研究報告Ⅱ　システム研究調査の概要と検討状況

牧野 淳一郎
神戸大学 大学院 理学研究科

次世代計算基盤に係る調査研究に関する合同ワークショップ ～ 次世代高性能計算基盤の開発に向けて ～
（2023年2月22日）

概要

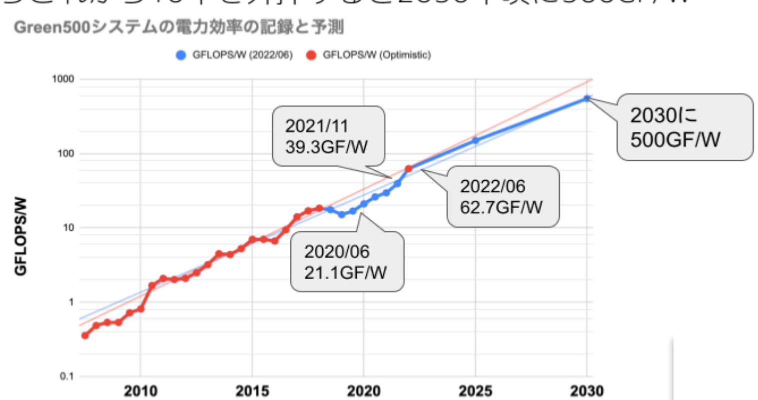
本調査研究では
- PFN・神戸大学の持つ
  - 世界最高性能のアクセラレータ実現技術
  - AI応用フレームワーク/アプリケーションソフトウェア技術

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
- 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

HPCトレンド:Green500システムの電力効率の予測

最近2年間 2020 21GF/W → 2021 40GF/W → 2022 60GF/W
- 過去10年からこれから10年を外挿すると2030年頃に500GF/W



本発表の構成

- 調査研究の実施体制
- 調査研究の概要
  - HPCシステムのトレンド
  - 日本の国家プロジェクトの経緯
  - 提案調査研究の概要
- 調査研究のスケジュール
- キーテクノロジー
- まとめ

調査研究の実施体制

研究代表機関
神戸大学

（研究分担機関）
◎Preferred Networks
東京大学
国立情報学研究所

A:アーキテクチャ調査研究グループ

（研究分担機関）
◎会津大学
松江高専
Preferred Networks

B:システムソフトウェア・ライブラリ調査研究グループ

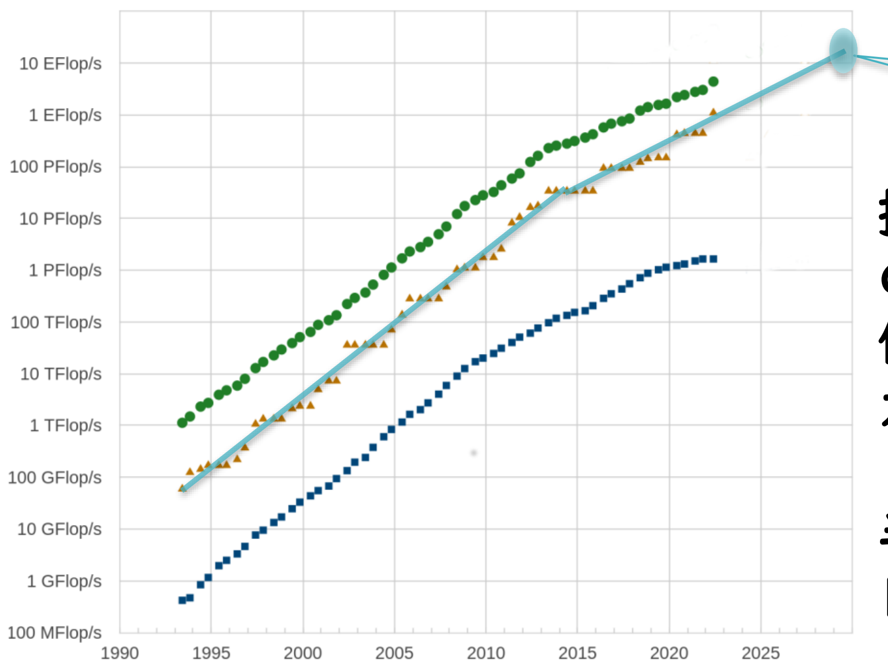
（研究分担機関）
◎順天堂大学
Preferred Networks
海洋研究開発機構
東洋大学
名古屋大学
広島大学
東京大学

