サイエンテフィック・システム研究会
科学技術計算分科会 2007 年度第 2 回会合資料

2007 年 11 月 28 日

GRAPE-DR とスーパーコンピューティングの未来

牧野淳一郎

概 要

GRAPE-DR は 512 個の演算ユニットを 1 チップに集積し、単精度 512Gflops、倍精度 256Gflops のピーク性能を持つ超並列プロセッサである。これまで東大で開発されてきた天文シミュレーション専用計算機 GRAPE の後継機だが、重力計算用の専用パイプラインではなくプログラマブルな SIMD プロセッサを採用した。2008 年度には、単精度 2Pflops のピーク性能を持つシステムを構築する計画である。講演では GRAPE-DR の意義、HPC 用システムの将来像にも触れたい。
キーワード: GRAPE、 超並列アーキテクチャ、ヘテロジニアス・マルチコア、専用アーキテクチャ

1 はじめに

GRAPE-DR は、我々が開発した 1 チップ超並列プロセッサ である。1 クロックサイクルに浮動小数点乗算と、浮動小数点加減算 (または整数演算) の 2 演算を独立に行う演算コアを 1 チップに 512 個集積し、500MHz 動作させることで 512 Gflops の理論ピーク性能を実現した。但し、パイプラインスループット 1 で実行できるのは単精度実数と倍精度実数の乗算であり、倍精度実数同士の乗算ではスループットが半分になる。加減算は常に倍精度実数に対して実行されるが、倍精度実数同士の乗算では 2 サイクルに 1 度加減算器も使うために倍精度実数乗算を実行中には乗算器、加算器ともにスループットがクロック当り 0.5 になり、ピーク演算速度は 256 Gflops に低下する。この、倍精度でピーク演算性能が 256 Gflops という数字はこの原稿を執筆している 2007 年 11 月初めの時点で世界最速である。

本稿では、まず、我々が GRAPE-DR を開発するにいたった背景を、計算機アーキテクチャの進化一般の観点、我々が従来開発してきた重力多体問題専用計算機 GRAPE の方向性の観点からまとめ、 GRAPE-DR のアーキテクチャ、性能について簡単にまとめた後、他のいくつかのアプローチ、具体的には FPGA を使った再構成可能計算、 GPGPU、汎用マイクロプロセッサを使った並列計算等と比較する。

2 背景 — 計算機アーキテクチャの「進化」

1940 年代末に ENIAC が稼働を開始してからの 60 年間にわたって、計算機の速度はほぼ 10 年に 100 倍のペースで向上を続けてきた。これは他の技術では類を見ないものである。この進化は、2 期に分けることができる。第一期は 1969 年まで、すなわちスループット 1 の浮動小数点演算ユニットを持つ CDC 7600 以前であり、第二期はその後現在までである。

この 2 つの時期で、計算機の性能向上のために必要な方策は根本的に異なる。どちらの時期においても、性能向上のための基礎になっているのはデバイス技術の進歩である。1950 年代においては真空管やパラメトロンからトランジスタへというデバイスそのものの変化であり、60 年代には個別トランジスタから集積回路へ、それ以降は集積回路の微細化、バイポーラから CMOS へといったトランジスタ構成方式の進化等多岐に渡るが、計算機アーキテクチャの観点からはこれらの進歩は基本的に以下の 2 つの意味を持つ。

- デバイスの動作速度の進歩
- デバイス集積度の進歩

動作速度の進歩は、基本的にはそのまま計算機の能力の向上につながる。これに対してデバイス集積度、すなわち 1 台の計算機で利用可能なスイッチング素子の数の増加は、なんらかの方法でそれを計算機構成方式に反映させ、より多数の素子を有効に使えるようにしなければ性能向上につながらない。第一期と第二期では、この方法が異なる。

第一期においては、利用可能な素子の数は完全にパイプライン化した浮動小数点乗算器を構成するのに十分なほどではなかった。従って、計算機アーキテクチャの構成方針はある意味単純なものであり、

- なるべくスループットの高い浮動小数点演算器を作る
- 残りの素子で、十分な命令・データを演算器に供給できる回路を作る

とまとめることができる。第一期の特徴は、**大きな計算機ほど価格性能比が良い**ということである。このことは当時グロシュの法則として知られていた。この法則は「計算機の性能は価格の 2 乗に比例する」というものである。

浮動小数点演算性能の観点からは、大きな計算機ではまず浮動小数点演算専用回路をつけることができ、さらにその性能は素子の数を増やすとそれ以上に上がる、ということからグロシュの法則は理解できる。スループットを倍にするにはトランジスタ数を倍増する必要はないからである。

しかし、スループット 1 の演算器が計算機に入ってしまうと、それからどうするか？という問題が発生する。後知恵では自明な回答は、演算器を増やして並列に計算させる、というものだが、歴史は素直にその方向に進んできたわけではない。CDC 7600 の後スーパーコンピューターの主流となったのは、 CDC 7600 の設計者であったシーモア・クレイが設計したベクトル計算機 Cray-1 であった。Cray-1 では、

