「次世代計算基盤に係る調査研究」システム調査研究・神戸大学
「オリジナルアクセラレータアーキテクチャによる超低消費電力次世代計算基盤の評価」

牧野淳一郎 神戸大学

第17回アクセラレーション技術発表討論会 2023/9/28

まとめ(1/2)

- ポスト富岳の目標として、商用展開できること、そのために世界一の (競合よりだいぶ上の) 電力性能、価格性能比を実現するべきである。
- そのためには、GPU アーキテクチャが性能の限界にきていることを認識し、その問題を解決できるポスト GPU アーキテクチャである必要がある。
- そのためには、3D DRAM を使用し、オンチップで分散メモリになっている必要がある。
- 神戸大学-PFN チームではそういう方向の検討を進めている。

まとめ(2/2)

- (ここの数字は非公式だけど例えば) アクセラレータ 3 万チップ、10EF、汎用 CPU 部分 500PF(但し A64fx の問題点はない、 Zen4c くらいの IPC で動くもの)、30MW。
- アクセラレータ部分のプログラミング環境はアセンブラでなければ OpenACC と OpenCL 方言。レジスタが無限にあるのでコンパイラによる最適化はわりとできる。

発表構成

- 調査研究の概要
- 我々の検討の概要
- 今年度の取組方針
- 次世代計算基盤の目指すべき方向

調査研究の概要

次世代計算基盤に係る調査研究

令和4年度予算額　　4.3億円（新規）

背景

- ◆ **データ駆動型科学が重要視される中で、シミュレーションやAI 等が連携した研究の重要性がより一層高まっている。**さらに、**世界的にも研究活動のデジタルトランスフォーメーション（研究DX）の必要性が高まっている。**
- ◆ スーパーコンピュータのみならず、データセンターからエッジコンピューティング、それらを繋ぐネットワーク等、様々な形態の社会情報基盤がますます重要となっており、また、これらの**基幹技術を自国で保有することは経済安全保障の観点からも重要**である。
- ◆ これらの情勢を踏まえると、**ポスト「富岳」時代の次世代計算基盤を、国として戦略的に整備することは必要不可欠**である。

次世代計算基盤検討部会　中間まとめ（令和３年８月）

◆ **次世代計算基盤検討の留意事項**

技術動向や周辺状況が急速に進化・変化

ムーアの法則の終焉等、関連技術が転換期にある、
性能の向上に伴い要求される電力量も増大

⇒　半導体やネットワーク等**国内外の周辺技術動向や利用側のニーズの調査**、**要素技術の研究開発**等必要な調査研究を行い、**多角的な検討**が必要。

◆ **次世代計算基盤の在り方**

次期「フラッグシップシステム」及び国内の主要な計算基盤、データ基盤、ネットワークが一体的に運用され、総体として持続的に機能する基盤

⇒　**調査研究（FS）を通じ**、技術的課題や制約要因を抽出しつつ、**実現可能なシステム等の選択肢を提案**

<ポスト「富岳」時代に目指すべき姿>

次世代計算基盤に係る調査研究

◆ 具体的には以下の取組を実施。

- **要素技術の研究開発**（併せて、我が国として独自に開発・維持するべき技術を特定）
- **評価指標**の検討（例：演算性能、電力性能比、I/O性能、コスト、運用可能性、生産性（アプリ開発のしやすさ）、商用展開・技術展開、カーボンニュートラルへの対応　等）
- **技術的課題や制約要因**の抽出　　等

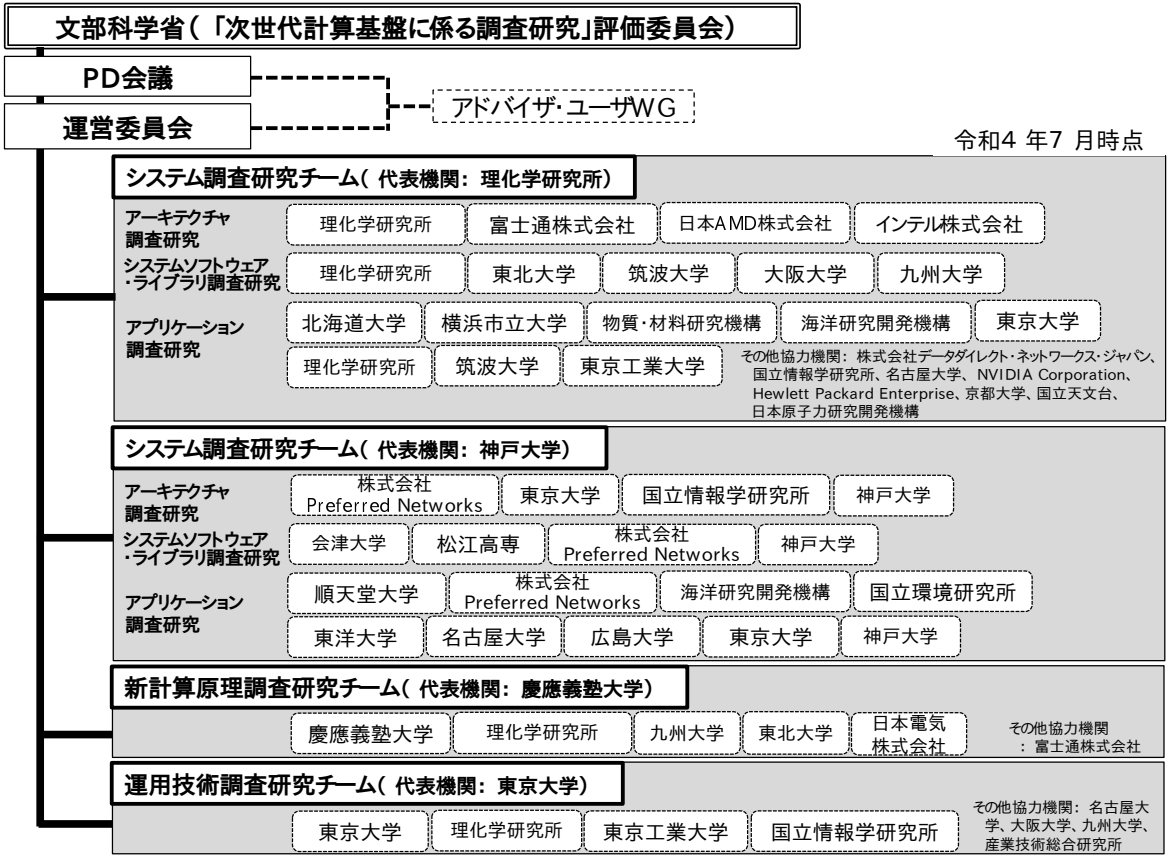
◆ 実施期間：令和4年度～令和5年度　※令和6年度以降の取組は、調査研究の進捗を踏まえ検討