C:アプリケーション調査研究グループ

提案調査研究の概要

- HPCシステムのトレンド
- 日本の国家プロジェクトの経緯
- 提案調査研究の概要

HPCシステムのトレンド



技術的にはかなり困難：半導体技術の進歩だけによる低消費電力化は4倍程度。設計の革新による低消費電力化が必須

半導体技術の進歩だけでは10EF程度にとどまる可能性もある

演算性能よりも大きな問題：実効性能、使いやすさ
多くのHPCアプリケーションの実行効率:5-10%

ーピーク演算性能だけ高くてもダメ

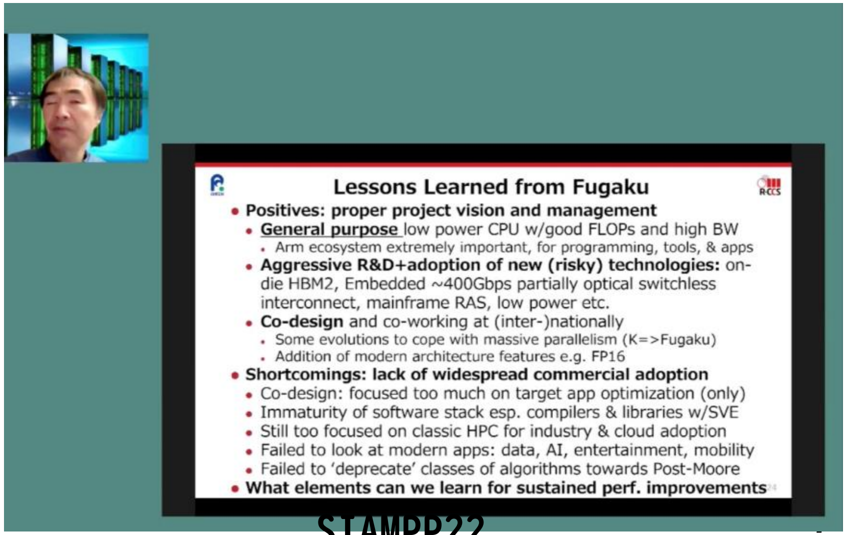
アプリケーションの新しい方向：AIの活用、低精度演算の利用
アプリケーションが何を実現できるかのほうが重要

日本の国家プロジェクトの経緯

システム	完成年	ピーク性能	消費電力	性能向上（対1世代前）	電力性能向上（対1世代前）
数値風洞	1993	280GF	1MW	-	-
地球シミュレータ	2002	40TF	6MW	143	24
京	2011	11PF	15MW	275	110
富岳	2020	500PF	35MW	45	19

富岳はエクサスケールシステムで唯一ホモジニアスメニーコア
次世代システムは何を目指すべきか？
京、富岳の教訓は？
富岳の大きな問題：
- 多くのアプリで実行効率が低い
- ピーク性能に対して高価

→商用展開が難しい
（長所：汎用CPUとしては高い電力性能）

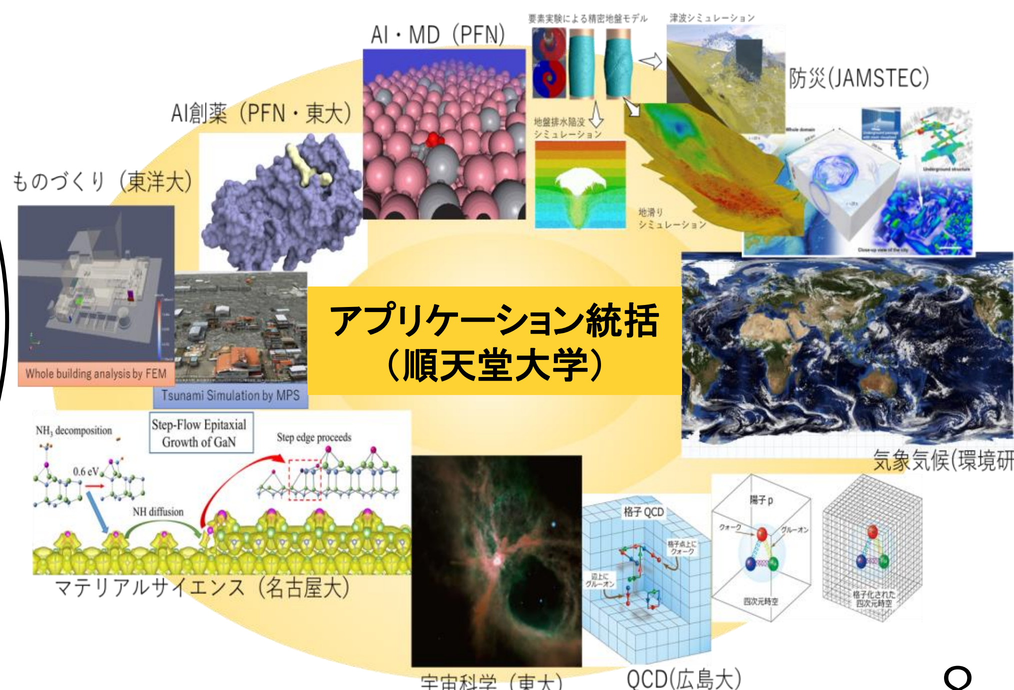


SIAMPP22

本調査研究の目標

- 国内で開発されるアクセラレータを含む複数のハードウェアアーキテクチャの組合せに対して
  - 演算性能、電力性能比、I/O性能、コスト、生産性、商用展開・技術展開等の指標を
  - 典型的かつ重要と考えられるアプリケーションタイプを対象に評価することで、総合的にハードウェアアーキテクチャを評価

CPU・アクセラレータの最適な組み合わせを評価



検討対象アプリケーション分野と担当機関

アーキテクチャ調査研究グループ

- ノードプロセッサアーキテクチャ
- ネットワークアーキテクチャ
- I/O サブシステムアーキテクチャ

のそれぞれについて、

- 分野自体の技術トレンド、
- 半導体プロセス技術動向からポスト富岳時代に実現可能なシステム

がどのようなものかを検討し、評価対象アプリケーションによるシステム評価を行う

ノードプロセッサアーキテクチャ：

汎用 CPU	アクセラレータ
商用 x86/arm	商用 NVIDIA/AMD/Intel
RISC-V	独自設計（MN-Core 後継）
RISC-V 独自設計 （キャッシュ構造 見直し）	

日本の半導体技術国家戦略と歩調を合わせ、日本が保有すべき技術を検討

システムソフトウェア・ライブラリ調査研究グループ

OS、コンパイラ、ファイルシステム、ライブラリ、フレームワーク、言語等およびそれに関連する技術分野が対象

現状の大きな問題：
実行効率の低下

- 階層キャッシュの特性とアプリケーションが必要とするメモリアクセスパターンの乖離
- キャッシュに最適化されたコード生成の困難さ
- システムごとに違うキャッシュ構成