- 半導体メモリとマルチバンク構成によるメモリのスループットの向上
- ベクトル演算制御による演算器の有効利用

という 2 つの技術革新により、 CDC-7600 から 3 倍弱のクロック向上でしかなかったにもかかわらず多くのアプリケーションでそれ以上の性能向上を実現した。特に競争相手であった CDC Star 100 や Cyber 203/205 に比べて圧倒的に高いスカラ演算性能とベクトルレジスタによる短いベクトルでの高い演算性能を持ったことでマーケットを支配した。

しかし、Cray-1 の後スーパーコンピューターの性能向上は長期に渡って低迷する。 1976 年に出荷が始まった Cray-1 が 160 Mflops の理論ピーク性能をもったのに対して、 1985 年にやっと出荷が始まった Cray-2 の演算速度はほぼ 10 年後であるのにわずかに 2 Gflops であった。Cray-2 の少し前に出荷が始まった Cray-X/MP では 4 プロセッサの最大構成でようやく 1 Gflops となった。この間に、日本の計算機メーカーは VP-200, S-810, SX-2 といったベクトル計算機を相継いで発表し、Cray のマシンを上回る性能を実現した。

この時期の大きな特徴は、浮動小数点演算器の数の増加が極めてゆっくりであったことである。 Cray-1 は 1 ユニットもっていたが、 Cray-2 でようやく 4 ユニット、 VP-400, S-810, SX-2 も 4 ユニットである。この 10 年間に利用可能な素子数は大雑把には 100 倍になっているので、素子数増加の殆ど全ては演算器数の増加以外の何かに使われたことになる。この、演算器に使われる素子数の割合の急速な低下が、第二期の大きな特色である。

ベクトル計算機において演算器の増加がゆっくりであった大きな理由は、複数の演算器の間、あるいは 1 つの演算器を持つプロセッサ複数の間で物理的にメモリを共有し、しかもメモリと演算器の間に十分な転送速度をもたせる、という設計方針にある。バイポーラ技術で製造されたベクトル並列機では、演算器の数は Cray C-90 の 32 が最大となる。 C-90 は 1991 年、すなわち Cray-1 の 15 年後に出荷が始まっている。

物理的な共有メモリと高いメモリバンドのどちらかあるいは両方を断念すればより多くの演算器をもった計算機を構成できる。このために 1970 年代から 80 年代にかけては多様な並列計算アーキテクチャ研究が行われた。しかし、1980 年代中頃にいたるまで、実際に Cray 等のベクトル並列計算機よりも高い性能を実現した並列計算機は殆どなかった。その基本的な理由は、それらの並列計算機では浮動小数点演算器のスループットが 1 よりもはるかに低く、グロシュの法則による損が大きかったためである。例えば Intel i286+287 を要素プロセッサにしたマシンでは、浮動小数点演算に数十クロックかかった上にクロックサイクルがベクトル計算機の 1/10 以下であったため、 10000 プロセッサ程度にならなければ 16 演算器程度のベクトル計算機に対抗できなかったが、それほどの規模のものは実現困難であった。

しかし、計算機全体における第一期から第二期への変化と同じことが 1980 年代中頃に今度は 1 チップ LSI 演算器で発生した。すなわち、1-2 チップで浮動小数点演算器を実現することが可能になった。これはまず浮動小数点演算専用 LSI である Weitek 1164/1165, LSI Logic L64132 といったチップで 1985 年頃に実現され、 Intel i860 において 1 チップマイクロプロセッサでも実現された。つまり、 i860 は CDC 7600 ないし Cray-1 にあたるものを 1 チップで実現した。

これらの演算器を使った並列計算機は筑波大学の QCDPAX(1)、TMC の CM-2、さらには Intel SSD の iPSC-860、Touchstone、Paragon 等があるが、どれも同じ時期のベクトル並列計算機に比べてピーク性能に関する限り圧倒的に高い価格性能比を実現した。これらでは、分散メモリアーキテクチャを採用することでノードあたりのコストを下げている。もっとも、マイクロプロセッサベースの超並列システムの商業的な成功は 1993 年に発表された Cray MPP (T3D) を待つことになる。これは DEC Alpha 21064 ベースであり、当初 1 チップ 150Mflops とほぼ Cray-1 に匹敵する演算速度と、 STREAM ベンチマーク実測で 400MB/s に及ぶ高いメモリバンド幅 (Cray-1 は理論ピーク 640MB/s) を実現した。さらに、高速なノード間ネットワークももたせたことで T3D では実用的なアプリケーションで高い性能を出すことが比較的容易になったことが商業的成功につながったと考えられる。

こうして、1990 年前後にマイクロプロセッサベースの超並列システムの古典的なベクトル並列計算機に対する優位性はほぼ確立した。1993 年には航技研数値風洞 (富士通 VPP-500) が発表され、ベクトル並列計算機も分散メモリアーキテクチャになったことで 1980 年代の停滞から脱出し、飛躍的な性能向上を実現した。これは地球シミュレータを経て NEC SX-9 につながる流れになっている。しかし、価格性能比でみる限りマイクロプロセッサベースのシステムと競合するのは困難になってきている。

