エクサスケールコンピュータとその上での分子シミュレーション

牧野淳一郎

理化学研究所 計算科学研究機構

エクサスケールコンピューティング開発プロジェクト

コデザイン推進チーム チームリーダー

今日はどんな話をするか
—前置きあるいは言い訳

5月にアブストラクトを送った時に書いたこと

- MD (特にタンパクMDとか) では沢山 (10^9 とかもっと) ステップ回したい
- 汎用並列計算機では大きくしても1ステップは速くならない。1ミリ秒切るのは難しいし、アーキテクチャが大きく変わらないと将来的にも速くならない
- とかいってたら、専用機 ANTON は 20 マイクロ秒を実現しちゃった
- ポスト「京」ではなんか考えてるかというあたりを紹介したい。

今日はどんな話をするか
—前置きあるいは言い訳

5月にアブストラクトを送った時に書いたこと

- MD (特にタンパクMDとか) では沢山 (10^9 とかもっと) ステップ回したい
- 汎用並列計算機では大きくしても1ステップは速くならない。1ミリ秒切るのは難しいし、アーキテクチャが大きく変わらないと将来的にも速くならない
- とかいってたら、専用機 ANTON は 20 マイクロ秒を実現しちゃった
- ポスト「京」ではなんか考えてるかというあたりを紹介したい。

この時点ではまだ公式には考えるのを止めてはいなかった

しかし、、、

フラッグシップ-2020プロジェクト

文部科学省研究振興局参事官 (情報担当) 付計算科学技術推進室
ポスト「京」で重点的に取り組むべき社会的・科学的課題についての検討委員会（第４回）
配付資料 4-1 2014/7/24

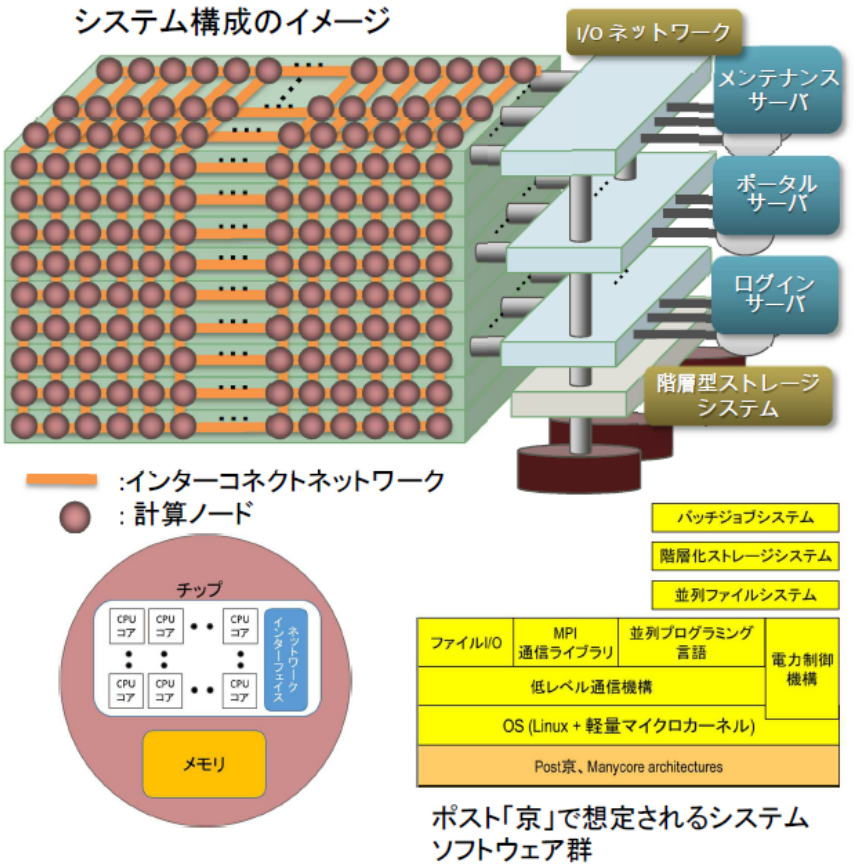
システムと開発の概要

＜システム構成＞

○CPU
- 汎用CPUを用いたメニーコア型アーキテクチャ
  →高い汎用性、幅広いアプリ実行に利点。
- ネットワークインターフェイスをチップ内に内蔵
  →高い通信性能、大規模並列処理に利点。

○ネットワーク構造
- 「京」の多次元トーラスネットワークトポロジを踏襲
  →高い移植性、京の資産を生かす利点。

＜要素技術開発の要点＞
○ 我が国が強みを持つコア技術は確保した上で、汎用品の活用や国際協力の推進により、効果的・効率的に開発。
○ 規格化を図ることによりユーザの利便性が高まるシステムソフトウェアは、米国と協力しながら開発。



ポスト「京」で想定されるシステムソフトウェア群

フラッグシップ-2020プロジェクト

○ＣＰＵ

- 汎用CPUを用いたメニーコア型アーキテクチャ
  →高い汎用性、幅広いアプリ実行に利点。
- ネットワークインターフェイスをチップ内に内蔵
  →高い通信性能、大規模並列処理に利点。

