

「演算加速機構を持つ将来のHPCIシステムに関する調査研究」状況報告

主管事業実施機関：筑波大学計算科学研究センター

共同事業参画機関：東京工業大学，理化学研究所，
会津大学，日立製作所

協力機関：東京大学，広島大学，
高エネルギー加速器研究機構

「演算加速機構を持つ将来のHPCIシステムに関する調査研究」

- ナノテクやライフサイエンスの進歩、気候気象予測や地震・防災への対処には計算科学は不可欠かつ有効な手段
 - そのためにはさらなる計算能力が要請されている。
 - 設置面積、消費電力等の制限からノード数の増加による並列システムの性能向上には限界
- ライフサイエンスの分子シミュレーション等、多様な分野で比較的小さい一定サイズの問題の高速化が望まれている(いわゆる強スケーリング)
 - 対応した研究開発の例: ANTON, MDGRAPE-4

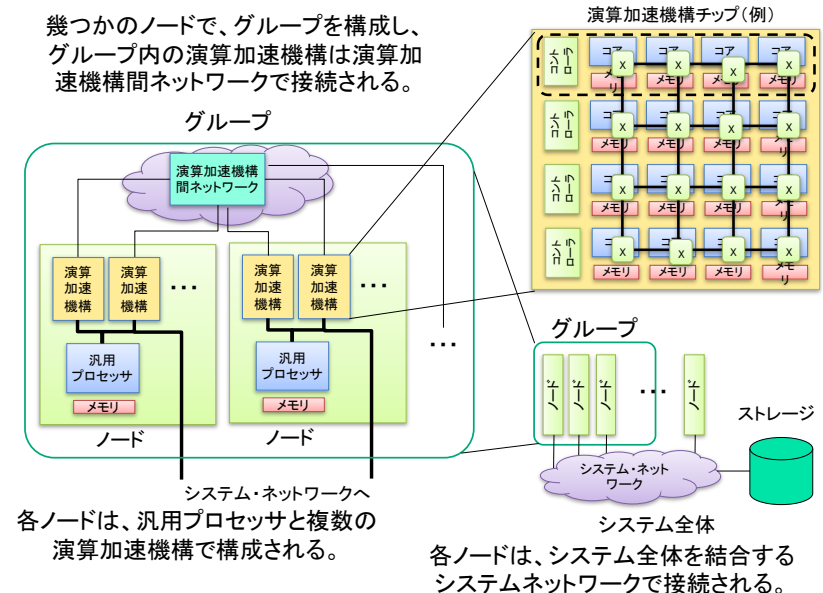


電力効率の大幅な効率化と強スケーリング問題の高速化による新たな計算科学の展開を目指して、演算加速機構による並列大規模システムについて調査研究を行う。

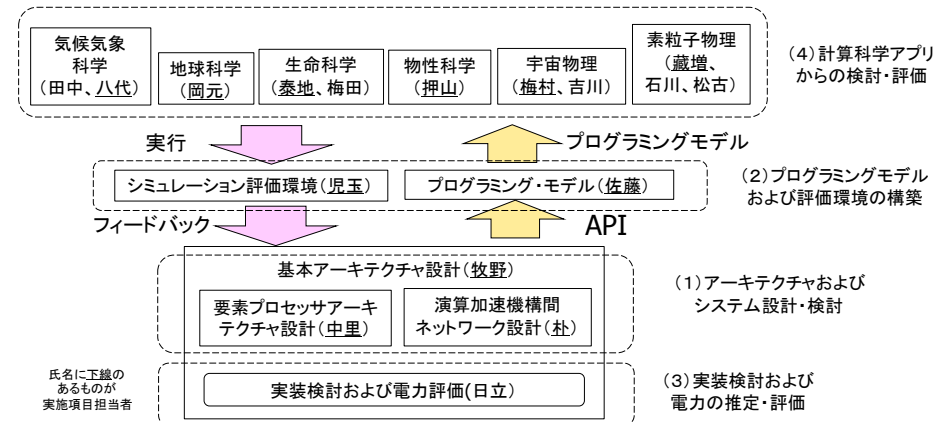
平成23年度文部科学省アプリケーション&コンピュータアーキテクチャ・コンパイラ・システムソフトウェア合同作業部会において、まとめられた「今後のHPCI技術開発に関する報告書」の中で、分類されたシステム構成のうち「**メモリ容量削減**」および「**演算重視**」のシステムを主な調査研究の対象とする。

