アプリケーションサイドから いうべきことを考える

牧野淳一郎

東京工業大学理工学研究科

理学研究流動機構

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
アプリケーション作業部会（第 2 回）2011/9/1

今日の話の構成

- 何故アプリケーションサイドからなにかいうべきなのか
- 計算機の「発展」の大雑把な歴史
- ハードウェアデザインの幅
- アプリケーション性能の考え方の例

何故アプリケーションサイドから なにかいうべきなのか

- ほったらかしにしておくと、使いにくい上に性能がでるアプリケーションが殆どないものができてしまうかもしれない
- (牧野の考えでは) 色々注文をつけ、なおかつソフトウェアの書き方を見直すなら、かなり色々なアプリケーション (「京」では性能がでないものを含めて) で高い実効性能がでて、なおかつそれほど高価でも電力食いでもないものができるかもしれない

何故アプリケーションサイドから なにかいうべきなのか

- ほったらかしにしておくと、使いにくい上に性能がでるアプリケーションが殆どないものができてしまうかもしれない
- (牧野の考えでは) 色々注文をつけ、なおかつソフトウェアの書き方を見直すなら、かなり色々なアプリケーション (「京」では性能がでないものを含めて) で高い実効性能がでて、なおかつそれほど高価でも電力食いでもないものができるかもしれない

といっても俄には信じ難いわけで、

- 何故そうか
- ソフトウェアの書き方を変えるというのはどういう話か

をしたいと思います。

計算機の「発展」の大雑把な歴史

- 1960 年代まで:スカラ計算機 (CDC7600)
- 1970 年代:ベクトル計算機 (Cray-1)
- 1980 年代:共有メモリベクトル計算機 (Cray XMP)
- 1990 年代:分散メモリ並列計算機 (CM-5, Cray T3x, 富士通 VPP500)
- 2000 年代:CPU の複雑化 (SIMD 命令、マルチコア)
- 2010 年代:？

何故そのような発展がおきたか

- 全体を通じて: トランジスタの微細化。使える数がどんどん増えた
- 70 年代: コアメモリから半導体メモリへの移行 (メモリバンド幅の増大)
- 90 年代初め: 1 チップでベクトルプロセッサが可能に (Intel i860)
- それ以降: メモリバンド幅と演算速度のギャップ増大 (memory wall)