本調査研究でのアプローチ

- 階層コヒーレントキャッシュのシステム
- ローカルメモリ的なものをもつシステム

の双方に対して高効率な実行コードを、アルゴリズム・数値スキームの高レベルの記述から自動生成するDSL/フレームワークを重視

◆ 深層学習では高レベルDSL/フレームワークが一般的なアプローチになった：TensorFlow, PyTorch, JAX
◆ アプリケーション開発者が個別アーキテクチャ向けのコーディング、チューニングする必要がなくなっている

(PFN は Chainer/Pytorch、CuPy 等で実績)

アプリケーション調査研究グループ

以下の分野で重要なアプリケーション、アプリケーションカーネルの抽出、次世代システム向けアルゴリズムの検討と性能評価を合わせて行う

- 創薬/MD/AI シミュレーション応用 PFN 岡野原・前田
- ゲノム科学 東京大学 鎌谷
- 防災・減災(地震・津波) JAMSTEC 堀
- 防災・減災(気象・気候) 環境研 八代
- ものづくり(CAE) 東洋大学 塩谷
- 物性/材料 名大 押山
- 基礎科学(素核) 広島大学 石川
- 基礎科学 (宇宙) 東大 藤井
- OSS/商用アプリケーション

AI 応用については PFN 内で様々な分野の実績がある
OSS/商用アプリについても十分な検討を行う
(例:GROMACS LAMMPS OpenFOAM FrontISTR等)

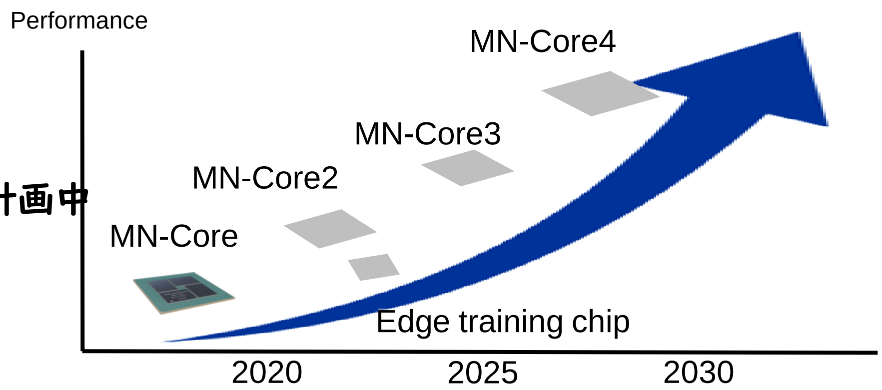
世界最高水準の性能の実現方法

汎用プロセッサ側:
RISC-V ベースの独自設計で世界最高レベルの性能を目指す。
このために
- ベクトル命令
- SMT
- プロセッサローカルメモリ
- プロセッサ間通信ネットワーク

を検討する

アクセラレータ側:
現時点で同じ半導体技術では世界最高の電力当り性能を実現している
MN-Core をベースに 一層高い電力性能、チップ面積あたり性能を目指すと共に、アプリケーションとのコデザイン により高い実行効率を実現する

MN-Core	MN-3システムとして稼働中
MN-Core2	現在ES評価中
MN-Core3	現在設計中
MN-Core4	2028～2030完成を目指して計画中



調査研究のスケジュール

アーキテクチャ調査研究グループ

事業開始-12月	現行 CPU/GPU のトレンド調査・独自アーキテクチャ検討
1-3月	商用プロセッサの将来予測、独自アーキテクチャ暫定版策定
2023	商用および独自アーキテクチャの性能・電力評価

システムソフトウェア・ライブラリ調査研究グループ

事業開始-12月	DSL/フレームワーク仕様検討
1-3月	プロトタイプ実装開発
2023	プロトタイプ実装開発継続およびアプリケーションカーネル評価

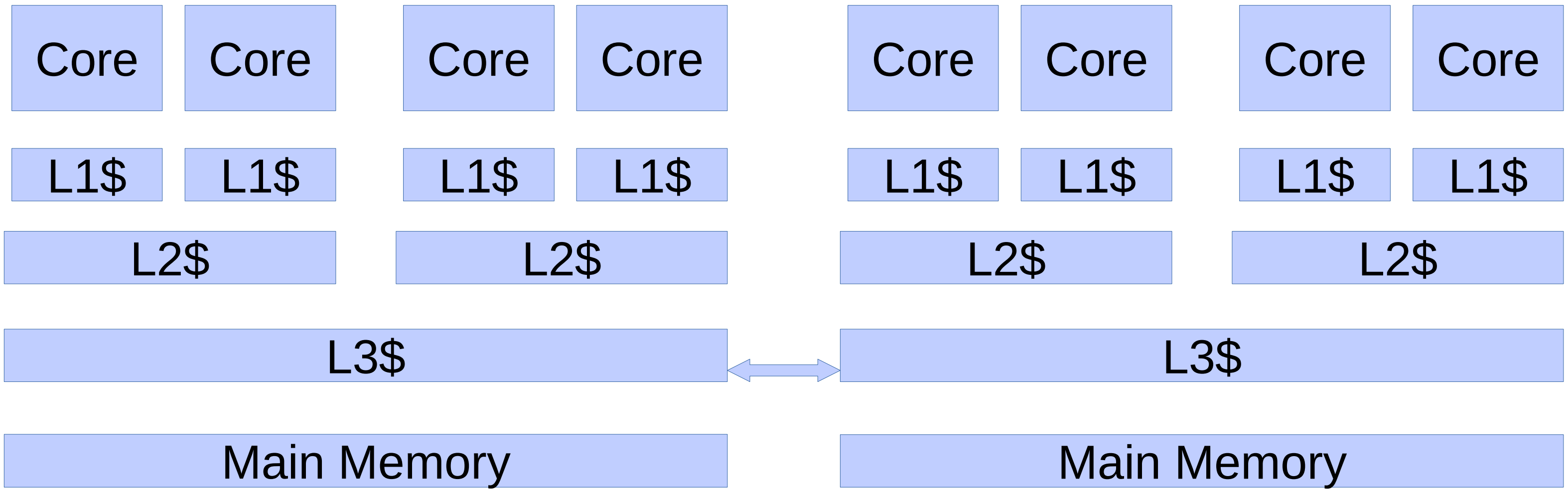
アプリケーション調査研究グループ

事業開始-12月	商用および独自コードのアルゴリズム調査
1-3月	商用および独自コードのカーネル抽出、ヘテロジニアスアーキテクチャでの性能評価
2023	商用および独自コードのアルゴリズム最適化と性能評価