一方、マイクロプロセッサベースのシステムも、1970 年代後半にベクトル並列機が停滞期にはいったのと全く同じ停滞が長く続いているのが現状である。これは、高性能マイクロプロセッサが持つ乗算器の数の過去 20 年間の増加速度をみれば明らかである。1989 年に 1 だったものが、2007 年の 4 コアプロセッサをもってようやく 8 まで増加した。これは、Cray-1 から Cray C90 までの 15 年間に 32 倍よりもさらにに遅い、18 年間にわずか 8 倍である。LSI の設計ルールはこの間に 1μm から 65nm になり、およそ 200 倍以上の数のトランジスタが使われるようになっている。

従って、1990 年代初めにベクトル並列計算機に対してマイクロプロセッサベースのシステムが持っていたような優位性を現在のマイクロプロセッサベースのシステムに対してもっている何かが可能なはず、と考えられる。これが、新しいアーキテクチャを考えるべきである背景である。

マイクロプロセッサにおいて演算器の数を増やすことの障害になっているのは、アムダールの法則といわゆる「メモリーの壁」の 2 つである。これらはどちらも 1980 年代にベクトルプロセッサの演算器の増加の障害になっていたものと本質的には同じである。しかし、大きな違いは、1990 年代にはメモリバンド幅の不足は共有メモリから分散メモリに移行することで解決できたのに対して、現在のマイクロプロセッサでは外付けのメモリバンド幅を大幅に増やす方法は存在しないことである。このため、問題は同じだが解決は同じではない。

3 GRAPE とその発展

我々は GRAPE(2; 3) の開発を 1989 年に始めた。これはちょうど i860 が発表されたのと同じ年であり、1 チップ LSI で浮動小数点演算が可能になった時期の数年後、技術的には複数の浮動小数点演算器を 1 チップに集積できるようになってきた頃である。GRAPE (GRAvity piPE) の基本的な考えは、天文学における重力多体問題のシミュレーションの中でもっとも計算量の多い部分、つまり粒子間の重力相互作用の計算機だけを専門に行うデジタル回路を作る、というものであった。

粒子間重力はニュートン重力で、粒子間の距離の 2 乗に反比例する。単純な計算法では、粒子が N 個あると一つの粒子の加速度は他の $N-1$ 個の粒子からの重力の合計になる。このために、計算量は粒子数の 2 乗に比例する。ツリー法や高速多重極展開法といった方法で計算量を $O(N \log N)$ や $O(N)$ に減らすことも可能である。しかし、重力多体問題では相互作用が引力であるために 2 つの粒子がいくらでも近付くことができる。また、重力不安定のために様々な空間構造が発達する。このため、粒子によって軌道運動のタイムスケールが大きく違い、粒子毎に独立に軌道積分の時間刻み幅を変化させるような方法が必要になる。この場合には、ツリー法等の有効な利用はそれほど簡単ではない。また、ツリー法や高速多重極展開法でも、近くにある粒子同士の相互作用は直接計算される。このため、粒子間相互作用を直接計算するようなハードウェアがあれば、これらの方法も高速化できる。

ハードウェアは、パイプライン構成としてそれぞれの演算器が 1 種類の演算しか行わないようにした。これには、プログラムによる制御が簡単になる、回路の大半が演算器になって並列動作するので高い性能を実現できる、演算毎に演算精度を最適化することで回路規模が小さくなる、計算の中間結果をメモリやレジスタに書き戻す必要がなくなるので高速のメモリが不要になる等の多数の利点がある。もちろん、これは、複数の演算回路を使った回路が容易に構成できるようになった時代、すなわち、前節の表現ではマイクロプロセッサの発展が第二期に入る前後にのみ可能であった。この意味で、我々は非常に良いタイミングで GRAPE の開発を始めた。数年後であれば、浮動小数点 LSI を並べた回路ではマイクロプロセッサ 1 つよりも高い性能を出すのは難しくなったし、逆に数年前では数十の演算回路を並べた回路は簡単に作れるものではなかった。

最初の GRAPE-1 は、計算精度を落として、粒子座標を固定小数点 16 ビット、その後の計算を対数表現 8 ビット、最終段の重力の積算を固定小数点 48 ビットでするものであった。これは、基本的には、我々のグループが初めて作るデジタル回路であったのでなるべく簡単なものにしたかったからである。しかし、この過程で、問題によってはこのような低い精度で良く、通常の倍精度浮動小数点演算が必要なわけではないということを発見した。さらに、どの程度の語長が必要かを決定するための理論的なモデルも構築できた。 GRAPE-1 は当時修士課程 1 年だった伊藤が設計、製作をした。

問題によっては高い精度が必要である。そのために、浮動小数点演算チップを使った GRAPE-2 を GRAPE-1 に続いて開発した。これも伊藤が中心になった。これらの開発過程は (4) に描かれている。