○ネットワーク構造

- 「京」の多次元トーラスネットワークトポロジを踏襲
  →高い移植性、京の資産を生かす利点。

つまり

「フラッグシップ-2020プロジェクト」で開発される計算機は

- **「京」、ポストFX10 の延長のメニーコア**
- **ネットワークも「京」、ポストFX10と同じ**

つまり

「フラッグシップ-2020プロジェクト」で開発される計算機は

- 「京」、ポストFX10 の延長のメニーコア
- ネットワークも「京」、ポストFX10と同じ

1年くらい前には違うことをいってたような気も、、、

「エクサスケール・スーパーコンピュータ開発プロジェクト」の概要

総合科学技術会議・評価専門調査会・ 第 103 回 2013/11/20 資料 6-2a

設計開発基本方針

課題：2020年から運用可能な高い性能電力比と幅広いアプリケーション実行環境を有するエクサスケールマシンの実現

計算機アーキテクチャ基本方針

汎用コアと**演算加速コア**を有するマシン
- ✓ 汎用コアでないと性能がだせないアプリ
- ✓ 演算加速コアで性能だせるアプリ
- ✓ 両コアを使って性能がだせるアプリ

を棲み分け、全電力時間積削減

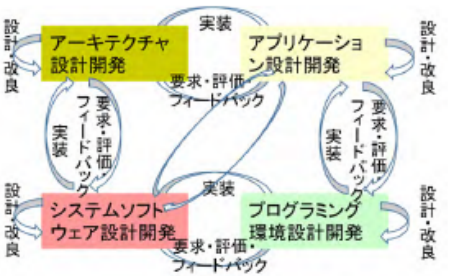
プログラミング環境設計基本方針

- ✓ 将来にわたって有効な統一プログラミングモデル、ライブラリ、フレームワークを提供
- ✓ 国際連携によるソフトウェア資産の国際的共有化促進

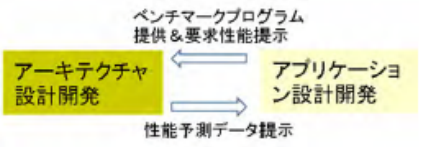
Co-design (協調設計)

- ✓ アプリケーションプログラムと計算機アーキテクチャの協調設計：演算性能・並列性能
- ✓ アプリケーションプログラムとプログラミング環境の協調設計：記述性・並列性能
- ✓ アプリケーションプログラムとシステムソフトウェアの協調設計：ファイルI/O性能・通信性能
- ✓ プログラミング環境とシステムソフトウェアの協調設計：ファイルI/O機構API・通信機構API
- ✓ 計算機アーキテクチャとシステムソフトウェアの協調設計：ファイルI/O性能・通信性能

Co-design



従来手法



続き

システム設計の基本的考え方

○将来動向

- 高い性能電力比と不規則非構造データ処理もできるCPUとして、汎用コアと演算加速コアが統合されていく。しかし、どのように統合されていくかは今後の研究成果に依存
- ノード単体の演算性能向上に比べ搭載可能メモリ容量はさほど増えない

○開発に関する考え方

- システム設計

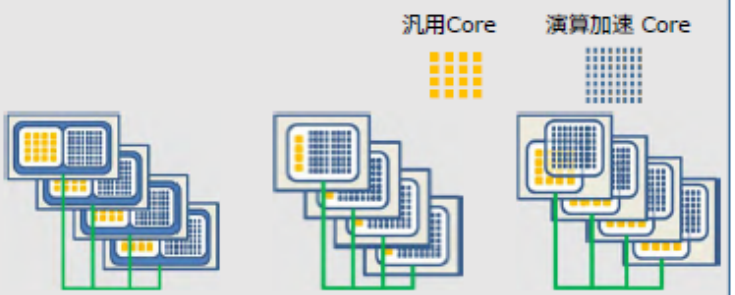
  汎用部（汎用コア）と**演算加速部（演算加速コア）**を有するマシン

  - 汎用コアでないと性能がだせないアプリケーション、演算加速コアで性能だせるアプリケーション、両コアを使って性能がだせるアプリケーションを棲み分け、全電力時間積削減
- 将来にわたって有効な統一プログラミングモデル、ライブラリ、フレームワークを提供
- 早期成果創出のためにキラーアプリケーションの同時開発を通してシステム設計に反映

	汎用コア (Server)	演算加速コア (GPU, SIMD)
データ構造	不規則・非構造	規則構造
性能電力比	低 Ex. 4.27GF/W, 22nm	高 Ex. 7GF/W, 28nm
メモリ階層の現状	キャッシュと主メモリ	GPU側とホスト側メモリ

将来マシンイメージ群

- 汎用コア、演算加速コア、ネットワークが統合されていく
- 統合方法および搭載メモリ容量とメモリバンド幅は、実装技術によって変わっていく
- 3D実装可能メモリ容量以上が必要な場合、低速メモリが導入される



去年の今頃は

汎用部 (汎用コア) と演算加速部 (演算加速コア) を有するマシン

- 汎用コアでないと性能がだせるないアプリケーション、演算加速コアで性能だせるアプリケーション、両コアを使って性能がだせるアプリケーションを棲み分け、全電力時間積削減

> 演算加速部: 単純なコアを多数並べ、大きなブロックで SIMD 動作、オンチップメモリを有効に使うことで高い電力性能とチップ間・コア間通信の低レイテンシ化を図る

と書いてあったけど、現在は
「汎用 CPU を用いたメニーコア型アーキテクチャ」

- 要するに、「演算加速部」がなくなった。
- 元々、MD の速い (1 タイムステップを速くできる) 実装はそっちでやるつもりだった

何故演算加速部はなくなったのか？

公式の説明

「次期フラッグシップシステムに係るシステム検討ワーキンググループ中間取りまとめ」

平成 26 年 7 月 4 日 HPCI 計画推進委員会 次期フラッグシップシステムに係るシステム検討 WG

◯“ 基本的なシステム構成 ”については，重点課題（案）が幅広い分野にわたり，幅広いアプリケーションが高い実行性能で利用できるシステムとする必要があることから，汎用部によるシステムを前提として，Co-design に基づく基本設計を進めることが提案されている。なお，安全・安心の実現や産業競争力の強化に資する課題では，総じて汎用部を中心とした利用が想定されることから，高い実行性能を得ることができるとの分析が示されている。