キーテクノロジーとその評価

- アーキテクチャのキーテクノロジー
  - メモリ階層
  - メモリ実装
- システムソフトウェアのキーテクノロジー
  - 既存HPCミドルウェアと深層学習むけ
  - フレームワークの融合

• アーキテクチャのキーテクノロジー：メモリ階層



現在の典型的なマルチコアCPUの構成（GPGPUも一見あまりかわらない）

下にいくほど多数のコアに共有される

- 下にいくほど容量大きい
- コア数が段々増えた「進化」の結果

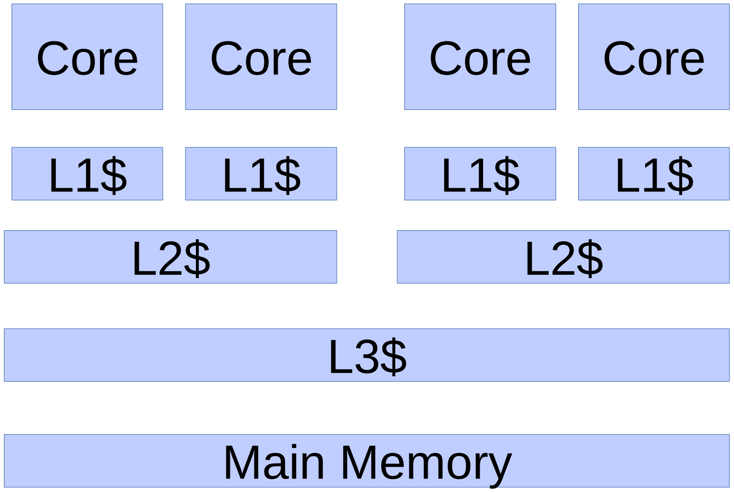
これは「一番よい」やりかたなのか？

アーキテクトからみたメモリ階層

L1って64kBくらいだよね？
L2は256KB? 512KB?

L3 大きいとアプリケーションの性能良くなるのはベンチマークででてるから、とにかくL3大きくしないといけないみたい

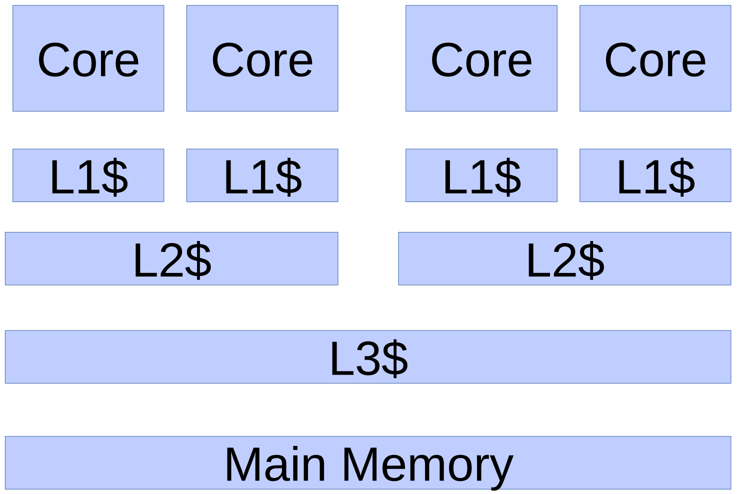
L3は共有だよね？LLCだし



アプリ側からみたメモリ階層

どうすればうまくキャッシュのデータ
再利用できる？

この２重ループそのままではのらない
から、ブロック化してちょっとづつ
L2にもってきて、さらにブロック化
してL１に、、、



ブロック化が非常にうまくいく例：行列乗算
　L１、L2、L3とブロックサイズ大きくする。必要バンド幅が下がる
それ以外のほとんどすべてのアプリケーション：
　階層キャッシュがあまり有効ではない。
　「L3には乗るけどバンド幅が」「レイテンシが」

