

# スパコン開発からサイエンスまで — GRAPE プロジェクトの経験から

牧野淳一郎

東京工業大学大学院理工学研究科

理学研究流動機構

シリコン超集積化システム第 165 委員会第 67 回研究会 2012/10/25

# 講演概要

- 自己紹介
- シミュレーション研究の方法
- GRAPE プロジェクトの歴史
- エクサスケールと今後
- スパコン開発と半導体
- まとめ

# はじめに：牧野って誰？

- 職歴

2011/4- 東工大理学研究流動機構 教授

2006/6- 国立天文台理論研究部 教授

1999/4- 東京大学大学院理学系研究科天文学専攻 助教授

1994/4- 東京大学教養学部 助教授

1990/4- 東京大学教養学部 助手

天文(天体物理)学者？

- 学歴

1990/3 東京大学大学院総合文化研究科広域科学専攻博士課程修了

1987/3 東京大学大学院総合文化研究科広域科学専攻修士課程修了

1985/3 東京大学教養学部基礎科学科第二卒業

# 牧野って誰？(続き)

研究はどんなことをやってきたか

- 主に理論・シミュレーションによる天体形成・進化の研究
  - \* 大規模構造の形成・銀河形成
  - \* 銀河中心・球状星団の力学進化
  - \* 惑星形成
- シミュレーションのための計算アルゴリズムの研究
- シミュレーションのための計算機の開発

# 計算機開発

20年くらいやっている。

- 1989- GRAPE (Gravity PipE):  
重力多体問題専用計算機
- 2004- GRAPE-DR (Greatly Reduced Array of Processor Element with Data Reduction):  
「汎用」アクセラレータ

# シミュレーション研究

シミュレーション: 数学モデルの方程式に対して、数値解を与えること

- 利点: 解析的に解けない方程式でも答はでる
- 問題点: なんでも解けるほど計算機は速くない。そのために答が間違ってる (信用してはいけないところの答を使ってしまっている) ことも多い。

なので、「より信用できる答を出す方法の研究」は極めて重要なテーマになる。

- 研究者個人にとって: 他の人と戦うための強力な武器
- 分野全体にとって: 確実にできることが広がる有用な成果

というわけで、方法の研究に結構集中してきたと思う。他の人より良い道具をもっているとできる研究はいくらでもある。

# 方法の研究等

1. シミュレーション用のプログラムを書く (デバッグする)
2. シミュレーション用の新しい計算方法を研究する
3. シミュレーション用の計算機を作る
4. シミュレーション用の計算機のための予算をなんとかする

# プログラムの開発

- 最近どんどん計算機が使いにくくなってきているので大変
  - 沢山の計算機の上での並列プログラム
  - GPGPU の利用、CPU 内複数コアの利用、キャッシュの有効利用、、、その他もろもろ
- 分野によっては、多数の研究者からなるチームでプログラム開発
- 方法やできる研究が制限される危険性がある。あんまり良い方向ではない



# 新しい計算方法の研究

- 計算機シミュレーションはそんなに長い歴史はない(せいぜい70年)。
- なので、方法はまだまだ未熟。改良の余地はいくらでもある
- 数学的な定式化、、物理的モデル、計算アルゴリズム、と様々なレベルでの研究がある。

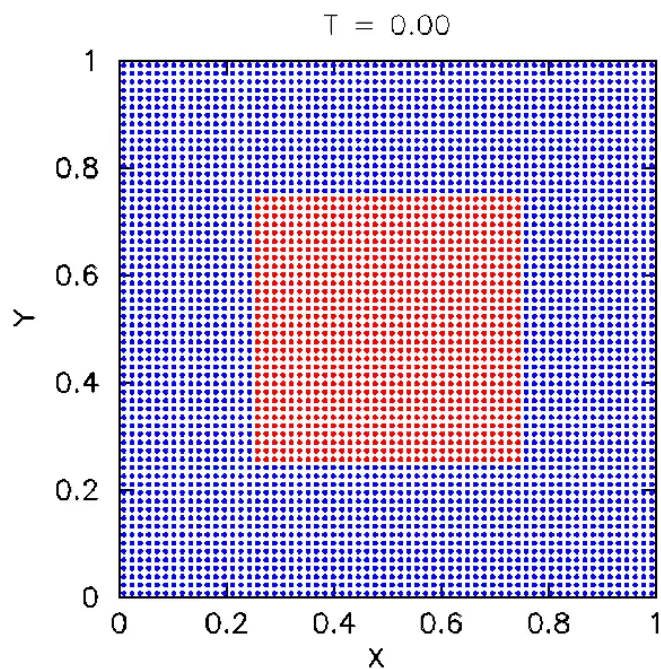
これは普通の理論研究に近い進めかた。解決したい問題があって、色々考えてアイデアをだして、それが実際に上手くいくかどうかさらに検討、という感じ。

# 方法の研究の例

最近やっていること (Saitoh and Makino 2012)

SPH法 (粒子によって流体を表す方法) の改良

密度が違うだけの2種類の流体



従来の方法

改良した方法

# GRAPE プロジェクト

- 基本的アイデア
- GRAPE-1 から 6 まで
- GRAPE-DR
- GRAPE-8

# 基本的アイデア—近田提案

1988年、天文・天体物理夏の学校

88/11(2)

A. S.

近田

「計算機を使う，計算機の技術を使う」

近田義広 (国立天文台・野辺山)

観山さんは理論の立場から見ると，現在の日本ではスーパーコンがどんどん使えるからイニシアチブをとる良い機会だと話されました。私は，その素晴らしい計算機を作り出す技術をうまく使いこなし，天文に応用すれば，計算機の利用者に止まる場合よりもっと大きな成果を期待できるぞ！ということをお話しします。

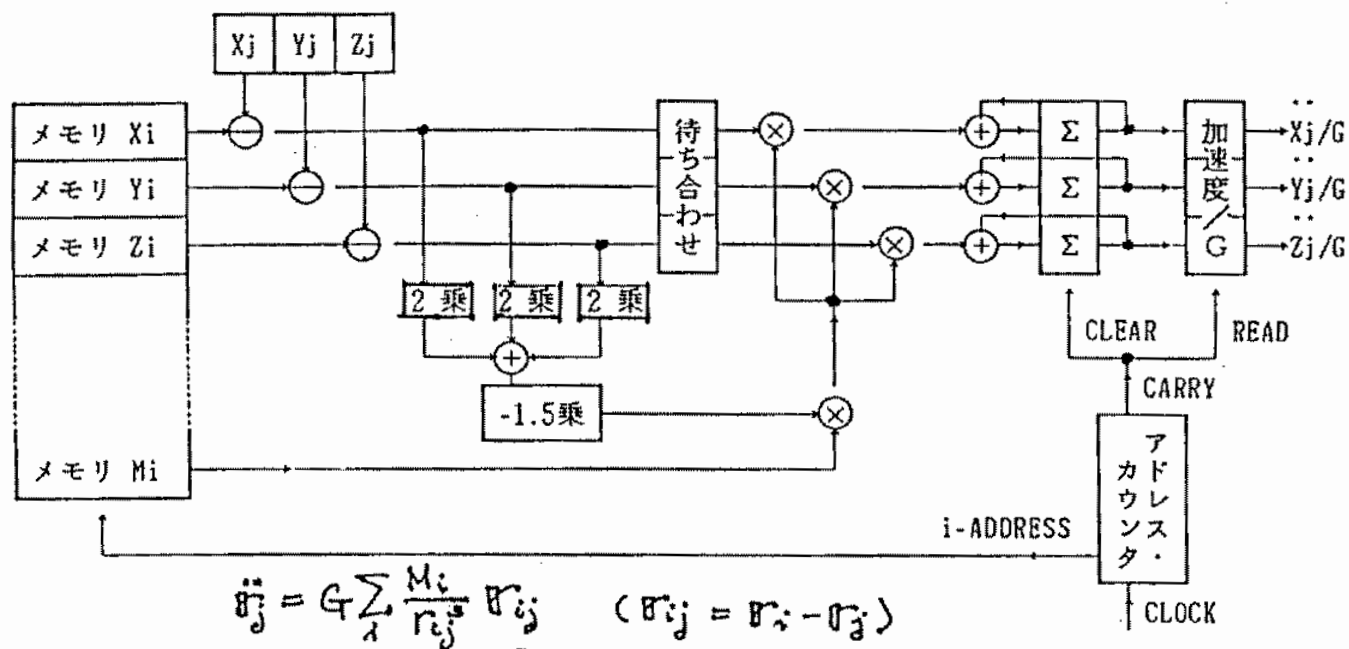
- 計算機買うんでなくて作れ!
- 近田さんは電波天文学の専門家。データ処理 (相関処理) 用の計算機を開発した。
- 1982年に100GOPS (Cray XMP が1Gflopsの時代)