令和５年度の取組：システム候補の性能評価、アプリケーションのコデザイン、新たな計算原理を適用すべき領域・分野の検討、多様な計算基盤の一体的運用、これらにおいて必要な要素技術の研究開発　等

令和４年度の取組：技術や利用分野の動向調査、評価項目・手法の検討　等

<FS実施体制（概略）>

「次世代計算基盤に係る調査研究」実施体制



調査研究の概要

本調査研究では PFN・神戸大学の持つ

- 世界最高性能のアクセラレータ実現技術
- AI 応用フレームワーク/アプリケーションソフトウェア技術

を最大限に活用しつつ、従来から重要な HPC アプリケーションと AI 利用等新しい応用の双方について高い実行効率を実現できるシステムの構成・評価をアプリケーショングループとの密接な協力によって実現する

調査研究の内容

PFN と神戸大学が開発するアクセラレータを中心に複数のハードウェアアーキテクチャの組合せを、典型的かつ重要なアプリケーションタイプを対象に評価

- **アーキテクチャ調査研究: 神戸大学・PFN が開発する MN-Core X とそれに適合した CPU による省電力化・効率改善に重点**
- **システムソフトウェア・ライブラリ調査研究: DSL・フレームワークによる高効率コードの自動生成に重点 (深層学習分野のアプローチを他分野へも)**
- **アプリケーション調査研究: これまでの伝統的 HPC アプリの他、AI 応用、商用ソフトウェアに重点 (新規: 生成 AI、LLM、マルチモーダル基盤モデルにも取組む)**

各サブグループの今年度の方針

- アーキテクチャ調査研究
- システムソフトウェア・ライブラリ調査研究:
- アプリケーション調査研究

アーキテクチャ調査研究

今年度目標:
（国立大学法人神戸大学）商用プロセッサについて、2028-2030 年までのコストパフォーマンス、電力性能を予測すると共に、独自アクセラレータアーキテクチャについて、前年度定めたレファレンス設計についてシミュレーションを行い、N5 ないし N3 以降のプロセスルールでの動作速度・電力性能、アプリケーションカーネルでの性能を評価する。

- **アクセラレータ評価**
- **CPU 評価**
- **ネットワーク評価**

アクセラレータ評価

独自アクセラレータアーキテクチャについて、昨年度定めたレファレンス設計についてシミュレーションを行い、 N5 ないし N3 以降 のプロセスルールでの動作速度・電力性能、アプリケーションカーネルでの性能を評価する。
LLM・生成 AI についても高い電力あたり性能・価格あたり性能の実現を目指したアーキテクチャ検討を進める。(PFN としては必須のアプリケーションであり、社として検討している)

メモリ階層と消費電力推定の詳細化中である。アプリケーション性能に対してもっとも大きな影響をもつのは DRAM の実装方式であるため、この部分の検討
を進めている。

CPU評価

RISC-V 等による独自実装の汎用 CPU および商用 CPU の評価のため、RISC-V アーキテクチャ設計のシミュレーションによるアプリケーション実行時性能評価を行う。

現在の状況:
アプリケーショングループと協力して、アクセラレータにのらない部分のうちボトルネックになると思われる部分の抽出を進めている。また、RISC-V アーキテクチャについて、ベンダと協力して評価を始めている。

ネットワークアーキテクチャ評価

今年度目標:
ネットワークアーキテクチャについて、レファレンス設計のアプリケーション性能に与える影響をシミュレーションないしは通信量の詳細な推定に基づいて評価する。

現在の状況:
アプリケーショングループと協力して、必要ネットワーク性能の評価を進めている。

- **想定するノード構成では fat tree 等の直径が小さくバイセクションバンド幅が高いネットワークが望ましい**
- **トーラスは大域通信 (放送・縮約・ alltoall) の性能、特にレイテンシが非常に悪いため**

システムソフトウェア・ライブラリ調査研究

目標:
前年度に定めた独自アクセラレータのレファレンスアーキテクチャ、アプリケーションを記述可能なデータ並列言語のプロトタイプに対して、アプリケーション性能を評価する。また、粒子系、構造格子系、非構造格子向けフレームワーク/DSL についてもアプリケーション性能を評価する。

現状:
OpenCL、 OpenACC の実装を始めている。OpenCL については、いくつかの制限と、MN-Core チップ内ネットワークを有効に使うための拡張を検討中である。
DSL について、粒子系、構造格子系について単純な実装はできている。
相互作用カーネルについて最適化コンパイラの検討を進めた。MN-Core アーキテクチャに対して人手によるアセンブラ最適化と同等の性能が得られており、他への適用を検討している。

アプリケーション調査研究

対象アプリケーションに対して、信頼できるレベルの性能推定を行う。このため、カーネル部分について実装ないし信頼できるサイクルカウントモデルを構築し、これによる推定を行う。