多数の演算コアを内蔵したチップによる演算加速機構が汎用プロセッサで構成された並列システムの各ノードに接続もしくは内蔵されているヘテロジニアスな並列システムを想定

演算加速機構は、多数のスループットコアにより構成。スループットコアは、チップ内ネットワークにより結合される。図に示したものは一つの例。

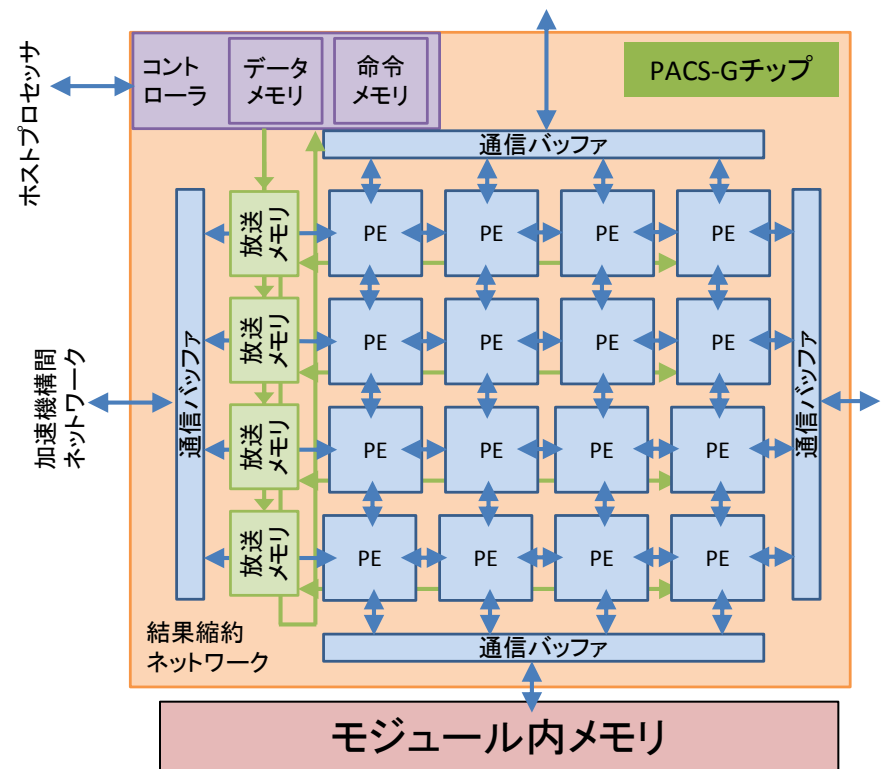


- 主管事業実施機関: 筑波大学 計算科学研究センター
- 共同事業参画機関: 東京工業大学、理化学研究所、会津大学、日立製作所
- 協力機関: 東京大学、広島大学、高エネルギー加速器研究機構
- 調査研究を、以下の4つの項目に分けて実施
 - (1)アーキテクチャおよびシステムの設計・検討
 - (2)プログラミング・モデルおよび評価環境の構築
 - (3)実装検討および電力の推定・評価
 - (4)計算科学アプリからの検討・評価



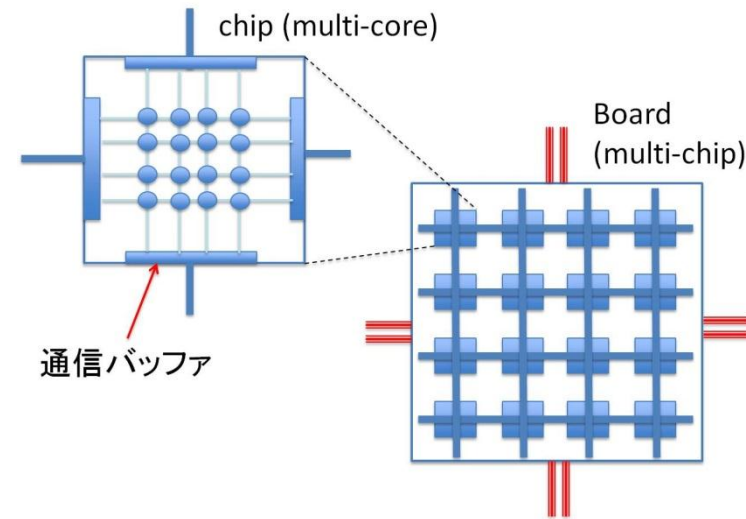
PACS-G アーキテクチャの概要： ノード(チップ)