# 近田提案のポイント

- 計算機は10年で100倍速くなる
- 同じ製品系列だと10倍しか速くならない
- 専用化すれば2－3桁価格性能比をあげられる
- 多体問題なら 400万円 250 Mflops
- 数億円なら 100 Gflops
- 「但し、近田電子製作所の見積もりは甘いという声もあることを付け加えます」

上の3個(と一番下)は今も変わらない

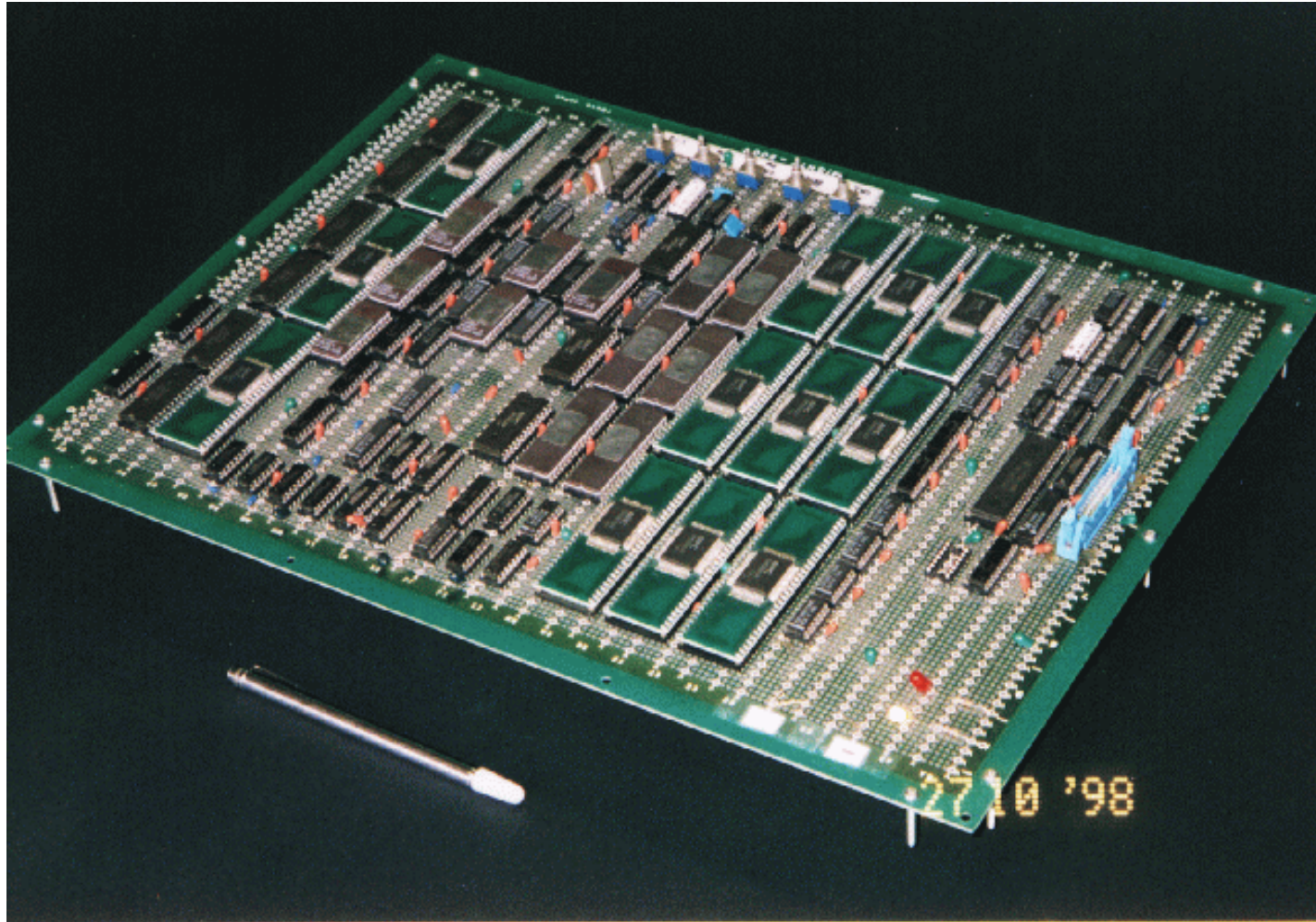
# 近田さんによるパイプライン概略図



+, -, ×, 2乗は1 operation, -1.5乗は多項式近似でやるとして10operation 位に相当する。  
 総計24operation.  
 各operationの後にはレジスタがあって、全体がpipelineになっているものとする。  
 「待ち合わせ」は2乗してMと掛け算する間の時間ズレを補正するためのFIFO(First-In First-Out memory).  
 「Σ」は足し込み用のレジスタ。N回足した後結果を右のレジスタに転送する。

図2. N体問題のj-体働く重力加速度を計算する回路の概念図。

# GRAPE-1(1989)



当時M1の伊藤君(現在千葉大工学部教授)が作った。

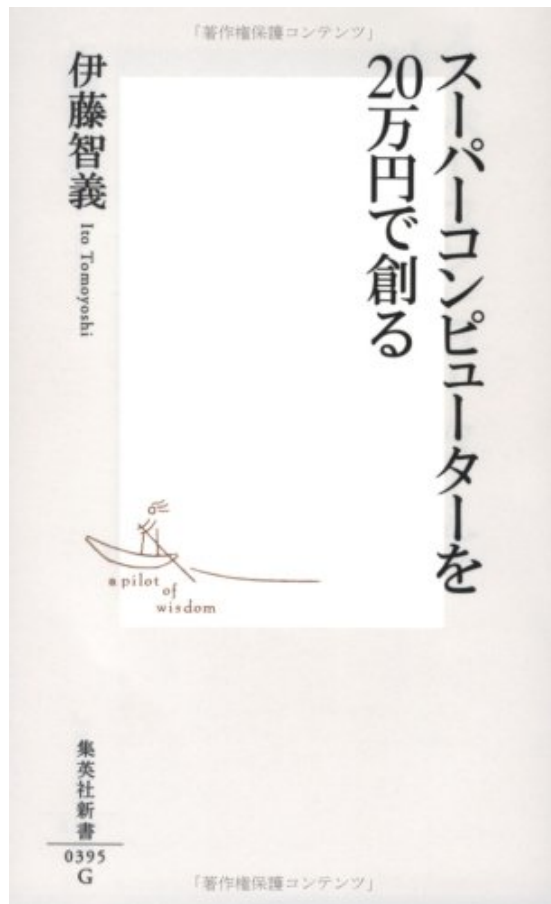
# 近田さんによる当初案との違い

中間での語長を8ビット、座標表現、加速度の積分は固定小数点でそれぞれ16, 48ビットにした

- 掛け算しないところは語長長くてもハードウェアは(それほど)大きくなる
- 2粒子間の力の必要な精度は低い。多数の粒子があることで統計的に高い精度が出れば十分(な場合がある、ということを経験的に示した)。
- 銀河合体による楕円銀河の形成(奥村ら)
- 「激しい緩和」(船渡ら)