現状
創薬・宇宙の粒子系アプリケーションでは、GPU 向け実装の現行 MN-Core での性能評価を行い、問題点を明らかにした。改良を検討する。
ゲノミクスについては、高速な商用実装がでてきており、その評価が可能かどうか検討中である。
他のアプリケーションではカーネル部分の抽出等を進めている。

ポスト富岳を進める上での基本的な考え方(であるべきと牧野が考えるもの)

- 「京」・富岳の考え方(公開情報による)
- この方向の問題点
- 本来目指すべき方向

「京」・富岳の考え方(公開情報による)

2014/12/8 テクニカルコンピューティングソリューション事業本部長 (当時) 山田昌彦氏インタビューより (https://www.hpcwire.jp/archives/6305)

> 私どもからすると、工程を開発と製造に分けてみた場合、開発の部分は国とメーカーでシェアするけれど、製造部分については製造原価は100%国が見るべきだと考えています。この部分だけはキープして欲しいというのが、一貫した考え方です。これが原則にならないと、今回のフラッグシップマシンの開発も次のステップに進んでいくことが難しいのではないかと。

つまり、富士通の外むけ見解

- 開発費用は一部社が負担
- 製造費用は 100% 国であるべき
- ((暗黙の？) 主張: 「京」はそうなってなかったのが問題である)

(暗黙の)前提？

- 日本のスパコン開発は国の援助が必要である。
- これを「開発プロジェクト」として行う、つまり、ある程度のマシンを構築・納入する場合、開発費用を一部国が負担しているわけだから納入価格はそれをを割り引いたものであるべき
- とはいえ、製造費用以下まで割り引くのはメーカーとしては難しい

別の制約: 同じ費用で外国から買ってきたほうがよりよいのでは意味がないのでは？

「京」・富岳の価格

マシン	時期	構成	価格	億円/PF
「京」	2012	SPARC64 VIIIfx 88128 ノード	600?	55
東大 FX10	2012	SPARC64 IXfx 4800 ノード	48	42
富岳	2020	A64fx 158,976 ノード	700?	1.3
JAXA FX1000	2020	A64fx 5760 ノード	61	3.1

- 「京」、FX10 はピーク性能あたりで同時期の Sandy Bridge ベースの機械の 2 倍くらいした
- 富岳ははありえないほど安く納入されている
- Frontier: 600M USD, Summit: 200M USD と比べると性能あたりでは高価
- FX1000 の販売価格は AMD EPYC ベースシステムより性能あたりで高価

「京」・富岳は価格競走力がない=商用展開は困難

本来目指すべき方向

そもそもなんのために「国策スパコン」を開発するのか？
「次世代計算基盤検討部会中間取りまとめ」から

科学技術の各研究分野からの研究ニーズに応え、世界最高水準の性能を有し、それが無ければ実現し得ない卓越した研究成果を創出するとともに、計算科学・計算機科学の技術と人材を維持・育成し社会情報基盤をけん引する役割を担う科学技術・学術情報基盤を我が国で開発・運用、製造・活用できる力を経済安全保障の観点からも確保する。また、これらの結果として、新たな科学技術の創出、産業競争力の強化、Society 5.0 の実現、国民の安心・安全の確保等の社会的課題の解決に貢献する。

重要と思われるポイント

1. 世界最高水準の性能
2. 科学技術・学術情報基盤を我が国で開発・運用、製造・活用できる力を確保
3. 産業競争力の強化に貢献

これらの実現には何が必要か？

世界最高水準の性能 を (おそらく) 与えられた予算と電力制約の中で実現する

予算内で高い性能を実現するアプローチ

× 赤字をだしても頑張る

○ 価格性能比 (と電力性能比) が高いものを開発する

これらの実現には何が必要か？(続き)

科学技術・学術情報基盤を我が国で開発・運用、製造・活用できる力を確保

- 一台大きなマシンができればいいわけじゃなくて、開発したマシンと同じモデルが色々なところで導入・利用されてはじめて「科学技術・学術情報基盤」といえる。
- 外国製プロセッサより安くて速いのでなければ誰も買ってくれない。

産業競争力の強化に貢献

- 外国製プロセッサより安くて速いのができればそれ自体が産業競争力だし、それを使うことも産業競争力の強化に貢献できる。

要するに

- 完成時点で世界一の価格あたり(と電力あたり)性能を実現すれば大体の目標は達成できる「はず」
- 逆に、価格あたり(と電力あたり)性能が世界一レベルでないと目標達成は困難。
- 実際、「京」、「富岳」と同一アーキテクチャのマシンはほとんど普及しなかった。
- もちろん、ある程度広い範囲のアプリケーションで実際に競合より高い性能でなければならない
- コストを下げるための種と仕掛けが必要

何故「京」・富岳は高価なのか？

根本的な理由: 半導体技術の進歩が有効に利用されるために必要となったアーキテクチャ変化に遅れている (「京」)、逆行している (富岳)

「ベルの法則」(ゴードン・ベル賞のベル)

Roughly every decade a new, lower priced computer class forms based on a new programming platform, network, and interface resulting in new usage and the establishment of a new industry.

大体 10 年くらいでアーキテクチャが変わる

スパコンだと

- 1976 年まで: スカラー計算機。最後は CDC 7600
- 1976 年から 1992 年まで: ベクトル (共有メモリ並列含む) 計算機。Cray-1 から C-90 まで。
- 1993 年から 2008 年まで: (マルチコア含む) マイクロプロセッサの分散メモリ並列計算 Cray T3D から、、、Red Storm あたりまで。
- 2008 年から: GPU アーキテクチャのアクセラレータとマイクロプロセッサの組合せ: IBM RoardRunner (Cell なので GPU じゃないけど)

