専用計算機に未来はあるのか？

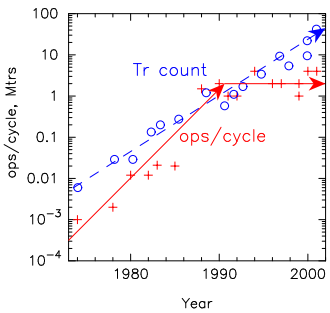
牧野

- 科学技術用高速計算機の現状
- 専用計算機の現状と困難
- 困難の回避
- まとめ

科学技術用高速計算機の現状

- 「汎用性」の低下
  アーキテクチャの複雑化
  - メモリ階層
  - スカラー → ベクトル → ベクトル並列 (分散メモリ)
    「あらゆる応用プログラムで万能に速い」計算機はなくなった
- 半導体利用効率の低下
  - チップあたりトランジスタ数は 10 年で 100 倍
  - チップあたりの演算器数は過去 15 年間で 2 倍程度

半導体利用効率の低下



マイクロプロセッサのトランジスタ数と 1 クロックあたりの浮動小数点演算数

演算数は 1 に達したところで進歩がほとんど止まっている

進歩が止まった理由

- ハード的な問題
  - memory wall: プロセッサ — メモリ間のデータ転送速度が足りないので、演算器を増やしても性能が上がらない
  - スーパースカラーでは実行ユニット数を増やすとハードウェアが急速に大きくなる。
- ソフト的な問題 — 命令レベルの並列性には限界

科学技術計算の多くは高い並列性を持つ — 「本来は」並列性の限界はあまり問題ではない

汎用プロセッサの進化の方向

- Simultaneous Multithreading
  - 基本的には、1 プロセッサを時分割して複数のプロセッサに見せる
  - キャッシュミスのペナルティをある程度隠蔽できる
  - メモリのバンド幅不足は解決しない
- On-chip multiprocessor
  - 2-4 個のプロセッサを 1 チップに載せる
  - メモリのバンド幅、レイテンシどちらもより大変になる

専用計算機

アプリケーションに専用化したプロセッサ:多数の演算器を使える

商用で成功しているもの

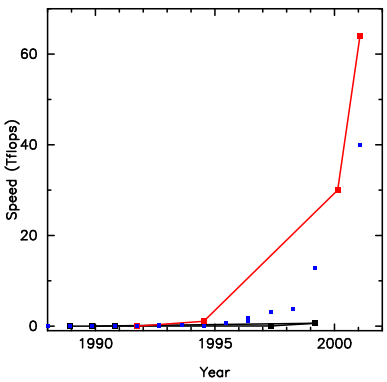
- PC 用グラフィックカード
- Sony PS2 などのゲーム機

演算器は 10 個程度

科学技術計算用専用計算機

- 多体問題専用計算機
  - 粒子間相互作用計算に特化した演算パイプライン
  - 400 個程度の演算器を 1 チップに集積 (GRAPE-6, MDM)

GRAPE の進化



赤: GRAPE

青: 汎用ベクトル/並列

速度はほぼ同等。
開発費は 2 桁違う。

専用計算機の「限界」

開発費の高騰

1990	1μm	1500万円
1997	0.25μm	1億円
2004	90nm	3億円以上？

ある程度広い応用を持つものでないと難しい

FPGA、「リコンフィギャラブルデバイス」は？

トランジスタ利用効率で大きく劣る
↓
データ語長が短い応用でないと高性能は難しい

別のアプローチはないか？

目標:

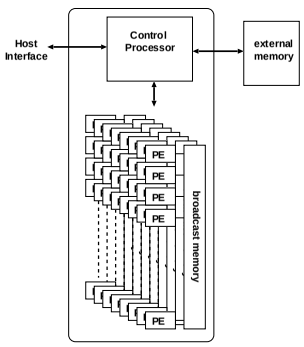
- 応用に特化し、多数の演算器を 1 チップに集積、並列動作させて高い性能を得た専用計算機の特徴を生かす
- しかし広い応用範囲を実現する

明らかな矛盾？

GRAPE の高性能: 多数の演算器を 1 チップに集積し、小さなメモリバンド幅で全演算器を有効に働かせたことによる

相互作用の計算式に専用化したパイプラインが必須というわけではない。

1 チップ 超並列プロセッサ



- 非常に多数のプロセッサエレメント (PE) を 1 チップに集積
- PE = 演算器 + レジスタファイル (メモリをもたない)
- チップ内に小規模な共有メモリ (PE にデータをブロードキャスト)
- 外部メモリへのインターフェースを持つ

