GRAPE-DR 計画と その応用可能性

牧野淳一郎
東京大学理学系研究科天文学専攻

大雑把な予定

1. GRAPE-DR 計画とは何か？
2. GRAPE-DR のアーキテクチャ
3. 応用可能性
4. まとめ

GRAPE-DR 計画とは何か？

「基本的には」次期 GRAPE 計画

- 2004 年度から 5 年計画
- 目標ピーク性能: 2 Petaflops
- チップ数 2048-4096
- 単体チップ性能 0.5-1Tflops

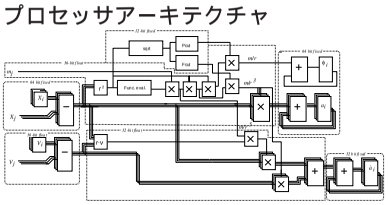
と、これだけなら今までの GRAPE が速くなっただけ。
実際のアーキテクチャ: 今までの GRAPE とは全然違う

- なぜ違うか
- それで何ができるか

というのが今日のお話

GRAPE とはどんなものだったか？

プロセッサアーキテクチャ



重力相互作用計算の順番に演算器を並べたパイプライン

- シリコンの利用効率は極めて高い
- 動作クロックも上げやすい
- アプリケーション限られる。多種類作るのはリソースがかかり過ぎる

「次期 GRAPE」の実際的な問題

天文だけ (しかも理論だけ (しかも N 体だけ)) でもらうにはチップ開発コストが大き過ぎる

チップ開発費

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予算獲得が難しい

ではどうするか

1. やめる

ではどうするか

1. やめる
2. 安くあげる方法を考える

ではどうするか

1. やめる
2. 安くあげる方法を考える
3. なんかお金を取る方法を考える

ではどうするか

1. やめる
2. 安くあげる方法を考える
3. なんかお金を取る方法を考える

GRAPE-DR では (3) を選択

基本的な考え

- チップに演算器を 2000 個くらい入れる
- それを (GRAPE が得意なタイプの問題に対しては) ある程度のプログラム可能性をもった形で使う。 GRAPE のようなハードワイヤードなパイプラインにはしない。

もうちょっとそれらしく言うと:

- 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

そんなことができるか？が問題

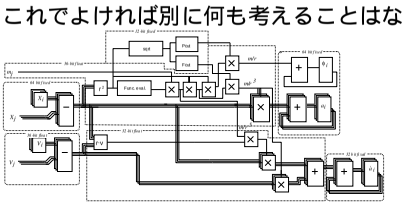
多数の演算器を詰め込む方法

境界条件: メモリバンド幅は増やしたくない (システムコストはほぼメモリバンド幅で決まる)
可能な方策

1. GRAPE 的専用パイプラインプロセッサ
2. 再構成可能プロセッサ
3. SIMD 並列プロセッサ

GRAPE的専用プロセッサ

これでよければ別に何も考えることはない。



再構成可能プロセッサ

FPGA ベース

- 任意のロジックを実現可能
- 集積度、速度は大きなペナルティがある
- 精度が低くてもいい応用には向く

いわゆる「動的再構成可能プロセッサ」
IPFlex DAP/DNA 等

- 8-32 ビットの単純な ALU を多数集積
- その間をプログラマブルな配線でつなぐ
- 集積度、速度はやはり大きなペナルティ

SIMD 並列処理

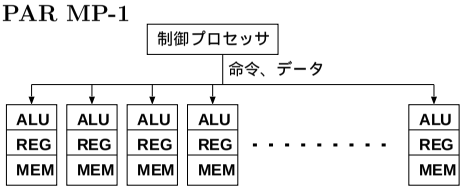
パイプラインプロセッサをやめにして、「プログラム可能なプロセッサ」を沢山載せる。
SIMD (Single Instruction Multiple Data): 全プロセッサが同じ命令を実行
基本的には、全プロセッサがソフトウェアで GRAPE をエミュレーションする。

SIMD 並列処理って？

- 古典的 SIMD 並列計算機
- SSE、MMX とかの SIMD 拡張命令
- GRAPE-DR におけるマトリックス SIMD(今作った言葉)

古典的SIMD並列計算機

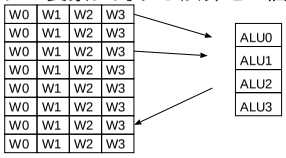
Illiac IV, Goodyear MPP, ICL DAP, TMC CM-2, MASPAR MP-1



1960 年代に発生、80 年代に絶滅
半導体技術の向上に対応できないアーキテクチャ: 計算速度とメモリアクセス速度が比例する必要あり。
メモリ階層をつける: プロセッサが複雑になりすぎて SIMD の意味が無くなる。