GRAPE-1, 2 の計算速度はそれぞれ 308Mflops, 51 Mflops であり、当時の最高速のスーパーコンピューターに比べると 1/10 から 1/100 の程度である。しかし、開発コストは 10-100 万円程度でありスーパーコンピューターの

価格の 1 万分の 1 から千分の 1 の間である。当時の東大大型計算機センターのスーパーコンピューターの使用料は 1 時間 1 万円程度なので、数十時間分のコストである。1ヶ月程度動けば元がとれることになるが、 GRAPE-2 はその後 3 年ほど使われた。

1990 年には GRAPE-3 の開発を始めた。これは、精度が低い GRAPE-1 をベースに LSI 化を試みたものである。GRAPE-1 に比べて若干精度をあげた演算回路を 1μm ルールでデザインした 11mm 角のチップに収め、 20MHz 動作させて 760Mflops の速度を実現した。24 チップをラッピングボードに収めたものを作ったが、これはクロック 10MHz 程度でしか動作しなかったので 1 ボードの理論ピーク性能は 9.12 Gflops、このボードを 2 枚並列動作させるシステムで 18.2Gflops の理論ピーク性能を実現した。これは GRAPE-1 の 60 倍であり、GRAPE-3 が完成した 1991 年当時のスーパーコンピューターの性能にほぼ匹敵する。

GRAPE-3 の成功により、さらに多数のチップを並列動作させる GRAPE-4 の開発に必要な予算を獲得できた。これは GRAPE-2 並の精度で、約 20 個の浮動小数点演算回路を 1 チップに収めた。32 MHz のクロックで動作し、 640 Mflops の速度となる。ボード 1 枚に 48 チップを収納した。これらのチップは 1 つのメモリユニットを共有し、全てのチップが同じ粒子データを受け取る。それぞれが違う粒子への重力を計算することで並列計算をする。さらに、36 枚のボードが並列計算をする。ボード間の並列動作は、基本的には別の粒子から同じ粒子セットへの力を計算することによる。このため、一つの粒子への力は多数のボードに分散して求まる。これを合計する回路を別につけることで、ホスト計算機との通信量を削減した。 GRAPE-4 の完成は 1995 年である。 ピーク性能は 1 Tflops を超えるものになり、同時期の航技研数値風洞を含め、どのスーパーコンピューターよりも高速となった。総予算は 2.5 億であり、典型的なスーパーコンピューターの価格の 1/10 以下である。

GRAPE-3 は、コピーを作りたいという研究者が多かったためにプリント基板化した量産版を設計し、製造・販売を委託した。これは 8 チップであるが 20MHz で動作し、6 Gflops のピーク性能をもつ。100 枚近くが製造され、世界中 (主にヨーロッパ) で使われた。

1996 年及び 1997 年に、 GRAPE-3 及び 4 の後継である 5 と 6 の開発を始めた。6 では 0.25μm ルールを使うことで 60 演算のパイプラインを 6 本 1 チップに集積し、 90MHz 動作させることで 32 Gflops のピーク性能を実現し、2048 チップを並列動作させて 64 Tflops のピーク性能を実現した。

また、GRAPE の考え方をタンパク質折り畳みシミュレーションのような分子動力学シミュレーションに応用することも 1992 年頃から始めた。これは GRAPE-6 とほぼ同時期に完成した理研 MDM (理論ピーク性能 75Tflops) からさらに 2006 年に完成した Protein Explorer に引き継がれている。後者のピーク演算能力 1 Pflops であり、現在世界最高速である。

以上の GRAPE の発展を見ると、前節で述べたマイクロプロセッサ発展の停滞期と丁度重なる時期に半導体技術の発展を直接演算性能の向上に使ってきたことがわかる。 これが可能であったのは以下の理由による。

- パイプライン構成のため多数の演算器を必要とした。制御回路も不要であった。
- 同じくパイプライン構成のため、高速なメモリが不要であった。
- 複数のパイプラインが単一のメモリに接続できるので、メモリバンド幅の必要量がさらに小さくなった。
- 仮想多重パイプライン (ハードウェアマルチスレッドとほぼ同じ) により 1 つの物理パイプラインを速度が遅い複数のパイプラインに見せることで、さらにメモリバンド幅の必要量を下げることができた

要するに、演算器の数を増やしてもメモリバンド幅を増やす必要がなかったのである。

4 GRAPE-DR へ

GRAPE-6 の開発プロジェクトが完了したのは 2002 年度である。次をどうするかが問題になる。分子動力学用専用計算機では既に述べたように後継の開発が始まっていた。しかし、天文シミュレーション用では一つ大きな問題があった。それは、 LSI 開発コストの指数関数的な上昇である。GRAPE-4 の時には 2000 万円前後で LSI の開発が可能だったが、 GRAPE-6 では億を超えた。 2003 年頃には 5 億円程度が必要になっていた。これは GRAPE-4 の開発予算総額の倍以上であり、 重力多体問題計算専用計算機に日本国内で獲得できそうな研究費ではない。

