将来の HPC アーキテクチャ

平木 敬†

本講演では、文部科学省が検討を進めている「今後のハイパフォーマンス・コンピューティング技術の研究開発の検討」の一環として 2018 年前後の HPC アーキテクチャの検討を行っている「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作業部会」における検討状況を紹介する。2018 年ごろのスーパーコンピューティングシステムにおける最大課題は、性能あたり消費電力とメモリバンド幅・メモリ量とのトレードオフと、性能あたりのコストであることは広く認識されている。「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作業部会」では、多数のメンバーの意見を集約し現時点で想定されるロードマップを数種類策定した。本講演では部会による検討資料をもとに、将来への方向性を述べる。

On next generation HPC architecture

KEI HIRAKI †

This talk introduces outline of the on-going discussion on architecture and software of the HPC systems around 2018. This is based on “Study on research and development of high-performance computing technology in future” by the Ministry of Education, Culture, Sports, Science and Technology. The issues on HPC architecture are discussed at “Working group on architecture, compiler and system software”. Main difficulty to construct supercomputing systems around 2018 will be (1) power efficiency, (2) memory bandwidth per flops, (3) memory size per flops and (4) cost performance. The working group formulated several roadmaps by summarize discussions among working group members.

† 東京大学情報理工学系研究科
Graduate School of Information Science and Technology, the University Tokyo

メンバー紹介（50音順）

- **取りまとめ（執筆者）**
  - 安島 雄一郎（富士通）
  - 石井 康雄（東大・NEC）
  - 井上 弘士（九大）
  - 加納 健（NEC）
  - 鯉渕 道紘（NII）
  - 近藤 正章（電通大）
  - 佐藤 幸紀（北陸先端大）
  - 佐野 健太郎（東北大）
  - 鈴木 篤浩（日立）
  - 曽根 猛（日立）
  - 萩原 孝（NEC）
  - 塙 敏博（筑波大）
- **アドバイザ**
  - 中村 宏（東大）、平木 敬（東大）、松岡 聡（東工大）

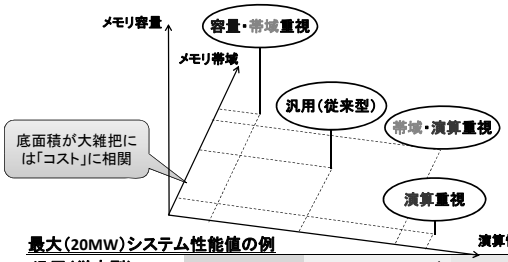
1

なぜ将来のHPCを検討するか？

- **2000年代でDenardスケーリングの時代は終わった**
  - テクノロジであらゆる性能が向上する時代は過去のものになった
- **2010年代はアーキテクチャ多様性の時代**
  - 特徴のあるアーキテクチャが競争力を持つ
  - **汎用アーキテクチャでは、どのアプリケーションでも一番になれない**
- **今後の計算科学向け計算機アーキテクチャ**
  - 重要サイエンス分野において高い実効性能が得られることが必須
  - 複数のアーキテクチャで計算科学領域の大半をカバー
  - 計算科学との密接な連携が必要
  - ⇒ **サイエンスドリブン型のアーキテクチャ開発**
    計算科学の要求と、利用可能なテクノロジのベストマッチ

2

アーキテクチャ分類の想定



最大（20MW）システム性能値の例			
汎用（従来型）	400 PetaFLOPS	40 PetaByte/s	40 PetaByte
容量・帯域重視	100 PetaFLOPS	100 PetaByte/s	100 PetaByte
帯域・演算重視	1000 PetaFLOPS	500 PetaByte/s	0.2 PetaByte
演算重視	2000 PetaFLOPS	10 PetaByte/s	10 PetaByte