◯ その他，以下の提案理由も示されている。

加速部については，技術自体の実現可能性は認められるが，必要となる開発・製造経費に比して，有効活用できる課題が限定される可能性が高い。

何故演算加速部はなくなったのか？

公式の説明

「次期フラッグシップシステムに係るシステム検討ワーキンググループ中間取りまとめ」

平成 26 年 7 月 4 日 HPCI 計画推進委員会 次期フラッグシップシステムに係るシステム検討 WG

◯“ 基本的なシステム構成 ”については，重点課題（案）が幅広い分野にわたり，幅広いアプリケーションが高い実行性能で利用できるシステムとする必要があることから，汎用部によるシステムを前提として，Co-design に基づく基本設計を進めることが提案されている。なお，安全・安心の実現や産業競争力の強化に資する課題では，総じて汎用部を中心とした利用が想定されることから，高い実行性能を得ることができるとの分析が示されている。

◯ その他，以下の提案理由も示されている。

加速部については，技術自体の実現可能性は認められるが，必要となる開発・製造経費に比して，有効活用できる課題が限定される可能性が高い。

守秘義務に反することはここではいいません。

というわけで

ポスト「京」ではMDの高速化についてなんか考えてるかというあたりを紹介したかったわけですが、

というわけで

ポスト「京」ではMDの高速化についてなんか考えてるかというあたりを紹介したかったわけですが、

現状の「フラッグシップ-2020プロジェクト」では特になにも考えてません。

おしまい

というわけにも
いかないので

ではどうするか
という話を

今日の本題

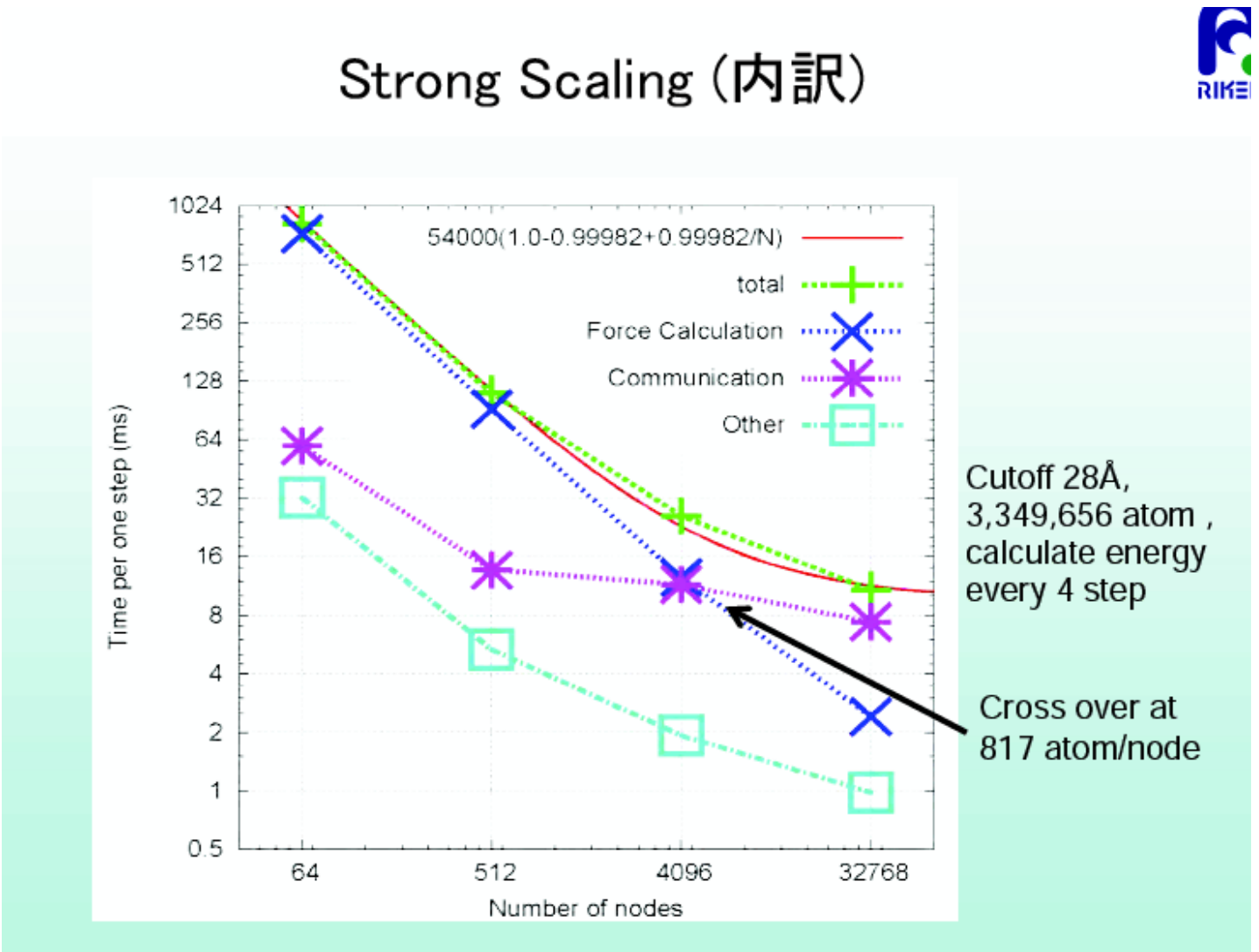
「これから 10 年くらいを想定して、MD 計算の高速化 (1 ステップを速くする方向) で何ができそうか？」
基本的な 3 つのアプローチ

1. **アルゴリズムをなんとかして計算量とか通信量自体を減らす**
2. **ハードウェアを有効に使うためのチューニング、コード書換えをする**
3. **ハードウェア自体をなんとかする**

話の構成

- 「京」での現状の検討
- ポスト「京」では？
- それ以外になんかあるか？

「京」の現状:性能スケーラビリティの例



(理研・大野さんの資料から)

「京」での分子動力学計算

- 1 タイムステップが 5ms を切らない
- 通信オーバーヘッドが問題

5ms で十分速いか？

- タンパクとかだとマイクロ秒くらいしか計算できない
- 専用機 ANTON は 100 倍以上速い

現在のアーキテクチャの延長では大きな改善は難しい?