大体 15 年毎。GPU の次はまだでてきてない。

アーキテクチャの変化が必要な理由

基本的な理由: そのアーキテクチャでは、半導体の進歩を有効に利用できなくなる

- **半導体技術の変化**
- **アーキテクチャのスケーラビリティの限界**

スカラー計算機→ベクトル計算機

- スカラー計算機の進歩: 増えるトランジスタを「1 つの演算器を速くする」に
- 主記憶は磁気コアで、半導体よりはるかに遅い
- CDC 7600 (S. Cray の設計) あたりが最後
- S. Cray はこの後、4 プロセッサ並列の CDC 8600 を開発していたが、完成前に CDC やめて Cray Research を設立。ベクトルの Cray-1 を開発する。

スカラー計算機では

- IC の開発で使えるゲート数が 1 演算器を超えて大きくなった
- 半導体メモリの開発でメモリが非常に高速になった

ことを生かせなかった。
ベクトル: 半導体メモリは有効利用できた。

ベクトル計算機→マイクロプロセッサ超並列

- ベクトル計算機の進歩: 1 プロセッサのパイプライン数や主記憶を共有するプロセッサの数を増やす。
- メモリもプロセッサも多数の IC から構成されて、沢山内部配線があるので、その間の配線も沢山あっても大丈夫、が前提。
- Cray-1 くらいのプロセッサが 1 チップにはいると話が破綻。
- 1 プロセッサチップと少数のメモリチップで 1 ノードを構成し、その間はネットワークでつなぐ構成が有利になる。
- メモリを演算器より遅くしないと価格性能比が下がる (memory wall)。
- 初期の代表: Cray T3D。

マイクロプロセッサ超並列→GPU

- 1チップの中に沢山プロセッサがはいるようになる。
- これは実はシステムレベルでのベクトル並列プロセッサの限界と同じ。
- マルチ(メニー)コアプロセッサでは、キャッシュコヒーレンシを維持した階層キャッシュとオフチップの共有メモリ。
- キャッシュコヒーレンシの維持のための電力が支配的になる。
- GPU では(OS 動かしたりしないので)キャッシュコヒーレンシを緩和したり、捨てたりできる。
- あと、巨大なレジスタファイルとマルチスレッド化でメモリレイテンシを隠蔽、要求バンド幅も多少は減らせる。

GPU→???

- コヒーレンシを諦めても、オフチップメモリや階層キャッシュで物理的に遠くにデータ移動させること自体が性能の制限になる。
- と書いた以上、階層キャッシュを諦めて、なるべく物理的に遠くにデータ移動させないようにすることが必要になるはず。
- PEZY-SC(キャッシュもあるけど)、Sunway SC26010、MN-Core 等では演算器の物理的に近くに大容量メモリ配置。

何故横方向のデータ移動が制約になるか？

配線遅延・配線消費電力の問題
微細化しても、「単位長さ辺りのキャパシタンス」はかわらない、ところが、「単位長さあたりの抵抗」は配線幅の二乗に反比例して増える:

- **配線長がスケールしても配線遅延は「デザインルールに反比例して」増える**
- **配線長一定だと遅延は二乗で増える。電力は減らない**

(最近だと、トランジスタ密度は上がっても配線はそんなに細くならないので配線が短くなれば遅延・電力が減る。長いと電力も遅延も減らない)

数字をいれてみると

- 10cm ($10^5 \mu\mathrm{m}$) の配線は 20pF のキャパシタンスがある。これは微細化してもかわらない (線と平面の間のキャパシタンス) 電圧 1V だと、この線は 10pJ/bit を消費する。
- DDR5 メモリは 20pJ/bit くらい。電圧 1.3V とか。。
- LPDDR5 と GDDR6x: 10 pJ/bit くらい。配線が DDR5 のモジュールより短くて電圧も低い。
- HBMx: 3-4 pJ/bit。概ね 25mm くらいまで配線短くなっている。

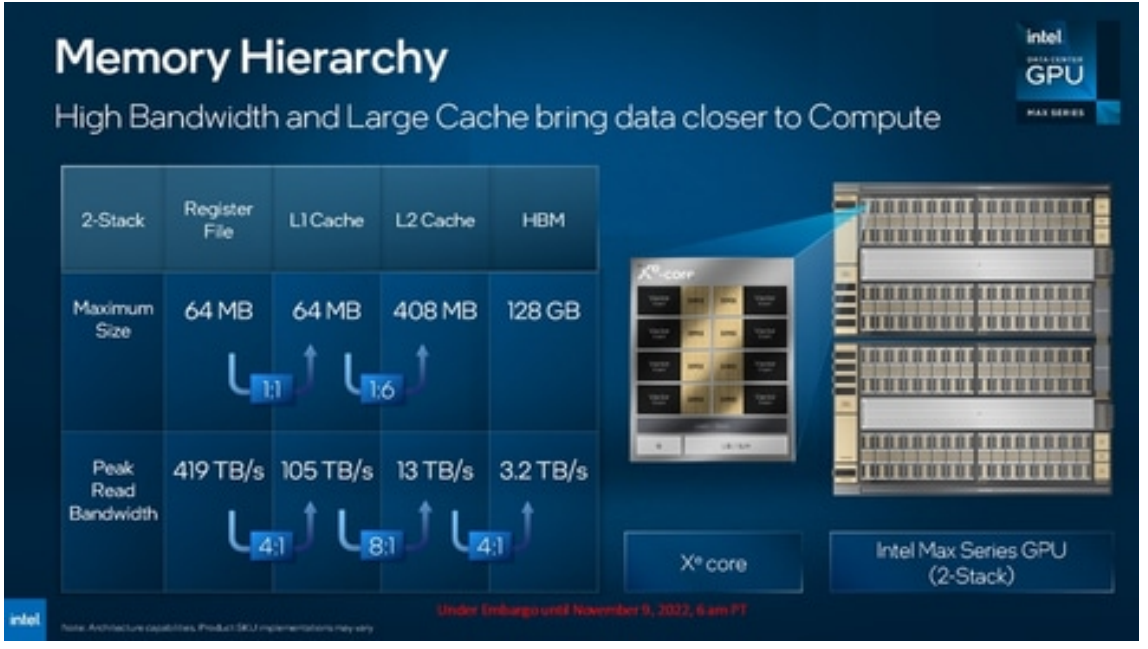
3pJ/bit は 64 ビットワード移動に 0.2nJ なので、HBM でも 5GW 移動すると 1J 消費する。 B/F=4 の計算機作ると 10GF/W しかでない。 0.1 にしても DRAM メモリだけで 400GF/W までしかいかないのでキャッシュの分とかいれると 100GF/W あたり。ポスト富岳は 500GF/W は必要、、、