# 伊藤君の本

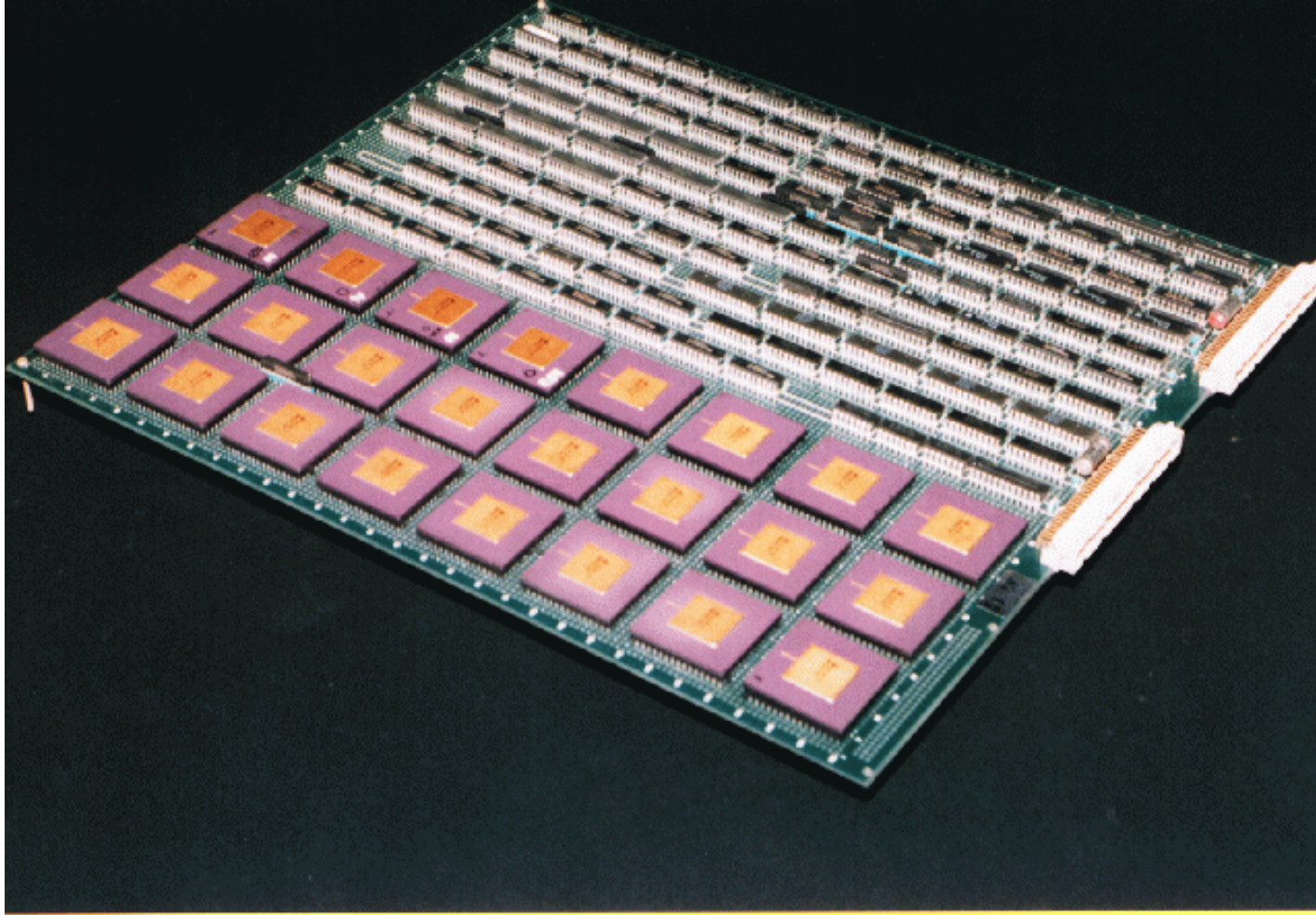


- 伊藤君はコミック「栄光なき天才たち」の原作をやった
- 数年前に本出した
- どんなふうの開発進んだか、は本を参照

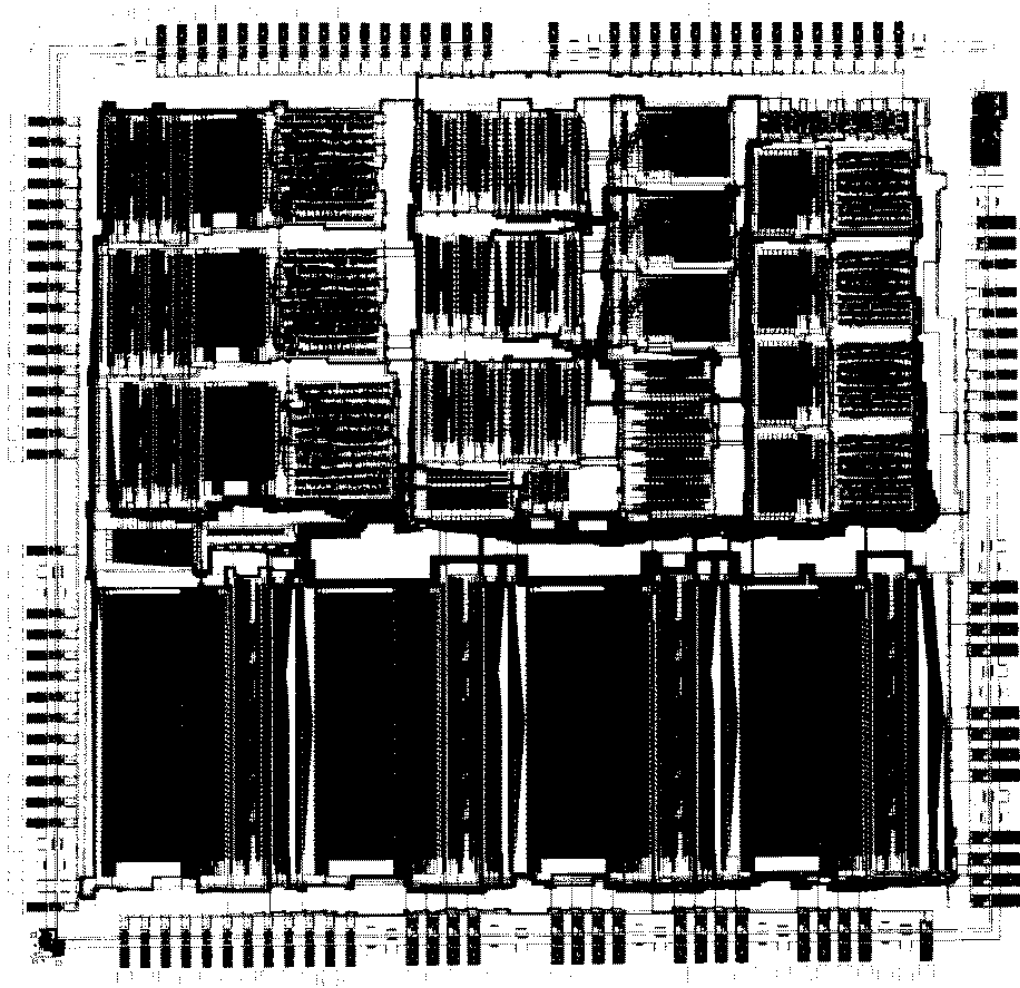
# GRAPE-3 — 低精度計算用 LSI

- 大雑把には GRAPE-1 パイプライン (但し、大きな ROM は使えないので対数での加算の実装は全く違う)
- 仕様決定、シミュレータ (C で記述) は牧野
- 論理設計以降は富士ゼロックス (橋本、富田)
- SCS Genesil で設計
- ファブは NS.  $1\mu\text{m}$
- ボードは奥村
- 1千万ちょっとの予算で 20Gflops を実現。