実際のところ何に時間がかかっているか？

- グラフを見ると通信がスケールしてないように見える
- とはいえ、カットオフありのMDで、グローバルな通信とかないので、1ステップにミリ秒単位で通信時間かかるのはちょっと不思議 (計算上は数百回とか通信してないとそうならないはず)
- とはいえ「京」の振舞いは必ずしも計算通りではないので、、、
- 重力多体のコード (私のチームの岩澤君が書いたツリー法) は1ノード1000粒子で $5\mathrm{ms}$ くらいだった。通信は1ミリ秒くらい。但しこれ以上ノード数増やすと通信量がそもそも増えた (空間アダプティブなツリー固有の問題)

なので

- ものすごく頑張ると、「京」での MD での実空間計算分の通信時間を 1 ミリ秒程度にはできるかもしれない
- そこから先はやはり難しい。通信、同期のオーバーヘッドは操作の種類によるが数マイクロ秒から数十マイクロ秒はある。(1000 ノード以上で alltoall とかはミリ秒単位)
- PME 使ってると、FFT の通信量 (通信時間) も見える。

ポスト「京」ではどうなるか

そもそもハードウェアはどう変わらるか

- まだ詳細は秘密
- FX10 とポスト FX10 の差分から外挿してみる

	FX10	ポスト FX10	ポスト「京」
コア数	16	32	64?
SIMD 幅	2	4	8?
B/F (片道)	0.5	0.24	0.12?
名目ピーク (TF)	0.2	1.1	5?
単精度ピーク (TF)	0.2	2.2	10?
通信バンド幅 (GB/s)	5	10(?)	20?

外挿しただけなので本当の値かどうかは？

最大の問題: 通信が相対的に 1/12.5 の速度に、、、

良いことはないのか？

- 補助コアがついて通信とか OS 割込みの面倒を見るはず: 同期や大域通信のオーバーヘッドはいくらか小さくなる
- 同じ速度を実現するのに必要なノード数は (単精度なら) $1/50$。同期や大域通信のオーバーヘッドがこちらは大きく減る ($1/4$ くらい？)
- つまり、ノードあたりの原子数を 10 倍にして、それでも 4 倍くらい速く計算できそう。
- 但し PME とかだと FFT で実行時間決まる可能性あり。FMM にするとかもっとアレゲな方法とかも要検討。

ポスト「京」での MD に関するまとめ

- ポスト「京」は、ノードあたりで計算速度は (単精度だと) 50 倍、通信は 4 倍というのが外挿値 (これが本当の値だというわけではない)
- 同じ原子数の計算で、FFT リミットでなければ「京」より 4 倍程度は速くできる。(ノード数 1/10) 10 倍は頑張るとできるかもしれないがそこから上は難しいかも。
- FFT は速くならないので、何か考える必要あり。

ポスト「京」以外の話

- ANTON はどんな機械だったか？(MD 用専用計算機の歴史)
- 汎用並列ではなんかないか？

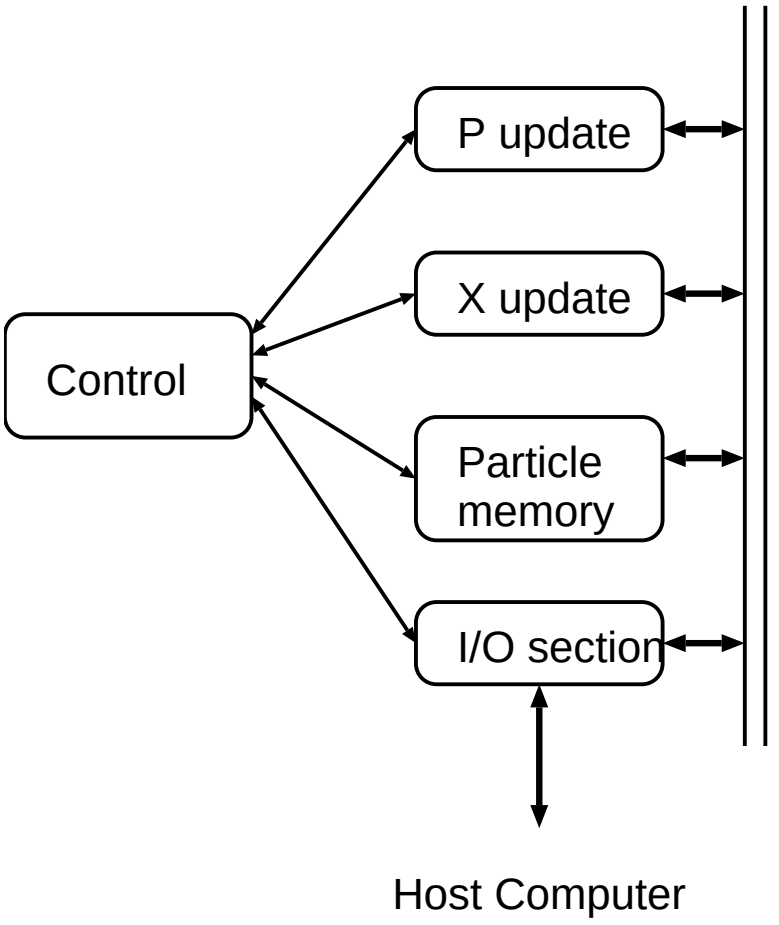
MDシミュレーション用専用計算機の歴史

- MD専用計算機の歴史は結構古い
- Delft Molecular Dynamics Processor: 1980頃完成
- しかし、(分子動力学計算の中だけ見ても)主流になったことはない