3

Part 1
HWトレンド予測に基づく
2018年ごろの計算機性能予想

4

アーキテクチャのバラエティ

- **リファレンス型**
  - PCクラスタや「京」と似た、CPUを中心とした計算機
  - B/F=0.1、システム側で検討していたアーキテクチャ
- **メモリバランス重視型（容量とBFのバランス）**
  - リファレンス型から演算器を減らし、メモリ帯域(B/F)を増やす
  - B/F=1.0、チップ上には多くのメモリインタフェース
- **演算重視型**
  - リファレンス型からメモリ帯域(B/F)を減らし、演算器を増やす
  - B/F=0.005、チップ上には多くの演算器
- **メモリ容量削減型（B/F重視型）**
  - メモリ容量を減らし(1/100)、メモリ帯域(B/F)を増やす
  - B/F=0.5、チップ上に主記憶を搭載

アーキテクチャの研究としては、上記4分類のどれかのアーキテクチャを研究開発すべきというメッセージではない。これをベースにアプリケーションがどういうアーキテクチャによりフィットするか検討し、さらによいアーキテクチャをFSすべきである。

5

システム性能見積もり結果

20MW制限下でのシステム性能(青字が弱み・赤字が強み)

	リファレンス	メモリバランス	演算重視	メモリ容量削減型
演算性能	200 PFLOPS〜 400 PFLOPS	50 PFLOPS〜 100 PFLOPS	1000 PFLOPS〜 2000 PFLOPS	500 PFLOPS 〜 1000 PFLOPS
メモリ帯域	20 PB/s 〜 40 PB/s	50 PB/s 〜100 PB/s	5 PB/s 〜10 PB/s	250PB/s 〜 500PB/s
メモリ容量	20 PB〜40 PB	50 PB〜100 PB	5 PB〜10 PB	0.1 PB〜0.2 PB

- **検討の前提**
  - 電力20MW・設置面積2500msq・コモディティ技術で実現できる数値
  - 数値は外挿で予測、電力・コスト・技術成熟度を考慮
  - より高度な計算機技術があれば上記性能は超えることは可能
    - システム側ではそれを目指すための研究開発ロードマップを作製中
      - 例) 電力制御でシステム特性を変化させ、複数特性を実現、など
    - そうしたFS・研究開発を実現することが望まれる

6

ネットワーク構成について

- **リファレンス構成 (Infiniband)** (青字が弱み・赤字が強み)
  - High-radix型トポロジ (Fat Tree型)
    - メモリ帯域の1/10のInjection帯域
    - 帯域: Injection 32GB/s, Point-to-Point 32GB/s, Bisection 2.0PB/s
    - 片道遅延: 隣接最短 200ns, 隣接最長 1000ns, システム直径 1000ns
  - Low-radix型トポロジ (4次元トーラス)
    - 総メモリ帯域の1/100のBisection帯域
    - 帯域: Injection 128GB/s, Point-to-Point 16GB/s, Bisection 0.13PB/s
    - 片道遅延: 隣接最短 100ns, 隣接最長 200ns, システム直径 5us
- **ネットワークのオプション**
  - グローバルバリア通信専用NW、リダクション演算専用NW 、・・・
  - 必要な機能はアプリ毎に様々である、FSで必要な機能を明らかにすることが望まれる

7

I/Oとストレージアーキテクチャ

- **階層型ストレージアーキテクチャは重要**
  - ローカルストレージとグローバルストレージの帯域・容量
- **総容量はメモリ容量の100倍**
  - 10PB ～ 40PB x 100倍 = 1EB ～ 4EB
- **帯域はメモリ容量を1000秒でローカルデバイスに退避する程度、グローバルにはコスト次第だが1/10以下**
  - 20PB ～ 50PB / 1000s = 20 TB/s ～ 50TB/s

8

Part 2
さらなる高性能(e.g., Exa Scale)を可能にする技術・研究アプローチ

9

エクサスケールを可能にする技術・研究アプローチ

- **ヘテロジニアスアーキテクチャ**
  - Latency CoreとThroughput Coreのデータ共有方式
    - 明示的⇔暗黙的
  - Latency CoreとThroughput Coreの結合方式
    - 主メモリ経由, オンチップメモリ階層経由, etc.
  - Throughput Core間のデータ共有方式
    - レイテンシ短縮とバンド幅向上
- **記憶階層**
  - 並列計算モデルの選定とそれに適したコア間接続・メモリ階層
  - アルゴリズム毎への最適化機能
    再構成メモリ階層、スマートメモリ、リコンフィギャラブル技術