# GRAPE-3(1991)



# GRAPE-3 chip



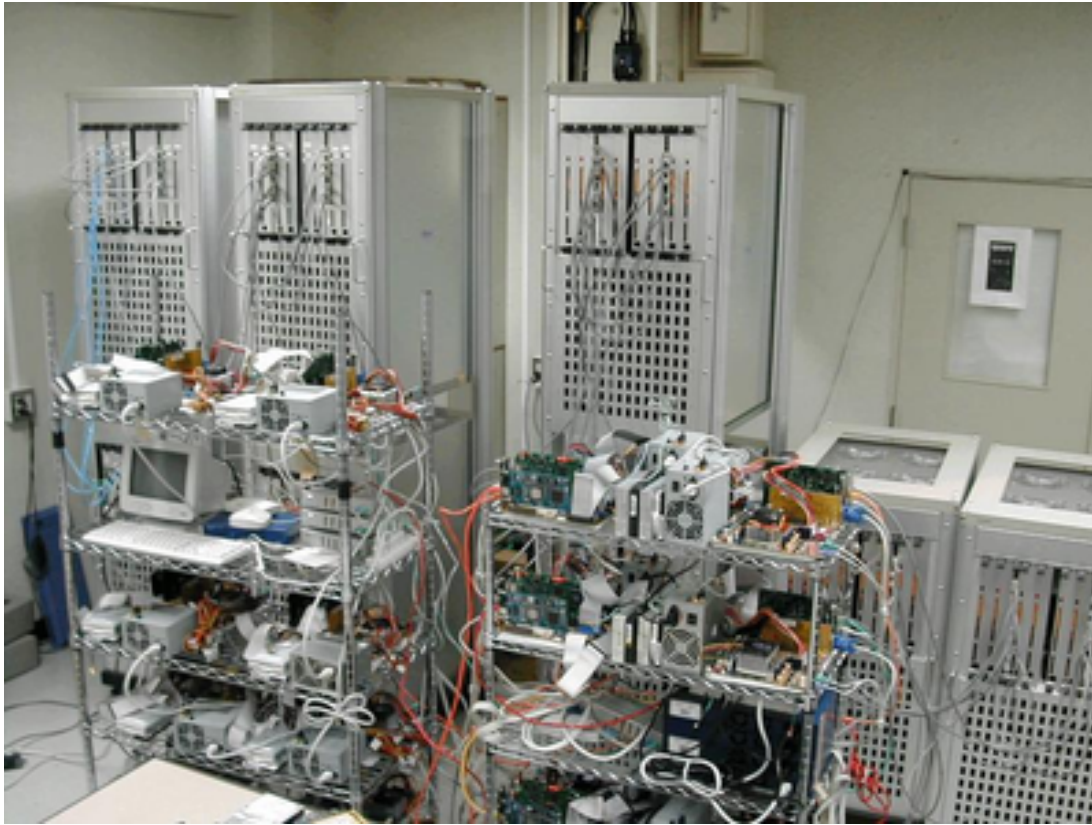
2 mm

# GRAPE-4(1995)



ファブは LSI.  $1\mu\text{m}$  で 2 チップ。難しいほうは泰地、簡単なほうを牧野が設計。1 Tflops, 1800 custom LSIs