歴史をみながら、何故かを考えてみる。

Delft Molecular Dynamics Processor

デルフト工科大学の D. Bakker らが 1980 年頃に完成



LJ ポテンシャルで相互作用する単原子分子の MD 専用機

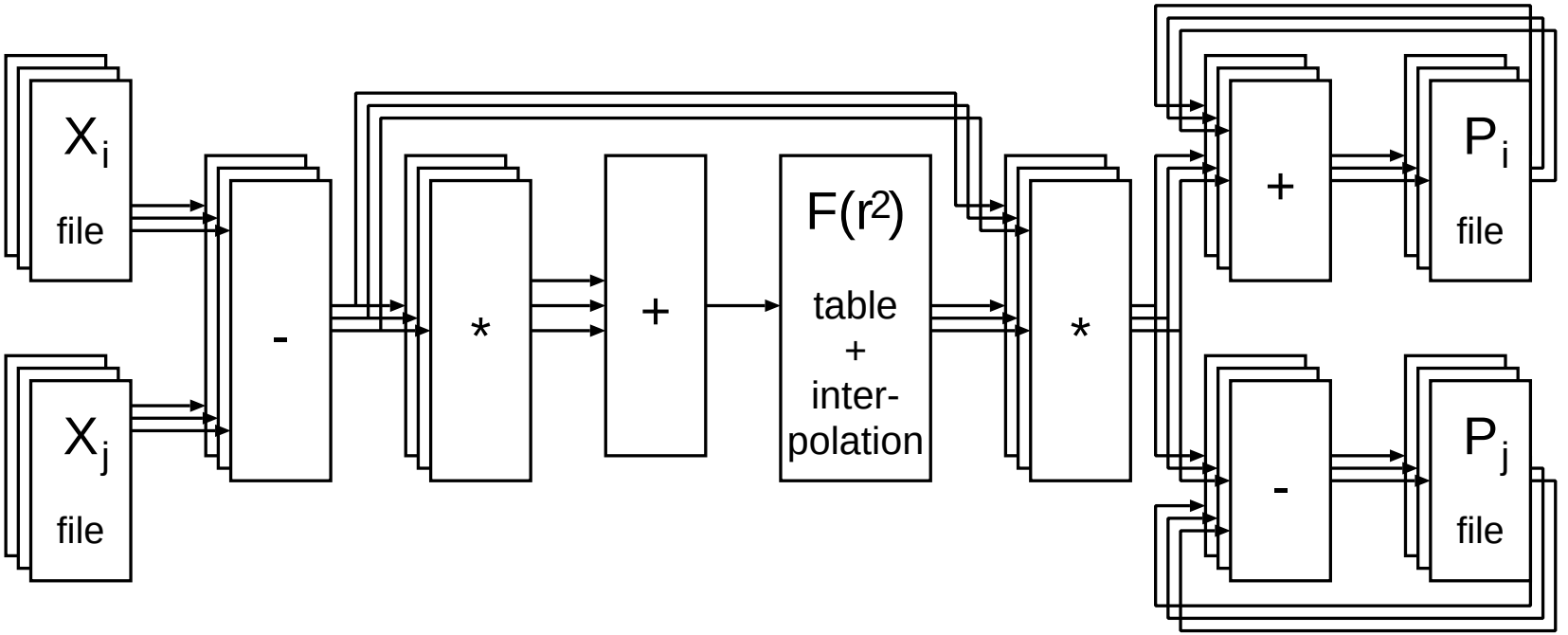
「座標アップデート」(leapfrog での積分)、「運動量アップデート」(加速度計算) それぞれに専用パイプラインプロセッサ

ホストはミニコン、シミュレーションはホストとは独立に走る

CMOS IC を使った 20 枚くらいのラッピング基板で構成。

牧野は 1990 年に見せてもらったことがある

P update



GRAPE と同様、2 粒子の座標をいれて相互作用をだす

P-update パイプラインの詳細

- 入力座標は 24 ビット固定小数点
- 距離の 2 乗は 32 ビットを保持、上位 10 ビットをテーブルにいれて一次補間して力をだす
- linked-list (cell-index) アルゴリズムをハードウェアで実装、隣接セルの粒子間の力を対称性を利用して積算

DMDP をどう評価するか？

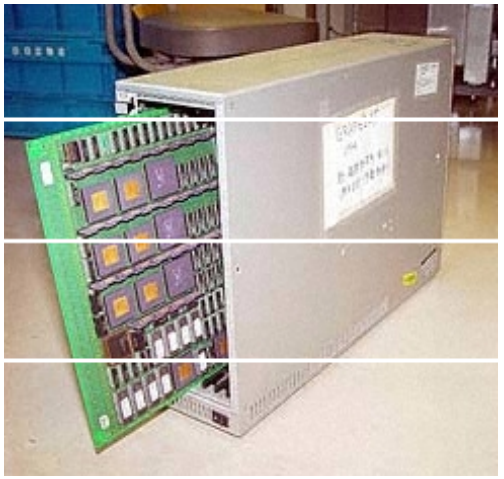
- 性能は素晴らしかった。
  - 性能は 160Mflops 相当: Cray-1 の理論ピーク程度
  - コストは 1000 万円以下くらい (Cray-1 の 1/100 以下)
- どれくらい科学研究に使えたかはあまり資料がない、、、
  - 単原子分子でそんなにできることはない
  - 結果解析もハードウェア追加しないとできない
  - 計算精度がちょっと微妙な気がする