10

エクサスケールを可能にする技術・研究アプローチ

- **Memory Wallを解消する三次元積層DRAMの活用**
  - プロセッサとDRAMの三次元積層(Wide I/O)
    - 複数の独立したチャネル → 主記憶データ局所性の制御
    - 冷却可能な低消費電力プロセッサ
  - 三次元積層DRAMキューブ(Hybrid Memory Cube)
    - インタフェースチップによる高機能トランザクション
    - インタフェースチップの省電力化
  - メモリ容量の低下に対するソフトウェアとのco-design
- **次世代NVRAM(PCRAM, MRAM, etc.)の活用**
  - 新しい故障回復(Fault Resilience)モデル、ジョブスケジュールモデル
  - 書き込み制限や信頼性への対処
- **耐故障性の向上**
  - ECC、CRC、Scrubbing

11

エクサスケールを可能にする技術・研究アプローチ

- **大規模並列・ストロングスケーリング・ネットワーク**
  - レイテンシ削減が重要
    - 通信時間: ノード内遅延＋ネットワークインタフェース＋インターコネクト
  - トポロジとルーティング
    - 隣接通信重視: Low-radixネットワーク (4次元トーラスなど)
    - 全体の帯域重視:high-radixスイッチを活用した低遅延トポロジ
    - QoSとCongestion Control
    - 小メッセージ(3KB以下)の低遅延化と処理の効率化(100万メッセージ/sレンジ)
    - 階層的なネットワーク、細粒度同期 の支援

    (アプリの通信パターンに応じた検討)
  - ネットワークインタフェース(NI)
    - オンチップNIとオフチップNIの統合 、軽量通信プロトコル、高度なNIC設計
    - Collective通信の効率化、細粒度同期 、converged NI
  - 特定ノード間の帯域最大化・低遅延化のためのサーキットスイッチ技術の部分的な利用検討

12

エクサスケールを可能にする技術・研究アプローチ

- **低消費電力**
  - CMOSデバイスの微細化に頼るだけではエクサは達成できない
    - 何らかの工夫で さらに電力効率(電力当り性能)を約60倍引き上げる必要あり
  - 新しいデバイス技術の活用
    - 半導体微細化、トライゲート、SOTB、不揮発メモリ、極低電圧回路、3次元積層
    - これらに対応したアーキ技術(ばらつき影響の緩和、信頼性低下への対策など)
  - 電力効率に優れたアクセラレーション技術の確立
    - GPU、FPGA、メニーコアなどのヘテロジニアスアーキテクチャの技術開発
  - メモリ・インターコネクトの大幅な低消費電力化
    - 新デバイスの採用、きめ細かな動作モード制御
    - 電力指向トポロジの考案や低消費電力ルータ／スイッチの開発
    - (搭載すべきメモリ量の削減、通信速度の低下も議論する必要あり)
  - システムレベル電力制御による性能／電力効率の大幅な改善
    - アプリ特性に応じて性能／電力効率を最大化する電力制御技術
    - 性能・電力のトレードオフを調整可能なPower Knobの提供
    - 各Power Knobの適切な空間・時間粒度の決定
    - システム全体の電力状況を把握可能な正確・高速な電力モニタリングの実現