一方、前々節でみたように汎用マイクロプロセッサは停滞してきており、GRAPE の考え方は一つの解決を与えている。解決といっても、それは、「外付のメモリバンド幅が必要でないアプリケーションなら 1 チップに多数の演算器がはいっていても有効に使える」というものであって完全に汎用な解決というわけではない。しかし、ベクトル計算機や分散メモリ並列計算機にしても完全に汎用なわけではなく、それらをが有効な問題、あるいは有効なアルゴリズムにしか使えないのだから、さらにもう一つ制約が増えてもそれで扱える問題がいくつかあれば十分ではないだろうか？

GRAPE の考え方であるメモリバンド幅を増やすことなく演算器を増やす、という方針のまま、ある程度汎用なシステムにする 1 つの方法は、演算パイプラインを多数の単純な要素プロセッサに置き換え、それらを SIMD 動作させることである。単純に GRAPE の代わりをさせるならば、要素プロセッサには大規模なメモリは不要であり、数十語のレジスタがあれば中間結果を格納できる。もっとも単純には、これらが全て 1 つのメモリにつながり、同じ粒子から複数の粒子への力を計算すればよい。

もっとも、これでは力を受ける粒子の数が大きくなりすぎる。要素プロセッサを適当にグループ化し、それらがメモリを共有するようにするほうが実際的である。この場合は、GRAPE-4 の複数のボード間でそうしたのと同じように、違うグループは違う粒子から同じ粒子セットへの力を計算する。粒子データは外付メモリから供給される必要があるが、粒子データ 1 つにつき数十演算するので、グループの数が 10-20 なら必要なメモリバンド幅は極めて小さい。

このようなグループ化をすると、1 つの粒子への力が別グループの複数のプロセッサに分かれて求まるので、これらを縮約してホスト計算機に送り返すための加算器ツリーが必要になる。この部分はグループの数程度のハードウェアなので、グループ内にある程度の数のプロセッサがあれば面積的には無視できる。

このようなプロセッサの概念を図 1 に示す。我々は、このアーキテクチャを GRAPE-DR (Greatly Reduced Arrey of Processor Elements with Data Reduction) と命名した。

2004 年度から実際にこのアーキテクチャに基づいたシステムの開発を始めることができた。これは、東京大学情報理工の平木教授との共同プロジェクトである。このプロジェクトでは

- 1 チップに 512 プロセッサを集積、 500MHz 動作で 512 Gflops を達成
- 4096 チップを並列動作させて 2Pflops のピーク性能を実現

を目標とした。現在、チップは完成しており (図 2 左)、チップ 1 つ、制御・通信用 FPGA、外付メモリを搭載したボードも完成し、チップ動作の検証も終了した。4 チップを搭載して PCIe 16 レーンインターフェースをつけたボードの設計がほぼ終了している。最終的には 1 台の PC にこのボードを 2 枚搭載したノード 512 台からなるクラスタ構成で 2 Pflops を実現する。

図 3 に要素プロセッサの構成を示す。プロセッサの構成は近代的なプロセッサ設計の考え方からはかなり常識外れに見えるかもしれない。まず、命令語、デコーダといった概念は存在しない。各部の制御信号が基本的にはその