つまり

GPU の次のアーキテクチャ に移行するためには、「データのチップ上・基板上の横方向の移動」を限りなく小さくする必要がある。

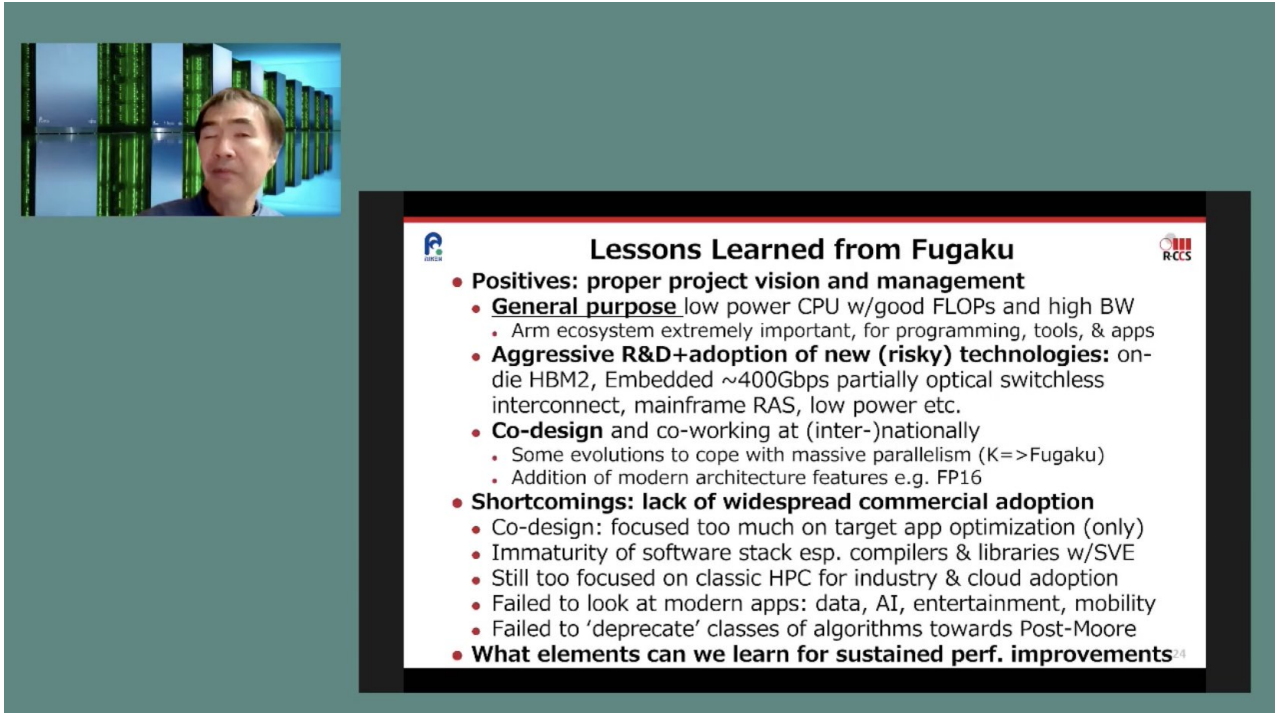
シリコンフォトニクスでとかいう話があるが、今のところ光変換のコストのほうが大きい

Intel GPU の例



B/F の値
レジスタ: 8? (普通 16 いる。行列乗算器だと 8 はありえる)
L1: 2
L2: 0.25
メモリ 0.06
A64fx に比べても数字が小さい
それでも電力消費が大きい原因になっている

ちょっと余談: A64fx の問題点復習



SIAMPP22

「昔のアプリケーションに最適化されすぎてモダンなアプリケーションに対応できてない」

とはいえ要するに:
「実用アプリケーションで性能でてないじゃないか」

「京」との比較の例
— 「昔の」アプリケーションならどうか

2022-09-16 HPC-Phys 第 16 回勉強会計算科学ロードマップの基礎科学分野の更新状況: 宇宙惑星から

- **PIC コード:「京」で 15%あった実行効率が数%まで低下、ループ分割等によって 6%まで改善。**
- **AMR 流体: キャッシュのレイテンシが大きく、バンド幅も小さいため効率が低下している (Intel Xeon Phi に比べて)**
- **格子流体: レジスタ不足のためキャッシュアクセスレイテンシの隠蔽が困難**

「京」ではそこそこの実行効率がでていたコードの効率が大きく低下。チューニングがんばっても戻らない。

A64fx 固有の問題

Skylake Xeon とのレイテンシ等の比較

	A64fx	Skylake	京
FP 演算レイテンシ	9	4	6
L1 レイテンシ	8-11	4	-
L2 レイテンシ	37-47	14	-
L3 レイテンシ	-	50-70	-
命令ウインドゥ	128	224	-
ロード/ストア	どちらか	同時実行	-
FP 論理レジスタ	32	32	256
FP 物理レジスタ	128	168	256

命令レイテンシが２－３倍、命令ウインドゥが半分、バンド幅も半分くらい

↓

Skylake では実行時にスケジューリングしてパイプラインを埋められるループでもリソース不足でパイプラインが埋まらない

レイテンシ大きくなった理由の推測: 低消費電力化のため低電圧で動作する低速・ドライブ能力の低いトランジスタを主に使ったので、動作周波数を保つためには深いパイプラインが必要になった。