# GRAPE-6 system



2002年の 64 Tflops シ  
ステム

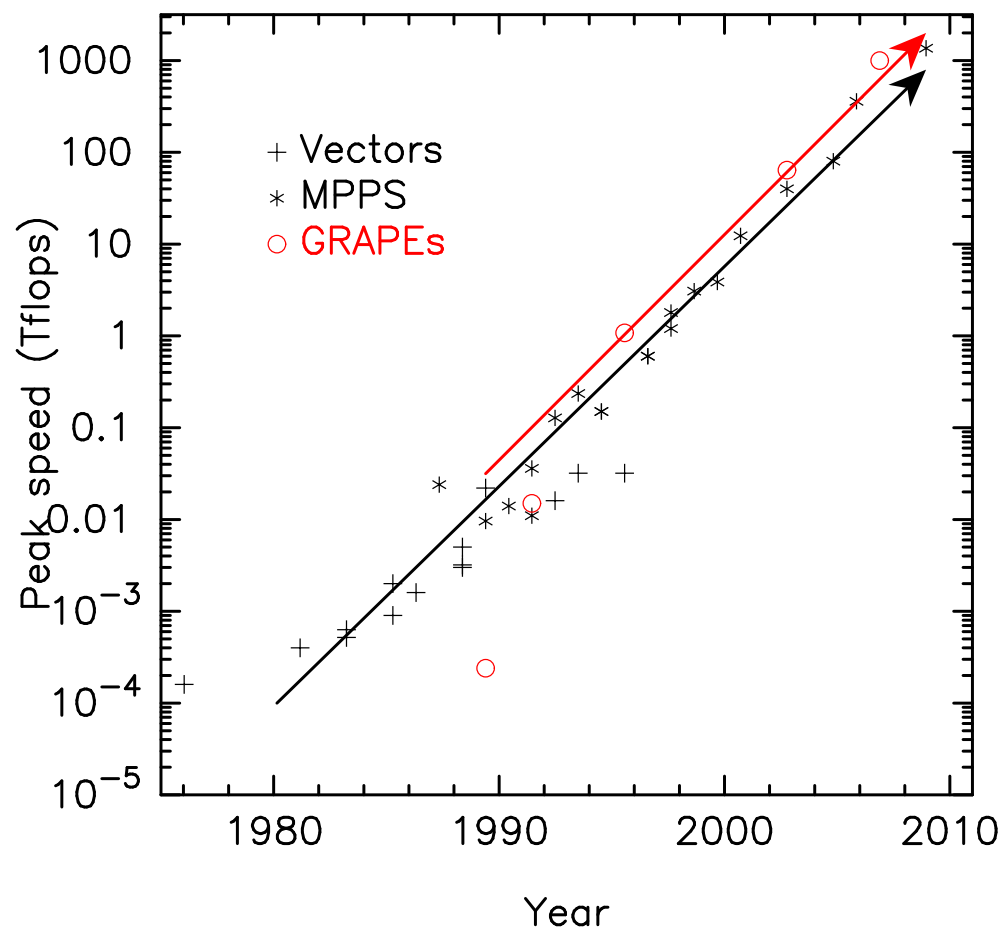
4 ブロック

16 ホスト

64 プロセッサボード

2048 カスタムチップ

# 性能の歴史



GRAPE-4, 6 は当時世界最高速の汎用スパコンより速かった。

MDGRAPE-3 も。

開発コストはおおよそ 1/100

# 天文学以外への応用

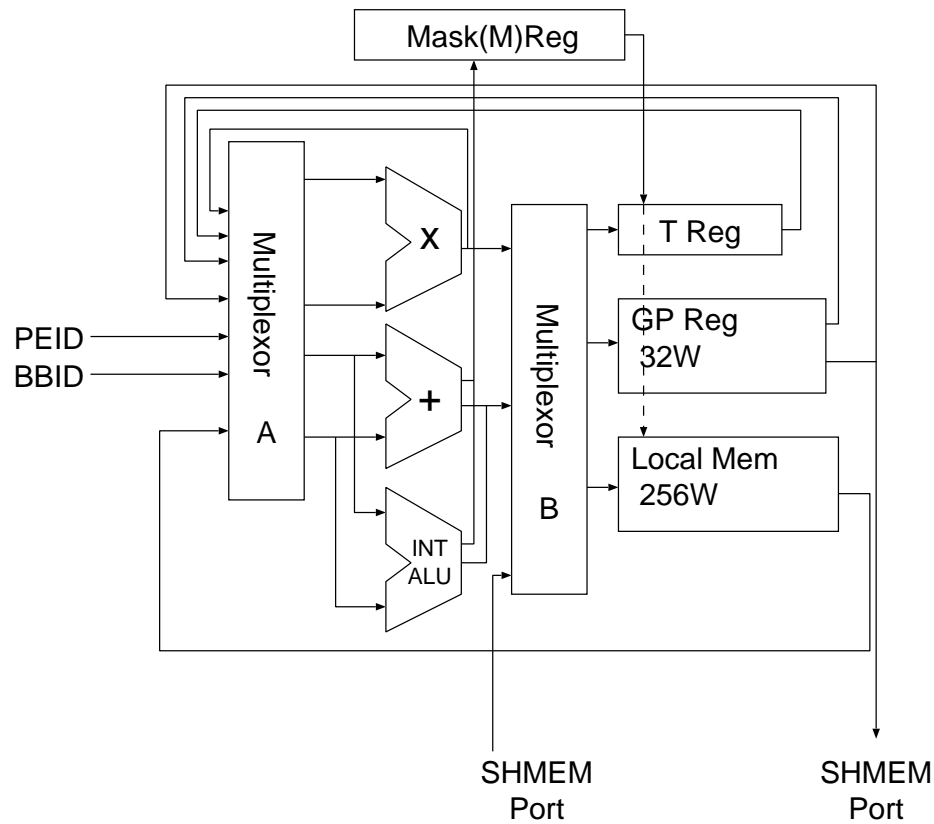
泰地さんが話すと思うので省略



# GRAPE-DR

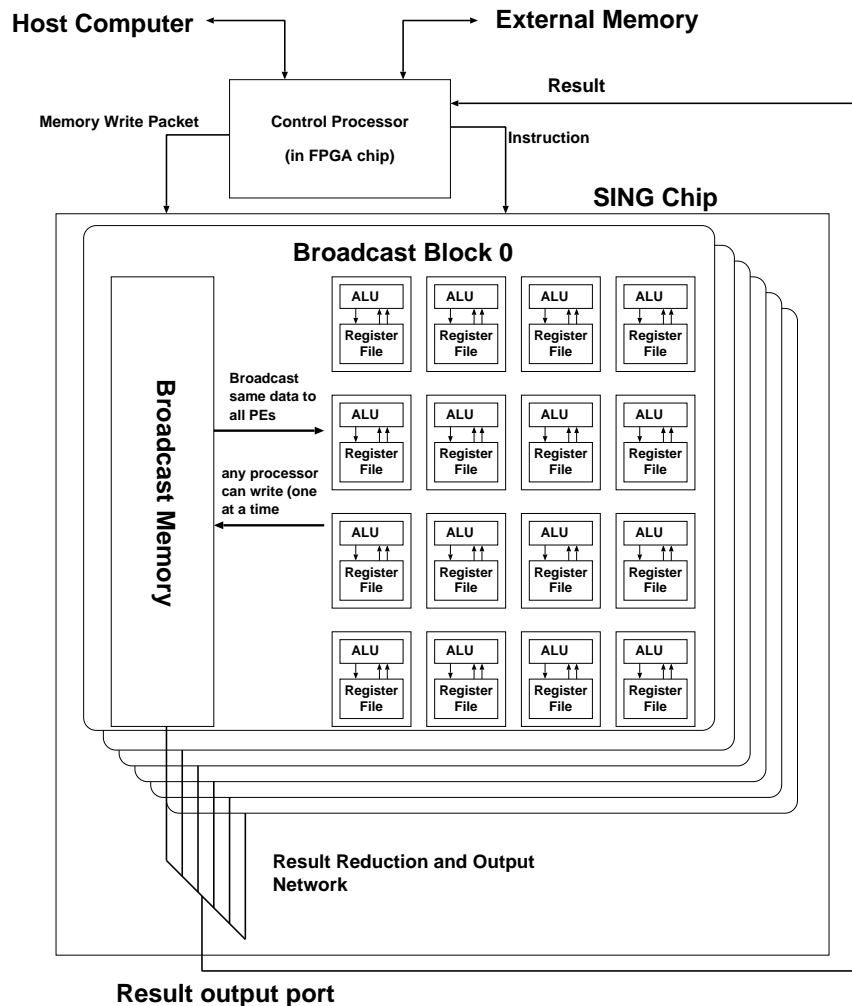
- 従来のGRAPE(重力計算の専用ハードウェア)とは違うアーキテクチャ
- 「汎用」ということで予算とる
- 非常に単純で小さなメモリだけをもつ演算プロセッサを多数並べるSIMD 構成
- 何故か最近のGPUと良く似た方向
- 開発予算: 振興調整費 (FY2004-2008)

# プロセッサアーキテクチャ



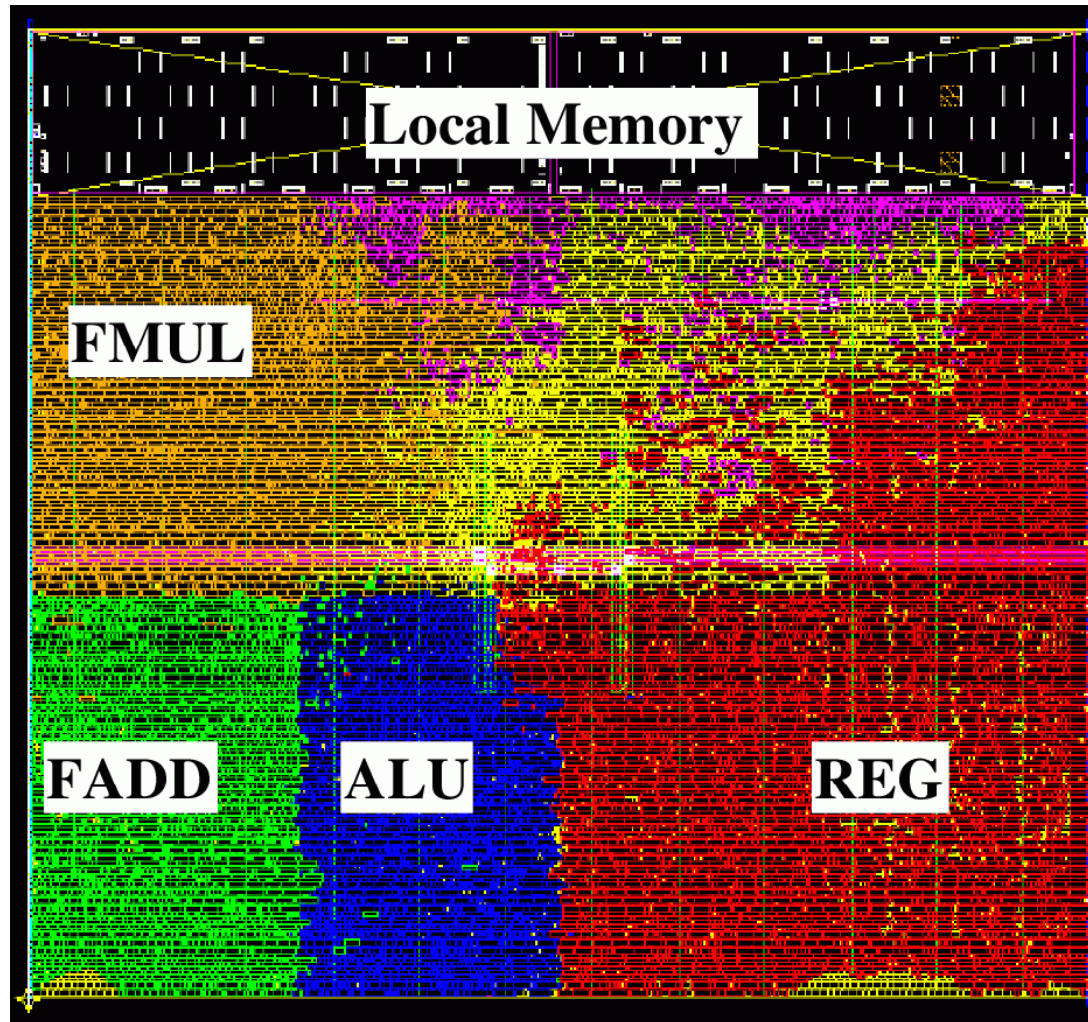
- 倍精度乗算器 (の半分)
- 倍精度加減算器
- 整数・論理演算器
- 32語のレジスタ
- 256語のメモリ
- 通信ポート

# プロセッサチップ構成



- 32個 (図では16) のプロセッサで「放送ブロック」を構成。放送ブロックは16個。
- 放送ブロック毎に共有メモリ
- 外からの入力データは全ブロックに放送
- 各ブロックからの出力データを合計等しながら出力可能