13

エクサスケールを可能にする技術・研究アプローチ

- **耐故障・信頼性**
  - システム構成部品の増大、高集積化、低電力化による故障率増加
    - MTBFが5分程度になるとの試算もある
    - 長時間に及ぶアプリ実行を支援する技術が必要
  - チェックポイント・リスタートの高速化
    - 数十万ノードのシステムではチェックポイント時間が数十分に
    - チェックポイント・リスタート時間の(数秒程度への)改善
      → コスト・消費電力を勘案した上での不揮発メモリの導入
  - 冗長実行の支援機構・電力バランス検討
    - チェックポイント・リスタートを避けるための冗長実行支援
      - ジョブのディスパッチや結果のコミットの支援
      - 電力バジェットとのバランスの検討
  - プロセスマイグレーションの支援機構
    - 故障予測に基づく事前プロセスマイグレーション、マイグレーション支援機構
    - 故障検知や予測を支援する高精度なモニタリング機構

14

まとめ

- **サイエンスドリブン型のエクサスケールシステム**
  - 科学的・社会的な課題の解決が目的
  - ターゲットアプリで高い実効性能が得られるシステム
- **エクサスケールシステム構成例**
  - いくつかのアプリケーションパラメータを基に4種類を提示
- **電力の壁による課題とそれを打ち破る研究アプローチ**
  - 高電力効率 → ヘテロジニアス・アーキテクチャ
  - データ移動削減 → 深い記憶階層、生産性低下への対処
  - 低電圧・低周波数→大規模並列化(Strong Scaling)、信頼性低下への対処
  - 三次元積層メモリ→ メモリ容量の低下、新しいメモリデバイスの活用
  - さらなる電力改善 → 新デバイス・バジェット管理
- **アプリ・システムソフトとの連携**

15

必要な帯域・エネルギー検討

Linpack電力見積もり

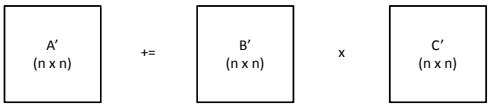
16

はじめに

- **電力性能の目標値のシナリオを作るためにLinpack(dgemm)で1exa flopsのシナリオを検討**
  - 他のアプリでも良いが適当なものが無かった、ということ
- **目標値: 10pJ / flop 20MWの50%をCPUに使うシナリオ**
- **結論: 普通にやると20pJ/flop~30pJ/flopが限界**
  - 10pJ / flopを諦めるか、3倍程度改善する方法を考える必要がある
    - システム:電力の割振変更してCPUへ70%割り振る ← 10pJ/flopを諦める
    - アーキテクチャ:SIMD幅を増やす
    - デバイス:電圧下げる

17

行列積の計算方式



ローカルメモリ上に配置
(必要なメモリ帯域は小さい)

上位の記憶階層からロード

- **N x Nの正方行列積をn x nのブロックに分割して計算する場合を考える**
  - Aを一旦ロードしてB/Cを入れ替えながら計算したあとにAをストアするブロッキングを考える
- **演算回数などのパラメータ**
  - 演算回数
    - 2 x N^3 FLOP
  - データのリード・ライト回数
    - (N/n) ^ 3回繰り返すこと: メモリリード回数 n^2 x 8B x 2行列
    - (N/n) ^ 2回繰り返すこと: メモリリード回数 n^2 x 8B / メモリライト回数 n^2 x 8B、中間階層の場合: 4 x n^2 x 8B
      - 必要なリード回数: (2 x N/n + 1) x N^2 x 8B, ライト回数 N^2 x 8B
      - 中間階層の場合: (2 x N/n + 2) x N^2 x 8B, 4 x N^2 x 8B
  - 必要なローカルメモリサイズ: n^2 x 8B x 6行列
    - A,B,Cの分と、A,B,Cのダブルバッファリングの分
    - 命令や制御用のメモリを含めて、きりのいいところでn^2 x 64Bの変と仮置きする
  - 必要なメモリ帯域
    - 必要なメモリ帯域: 8 x (1/n + 1/N) byte/flop (readの帯域), 8 x 1/N byte/flop (ライトの帯域)
    - 中間階層の場合: 8 x (1/n + 2/N) byte/flop, 8 x 4/N byte/flop

