粒子系計算における多階層性

牧野淳一郎

東京大学大学院理学系研究科

講演概要

1. 多階層性とは？
2. アルゴリズムの多階層性
3. $O(N^2)$ 直接計算における並列化

階層性とは？

階層性といった時に、そもそもなんの話をしているのか？ということが問題

- 物理の多階層性 (multiphysics)
- アルゴリズムの多階層性 (multigrid, FMM, multiple timescale)
- メモリ階層
- 並列化の階層

物理の多階層性

見ている空間,時間スケールによって考えないといけない物理が変わる。

重力多体問題:「質点」とみなせる範囲では物理法則は同じ。

星の大きさが無視できない領域

近接散乱、衝突、近接連星

パラメトライズして N 体モデルにいれるというのが(世界的にみて)現状。

課題：

- 非弾性散乱にたいして信頼できる計算結果を与えること
  星自体の構造が一様から遠い：計算が困難
- 統合したシミュレーションコードの開発
- 星の内部構造進化のコード、、、「自動化」自体がまだ課題（主系列から白色わい星になるまでを通しで計算できるコードはあんまりない）

統合：アルゴリズムというよりは「ソフトウェア工学」みたいな問題。

アルゴリズムの多階層性

去年のワークショップでした話の繰り返しになるので今日は省略

メモリ階層

並列化の効率とかを考える上では重要
多分ちょっと細か過ぎる話、、、

並列化の階層

ここでは、「単純な」例としてツリーとかを使わない直接計算の多体問題プログラムを考える。
時間方向：独立時間刻みでも本質的には同じ話。

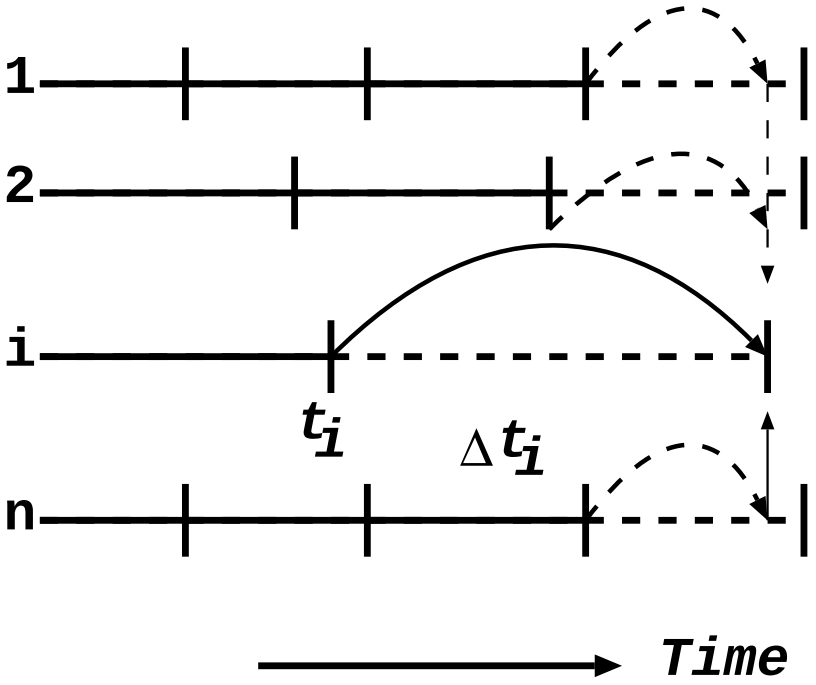
基礎方程式

$$\frac{d^2 x_i}{dt^2} = \Sigma_{j \neq i} G m_j \frac{x_j - x_i}{|x_j - x_i|^3}$$

- 要するに、ある星（惑星、小惑星、あるいはダークマター粒子、、、）は、系の他のすべての粒子からの重力を受けて運動するということ。

独立時間刻み

(Aarseth 1963)



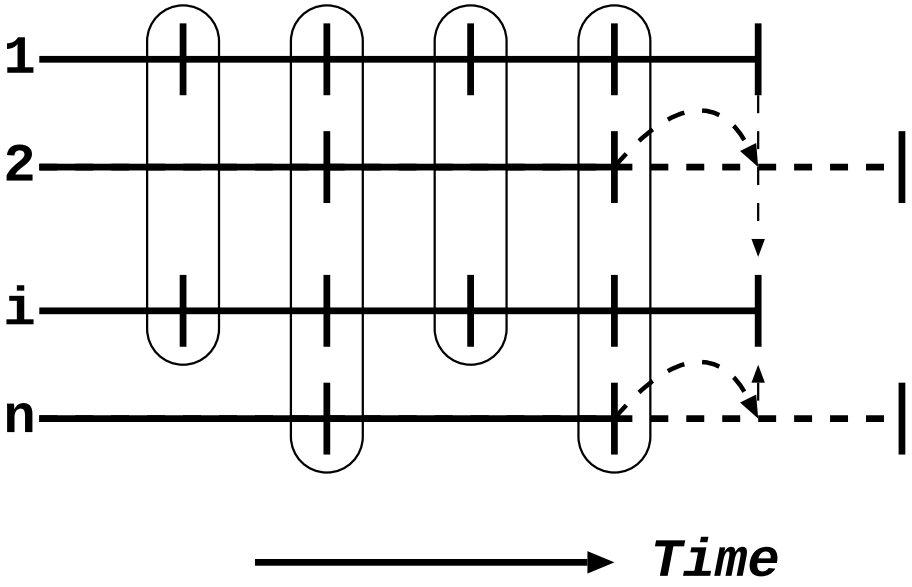
- 各粒子にそれぞれ時刻と時間刻みを与える
- 「イベント駆動」時間積分 — $t_i + \Delta t_i$ がもっとも小さい粒子が積分される