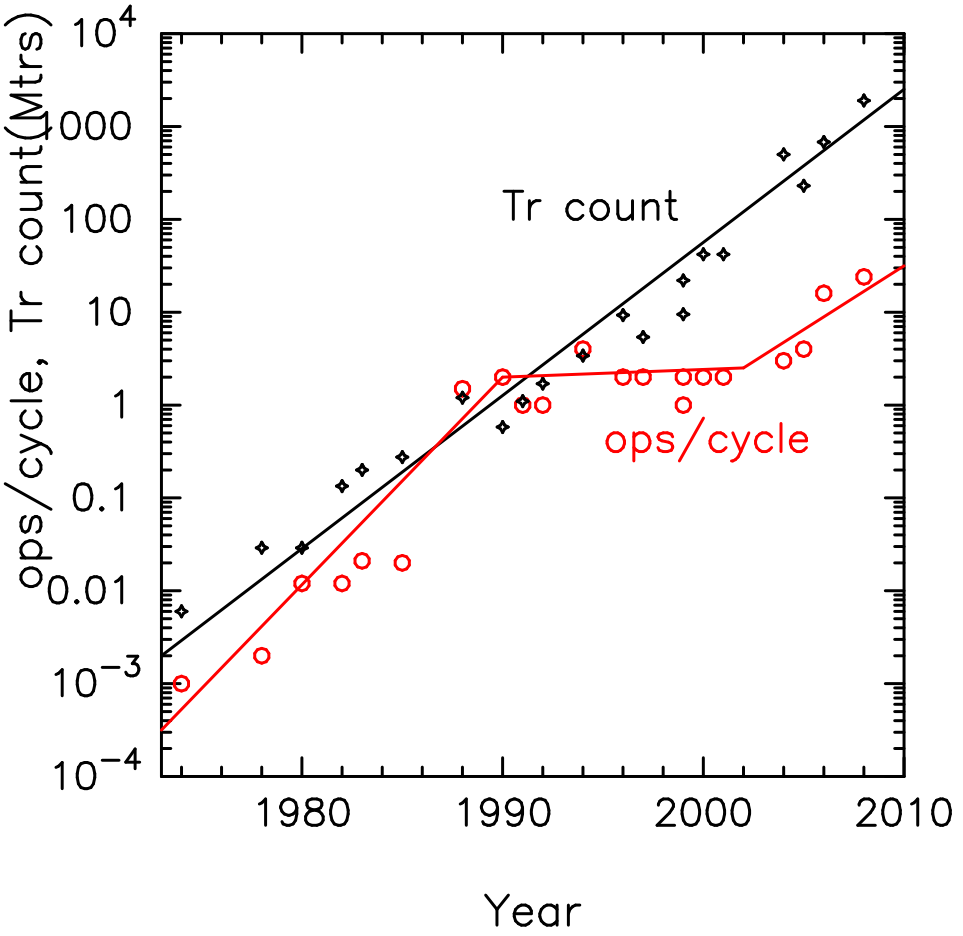
B/F の変化

マシン	年	B/F
Cray-1	1976	4
Cyber-205	1982	12
VPP-500	1993	4?
CP-PACS	1996	2
ES	2002	4
Pentium-4	2001	1
“Dunnington” Xeon	2008	< 0.2
NVIDIA Fermi	2010	0.3
Sandy Bridge	2011	0.5

20 年間で大体
1/10 になった。

最近 10 年間の減少
が大きい。

B/F だけが減っているわけではない



黒: マイクロプロセッサのトランジスタ数。10 年で 100 倍弱

赤: 演算器の数 (90 年以前は必要サイクル数の逆数)。90 年代に増えなかった。ここ数年も停滞

整理しなおすと

- 90 年代に起こったこと: トランジスタ利用効率の急激な低下。演算器を増やさないで頑張ってクロック上げた。
- 2000 年代に起こったこと: $\mathrm{B/F}$ の急激な低下。電力的にクロック上げるの無理になった。演算器増やすがメモリバンド幅は増やせない。

10年後まで同じ傾向なら

- トランジスタ利用効率は上がらないかジワジワ下がる
- B/F は段々下がる

アーキテクチャ部会のベースライン的イメージはこれ

- 10Tflops、 1TB/s、 キャッシュ 128MB、
- 通信 100GB/s、レイテンシ 100ns
- 50Gflops/W、エクサフロップスで 20MW

想定している技術との関係

- 想定技術: 11nm、500億トランジスタ、5GHz クロック
- 演算器は 1024 個、組合せ論理には2億トランジスタ
- キャッシュには60億トランジスタ、面積の 10% 程度
- メモリインターフェース: 電力の 10% 程度？
- ネットワークインターフェース: やはり電力の 10% 程度?

つまり、チップ面積、電力を大きく変えないでも

- 演算器数を100倍
- オンチップメモリ量を10倍
- 外部メモリ、ネットワークのバンド幅を10倍

の「どれか」、あるいはそれぞれの適当な組合せを実現可能

消費電力、価格性能比

消費電力あたりの性能を上げるには、

- チップ面積当りの演算器数を増やす
- クロックを落とす
- メモリバンド幅を落とす
- ネットワークバンド幅を落とす

のが有効。価格当りの性能もほぼ同じ。多くのアプリケーションで、オンチップメモリサイズと必要なバンド幅に関係がある。

通信レイテンシは？

- 100ns/hop くらいまではやればできる。電気余計に喰うとかもない
- それより減らすには色々考える必要あり。

「私がしたいこと」は何？

×今あるプログラムを (ちょっと大きく問題で) 速く

×10年後にありそうな計算機でできる中で役に立つことを考える

○10年後に作れるはずの計算機、アルゴリズム、プログラム実装、解く問題の多次元空間の中で最適解を見つける

アプリケーションによって欲しい計算機が全然違っては困るが、本当にそんなに違ってくるのか?が、アプリケーション部会で議論するべきことでは？

「アプリケーションの性能」の考え方の例

規則格子、陽解法流体計算 (宇宙分野での相対論的 MHD 計算、松元@千葉大によるオーダー推定)

- 1 格子点、1 ステップ当りの演算数 10 万
- 変数 100 個=1000 バイト

理想的に賢いキャッシュがあれば、1 ステップで 1 つの変数は主記憶から 1 度読んで 1 度書くだけ。

この場合、10 万演算に対して 2000 バイト読み書きなので、B/F は 「0.02」あれば足りる。

オンチップメモリにデータが全部のるなら 0.02 も不要。

通信に対しても同様に考えることができる。

そんなことをいっても

- 実際のプロセッサのキャッシュは理想的からは限りなく遠いのでは？
- 陰解法で multigrid 反復なんだけど、、、
- 不規則格子なんだけど、、、
- AMR は、、、
- FFT は、、、