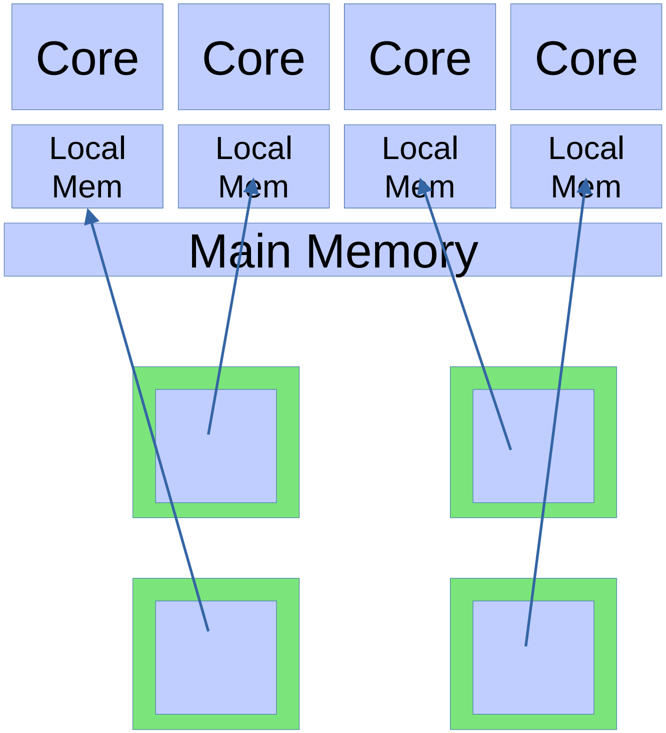
• （プログラム書く手間を考えなければ）アプリ側の希望

差分法の例

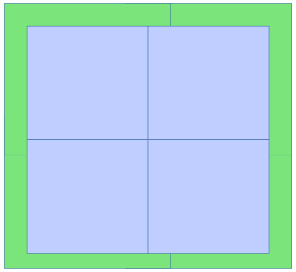
あるループでは各コアはそれぞれ
この領域を分担させたい

必要なデータを袖領域もいれて Local Memory にもってくる

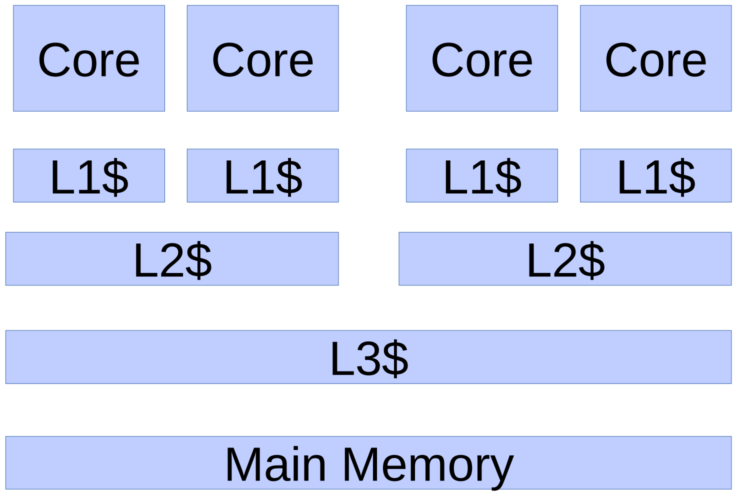
袖交換すれば Main Memory に戻さないで
計算継続できる



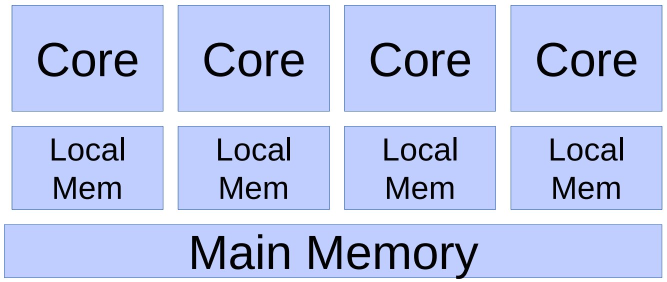
要するに：
- B/Fの高いメモリがなるべくたくさんあってほしい
- プログラム書くのはいやだけど、どうデータを動かしたいかはわかってる
- プログラム全部書き直すのはいや



• アーキテクトからみると

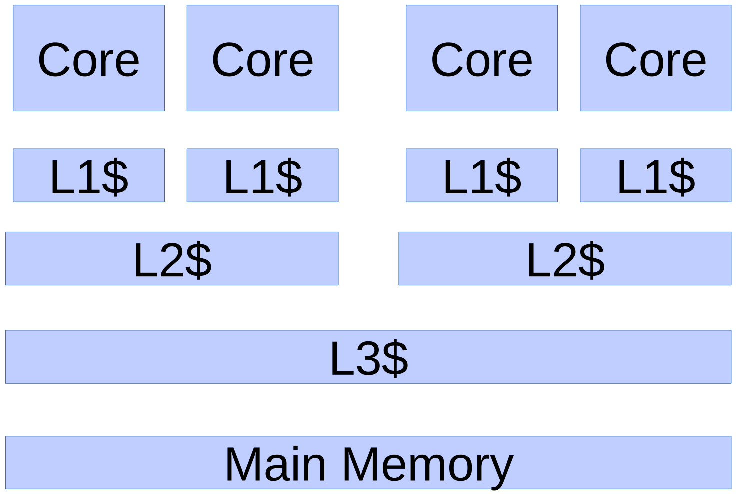


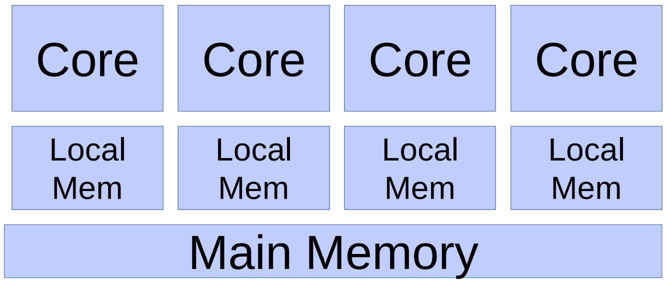
- とりあえずいままでのアーキテクチャの延長
- 作るのすごい大変だけどしょうがない
- これまでのプログラム動く
- コア数とかSIMD幅が広くなると段々効率落ちるけどそういうもの
- 電力性能も悪いけどキャッシュはそういうもの



- 作るの難しくないしこれでいいならそれは楽
- でも、アプリそのままではうごかないしだれも使ってくれないんじゃ？
- 前例あるけどみんな失敗じゃない？ CDC 7600, Cray-2 Sony/IBM Cell
- 最近だと Sunway SW26010。すごく使いにくいという噂だし