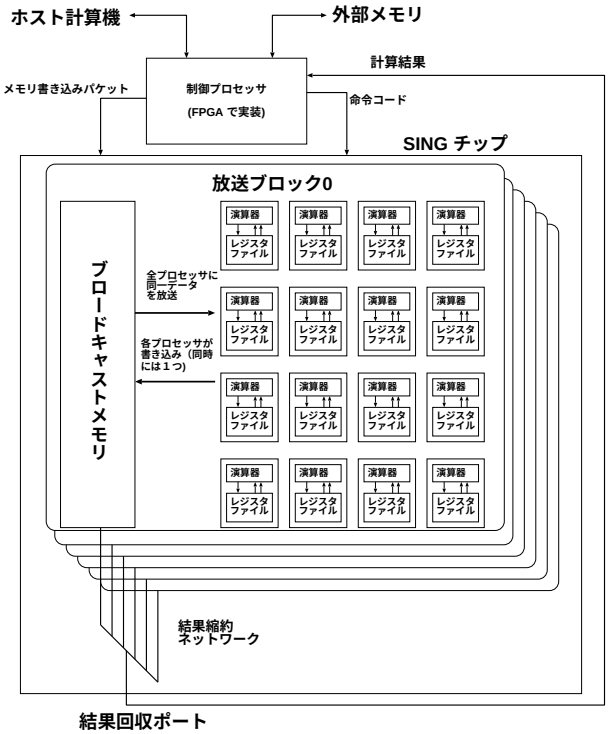

図 1: GRAPE-DR の基本構成

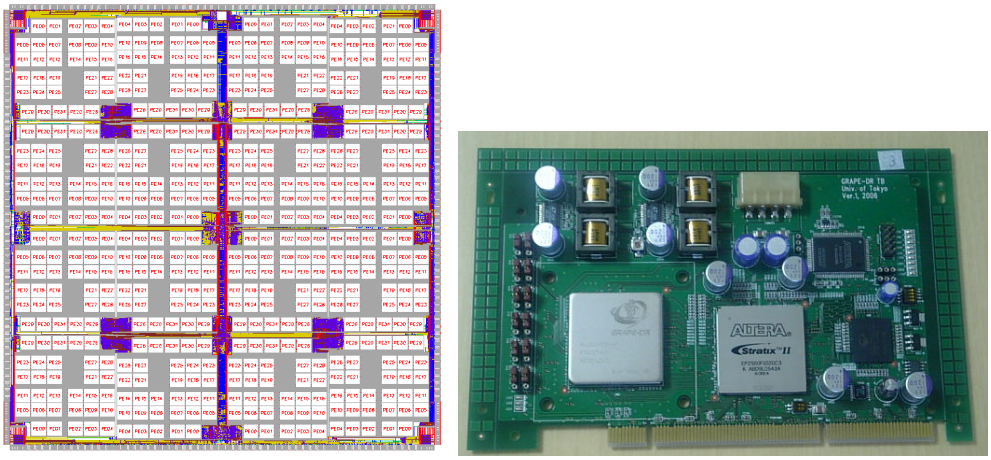

図 2: GRAPE-DR チップレイアウト (左) と チップ評価ボード (右)

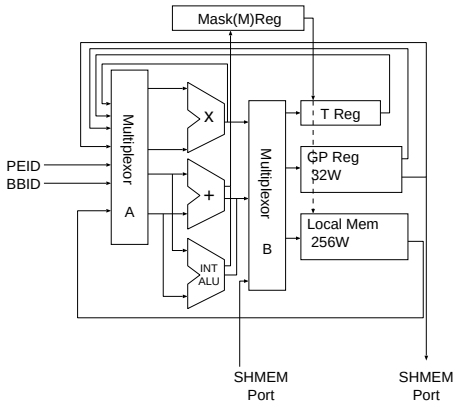

図 3: 要素プロセッサの構成

ままチップ外の制御回路から全要素プロセッサに放送される。但し、そのままでは命令に必要なチップ間通信バンド幅が大きくなりすぎるので、4 サイクル単位の固定長ベクトル命令とし、レジスタ、メモリ等のアクセスには連続アクセス、固定アドレス、ストライドアクセス、間接アクセス等のモードをつけた。また、全ての演算器は固定かつ同一ステージのパイプライン動作をする。演算器の内部ステージは 1 段であり、4 サイクル後の次の命令では前の命令の実行結果を直接利用することが可能である。このために中間結果にレジスタを消費することがなく、また命令スケジューリングの必要もほぼなくなっている。

浮動小数点加減算器は倍精度である。乗算器は単精度 × 倍精度であり、倍精度語の下半分を演算器に供給する制御フラグがある。これにより、2 命令で倍精度乗算を行うことができる。また、演算結果からフラグビットを生成してマスクレジスタに格納し、その値によって結果をメモリ等に格納するかどうかを制御できる、これにより条件付き実行ができる。

レジスタファイルは 3 ポート (2R1W) 32 語である。これに対してメモリは 1 ポート 256 語であり、読み書きは同時には行えないし書いた直後の命令では読出しはできない。T レジスタは汎用レジスタのポート数の不足を補うための補助レジスタである。

チップ物理設計は Alchip 社、製造は TSMC の 90nm プロセスで行った。チップサイズは 18mm 角とかなり巨大であるが、消費電力はフル動作時の実装で 65W とそれほど高くはない。クロックが 500MHz と低いのでこの程度であろう。むしろ、クロックが低い割には消費電力が大きくなっている。これは、チップ上のトランジスタのかなり大きな割合が演算器であり、毎サイクルスイッチングするからである。

しかし、演算能力あたりの消費電力では、倍精度で 0.25W/Gflops となり現時点で存在している何に比べても数倍良い。例えば Intel や AMD の 4 コアプロセッサは 40Gflops に対して 100W 近くであり、ほぼ 10 倍の電力を消費する。

5　アプリケーション

さて、 GRAPE-DR で GRAPE がやってきたことができるのは当然として、他の何かの役に立つのか？という疑問をもつ読者もいるであろう。正直なところ開発者である我々も当初はどれくらいのアプリケーションがあるのだろうと思っていたが、詳細な検討を進めていくと予想以上に応用範囲が広いことがわかってきた。

個別の例をあげても意味がないので、以下では多少概念的な説明を試みる。

PAX の開発をした星野・川合らは、並列化できるシミュレーションプログラムを以下の 3 種類に分類した (5)

1. 近接型 (連続型): 計算要素 (例えば格子点) が近隣の要素とのみ相互作用する
2. 遠隔型 (粒子型): 計算要素 (例えば粒子) が他の全ての要素と相互作用する
3. 不規則型: 単純な空間構造をもたない。回路シミュレーション等

最近では要素間に全く相互作用がないものも並列アプリケーションの例に数えられるが、それは (メモリが足りるなら) GRAPE-DR で実行可能なことはいうまでもない。

