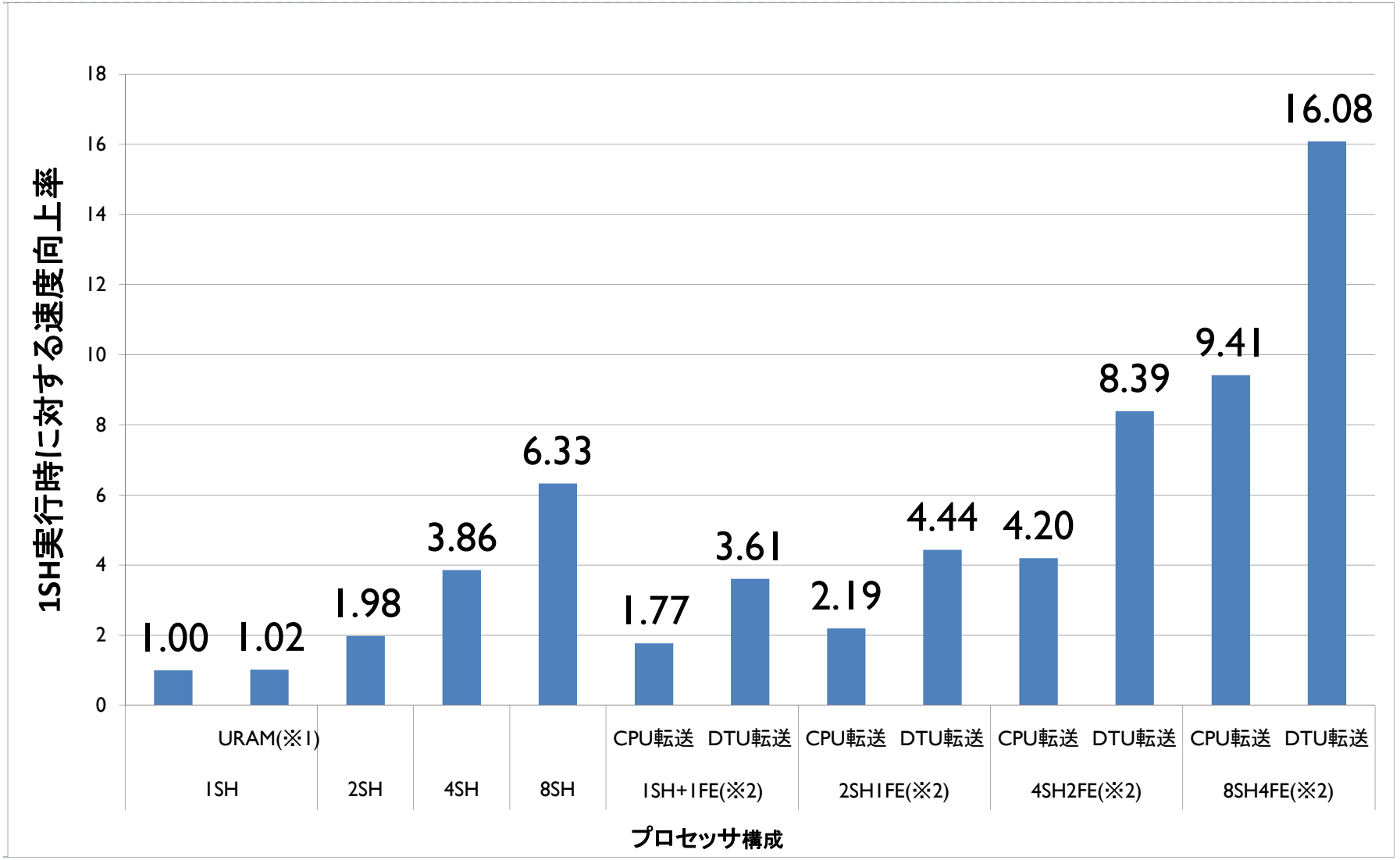
新規開発ヘテロジニアスマルチコアRP-Xでのヘテロジニアス並列性能評価



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

RP-XでのAACエンコーダを用いたヘテロジニアス並列性能評価



まとめ

- OSCAR API
  - v1.0を2008年11月に発表
  - OSCARコンパイラによる並列化を様々なマルチコアに展開
  - 高いスケーラビリティと低消費電力化
- OSCAR APIのメニーコアへの拡張
  - ソフトウェアによるキャッシュ制御機構
  - 今後のメニーコアでさらなるスケーラビリティでは必須
- OSCAR APIのヘテロジニアスマルチコアへの拡張
  - 汎用コアからアクセラレータコアを制御
- 仕様書を近日公開予定
  - 笠原研webにて

NEDO 「メニーコア・プロセッサ技術の先導研究」
メニーコア・アーキテクチャ・API検討委員会

- 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)
- 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)
- 神谷 幸男(富士通株式会社)