「京」との比較

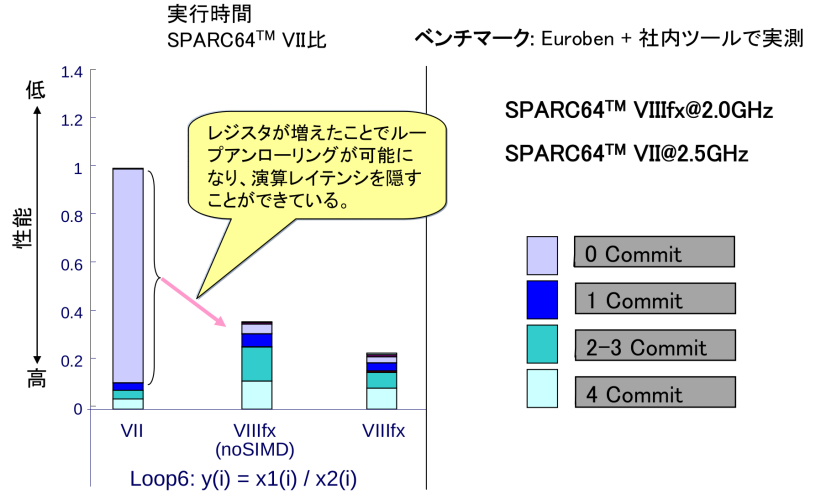
- 演算レイテンシが 1.5 倍
- 物理レジスタ数が半分、論理レジスタ数が 1/8

「京」では非常に有効だったループアンロール、ソフトウェアパイプライニングによる最適化では十分な性能を実現できなくなった。

レジスタ数と性能

レジスタ拡張の効果

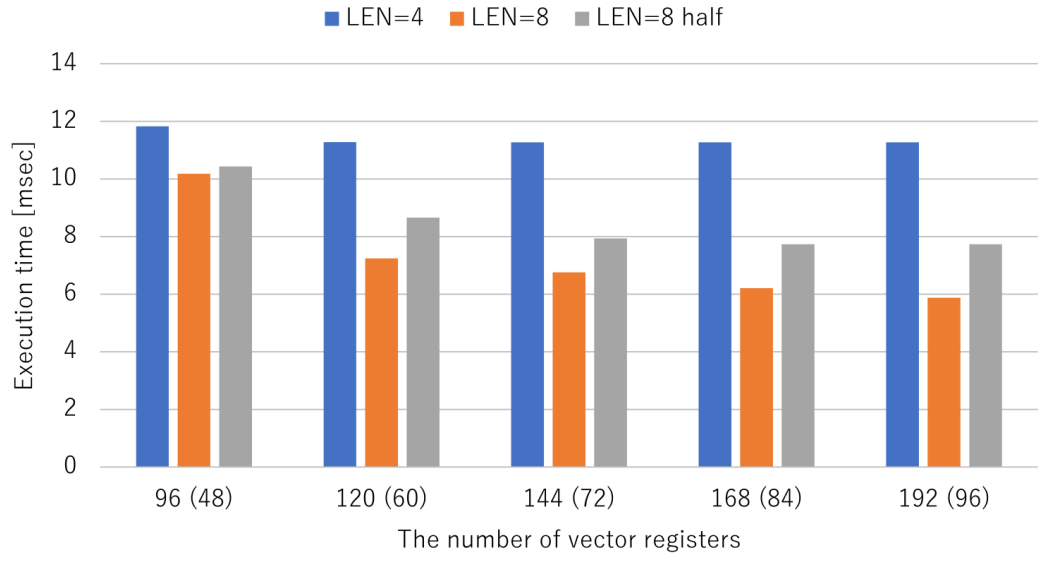
FUJITSU



「SPARC64 VIIIfx 富士通次期スーパーコンピュータプロセサ
2010年1月28日
富士通株式会社
次世代テクニカルコンピューティング開発本部
山下 英男」
スライド27

理研側の認識

公開資料から言える範囲で、、、



(演算器等のレイテンシは A64fx のよりずっと小さい)

Odajima, Kodama and Sato 2021, J Supercomput 77, 5757–5778 (2021).
元々は Cool Chips 21 (2018/5) で発表=2017 年の研究
SVE 命令を 1024 ビット幅にして 2 サイクルで処理するほうが「物理レジスタの面積が同じでも」性能がずっと上がる

理研は問題を認識していた

一般的問題

ワイド SIMD 命令セットをもつ共有キャッシュアーキテクチャメニーコア共通の問題

ほとんどのアプリケーションで実行効率が低い

(密行列乗算が演算カーネルでないとなかなか性能でない、、、)

理由は沢山の色々なものの組合せだが、

- **メモリへのストライドアクセス、ランダムアクセスが相対的に非常に遅い**
- **キャッシュサイズが相対的に小さくなっていて、メモリアクセスを隠蔽できない**
- **メモリバンド幅が相対的に下がっている**

どうするべき(だった)か

- 商用展開には実用アプリケーションでの価格性能比、電力性能比で競合より優れていることが必須。
- 1990年代までと違って半導体技術における優位は失われていて、むしろ不利になっているので、それ以外の技術で性能の優位性を実現するべき。
- 価格性能比、電力性能比に直結できる日本独自技術が必要。

要するに

今回はポストGPUアーキテクチャを世界に先駆けて実現することが必須

ポストGPUアーキテクチャ

ポストGPUアーキテクチャでは

- チップ内部の演算器が物理的に共有するオフチップメモリはもたない
- チップ内部の演算器が物理的に共有するキャッシュはもたない

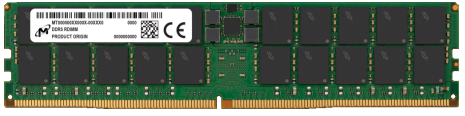
必要がある。これは1990年代前半の共有メモリベクトル並列から分散メモリ超並列への移行と同じ。