アプリケーション

「GRAPE として」使う

- 全ての PE が、自分の粒子への、同じ粒子からの力を計算
- 力を及ぼすほうの粒子データは外部メモリから供給

「多対多」の相互作用計算ならばなんでも扱うことができる

- 粒子法: 重力、分子間力、流体的相互作用、、、
- 境界要素法: ポアソン方程式、ヘルムホルツ問題、、、
- ルジャンドル展開による極座標流体計算
- 密行列ソルバ (内積型を利用)

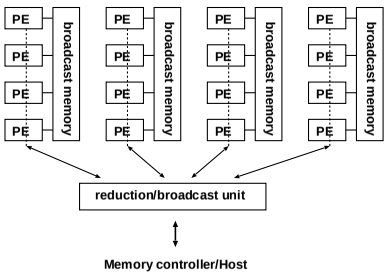
PE の (もうちょっと) 詳細

演算:(最低限) 以下をサポート

- 倍精度浮動小数点加減算
- 倍精度固定 (ブロック) 小数点加減算
- 単精度浮動小数点乗算
- 区分多項式近似・スケーリングに必要なビット操作
- 条件付き実行

メモリオペレーション
ローカル共有メモリ: ブロードキャスト、シリアルな読出し、書き込み
汎用ポート: リダクション演算、放送

ローカル共有メモリとリダクション



GRAPE として使う時: 同じ共有メモリにつながったプロセッサは同じ粒子から別の粒子への力 (i 並列)、違う共有メモリブロック間では j 並列

メモリコントローラ、シーケンサ

なるべくチップは設計を簡単にしたい
シーケンサは設計ミスがもっとも起きやすい場所
メモリコントローラはチップに作り込みたくない

面倒なものは全部 FPGA に任せる。
問題:命令ストリームを十分高速に渡せるか？
手抜きな対応: ベクトル命令なら問題なし。

GRAPE に比べるとどれくらい損か？

GRAPE-6: 200万ゲート、 400 演算 = 5K ゲート/演算
PE の大きさ (推定):

ユニット	サイズ
fadd	7K
ALU	8K
register file	10K
合計	25K

場合によっては乗算・加算同時実行可能だとすると、合計では 12.5K ゲート/演算。純粋な GRAPE に比べると 2.5 倍損
但し、レジスタファイルが 10K ゲート程度で十分かどうかは微妙。
共有メモリやインターフェース回路はサイズとしては無視できる。

類似のアプローチとの比較 (1)

- SIMD 超並列計算機 (Goodyear MPP, CM-1/2)
  - 外部メモリへのバンド幅、通信ネットワークでプロセッサ数が制限
  - プロセッサがメモリを共有することでプロセッサ数の制限を回避。応用範囲には制限。
- FPGA
  - トランジスタ利用率が低い
  - ソフトウェア開発が困難
  - どちらも回避

類似のアプローチとの比較 (2)

再構成可能計算機

- DSP ブロック入り FPGA
  - ゲート効率 FPGA よりはちょっといい？
- DAP/DNA, DRP
  - ゲート効率悪い
  - 動作速度遅い

他のアプローチとの基本的な違い:

チップ内にスイッチングネットワークをもたない

- (多少) できないことも発生
- デザインの単純化 → 高集積、高速動作

ソフトウェア

PE のプログラムは、基本的にシーケンシャル
(SIMD アプローチの利点)

- アセンブリ言語
- 単純なコンパイラ

上位のソフトウェア

- 複数ホスト+超並列ボード上の並列化
- ホスト計算機/超並列ボードのタスク割り当て

まとめ

- 普通のマイクロプロセッサのアプローチ:現在でもトランジスタ利用効率が極めて低い
- GRAPE のような専用計算機では、効率は高いが半導体技術の微細化によって開発費が高騰
- 多用途で GRAPE 並の効率を実現する一つの方法:1 チップ超並列プロセッサ

将来展望

- 多対多の相互作用向け SIMD 超並列計算機という新しい概念
  - 専用計算機並の高いコストパフォーマンス
  - 特定の物理系に縛られない広い適用範囲
  - 将来の半導体技術の発展を有効に利用可能
- ベクトル並列、スカラー並列と相補的
- compute-intensive な問題にブレークスルーをもたらす

COTS アプローチ

Intel Pentium 4 等の “off the shelf” プロセッサ

- 大量生産 → 低コスト
- 大量生産 → 莫大な開発費 → 高い性能

単一スカラープロセッサで COTS の性能を超えるのは非常に難しい
性能が memory limited — プロセッサを別のものにしても速くならない

COTS の限界: backward compatibility.
VLIW すら難しい (Itanium の失敗)。

チップ内部構成

- 共有メモリユニットの数 (共有メモリユニットあたりの PE 数)
- PE の構成
  - 演算方式 (語長、データ形式、関数計算のサポート)
  - 制御方式 (ベクトル命令/スカラー命令)
  - レジスタ構成
- メモリインターフェース方式
- 汎用プロセッサとのインターフェース