- これまでの歴史（２０１５年くらいまで）

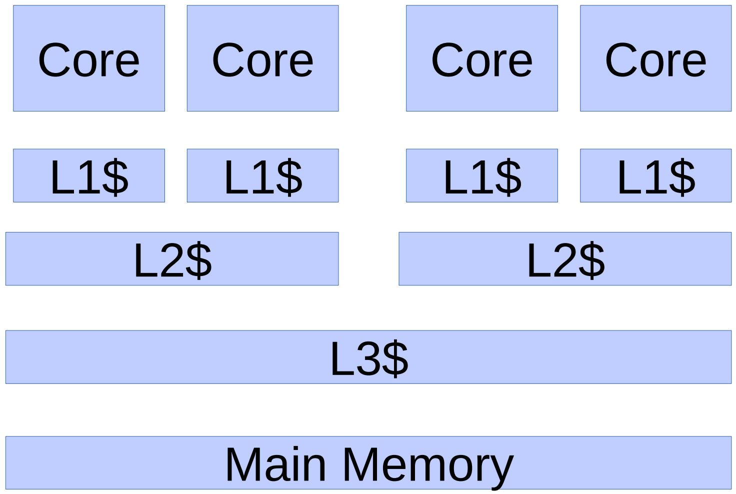




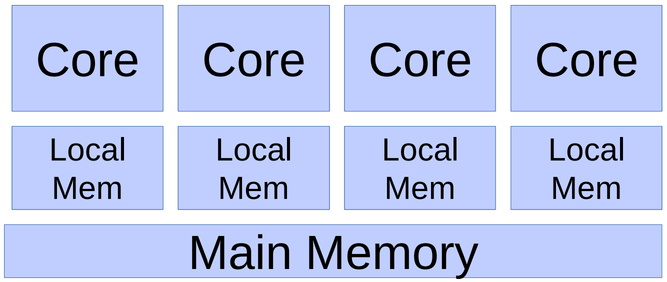
- Intel x86が典型
- 最先端のプロセスで次々に新しい世代がでる
- ムーアの法則でどんどん性能上がる
- いままでのプログラム動くし速くもなるのでちゃんとみんな買ってくれる

- あるプロジェクトで採用されても、アプリケーションがそろうまでに時間がかかる
- その間に x86 においつかれる
- 後継プロジェクトがキャンセルされる

• 今後は？



- 2018年以降、CPUの電力性能の上昇は極端に遅くなった
- デスクトップ用CPUもサーバー用も消費電力がどんどん上がっている。サーバー用の最近のは 400W。デスクトップも 230Wとか
- これで Green500 トップレベルのスパコンを、はもう無理



- SW26010 は後継が開発された。1EF以上をそこそこ高い電力性能で実現したと想像される
- PEZY-SC も3世代に渡って開発
- GPU は巨大なレジスタファイルが物理的にはローカルメモリ。なのでローカルメモリもキャッシュもある構成
- MN-Core は全体SIMDでキャッシュもない

- **比較**

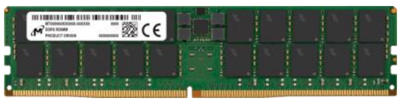
	MN-Core	Sunway	PEZY	GPU (A100)
SIMD/MIMD	SIMD	MIMD	MIMD	SIMDブロック
キャッシュ	なし	なし	あり	あり
コア間通信	ツリー	ハイパークロスバー	キャッシュ経由	キャッシュ経由

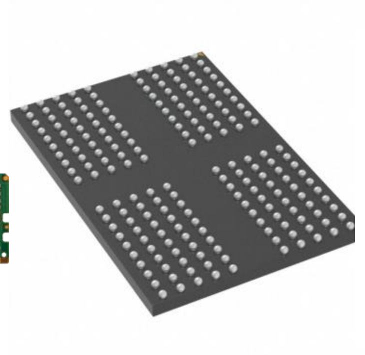
（プログラム書けたとして）ローカルメモリ構成のメリット

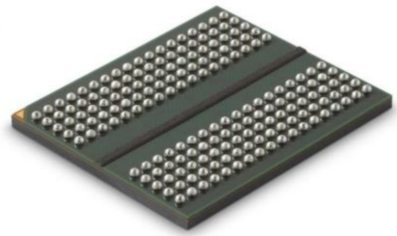
- 共有キャッシュまでの長い配線がないので電力性能をあげやすい
- 行列乗算以外の多様なアプリケーションでローカルメモリを有効につかって実行効率を高くできる
- コア間通信（同期・放送・縮約等を含めて）を導入すれば、キャッシュ経由の通信よりずっと並列化オーバーヘッドを小さくできる＝強スケーリング性能が高い
- ハードウェア開発、検証はキャッシュありのシステムに比べるとはるかにかんたん

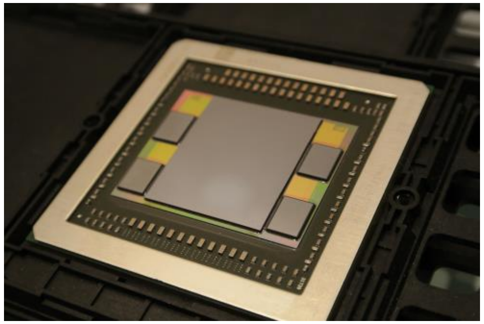
（プログラム書けるなら）キャッシュに比べてよいことしかない
どうやってプログラム書くのか＝システムソフト・コンパイラのキーテクノロジー