# GRAPE-DR プロセッサ



# プロセッサボード

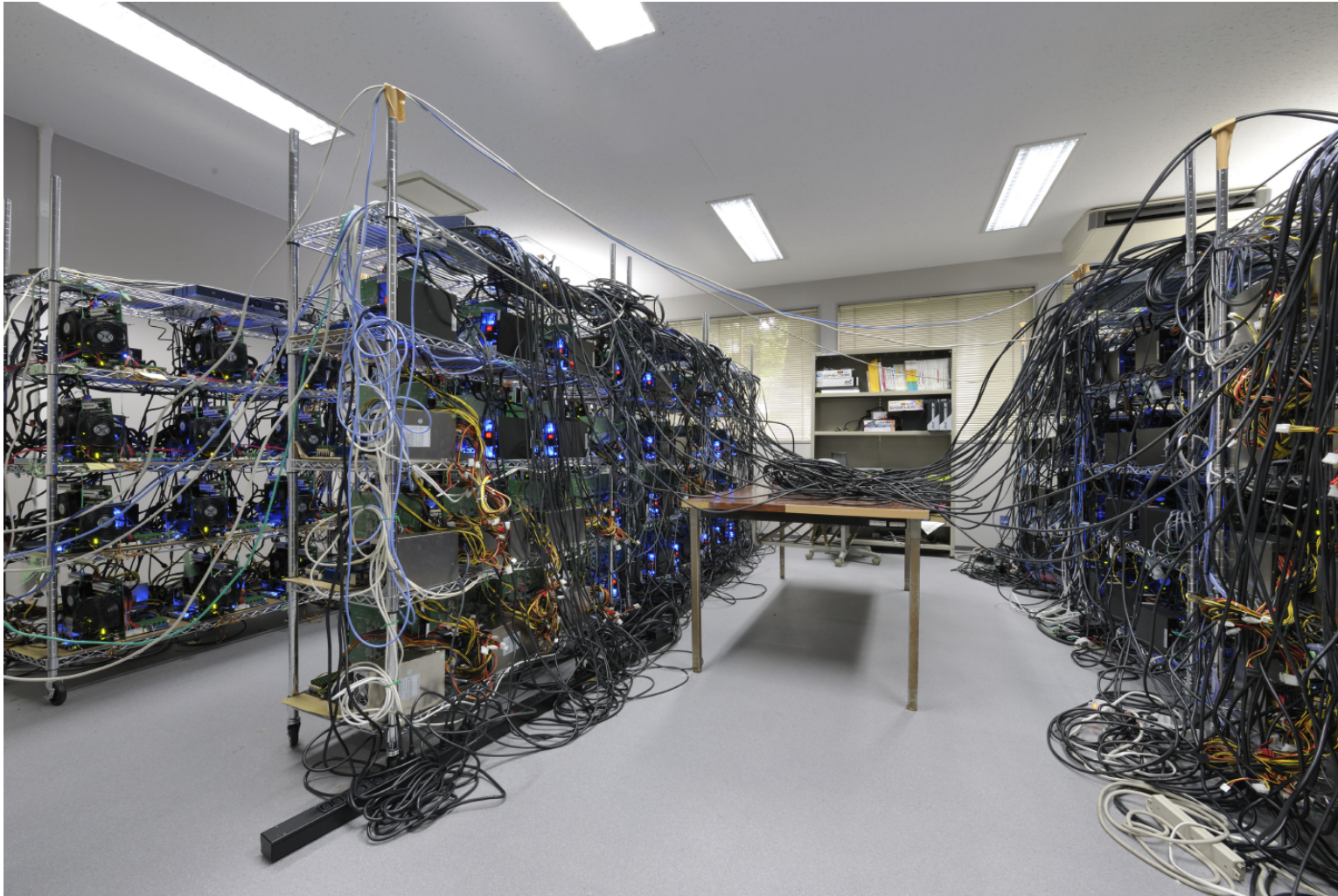


PCIe x16 (Gen 1) インターフェース

Altera Arria GX を DRAM 制御、通信インターフェースに利用。

- 最大電力消費 200W くらい
- 433MHz くらいまでは問題なく動作
- 400 MHz での理論ピーク性能は 819Gflops

# GRAPE-DR クラスタシステム (2008)



# Little Green500 (2010/6)

Listed below are the Little Green500's Top 10 most energy-efficient supercomputers in the world as of June 2010.

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	815.43	National Astronomical Observatory of Japan	GRAPE-DR accelerator Cluster, Infiniband	28.67
2	773.38	Forschungszentrum Juelich (FZJ)	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
2	773.38	Universitaet Regensburg	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
2	773.38	Universitaet Wuppertal	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
5	536.24	Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw	BladeCenter QS22 Cluster, PowerXCell 8i 4.0 Ghz, Infiniband	34.63
6	530.33	Repsol YPF	BladeCenter QS22 Cluster, PowerXCell 8i 3.2 Ghz, Infiniband	26.38

- 消費電力当りの性能でランキング
- ちょっと小さいめのシステムまであり、のリストで1位
- 何故かNHK まで取材にきた。

# GRAPE-8

## GRAPE:

- 自己重力粒子系シミュレーションの、重力部分の高速化
- 直接計算の他、ツリー法、FMM でも

## 歴史:

- 1989 年からの GRAPE プロジェクト: 1991 の GRAPE-3 から 2002 の GRAPE-6 まで専用プロセッサ LSI 開発。
- 2006 GRAPE-7: FPGA 利用、2008 GRAPE-DR: プログラマブルな SIMD プロセッサ

GRAPE-6 以降、専用 LSI は作っていない: **初期コストの指数関数的増加のため**



# Structured ASIC

カスタム LSI と FPGA (論理回路変更可能な LSI) の中間

	ASIC	FPGA	Str. A.
初期費用 (10 億円)	1	0.001	0.02
回路規模 (1 千万ゲート)	20	1	1
ゲート単価 (円/1000ゲート)	0.3	10	2

良くいうと、ASIC より初期費用安くて FPGA より量産費用安い

上の数字の例では、予算 2400 万から 12 億の間では「ベストソリューション」  
(2009 年の皮算用)

# そもそも作る意味はあるか？という問題

- GPGPU を買ってくるほうが安くないか？
  - 安い GPU 買ってくるならそう
- 重力だけで役に立つか？
  - 無衝突系だけだと？
  - 衝突系なら役に立つ

しかし、高精度版は電気も食うし大変。 Particle-Particle Particle-Tree 法 (押野他 2011) 用に作る。

# PPPT スキーム

$$H = H_{Hard} + H_{Soft},$$

$$H_{Hard} = \sum_{i=1}^N \left( \frac{p_i^2}{2m_i} - \frac{Gm_i m_{\odot}}{r_i} \right) - \sum_{i < j}^N \frac{Gm_i m_j}{r_{ij}} W(r_{ij}),$$

$$H_{Soft} = \sum_{i < j}^N \frac{Gm_i m_j}{r_{ij}} (1 - W(r_{ij})),$$

- P<sup>3</sup>M スキームと同様に近距離力と遠距離力を分割
- 遠距離力はツリー、近距離力は独立時間刻み直接計算
- 分割には高次スプラインを使う

# GRAPE-8 ボード



# GRAPE-8 ボード

- ボード論理回路設計は台坂
- PCI-Express インターフェースは FPGA を使う。  
GRAPE-DR の設計流用
- GRAPE-8 チップ2個搭載
- 理論ピーク性能 960 Gflops 相当、消費電力 47W。  
20Gflops/W (インターフェースFPGAが半分  
食う、、)
- PCIe インターフェースにはKFCR 社 GPCI を使用。  
Gen 1 8レーン

# ポスト「京」プロジェクト

どうしてそうなったのかよくわからないが、ポスト「京」のスパコンプロジェクトをまとめるお手伝いをしているので少しその話を

- 昨年度のアプリケーション部会
- 4つのアーキテクチャイメージ

# 昨年度のアプリケーション部会

- 去年の今頃に突然つくれという話が発生した。
- アーキテクチャ・コンパイラ・システムソフトウェアのほうは SDHPC 検討グループが横すべり
- アーキテクチャは戦略分野等から人を集めて急拠立ち上げ。9-11 月に集中的にミーティング、検討。