上の分類では、不規則型はケースバイケースだが、粒子型は基本的に GRAPE-DR で実行できる。これは GRAPE が有効であったのと全く同じことである。連続型については、相互作用の複雑さ次第、ということになる。空間 3 次元の計算では、空間差分の次数を上げると計算量は急速に増大する。そうなると、意外に高い実行効率がでるケースもある。

なお、空間差分の次数を極限まであげたものは FFT だが、これには適していない。これは、GRAPE-DR が適していないというよりも、クラスタ構成のホスト計算機自体が大規模 FFT には不向きであるからである。不向きな理由は、FFT では大域的な通信が高速に行えることが必須だが、現時点ではギガビットイーサネットや、あるいはインフィニバンドといった高価なネットワークを使っても演算速度に対して通信速度は全く不足なことであり、GRAPE-DR によって演算速度が向上しても FFT は速くならない。 GRAPE-DR の速度に見合ったノード間通信速度が得られるならば FFT にも有効である。

上の近接型、遠隔型の中間的な特性をもつものに密行列の処理がある。密行列の反転、固有値計算等はアルゴリズムの進歩によって殆どの演算が行列乗算になるようになっている。行列乗算の計算量はデータ量の 1.5 乗であり、遠隔型の 2 乗、近接型の 1 乗の中間にくる。外部メモリ、内部のローカルメモリ、外部メモリとの通信速度、ホスト計算機との通信速度を適切に選択することで、それほど無理をしないで行列乗算で理論ピークに近い性能を実現できる。転送速度を増やすと必要なメモリ量をその 2 乗に反比例して減らすことができるが、例えば PCI-Express のような新しい技術の場合には実効的な転送速度と名目ピーク速度のギャップがかなり大きくなる傾向があるので注意が必要になる。

密行列計算は、局在基底を使った量子化学計算等でも発生する、応用上は極めて重要な処理である。また、いうまでもなく LINPACK ベンチマークでも重要な役割を果たす。

6　類似アプローチとの比較

さて、1 チップに非常に多数の演算器を入れる方向を目指すアーキテクチャは GRAPE-DR の他にもいくつかある。以下では、その例として FPGA による再構成可能計算、 GPGPU、 類似の超並列 SIMD プロセッサ (6)、タイルプロセッサ等の MIMD 超並列プロセッサをとり上げる。

6.1 FPGA

FPGA による再構成可能計算については天野の講演で詳しく述べられるはずなのでここでは繰り返さない。ビット長が短いデータに対する処理では FPGA は優れている。しかし、浮動小数点演算、特に倍精度演算が必要になると、専用乗算回路を持つ大規模な FPGA でも実装可能な演算器の数は少なく、汎用マイクロプロセッサに比べて動作クロックも桁で低いために性能で上回るのは困難になる。 GRAPE-DR の場合には初めから多数の演算器を持ち、クロックもそこそこなので性能はかなり有利になる。

外付メモリバンド幅については FPGA でも高くするのは困難であり、適した問題の性質は同様になる。

6.2 GPGPU

GPGPU の歴史、発展については 8 月の HPC フォーラムで伊野から発表があった。GPU は元々は画像表示のための専用回路であり、座標変換、Z バッファ、テクスチャマッピング等を専用回路で行ってきたが、これらの処理が次第に複雑になったためにプログラマブルなプロセッサに多数が複合した動作をさせることで高速性と柔軟性を両立させようとしている。その結果、汎用計算にも対応できるものになってきた。

これは GRAPE の進化と並行的なものであるが、いくつかの違いがある。

- GPU は高速な外付メモリを持つ
- GPU は GRAPE-DR よりもはるかに複雑なハードウェアであり、チップ当りの演算性能は高くない
- GPU ではメモリバンド幅に対してそれほど演算性能を高くできないので、今後の発展の方向は不明である。実際、 90nm の nVidia G80 から 65 nm の G92 になって、トランジスタ数は 1.5 倍になったにもかかわらず演算器の数は 128 から 112 に減少している。それでもチップ面積は 300 平方ミリに及ぶ巨大なものである。
- GPU はその設計がグラフィック処理以外の具体的なアプリケーションを念頭においていないので、意外なところで性能低下がある
- GPU は大量生産されるのでチップ単価が安い
- GPU は 1 年程度のサイクルで新製品がでるので、テクノロジー的には有利である

それぞれの要因はかなり大きなものである。例えば、 GPU のオンボードメモリは GRAPE-DR のオンボードメモリの 20 倍以上高速である。これに対して演算器の数は GRAPE-DR と nVidia の現時点で最新プロセッサである G92 を比べると、 GRAPE-DR のほうが 4 倍以上多い (その代わりクロックは 1/3)。倍精度演算に関しては 2007/11/7 現在では G92 の性能は全く不明である。

性能低下は、行列乗算、N 体計算等様々な問題で報告されている。現在のところ、 nVidia G80 プロセッサの重力多体問題での性能は、 GRAPE-6 チップ 4 個のボードよりも若干遅い程度であり、 GRAPE-DR チップに比べてもかなり低くなる。しかし、価格の違いは 10 倍近い。