- 以下のアーキテクチャを、Straw man(たたき台) アーキテクチャとして設定
- 演算集約型とメモリ削減型のステンシル計算を両立させるアーキテクチャ(プロセッサ、ネットワーク)をターゲットに設定
- 2018~2020年のLSIテクノロジーとして、14nmを想定。チップサイズを20mm²として、メモリ(SRAM)換算で1GB/チップを想定
- チップの基本アーキテクチャは、SIMD
- チップ内は、2次元のメッシュ・ネットワークを(当面)想定 (+ブロードキャスト・リダクションネットワークを検討)、コア間 16GB/s (双方向)
- コアとメモリの比を1:1 として、チップあたり4096 コア(PE) = 64 x 64
- チップ内メモリ 512MB/チップ, 128KB/コア
- コアの基本性能は2FMA@1GHz, したがって、4GFlopsx4096 = 16TFlops/チップ (64Kチップ/1EF)
- TSV 2.5次元実装によるモジュール内メモリを想定。HMC もしくはWide IO DRAM で、
バンド幅は1000-1500GB/s
サイズは、16-32GB/chip程度
- チップ外付けメモリ(DDR/DIM)は、想定しない
- 電力は250W/チップを目標 (16MW/1EF)
- 2048 チップ/group, Group内のチップは演算加速機構ネットワークで結合



PACS-G アーキテクチャの概要： ノード間、ホスト間

- 1024～2048チップ(グループ)ごとに演算加速機構間ネットワークで結合
- チップの2次元メッシュネットワークをボード上のチップ間ネットワークに展開する際、ボード上のチップ(例えば4x4=16個)を同様に2次元メッシュ結合すると、隣接チップ間(数cm～10cm程度)接続のバンド幅は、チップ内隣接ネットワークの20～40%程度で実現可能(電気配線)
- さらに、ボード間ネットワークまでも2次元(あるいはより高位の多次元)メッシュ展開とすると、インタフェースチップからの光コネクションが使えればチップ間バンド幅と同等(=チップ内ネットの40%程度)が実現可能。



チップ内コア間:

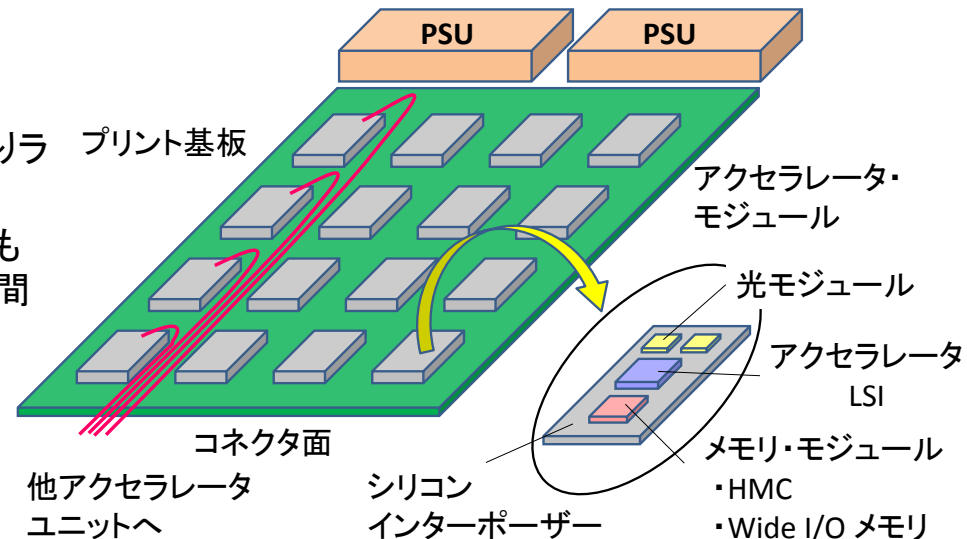
16GB/s (= 1GHz x 8B x 2(双方向))@コア間→1024GB/s (= 16GB/s x 64コア)

チップ間(ボード内、ボード外):

200～400GB/s (= 32ch. x 25～50Gbps x 2(双方向))

- QCDのような隣接通信のアプリであれば演算のB/F値よりラックサイズのシステムまではメッシュのままでも対応可能
- 当面、2次元メッシュで考えるが、もう少し高次元の実装も検討。また、メッシュをトーラスに変更することは、チップ間配線の実装で対応可
- 検討しているシステムは、汎用CPUを基本とした超並列システムにアタッチされることを想定
- ホストとのインタフェースは、PCI Express Gen4 x 16 相当の性能を期待

チップ間ネットワークの検討例



システムの実装イメージの検討例

計算科学アプリからの検討・評価(1/2)