以下 11/15 の合同部会での牧野の報告から抜粋

# 予備検討の方針 (1)

- アーキテクチャ部会での検討では、B/F、メモリ量、ネットワークバンド幅等について非常に狭い範囲しか想定していないように思われた (手本にした米国のレポートがそもそもそうなっている)
- 消費電力当り性能は、エクサスケール実現にとって大きな壁である。
- B/F はアプリケーションの効率に大きく影響する一方、消費電力当り性能にも大きな影響をもつ
- メモリ量、ネットワークバンド幅も、大きく変えれば電力に影響する

アプリケーション側で、B/F、メモリ量、ネットワークバンド幅の必要量をだしておこう

B/F: Byte per flops. メモリバンド幅と演算速度の比



# What I learned from Steve Jobs

— Guy Kawasaki

1. **Experts are clueless**
2. **Customers cannot tell you what they need**
3. Jump to the next curve
4. The biggest challenges beget best work
5. Design counts
6. Changing your mind is a sign of intelligence
7. "Value" is different from "price"
8. A players hire A+ players

(いくつか省略)

# 予備検討の方針 (2)

- アプリケーション、アルゴリズムにより B/F 等への要求は変わる
- 特に、同じアプリケーションでもアルゴリズムが変われば、またアルゴリズムが同じでも系のサイズ等だけによっても要求は変わる

といった問題があるので、

- 各分野に、重要なアプリケーション (計算法、系のサイズ等含めて) を選定してもらい、それぞれについて要求を見積もってもらう
- それらをいくつかのタイプに分類できるかどうか検討する

という方針を考えた。

# 検討結果

(詳しい話は今日は省略)

- B/F とネットワークバンド幅は関係あり。
- B/F 要求高いがネットワークは弱くていいものはある。逆はない
- ランダムアクセス、非数値計算等、この軸ではよく表現できない要求もある
- 大雑把に数種類にタイプわけできそう

# タイプわけの観点

## 観察:

- B/F は 0.1 以上の高いものと、桁で小さいものに分かれる
- メモリ要求にも非常に幅がある

## 注意事項:

- 分野によってはまだ十分な検討が進んでいない
- 分野によっては、そもそもアプリケーション・アルゴリズムの進化が速いために定量的な要求を明確にしにくいところもある

# タイプわけの観点

「アプリケーションタイプ」でなくて「アーキテクチャタイプ」として  
みた。

- そのほうが物理的制約をイメージしやすい
- アーキテクチャ側との議論もしやすい？

# タイプわけ

以下の4タイプ

タイプ	B/F	メモリ量 (1TF)	消費電力 (1EF)	演算性能) (20MW)	バンド幅 (20MW)
ベースライン	0.1	10-100GB	20MW	1EF	0.1EB/s
SoC	4	5-10MB	2-5MW	4-10EF	16-40EB/s
アクセラレータ	0.001	1-10GB	4-10MW	2-5EF	2-5PB/s
バンド幅重視	1	1TB	120MW	0.15EF	0.15EB/s

# 最終レポートではこんな感じ(1)

## 2. サイエンスロードマップ「アプリケーションからの要求の概要」(9/9)

### ネットワークレイテンシ

タンパクMD	1時間ステップがマイクロ秒程度。同期等がこれより十分短い必要あり
格子QCD	大域縮約をマイクロ秒程度
他の多く	もう少し余裕あり

### ネットワークバンド幅

格子QCD	隣接ノードとの通信速度が B/F で 0.01 程度
大域FFT	バイセクションバンド幅で性能が決まる。普通の構成では効率 1% 以下 ハードウェアだけでなく、アルゴリズム面からの検討も重要

### メモリ容量・バンド幅

地震波動解析 圧縮性流体計算 有限要素解析の 防災・工学応用	100ペタバイト前後のメモリ、高いメモリバンド幅(B/F 0.5 以上)が必要
タンパクMD 格子QCD	メモリ必要量は極めて小さい。大きなバンド幅(B/F 1以上) が必要
大規模粒子系計算 量子化学計算	バンド幅、メモリ量とも比較的要求小さい

### ストレージ容量 速度

DNA	シーケンサデータ処理 50EB, 500TB/s 程度が必要
他の多く	1桁程度下の要求

### 多様な要求

- ・複数アーキテクチャも視野にいれる必要あり?
- ・メモリ・ネットワークバンド幅については新しいアルゴリズムの研究開発も重要

# 最終レポートではこんな感じ(2)

## 4. ロードマップ達成に向けて(アプリ要求性能)

▶サイエンスロードマップに基づいて2018年ごろのアプリケーションに必要な性能を調査

▶演算性能要求・総メモリ容量・演算性能あたりメモリ帯域・ネットワーク要求を調査

要求性能の解析結果

▶演算性能・メモリ容量・メモリ帯域に関する要求

▶演算性能は800PFLOPS~2500PFLOPS

▶メモリ容量は10TB~500PBの幅があり、帯域も1000倍程度の差がある

▶特徴的なもの

▶メモリ容量が少なくても良い: MD・気候・宇宙物理・素粒子物理

▶メモリ帯域が少なくても良い: 量子化学・原子核物理

▶メモリ容量・帯域が両方必要: 構造解析・非圧縮流体解析など

▶ネットワークに対する要求 ※トポロジに依存する部分もあり継続検討が必要

▶レイテンシ・帯域とも強い要求はないが、性能必要なアプリもあった

▶タンパク質の構造解析などでは1us以下の通信が必要な見込み

▶物質化学分野ではBisection帯域が必要なアプリもある

▶1us以下での高速な同期・放送・縮約などが要求されるアプリケーションもあり、専用のハードウェアによるサポートが必要になる可能性もある

▶ストレージに対する要求

▶要求容量に対して他の課題に比べて大きな課題はない

▶性能要求に対しては今後のストレージデバイス技術に応じて構成方法を検討

▶要求性能をトレンドから予想される性能にマッピングした(図1)

▶サイエンスロードマップの達成には、前スライドの4分類とも、技術トレンドから予想される性能よりも高い数値が要求されている

▶ロードマップ達成のためにアプリケーションの特性をさらに詳細化・定量化し、将来のスーパーコンピュータの設計目標を提示していくことが必要である

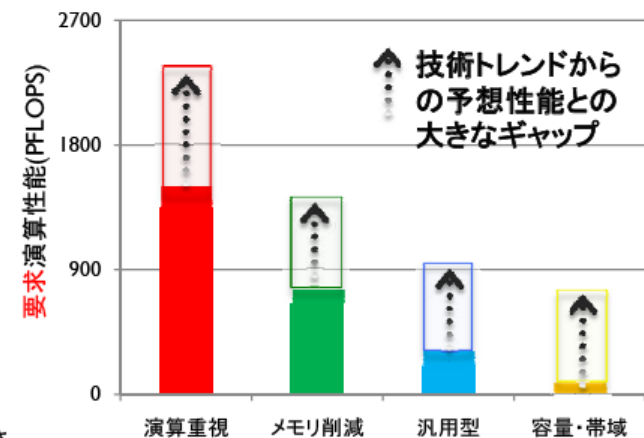
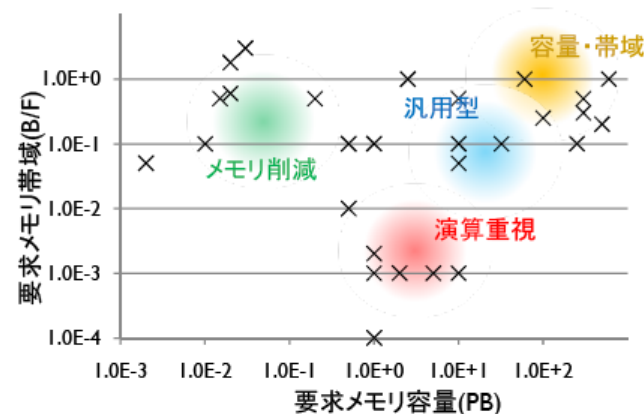


図1. 要求性能と予測性能の相関  
上: 各アプリの要求するB/F値とメモリ容量  
下: 各分類に対する要求性能値と予想性能



# 要するに

報告書での用語	アプリ部会のつけた名前	具体的イメージ
汎用 容量・帯域 演算重視 メモリ削減	ベースライン バンド幅重視 アクセラレータ SoC	「京」の延長 NEC SX-9 の延長 GRAPE-DR,GPGPU ...