つまり、チップ内で分散メモリ超並列を実現すればよい。

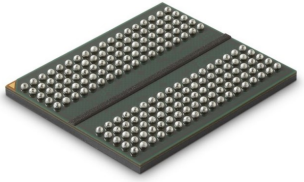
但し、普通にオフチップDRAMをつけたのでは駄目(バンド幅足りない)

普通でないオフチップDRAMは？

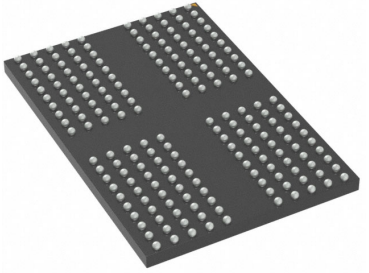
メモリアーキテクチャの変化



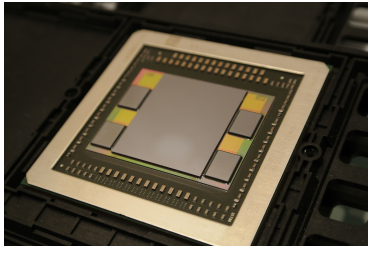
DDR(2000〜)



GDDR(2003〜)



LPDDR(2008〜)



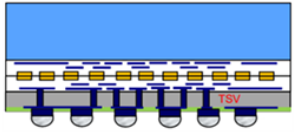
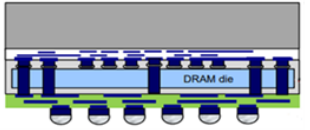
HBM(2015〜)

- DRAM セル構造、動作速度とかはあんまり変わってない
- 大きく変わったもの: インターフェース部分の速度、電圧、物理的な構造
- 電力を食うのは DRAM メモリセルではなく、チップ間配線のドライバと配線そのものになっている。(前に書いた $1\mathrm{pJ/bit \cdot cm}$)

HBM の次は？

- HBM は平面的実装としてはほぼ究極。プロセッサダイのすぐ隣に DRAM ダイがある。
- それでも、電力消費はプロセッサダイと DRAM ダイの中を横に動く分、下のインターポーザの中を横に動く分が大きい。
- これらを減らすには、DRAM ダイをプロセッサダイの上にのせて、さらに面的な配線で演算器とその直上 (下？) の DRAM メモリブロックを接続すればよい。

現状の例: AP Memory/Powerchip

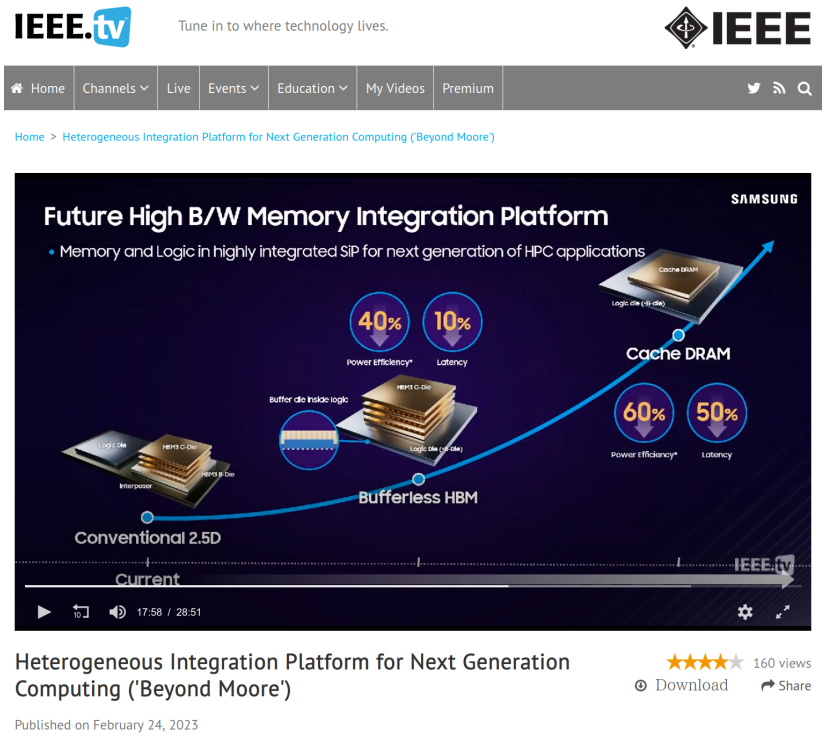
	WoW Hybrid Bonding	CoW uBump Bonding
		
Stacking partner	Front End	Back End
pitch	Pitch < 3um >>100K IO	Pitch ~ 40um 1-2K IO
Base technology	BSI CIS	HBM
Application	Supercomputing Mining Networking	CPU/GPU AI AR/VR

WoW Hybrid Bonding 自体は既に実績がある技術。この会社は現状 mining chip を作っている模様

0.7pJ/bit 程度

https://www.apmemory.com/ AP Memory 社のウェブサイトから

将来の例: Samsung



HBM に対して 40-60% の電力削減

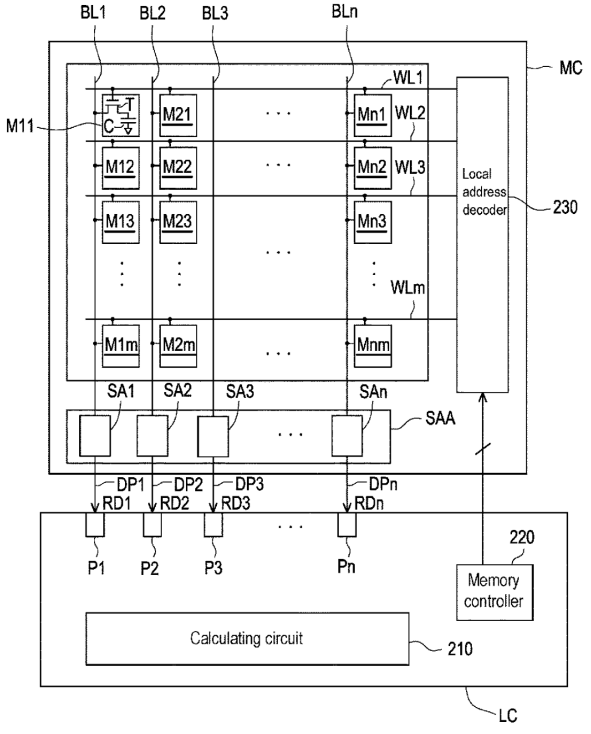
Cache DRAM で 1pJ/bit 程度

(現状の HBMx は 2.5-3pJ/bit 程度)