FASTRUN

Fine *et al.* **1991**

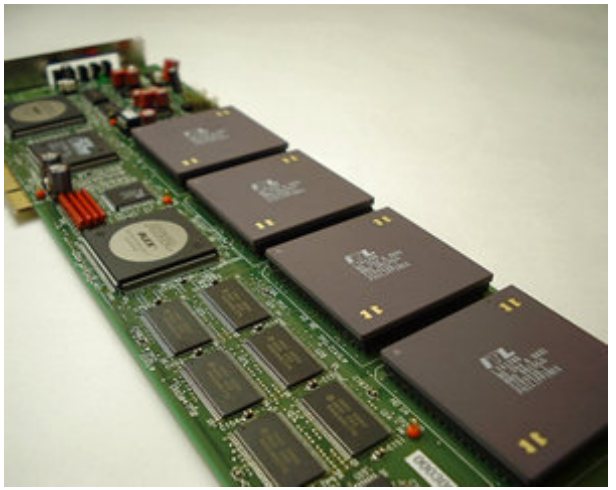
- コロンビア大学と BNL の共同開発
- タンパク MD 用
- コスト、速度は DMDP と同程度 (10 年たってるのに、、、)
- 相互作用計算部分のみハードウェア
- ネイバーリストをハードウェアで実装した (らしい)

GRAPE-2A



- 1991 年に開発開始。駒場の杉本グループと、筑波 (当時) の永山グループの共同開発
- DMDP の相互作用パイプラインだけを取り出して汎用計算機につないだようなもの
- 92 年には完成、費用 (人件費以外)100 万くらい、200Mflops
- 当時のワークステーションはまだ 10Mflops くらいだったのでまあ速かった

MD-GRAPE



- 93 年くらいに開発始めた
- 駒場の杉本グループと画像技研 (株) の共同研究。都からの研究費で
- 基本的には GRAPE-2A をカスタム LSI 化、4 チップのせたボード開発。
- エバルド法用の DFT パイプラインにもなる設計 (泰地による)

Gflops の性能。

この他に、富士ゼロックス+大正製薬で「MD-Engine」というのも。

MDM と PE

- 戎崎が理研に異動したあと、理研でスタート
- MDM は 2001 年くらいに完成、75Tflops
- PE は 2006 年に完成、1Pflops
- どちらも、基本的には MD-GRAPE の大規模並列化
- どちらも 1 万チップ程度の巨大システム、ホスト計算機も数十台。インフィニバンドで並列化。

ここまで振り返ると、、、

- 分子動力学用専用計算機 (MD-GRAPE とその後継含めて) は性能 (少なくとも価格当りの計算速度) は高い
- でも、どうサイエンスの役に立ったかは色々意見もあるかもしれない

MDM、PE の「問題点」

- 10 万原子では性能でない = 小さい系の長時間計算にはむかない
- Direct Ewald だけなので計算量が粒子数の 1.5 乗で増える (PME をホストでやればいいが、、、)

ANTON

D. E. Shaw の個人研究所「DESRES」が開発。

D. E. Shaw って何者？

Financial Times 2010/3/8 の記事から:
DE Shaw broadens Asian reach

DE Shaw, the $24bn hedge fund founded by mathematician David Shaw, is to open offices in Shanghai and Tokyo as part of an expansion in Asia, according to people familiar with the situation.

The Shanghai office, to house a team of private equity analysts, will increase the group 's presence in the region and mark its first expansion into mainland China. It will focus on acquisition opportunities in China.

モルガン・スタンレーに 1986 年に入る前はコロンビアの計算機科学科のファカルティ。並列計算機 Non-Von の開発を主導。1951 年生まれ。

ARCHITECTURE AND APPLICATIONS OF A HETEROGENEOUS, MASSIVELY PARALLEL MACHINE

David Elliot Shaw

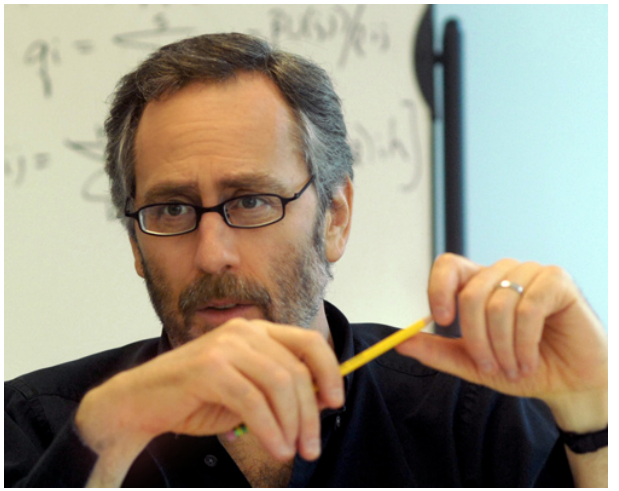
Department of Computer Science, Columbia University, New York, New York 10027

1985 年のレビュー論文

The organization of an experimental, massively parallel machine called NON-VON is described, along with some of the typical artificial intelligence applications for which the machine is intended to provide significant performance and cost/performance advantages over conventional computer systems. The machine incorporates an *active memory*, which is constructed using custom, very large scale integrated (VLSI) chips. Each chip contains a number of simple processing elements and a small number of larger processing elements, each capable of controlling the operation of a subset of the active memory. A simplified, preliminary prototype of the NON-VON architecture is now operational at Columbia University.