とりあえずキャッシュの話だけ

- 1つの方向:ハードウェア自動制御のキャッシュを諦めて、ソフトウェア制御にする
- 例: GPU の共有メモリ、CDC7600、Cray-2、Venus のセクタキャッシュ
- 問題点: プログラムが煩雑、保守性低下
- 可能な対策: ステンシル記述からのソフトウェア自動生成。昔なら日立の DEQSOL

FFT

グローバルなFFTが計算量の大半を占めるような解法をどうするか？

FFT

グローバルなFFTが計算量の大半を占めるような解法をどうするか？

無理。

アルゴリズムの変更を考えるべきでは？

不規則格子、Multigrid、AMR

- すみません、私はよくわからないです
- 「Co-design」のためにはアルゴリズム・サイエンス・アーキテクチャを見通した議論が必須
- エクサの本格設計に入る前に検討進めるべき課題

「複数アーキテクチャ」の一つの意味

- 今の延長のものはそれとしてやる
  - 「保険」
  - 「過去の資産」の維持
  - 最低限の性能向上
- 新しいものは過去と切り離してゼロから絵を書く

そんなお金はない？

新しいのを安く作る方法を考える。

まとめ

- ハードウェア屋さんに任せておくと「トレンド」でなんかでてくる
- 実際に可能な設計空間はずっと広い
- 但し、電力を増やさないでなんでもできるわけではない。メモリ、通信バンド幅数倍、演算速度数十倍
- アプリケーション側の原理からの検討が必要
- 「トレンド」にのったもの「も」作るべきかも

宇宙・粒子系シミュレーションでの大雑把な見積り

- 計算規模は:1 週間使うとして 10^{24} 演算
- 粒子・ステップ数として 10^{19} 程度 ($O(N \log N)$ 法が使えるとして)。粒子数が
  - ダークマターハロー形成等 10^{14-15}
  - 銀河形成 10^{11-12}
  - 惑星形成、星形成 10^{11-10}
- 最大 1 秒に 10^3 ステップ

目標:

「10^{10} 粒子に対して 1 タイムステップ 1 ミリ秒で粒子間の重力や流体的な相互作用が評価する」

時間積分法: Oshino and Makino 2011 のハイブリッド法

評価するもの、仮定

- 演算器精度
- ネットワーク速度
- ネットワークレイテンシ
- 主記憶バンド幅
- 主記憶サイズ
- オンチップメモリサイズ

の必要度を評価する。簡単のため、1 ノードはピーク演算性能 100 Tflops で、1 万ノードから構成されるとする。
(粒子数が多い時には基本的に色々楽になる)

演算器精度

- 時間積分のための粒子座標の表現、近接粒子間の引き算: できれば 4 倍精度でやりたい (演算量の 0.1% 以下)
- 引き算のあとの重力相互作用計算:殆どは 10 ビット程度の精度でよい。単精度も不要。「半精度」演算があると極めて有用

ネットワーク速度、レイテンシ

ネットワーク速度

- **ノード当り** 100 **万粒子**
- 1 **ステップで他のノードから受け取る必要がある粒子数は** 20 **万程度** (Makino 2004 **による**)
- **データ量は**6M **バイト、これを** 1 **ミリ秒以下で、とすると最低** 6GB/s
- 30GB/s **あれば十分**

レイテンシ: 1ms **に比べて** 3 **桁程度小さければ十二分である。**

主記憶バンド幅、サイズ

- 現在の実装では座標データについて 1 ステップで 10-30 回読出しが発生する。
- 最小では 300 バイト/粒子、300MB/ステップ、300GB/s
- B/F 3e-3

主記憶サイズ

- 10^{14} 粒子の計算で 100GB 程度
- 10^{11} 程度の計算では 100MB/ノードとなるので、オンチップで実現したい。
- より大粒子の場合も、ブロック化を工夫することでメモリアクセスを 1/5 程度にはできる

まとめ

1 ノード (100Tflops) の性能として

ネットワーク速度	$> 30\mathrm{GB/s}$
ネットワークレイテンシ	$<1\mathrm{us}$
主記憶バンド幅	$> 60\mathrm{GB/s}$
主記憶サイズ	$> 100\mathrm{GB}$
オンチップメモリサイズ	$> 100\mathrm{MB}$

が、効率的に宇宙物理粒子系シミュレーションを行うための必要条件となる。

4 倍精度、半精度が使える (特に半精度で高速 or 低電力になる) と有効

実際にはさらに (i) プロセッサ内部の結合方式 (ii) ノード速度、台数が変わる時のスケーリング関係 (iii) プロセッサチップ内部のアーキテクチャを考える必要あり