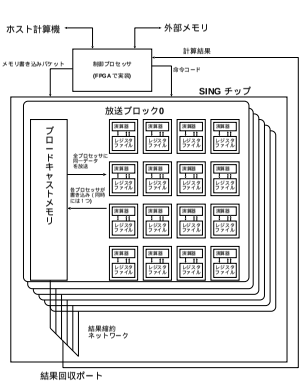
SIMD拡張命令

128 ビットなり 64 ビットのデータを 4 語に区切って、それぞれの要素に対する演算を 4 個の演算器で同時に処理



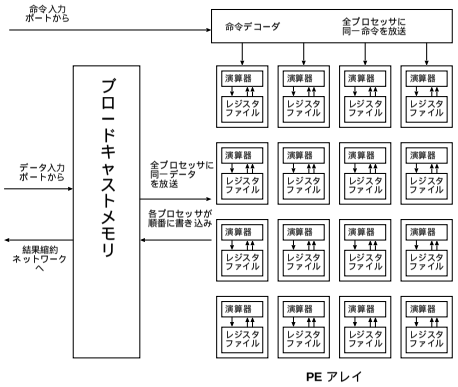
1 つのプロセッサの中の話: キャッシュとデータをやりとり
並列度 4 程度が限界？それ以上増やすとキャッシュの速度が追いつかなくなる。

GRAPE-DR におけるマトリックスSIMD



- 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)。これを共有する PE をブロードキャストブロック (BB) と呼ぶ。
- 制御プロセッサ、外部メモリへのインターフェースを持つ

ブロードキャストブロック



使い方: GRAPE としての場合

最も単純には

- 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

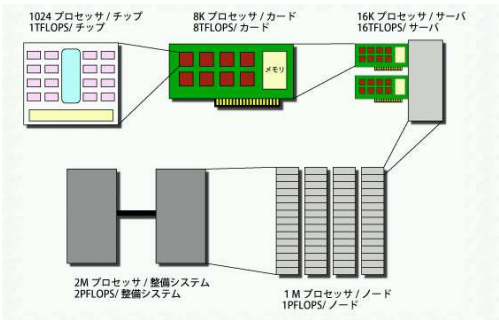
現実問題としては、実用アプリケーションで性能を出すのはもうちょっと工夫がいる。

- 違うブロードキャストユニットの同じ位置の PE には同じ粒子データを書く
- 力を及ぼすほうの粒子データはブロードキャストユニット毎に違うものにする。

最終システムのイメージ

2007 年度に基本的には完成 (2008 年度に増強)
ピーク性能 2Pflops (単精度)
プロセッサチップ 2048 個。
プロセッサボード当り 4 チップ (PCI-X/Express)
ホストは 512 ノードの PC クラスタ

最終システムのイメージ



従来の SIMD との違い

とにかくチップ一杯に演算器を詰め込む。
このために、演算器についているメモリを極限まで小さくする (現在の設計では 2KB)。
ピーク性能は高い
GRAPE として使う分にはピークに近い性能が出る
それ以外のものはどうか？

アプリケーション

今までの GRAPE 的なものの置き換え

- 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

プロジェクトとしての目標

- 密行列ソルバ — Linpack で 500 Tflops 以上

もうちょっとまともな目標

- 分子軌道法での 2 電子積分
- 密行列の対角化、直交化

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
fmul	15K
register file 等	30K?
合計	47K

乗算・加算同時実行:合計では 25K ゲート/演算

純粋な GRAPE に比べると 5倍損

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。

天体力学への応用

Few-body problem はどうか？

1つ解くだけならあんまり意味はない。プロセッサ1つは最新の Intel とかのプロセッサより数倍遅い

1000 個の3体問題を並列に計算するなら 1000 倍速い

放送ブロック内で1つの系の積分を並列化することも可能

原始惑星100個くらいの系を高精度、長時間積分

(誰かが頑張れば多倍長計算もできるはず)

天体力学への応用

Few-body problem はどうか？

1つ解くだけならあんまり意味はない。プロセッサ1つは最新の Intel とかのプロセッサより数倍遅い

1000 個の3体問題を並列に計算するなら 1000 倍速い

放送ブロック内で1つの系の積分を並列化することも可能

原始惑星100個くらいの系を高精度、長時間積分

(誰かが頑張れば多倍長計算もできるはず)

今回の天体力学研究会の口頭発表で紹介された3体問題、少数多体問題はほとんど GRAPE-DR でできそう

まとめ

- 次世代 GRAPE は予算がついた。
- 作るものは基本的には SIMD 動作する1チップ超並列プロセッサ。これで従来の GRAPE、MD-GRAPE の他、密行列計算も可能にする。
- GRAPE 的なメモリ階層をもたせることで、 SIMD 計算機の難点を解消して高い価格性能比を実現する。
- 上のような応用の他、少数自由度の系を大量に並列時間積分するとかいった応用も考えられる。

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
fmul	15K
register file 等	30K?
合計	47K

乗算・加算同時実行可能だとすると、合計では 25K ゲート/演算。純粋な GRAPE に比べると 5倍損

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 (理論上は、、、)

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

大規模科学技術計算の今後

GRAPE-DR のような非常に計算量の多い規則的計算に向いたアーキテクチャ:
多分結構順調に進歩する (CELL みたいなのも含めて)

規則的だけどメモリバンド幅が必要な (ということになっている) 種類のアプリケーション
本当に不規則なアプリケーション

- 普通の PC (クラスタ) で我慢？
- もうちょっと違う何かを考える？

ポスト地球シミュレータ ???