# 今後の方向

- 今年度、来年度 feasibility study
- その後？5年くらいで開発？

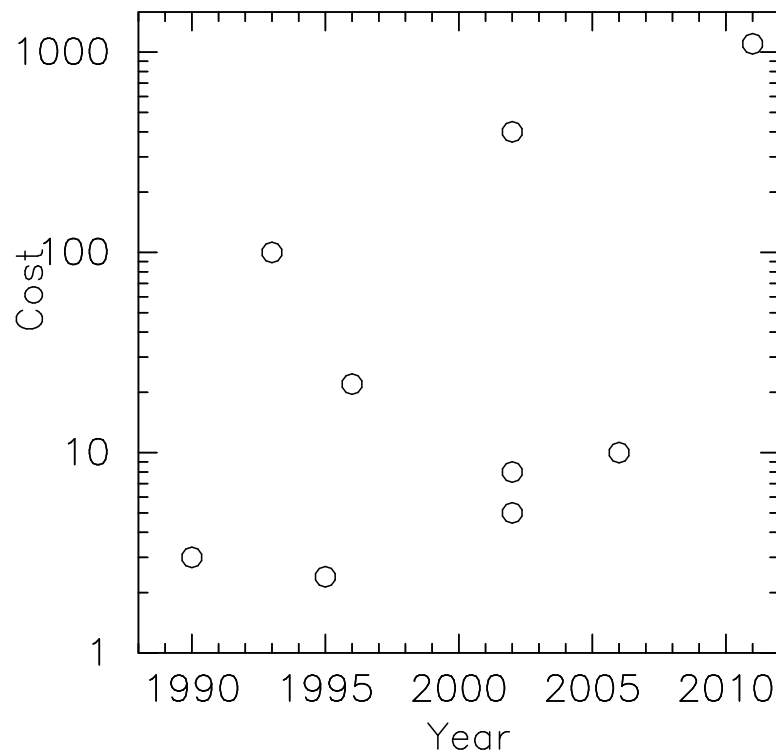
# 牧野の印象

- 目的が絞られていない
- なので、方針が定まりようがない
- GRAPE は単一目的、 GRAPE-DR も想定アプリケーションは少数。それでハードウェアも必要なソフトウェアも単純化した
- 複雑なハード・ソフトのほうが雇用創出にはいいのかもしれないが、、、

# 歴史的に見ると

## — 「計算速度世界一」の値段

- 1990? QCDPAX 3億?
- 1993 数値風洞 100億
- 1995 GRAPE-4 2.4億
- 1996 CP-PACS 22億
- 2002 地球シミュレータ 400億
- 2002 GRAPE-6, MDM: 5, 8億
- 2006 Protein Explorer 10億?
- 2011 「京」 1200億



# 歴史的に見ると — LSI 開発のコスト

- 1990 GRAPE-3  $1\mu\text{m}$  1000万
- 1992 GRAPE-4  $1\mu\text{m}$  1500万+2500万 (2チップ作った)
- 1996 GRAPE-6  $0.25\mu\text{m}$  1億
- 2004 GRAPE-DR 90nm 4億前後

汎用プロセッサはこの 1.5-2桁上。何故かは私にはよくわからない

# LSI 開発コストの動向

- 実はマスクコストはそんなに大きくない。少量試作ならマルチプロジェクトウェハが使える。
- IP が高い。メモリ、レジスタファイル、I/O セルといった基本的なものだけでも。CPU コアとかインターフェースをいれていくとすぐに億を超える
- 物理設計と検証の期間・人件費が高い。設計に人手がかかりすぎている。(中国・インドにアウトソースしてる)

# 個人的な意見

- 今後のスパコンの発展のためには、アカデミアで(汎用・専用両方の)先端プロセッサを多数(年に数個とか)開発できるためのサポート体制が必要
- VDEC では不足。開発予算とプロジェクト選定機能をもった組織ないしCREST のような予算枠
- 半導体設計(論理設計からタイミングクロージャまで)の自動化(超並列計算でとか)ができれば大きな競争力になるはず、、、

# まとめ

- GRAPE では、アプリケーションに対してアルゴリズムとプロセッサを合わせて開発。
- 汎用計算機に比べて開発費で大体 1/100。半導体の利用効率、電力効率もそれくらい。
- 半導体開発費の指数関数的上昇で、例えば天文学みたいな役に立たない分野で専用計算機を作るのは予算的に困難になった
- 汎用計算機用プロセッサは100倍コストかかり、プロセッサ開発はやはり困難に。
- LSI 開発コストのうち人件費の部分を下げる技術開発が重要ではないか？



# 20年後

- 計算能力が上がるのに対して使う需要はある。
- 何をやってるかはあんまり変わらないような気がする。
- デバイス技術がどうなるかによるので、、、
- 半導体の 20 年後は今とあまり変わらないとすると、、、

# 現在の計算機の電力効率

- 最近 10 年、今後 10 年の問題はとにかく電力
- 汎用プロセッサは無駄が多い
- まあ 100 倍くらい効率あげる余地はある
- でも、例えば浮動小数点処理に対して、効率向上の限界はある

# 今後の方向

- 動作速度がいくらさがってもいいから、スイッチ素子の演算あたりの消費電力を下げて欲しい。
- 並列度の問題は、動作速度が遅いこと前提ならデータフローで、、、
- 時間方向の並列化というアプローチもあるし、、、
- あとはやはり設計の自動化を、、、

# おまけ： 前回の経緯、 アクセラレータ

## アーキテクチャ案の概要 (アクセラレータ) 【月末時点】

- 基本仕様
  - アクセラレータ部の理論ピーク性能：10PFLOPS
  - 汎用サーバ (ホスト) のI/Oインターフェースに接続
  - ホストより指定された演算処理をアクセラレータで実行し、結果をホストに格納

アーキテクチャ案		国立天文台	東京大学
アクセラレータ	アーキテクチャ	SIMD型 プロセッサアレイ	
	プロセッサチップ数	約 15,000	約 20,000
	ポート数	4,000	2,500
ホストサーバ数		2,000	2,500
アクセラレータ部 消費電力		<div style="border: 1px solid black; padding: 2px; display: inline-block;">-10 MW</div> <small>※平成24年6月公開時の注意書き 消費電力については、10MW以下、10-20MW、20-30MWの範囲でまとめたもの。提案は、ホスト部を除いて、1.7MWであった。</small>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">-10 MW</div> <small>※平成24年6月公開時の注意書き 消費電力については、10MW以下、10-20MW、20-30MWの範囲でまとめたもの。提案は、ホスト部を除いて、0.68MW (案1)、0.88MW (案2)であった。</small>

(2012/6 公開)

# 概念設計評価時の判断

- アクセラレータ案の説明資料には消費電力「-10MW」という謎の表現が。議事録には「10MW以下くらい」とある。
- 実際の提案の数字は公開時注意書きにあるように0.88-1.7MW

# アクセラレータを採用しなかった理由

## 公式の説明

2者のシステム構成により、目標性能達成の見込みが確認できたため、アクセラレータの採用は考慮しない

- 目標 = LINPACK 10PF + HPCC 4種1位
- アプリケーションのことはすっかり忘れられた、、、

# 教訓的なもの

- プロジェクトは (実現可能なら) 設定した目標通りのものが実現される
- 「プログラムは思った通りではなく、書いた通りに動く」と同じ
- なので、Top500 1 位とか HPCC 1 位とかを目標にしてアプリケーションのことを忘れられては困る
- 目標をちゃんと決めよう