18

必要なローカルメモリ量

主記憶階層									
行列サイズ	ブロックサ	演算回数	必要な帯域	メモリ量	リード回数	ライト回数	リード回数	ライト回数	参照量
		FLOP	B/F	KB	Words	Words	W/FLOP	W/FLOP	bit/FLOP
8192	1	1.1E+12	8.001	0.0625	1.10E+12	6.71E+07	1.00E+00	6.10E-05	6.40E+01
8192	2	1.1E+12	4.001	0.25	5.50E+11	6.71E+07	5.00E-01	6.10E-05	3.20E+01
8192	4	1.1E+12	2.001	1	2.75E+11	6.71E+07	2.50E-01	6.10E-05	1.60E+01
8192	8	1.1E+12	1.001	4	1.38E+11	6.71E+07	1.25E-01	6.10E-05	8.01E+00
8192	16	1.1E+12	0.501	16	6.88E+10	6.71E+07	6.26E-02	6.10E-05	4.01E+00
8192	32	1.1E+12	0.251	64	3.44E+10	6.71E+07	3.13E-02	6.10E-05	2.01E+00
8192	64	1.1E+12	0.126	256	1.72E+10	6.71E+07	1.57E-02	6.10E-05	1.01E+00
8192	128	1.1E+12	0.063	1024	8.68E+09	6.71E+07	7.87E-03	6.10E-05	5.08E-01
8192	256	1.1E+12	0.032	4096	4.36E+09	6.71E+07	3.97E-03	6.10E-05	2.58E-01
8192	512	1.1E+12	0.017	16384	2.21E+09	6.71E+07	2.01E-03	6.10E-05	1.33E-01
8192	1024	1.1E+12	0.009	65536	1.14E+09	6.71E+07	1.04E-03	6.10E-05	7.03E-02
8192	2048	1.1E+12	0.005	262144	6.04E+08	6.71E+07	5.49E-04	6.10E-05	3.91E-02

- **行列計算をする"コンテキスト"毎に必要なリソース**
  - "コンテキスト"とは分散で計算する場合の"GPUのSM"や"CPUのコア"単位
  - コア0がパネルの一部分(A')を計算して、コア1が別のパネル(A'')を計算する場合には2倍のリソースが必要、ということ
  - B/F=0.501ならば16KBのLMが必要 (京の場合: 8コアでL2=128KBが必要)
  - B/F=0.063ならば1MBのLMがコンテキストあたり必要
  - B/F=0.017ならば16MBのLMがコンテキストあたり必要

19

それぞれのメモリ階層での参照数

		必要な帯域(W/F)										
		コピー元の行列サイズ										
ブロックサイズ	容量	2	4	8	16	32	64	128	256	512	1024	2048
		0.25	1	4	16	64	256	1024	4096	16384	65536	262144
	KB	MRF			L1		L2			L3		
1	0.06	2.500	1.750	1.375	1.188	1.094	1.047	1.023	1.012	1.006	1.003	1.001
2	0.25	2.000	1.250	0.875	0.688	0.594	0.547	0.523	0.512	0.506	0.503	0.501
4	1.00	1.750	1.000	0.625	0.438	0.344	0.297	0.273	0.262	0.256	0.253	0.251
8	4.00	1.625	0.875	0.500	0.313	0.219	0.172	0.148	0.137	0.131	0.128	0.126
16	16.00	1.563	0.813	0.438	0.250	0.156	0.109	0.086	0.074	0.068	0.065	0.064
32	64.00	1.531	0.781	0.406	0.219	0.125	0.078	0.055	0.043	0.037	0.034	0.033
64	256.00	1.516	0.766	0.391	0.203	0.109	0.063	0.039	0.027	0.021	0.019	0.017
128	1024.00	1.508	0.758	0.383	0.195	0.102	0.055	0.031	0.020	0.014	0.011	0.009
256	4096.00	1.504	0.754	0.379	0.191	0.098	0.051	0.027	0.016	0.010	0.007	0.005
512	16384.00	1.502	0.752	0.377	0.189	0.096	0.049	0.025	0.014	0.008	0.005	0.003
1024	65536.00	1.501	0.751	0.376	0.188	0.095	0.048	0.024	0.013	0.007	0.004	0.002