一般的なテクノロジーの方向、特に大量生産によるコストメリットと最新のテクノロジーが使えるメリットは GPU のほうで極めて大きく、 GRAPE-DR のような HPC 専用プロセッサでは勝負にならないようにも思われる。

しかし、GPGPU の最大の敵は汎用マイクロプロセッサである。例えば、4 コアの Intel/AMD プロセッサは単精度演算では 100Gflops 近いピーク性能をもち、nVidia G80 の 256 Gflops と大差ない。バンド幅が低い PCI-Express でつながっていることによる性能低下を考えると、 GPU の側で演算するのと CPU の側でそのままやるのでどちらが得かは難しい、というより、 GPU で性能向上ができるケースはそれほど多くない。

この意味では、 GPGPU の将来がどれほど明るいかは自明ではなく、これは GRAPE-DR も同じかもしれない。

6.3 ClearSpeed CSX600

ClearSpeed CSX600(6) は GRAPE-DR に極めて良く似たアーキテクチャをもった SIMD 超並列チップである。大きな違いは演算器の数 (96) と動作クロック (200MHz) であり、このために CSX600 は理論ピーク性能が 1 チップ 50Gflops 程度と低く、発表時点でマイクロプロセッサとの競合が難しいものになっていた。これは、テクノロジーが 1 世代古いこともあるが、 GRAPE-DR に比べるとチップ面積に対する演算器の割合は小さい。

GRAPE-DR では、かなり極限まで演算器の割合を増やすことでピーク性能を上げている。

6.4 タイルプロセッサ等

Intel が発表した 80 コア超並列プロセッサ等、比較的単純だがそれぞれが独立にプログラムを実行するプロセッサを多数集積したチップでアプリケーションを実行する研究はかなり以前から非常に沢山ある。

現在のところ実用的なアプリケーションで性能がでたという報告がないので、評価は難しい。

7 HPC の今後

現在はベクトル並列計算機の性能向上が停滞期にはいってから 30 年、マイクロプロセッサの性能向上が停滞期に入ってから 15 年程度たった時期になる。つまり、ベクトル並列計算機からマイクロプロセッサベースの分散メモリ計算機への移行にあたる、新しいアーキテクチャへの移行が始まっているはずの時期であり、その候補になりえるものは前節でみたようにないわけではない。

しかし、汎用マイクロプロセッサはベクトル計算機が持っていなかった大量生産によるコスト・性能面のメリットを持つため、理論的には優れているアーキテクチャでもかなりの投資をする、あるいは非常に大きな利点を持つものでなければ汎用マイクロプロセッサと競合できない。このことが、あまり有望な新アーキテクチャがないように見える大きな理由であろう。

これは、言い換えると HPC 専用プロセッサはコストメリットがない、ということである。しかし、良く考えてみるとこれはそれほど自明なことではない。例えば日本の次世代スーパーコンピュータープロジェクトは年当り約 150 億もの予算を使っている。これに対して例えばグラフィックチップ専業メーカー nVidia の年間総売り上げは 2000 億円程度であり、10 倍程度の差でしかない。つまり、次世代スーパーコンピューターが、商品化されてその総売り上げが投入された税金の 10 倍程度になるなら、グラフィックチップの業界最大手と同等の売り上げになるわけである。ちなみに、インテルの売り上げは nVidia の数十倍である。

そんな売り上げは夢のような話で現実的ではない、という意見もあるかと思うが、それは、実際に開発されるものに市場競争力がない、ということを前提に考えているからである。HPC マーケットは意外に大きく、 Top 500

リストに現われるものだけでも Top 1 の機械の 20 倍以上の能力になる。さらに小規模なサイトまで数えるなら 2 桁程度は大きいであろう。現在は、小規模なシステムは殆ど PC クラスタになっている.。PC クラスタよりも価格性能比でメリットがあり、いくつかの重要なアプリケーションが動作するなら期待されるマーケットは小さくはない。

もちろん、このためにには、ある程度量産したなら単体でマイクロプロセッサよりも価格性能比が良い、という条件が必須である。これはしかし、マイクロプロセッサのトランジスタ利用率をみれば現時点ではそれほど難しい条件ではないはずである。 GRAPE-DR およびその後継では、我々はそういう方向を目指している。

参考文献

[1] Y. Iwasaki, T. Hoshino, T. Shirakawa, Y. Oyanagi, Qcdpax: A parallel computer for lattice qcd simulation, Comp. Phys. Commun. 49 (1988) 449–455.

[2] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, M. Umemura, A special-purpose computer for gravitational many-body problems, Nature 345 (1990) 33–35.

[3] J. Makino, M. Taiji, Scientific Simulations with Special-Purpose Computers — The GRAPE Systems, John Wiley and Sons, Chichester, 1998.

[4] 伊藤智義, スーパーコンピューターを 20 万円で創る, 集英社, 東京, 2007.

[5] 星野力, PAX コンピュータ, オーム社, 東京, 1985.

[6] ClearSpeed Technologies, plc., CLEARSPEED WHITEPAPER: CSX PROCESSOR ARCHITECTURE, 2007.