- 昨年度は以下の5つのアプリのカーネルをターゲットとしてアーキテクチャとのco-designを進めた.
 - 格子QCD(素粒子分野)
 - 量子多体計算 N-body(宇宙物理分野)
 - 磁気流体コード HMD(宇宙物理分野)
 - 分子動力学 MD(生命科学)
 - 地震波計算コード(地球物理)
- 強スケーリングによる分子動力学アプリケーションの大幅な高速化
- 電力効率の大幅な効率化による格子QCD等のメモリ削減型アプリケーションの大規模・効率的実行

計算科学アプリからの検討・評価(2/2)

- 本システムは, 汎用ホストに接続されて利用されることを想定.
- コードのカーネル部分を演算加速機構部(グループ)にオフロードして, アプリケーションの実行を加速する. そのためには, どの部分がオフロードできるかを見極めることが重要
- 社会的課題に関するアプリケーションについても(間接的に)一部を加速できる可能性がある.
- 今年度の予定: NICAM(気象), RS-DFT(物性), FMO(化学)

プログラミング・モデルの検討および評価環境の構築の状況

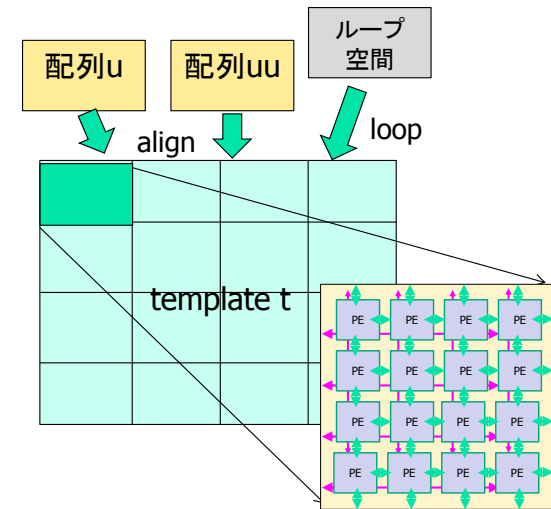
- 性能評価のためのクロックレベルのシミュレータと、ソフトウェア開発用命令レベルシミュレータを開発し、評価を進めている。
 - 命令セットの設計、ネットワーク構成などの設計・評価、モジュールメモリ内の利用、
 - 具体的コードによる定量的評価

■ プログラミングモデルの検討

- アセンブラレベルのSIMDプログラミングをするためのCのsubsetのような言語
- ユーザに提供するための言語として、XcalableMPの拡張を検討
- (C * などのデータ並列言語)

■ XcalableMP + OpenACCによるプログラミングモデル

- ホストからチップにオフロードするために、OpenACCの指示文を用いる。
- チップの中のプログラミングに、XcalableMPのtemplateを利用
 - Templateは、データやindex空間をマップするための仮想格子
- ループのプロセッサと配列を整合させることで、PEあたりのコードを生成することができる。
- データを積層メモリに置く場合は、仮想プロセッサという形でマッピング(?)



- Template directiveで、宣言
- (distribute directiveで、templateをpeにmapping)
- align directiveで、配列を整列
- loop directiveで、ループの実行プロセッサを割り当て

```
#pragma xmp template t(0:XSIZE+2, 0:YSIZE+2)

double u[XSIZE+2][YSIZE+2],uu[XSIZE+2][YSIZE+2];
#pragma xmp align u[i][j] with t(i,j)
#pragma xmp align uu[i][j] with t(i,j)

#pragma xmp loop on t(x,y)
for(x = 1; x <= XSIZE; x++)
  for(y = 1; y <= YSIZE; y++)
    u[x][y] = (uu[x-1][y] + uu[x+1][y]
              + uu[x][y-1] + uu[x][y+1])/4.0;
```

H24

実施スケジュール H25

(1) 基本アーキテクチャおよびシステムの設計検討

概念設計(概略)

基本アーキテクチャの設計

提案アーキテクチャの電力・機能評価用
テストチップ(1コア相当) 実装

提案アーキテクチャの
評価(電力・機能)・課題抽出

チップ間ネットワークの基本設計

チップ間ネットワークの評価・課題抽出

(2) プログラミングモデルおよび評価環境の構築

プログラミングモデル
の検討

基本コンパイラの検討・設計

プログラミング・モデルおよび
コンパイラの試作・実装

プログラミング・モデルおよび
コンパイラの評価・課題抽出

ソフトウェアシミュレータ
の設計・実装(チップ内)

ソフトウェアシミュレータ
の設計・実装(チップ間)

シミュレータの改良・高速化・FPGA実装

(3) 実装検討および電力の推定・評価

実装検討および電力の推定・評価

(4) 計算科学アプリからの検討・評価

アプリケーションの選定・解析・概算性能算定

提案システム向けコード開発

選定アプリの評価・課題抽出