Performance projections, derived through detailed analysis and simulation, are summarized for applications in the areas of rule-based inferencing, computer vision, and knowledge base management. The results, most of which are based on benchmarks proposed by other researchers, suggest that NON-VON could improve performance of such tasks by as much as several orders of magnitude, compared to a conventional sequential machine of comparable hardware cost.

D. E. Shaw って何者？



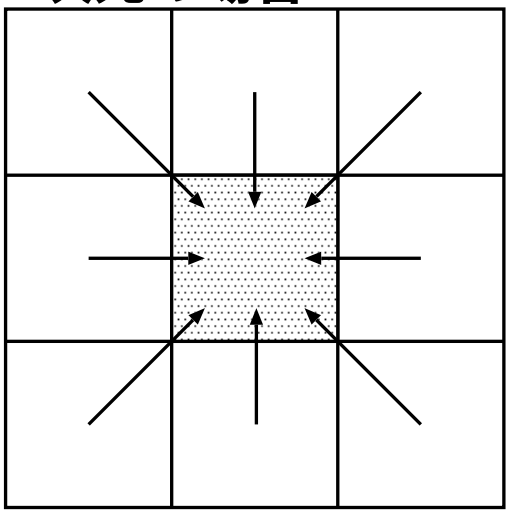
- 元々並列計算機研究者・アーキテクト
- 世界有数のヘッジファンドの創設者
- 個人資産から計算生物学、特にタンパクの機能シミュレーションのための研究所を作った
- かなり良い給料と魅力的な仕事でよい研究者を集めた。牧野が知っているところでは Caltech にいた John Salmon (並列ツリーコードで有名) とか

ANTON

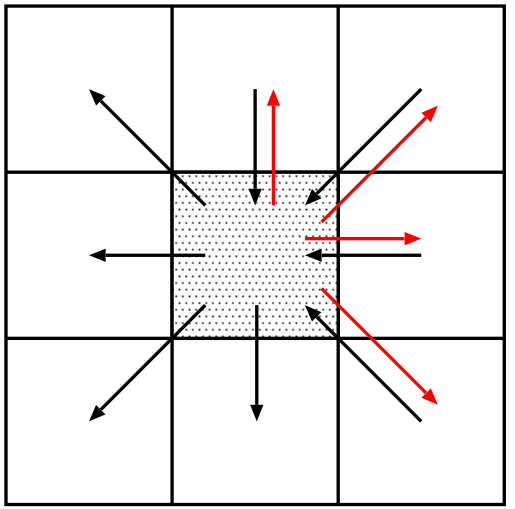
- 数万原子程度のあまり大きくない系で高い性能がでることを目標に設計
- 相互作用計算の新しい並列アルゴリズム (NT 法) を開発、それ用にハードウェアを設計
- 相互作用計算パイプライン+プログラム可能プロセッサ+ネットワークプロセッサを 1 チップに集積