メモリ実装









DDRx
2000～

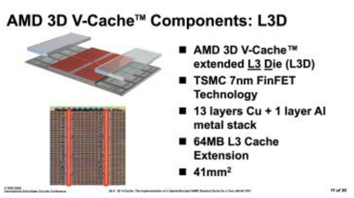
LPDDRx
2008～

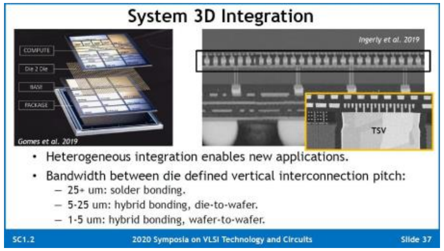
GDDRx
2003～

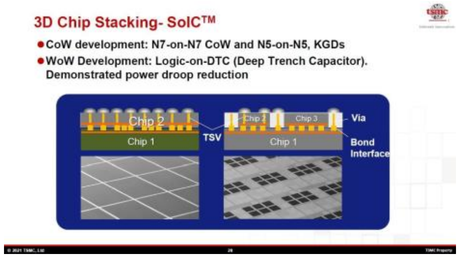
HBMx
2015～

HBMのメリット:短い、多数の配線:高バンド幅・低消費電力
デメリット:横方向の配線がそれでも長い。TSVでの積層で高価、シリコンサブストレートや高度な実装技術でやはり高価

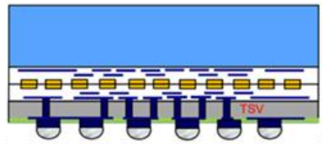
さらに高バンド幅、低消費電力を実現する方法=3D積層

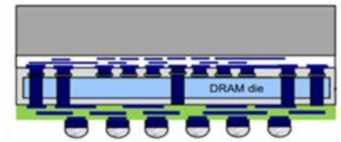






3D実装技術





	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

HB: Sony CMOSセンサで使われている非常に高密度な接合。2枚貼り合わせ。
　ロジック+メモリ等は各社開発中。電力的には有利。
マイクロバンプ+TSV: HBMで実用に使われている。ピッチに限界=高クロックが必要、電力増加

• システムソフトウェアのキーテクノロジー

既存HPCミドルウェアと深層学習むけフレームワークの融合

（ややこしいハードウェア向けの）プログラムを「書かない」こと：
深層学習分野ではすでに実現している。

- ネットワークは PyTorch, TensorFlow, JAX といったフレームワークで書く
- それらのフレームワークが各オペレーションを個別のハードウェアで高速に実行できるランタイムをもっている
- CPUでもGPUでもMN-Coreでも同じPyTorch のコードが高い実行効率で動く

HPCでも同じ考え方が使えるところはある

- 規則格子差分法とかは普通に書ける。ただし、現在のランタイムの実装はそのままでは性能はあんまりでないこともある
- OpenACC（あるいは F90の配列表現）と NumPy, PyTorch は同じようなことができる。
- 有限要素法とかも意外に書ける