原理的な限界は？

- HBM 等でも、 DRAM 側の現状の設計は DDRx の DRAM と同等。クロックがはいり、FIFO やセレクタを通してデータがでてくる。部分書き込みのためのデータ保持回路等もある。
- DRAM チップ内でみても、実際の DRAM セルやセンスアンプの消費電力は結構小さい (O’Connor et al. Fine-Grained DRAM: Energy-Efficient DRAM for Extreme Bandwidth Systems, MICRO-50)
- ハイブリッドボンディングでは非常に大きなパッド数が可能であるので、回路構成から再検討が可能。

PSMC 特許(出願中)



PSMC 特許 US20230186972A1
Memory device with high data bandwidth

これは多分一番極端な回路構成案。センスアンプから先をすべてプロセッサダイ側にもってくる。

(特許申請自体は DRAM をロジックダイにのせますと書いてあるだけなのでこのまま成立はしなさそう)

実装上の懸案事項

一杯ある。

- HBM みたいに DRAM スタックできるし、 TSV middle とかですごく沢山パッドだせるんだけど、DRAM スタック上につんで冷却できるか？
- face up でロジックダイ使って電源とか信号線引き出せるか？
- DRAM スタック下においたら、DRAM スタック通して信号とか電源だせるか？

とはいえ、解決できない問題という感じでもない。物理的に金属配線でつながるならまあなんとかなる。

MN-Core アーキテクチャと 3D DRAM

- MN-Core アーキテクチャはチップ全体が SIMD アレイ。それぞれの PE がローカルメモリをもって、階層的なネットワークで結合。
- 3D DRAM は、基本的には各 PE に DRAM ブロックがつく。例えば 16MB とか。
- 80 年代風の超並列 SIMD プロセッサとほぼ同じ。ネットワークはさぼっている。概ね同じようにプログラミングもできる。
- つまり: ほぼ HPF (OpenACC ともいう) でプログラムできる。(但し、DRAM とオンチップメモリの違いを意識する必要はある)

MIMD 超並列できないの？

- 私の頭では無理なのでもっと賢い人が考えればいいと思う。
- Sunway みたいな、オフチップ DRAM に iCache だけあるアーキテクチャはありえるかもしれない。
- Sunway の使う側からみたメリット: OpenCL も OpenACC も××でも低レベルのスレッドライブラリ使えばその辺のオーバーヘッドは避けられる。コンパイラありますとはいえる。
- 作る側からみた MIMD のデメリット: 大変である。ハードウェアあがってこないとアプリケーション性能わからない。命令セットによる制約が無数にはいってきて無駄にハードウェアが複雑になる。

ポスト富岳システムの「初期イメージ」

これは個人としての発言で FS チームとしての公式のものではありません。

- N2 での NVIDIA GPU の予想性能: 173TF/800mm^2, 163GF/W。
- 従って: 我々の目標: 面接あたり性能、電力あたり性能ともに最低でこの 3 倍。
- 例えば、300TF/400mm^2, 500GF/W。これだと 600W。
- 3D DRAM は容量 xxx GB(秘密)、バンド幅 20-30TB/s とか(B/F =0.1 あたり)。
- これと、可能なら同一パッケージで CPU。 1TB/s くらいで接続。CPU フルに動くと 300W くらい。50GF/W として 15TF。B/F=0.1 として 1.5TB/s。これは LPDDR で作れる。
- この演算ノードはさらに (初期案としては)8 ノード程度が x86 のホストに 100GB/s 程度 (あるいは相互にその程度) でつながる。 x86 はネットワークとファイル I/O の面倒を見る。

ポスト富岳システムの「初期イメージ」(続き)

- 10EF とすると 3 万チップ。ネットワークノードは 1/8 なので 4000 ノードとか。これくらいだと 1.6Tb/s くらいでもまあ予算的にはいるかも？
- 消費電力最大 30MW。
- CPU コアは 256bit SIMD ユニット 4 個とか (1024 x 1 でもいいと思うけど、、、) で 4GHz 動作、1 コア 128GF 128 コアとか。AMD Zen4c のちょっと先くらい。そこまで FP 性能いらないかもと思うと 256bit x 2 ユニット。Zen4c そのもの。

まとめ(1/2)

- ポスト富岳の目標として、商用展開できること、そのために世界一の (競合よりだいぶ上の) 電力性能、価格性能比を実現するべきである。
- そのためには、GPU アーキテクチャが性能の限界にきていることを認識し、その問題を解決できるポスト GPU アーキテクチャである必要がある。
- そのためには、3D DRAM を使用し、オンチップで分散メモリになっている必要がある。
- 神戸大学-PFN チームではそういう方向の検討を進めている。

まとめ(2/2)

- (ここの数字は非公式だけど例えば) アクセラレータ 3 万チップ、10EF、汎用 CPU 部分 500PF(但し A64fx の問題点はない、 Zen4c くらいの IPC で動くもの)、30MW。
- アクセラレータ部分のプログラミング環境はアセンブラでなければ OpenACC と OpenCL 方言。レジスタが無限にあるのでコンパイラによる最適化はわりとできる。