近距離相互作用の並列計算

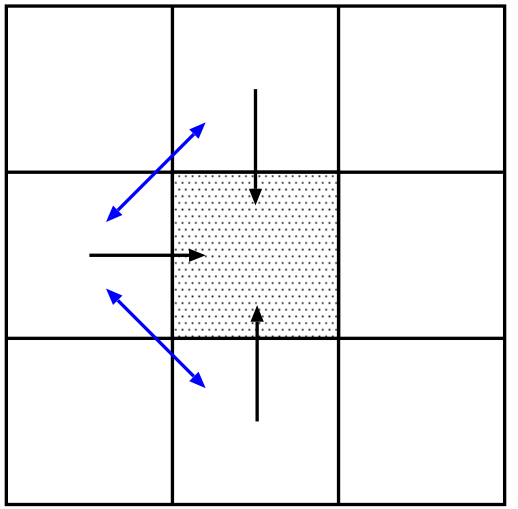
2 次元の場合



対称性を利用しない。周り 8 個から座標をもらって力計算



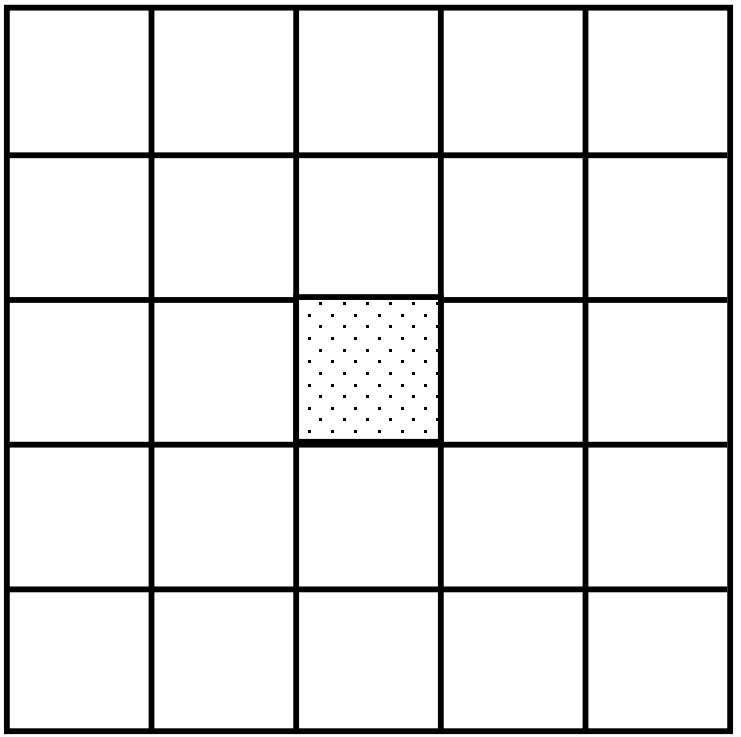
対称性を利用する。4 個からもらって、相手への力は送り返す

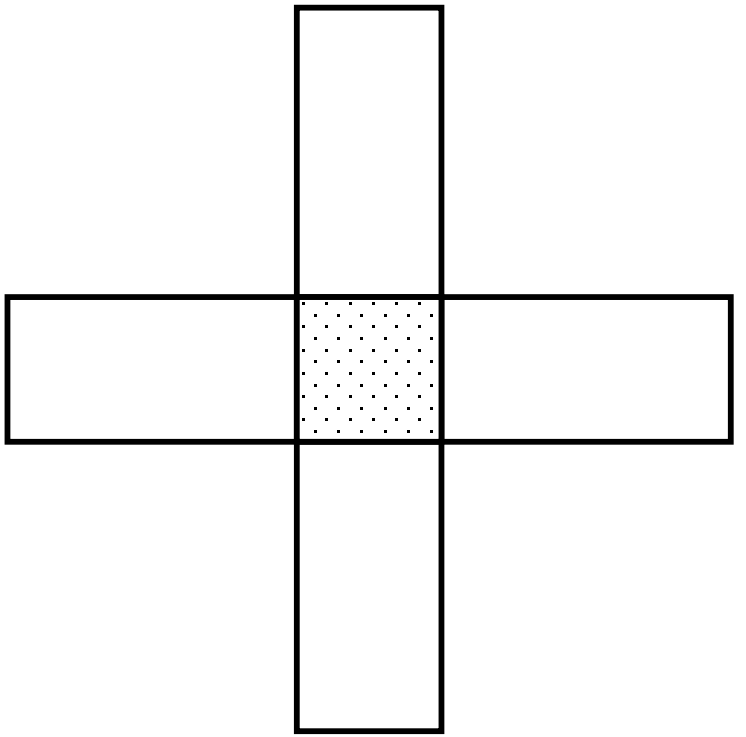


NT 法の考え方。軸方向だけからもらって、斜めの力もついでに計算する

3 次元だと 26 と 6 でだいぶ違う。

基本セルを小さくすると





24 と 6(3 次元だと 124 と 16) で、大きな違い。

漸近的な振る舞い: 問題サイズを固定して、セル数 p が無限大の極限

- 普通の方法: 通信量は一定値に収束
- NT 法:通信量が $p^{-1/2}$ でゼロに収束

ANTON のハードウェアと性能

- 1 チップに DMDP と似たような相互作用パイプライン 32 本、800MHz 動作、1Tflops くらい
- 512 チップを 8^3 トーラスネットワークに接続
- 速度は 500TF くらい、専用機としてはそんなに速くない
- 2 万原子の系で 1 日に 10 マイクロ秒を実現。(1 ステップ数十マイクロ秒)。「京」でできるより 100 倍速い。

何が高速化に効いているか？

- 通信量、通信回数を減らす新しいアルゴリズム
- プログラム可能部分、ネットワークも独自開発することでの通信レイテンシの削減
- その他計算量を減らすための沢山の細かい工夫

ANTON とそれまでの「MD 専用計算機」の違い

- 原子数少ないところで性能をだすことを目標にした
- そのために新しいアルゴリズムを開発した
- 元々汎用並列計算機アーキテクトだったので、まともに複雑なものを作った

これまでの MD 専用計算機では新しいアルゴリズムという話はあまりなかった、、、

じゃあ我々はどうするか

- MDGRAPE-4 というものはある。組みあがって現在調整中。ANTON くらいにはなるはず。(ANTON-2 のことはまあその)
- 演算加速部での評価の時には 10 マイクロ秒/ステップがとりあえずの目標。できないわけではなさそうだった。これは一応汎用 (プログラム可能) な機械。
- 日本語のベンチャー企業が、MIMD で 1024 コアの超メニーコアチップを開発、256 チップ、384TF のシステムが KEK で稼働している (はず)。こういうのも検討する価値はありそう。

ExaScaler-1 と PEZY-SC ボード

公表されている性能

- HPL で 154TF (まだ実行効率は低い)
- 4.02GF/W (それで既に世界最高クラスの電力性能)

PEZY-SC 超メニーコアチップ

- NEDO からの助成で開発していた (我々も同じ頃応募したんだけど、、、)
- 1024 コア。1 コアで倍精度 2 演算 or 単精度 4 演算。750MHz 動作で単精度 3TF ピーク
- 階層キャッシュメモリ L1/2PE, L2/16PE, L3/256PE
- 外部メモリは DDR4 8ch
- 通信インターフェース Gen3 PCIe x8 4 組 (多分)。PCIe 以外としても使えるらしい (?)
- 通信等制御用の ARM コア 2 つ

現在の ExaScaler-1 では GPGPU クラスタと同様、ホスト経由での通信がボトルネックになるが、PEZY-SC 同士を直結したシステムつくればレイテンシの問題も回避できる (かもしれない)

PEZY Computing 齊藤社長

2008 年の記事から

「テラリコン・インコーポレイテッド代表取締役会長兼最高経営責任者」と書いてある

テラリコンは

「1997 年 東京大学医学部付属病院の放射線科医師だった齊藤元章が、日本における 3 年間の基礎開発を経て、最高技術責任者の高橋一夫と共に米国シリコンバレーに TeraRecon, Inc. を設立。」

商業ベースで画像処理技術 (プロセッサ含む) をずっとやってきた人らしい。

まとめ

- ポスト「京」は加速部がなくなったので、概ね「京」の延長。
- 1 ステップにかかる時間が劇的に短くなったりはしない。10 倍くらいは速くできるかも。
- MDGRAPE-4 はできてきつつあるので、期待してもよさそう。
- 国内ベンチャーがこういうのに使えそうな低消費電力メニーコアプロセッサを開発している。検討する意味はありそう。