• システムソフトウェアのキーテクノロジー

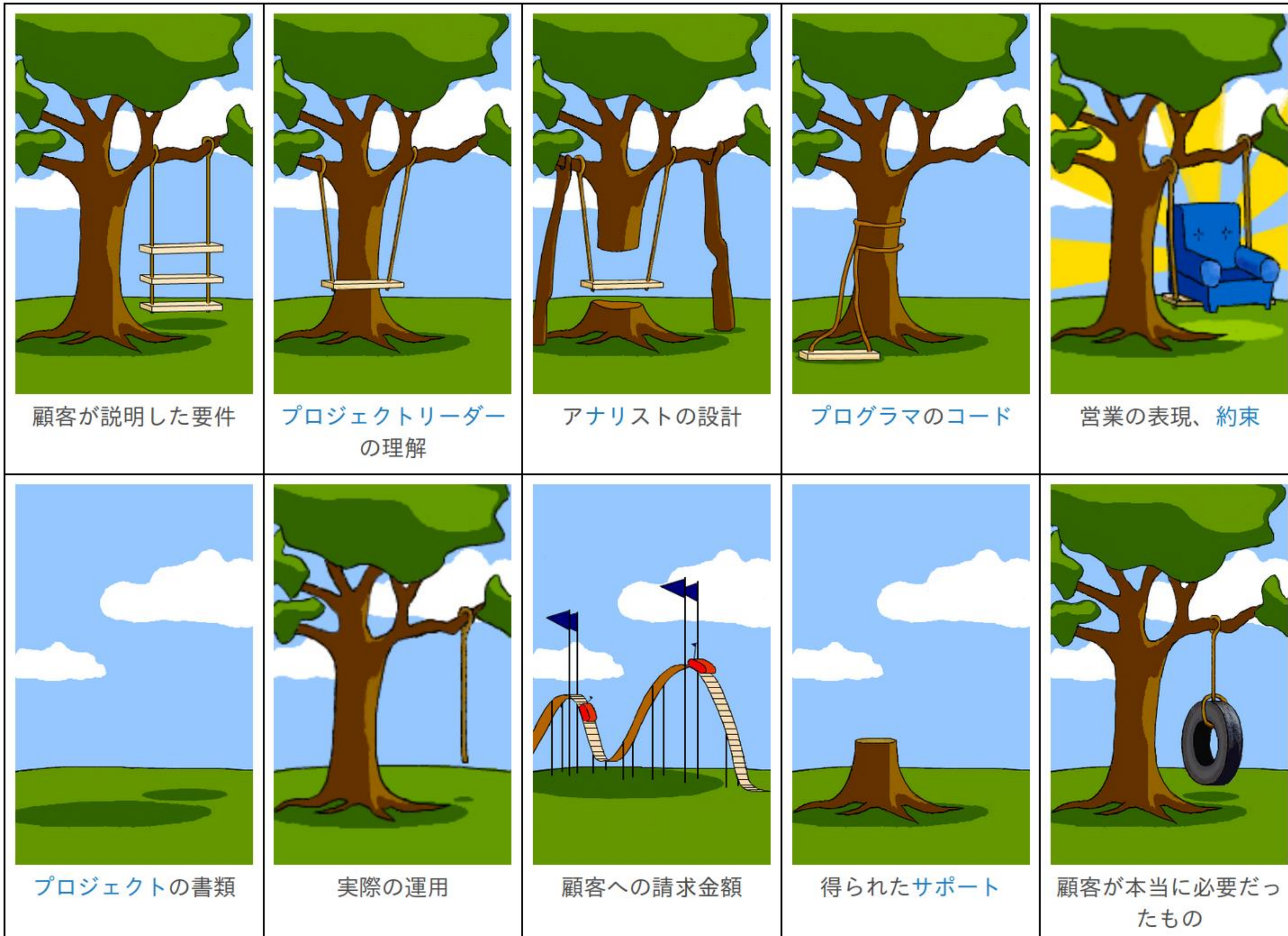
前のスライドは深層学習むけフレームワークそのものを使うという方向

別の方向：同じような考え方でフレームワークを作る

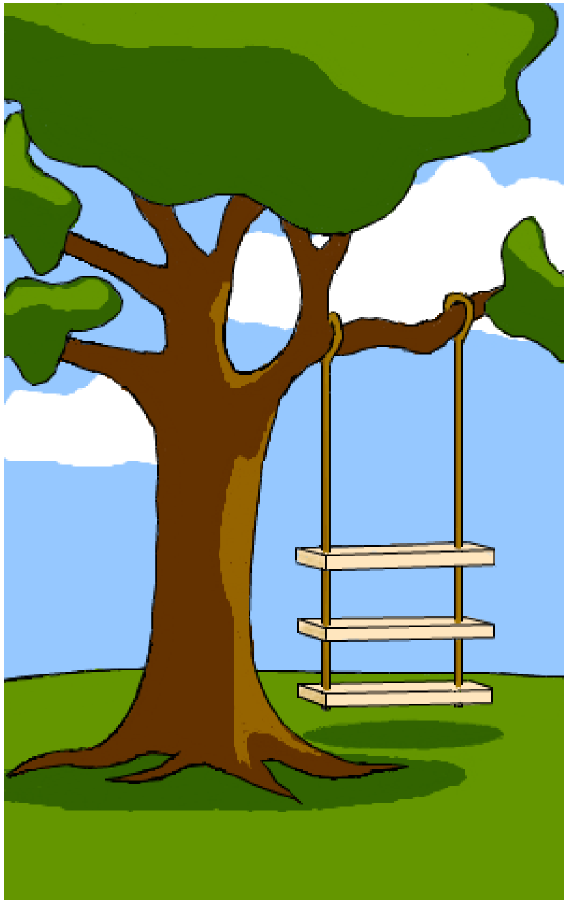
- **粒子系：相互作用の式と時間積分公式を与えるとあとはよしなにとか**
- **理研粒子系チーム(2021まで)で開発したFDPSはそれに近い。相互作用の高レベル表現から、動的負荷分散してMPIで並列に相互作用計算するコードがでる。CPUでもアクセラレータでも動く**
- **量子化学等だと必要なのは密行列ライブラリ：わりとはじめから何でも動く**

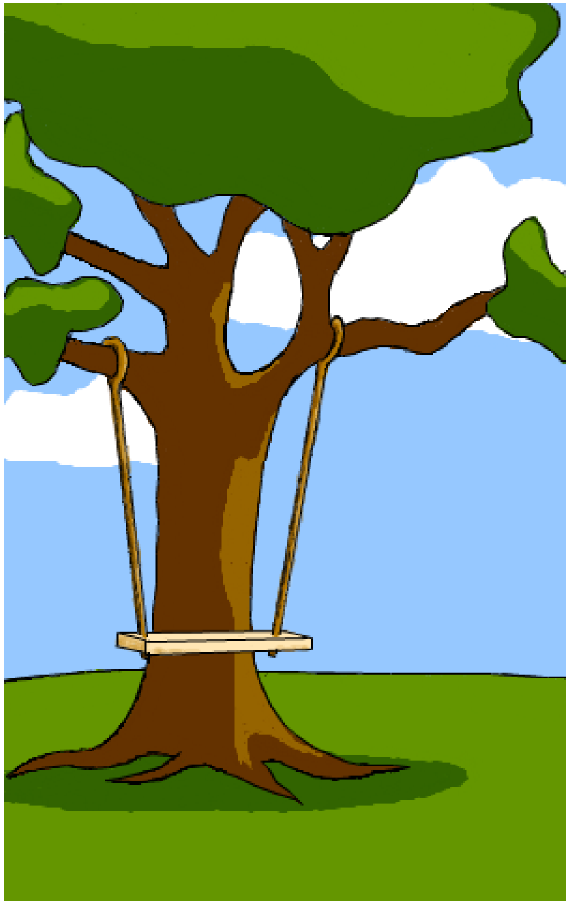
「CPU向けに書いたコードはアクセラレータにもってくのは大変と考えるのではなくて、このさいフレームワークやDSLや各種ライブラリを使ってマルチプラットフォームなコードにしよう、と考えるとしあわせになれるかも

- **アプリケーションサイドが本当に必要だったもの？**



• アプリケーションサイドが本当に必要だったもの？





こういうこともあるのでだまされない必要はある

まとめ

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
- 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

HPCトレンド:Green500システムの電力効率の予測

最近2年間 2020 21GF/W → 2021 40GF/W → 2022 60GF/W

- 過去10年からこれから10年を外挿すると2030年頃に500GF/W