- **各階層で必要なメモリ参照の回数**
  - L3: 0.025W/F (B/F=0.20)
  - L2: 0.141W/F (B/F=1.13)

20

レジスタファイル/L1の参照回数

		L1 Cache				Register File			
ベクトル長	レジスタ容量	Read	Write	Read	Write	Read(MAX)	Write(MAX)	Read(min)	Write(min)
	要素数	B/F	B/F	W/F	W/F	W/F	W/F	W/F	W/F
2.00	16.00	8.00	8.00	1.00	1.00	1.50	0.75	1.25	0.50
4.00	96.00	6.00	4.00	0.75	0.50	1.25	0.63	0.88	0.25
8.00	640.00	5.00	2.00	0.63	0.25	1.13	0.56	0.69	0.13
16.00	4608.00	4.50	1.00	0.56	0.13	1.06	0.53	0.59	0.06
32.00	34816.00	4.25	0.50	0.53	0.06	1.03	0.52	0.55	0.03
64.00	270336.00	4.13	0.25	0.52	0.03	1.02	0.51	0.52	0.02

- **L1とレジスタファイルの必要な帯域**
  - L1: read 0.63W/F, write 0.25W/F
    - B/F比率: Read 5.00 B/F, Write 2.00 B/F
  - RF参照回数(最大値): Read 1.13W/F, Write 0.56W/F
  - RF参照回数(最小値): Read 0.69W/F, Write 0.13W/F

21

電力値の計算

ベクトル長:8、L1:16KB, L2:256KB, LLC:16MB, 8192x8192の行列積

L3/L2/L1でブロッキング

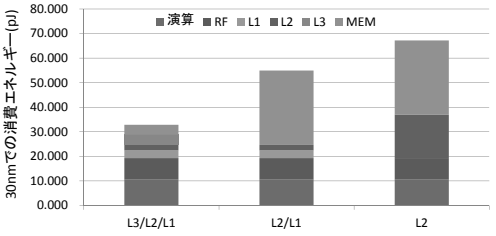
- 条件
  - メモリ帯域: 0.017 B/F以上
- 1演算あたりの電力試算
  - 演算器: 1 flop
  - RF: 1.13 read, 0.56 write
  - L1: 0.63 read, 0.25 write
  - L2: 0.109 read/write
  - L3: 0.020 read/write
  - メモリ: 0.136 bit read/write
- 合計: 32.8 pJ / flop

L2/L1でブロッキング

- 条件
  - メモリ帯域: 0.126 B/F以上
- 1演算あたりの電力試算
  - 演算器: 1 flop
  - RF: 1.13 read, 0.56 write
  - L1: 0.63 read, 0.25 write
  - L2: 0.109 read/write
  - L3: なし
  - メモリ: 1.01 bit read/write
- 合計: 54.9 pJ / flop

22

各方式の演算あたりエネルギー分布



- **ブロッキングする階層を切り替えて電力の差分をプロット**
  - 深い階層化をすることで消費電力が劇的に下がる
    - メモリ階層ごとの電力がほぼ均一になるのが良いバランスにみえる
    - 階層を減らすと生産性が上がるがトータル電力も上がる

23

まとめ

- **30nm世代で約30pJ/flopのエネルギーがDGEMMに必要**
  - ブロッキングをサボると60pJ/flop程度必要になる
  - レジスタ周辺の検討が最適でない部分があるがずれても2倍程度
- **テクノロジトレンドなどの予測**
  - 制御系の電力:データ移動のみの電力算出をしているため、その他の要因で最低2倍程度の電力悪化が見込まれる
  - 2018年のプロセス:30nm世代→15nm世代のプロセス進歩は見込めるため、約2倍程度の電力効率改善が見込まれる
- **15nm世代で30pJ/flopが最低限必要なエネルギーと予想**
  - 10pJ/flopとの乖離があり埋めるための努力が必要

24