問題点

- 並列化が困難 (一度に一粒子)

ブロック時間刻み法

(McMillan 1986) ベクター / パラレル計算機のための改良
最初は Cyber205 のために導入された



- 時間刻みを 2^{-k} に制限する。
- $t_i + \Delta t_i$ が同じ (Δt_i は違ってもよい) 粒子は同時に積分される。

$O(N_c^{2/3})$ 個の粒子 (N_c は高密度コアのなかの粒子数) を並列計算

— そんなに大きな数ではない。

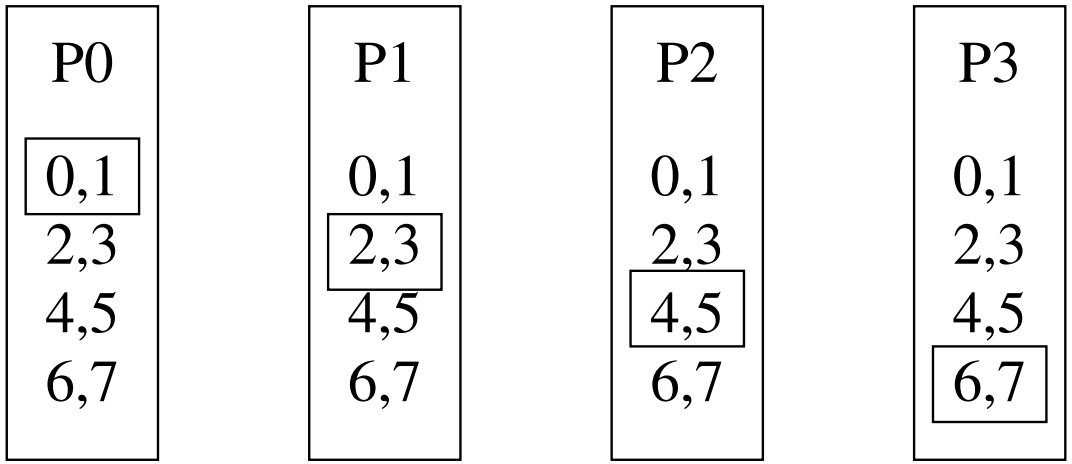
並列化の 2 方法

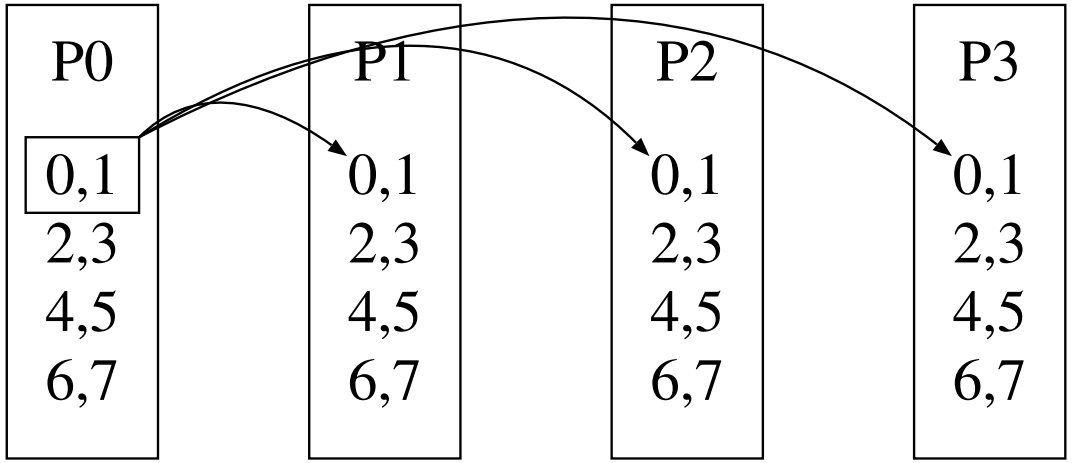
- 各ノードが全粒子のコピーを持って、重力計算、時間積分は分担する
- 各ノードが自分の分担の分だけを持つ。重力計算のためには他のノードから粒子を貰ってくる。

どちらにしても、各ノードが全粒子の情報を使って力を計算する。

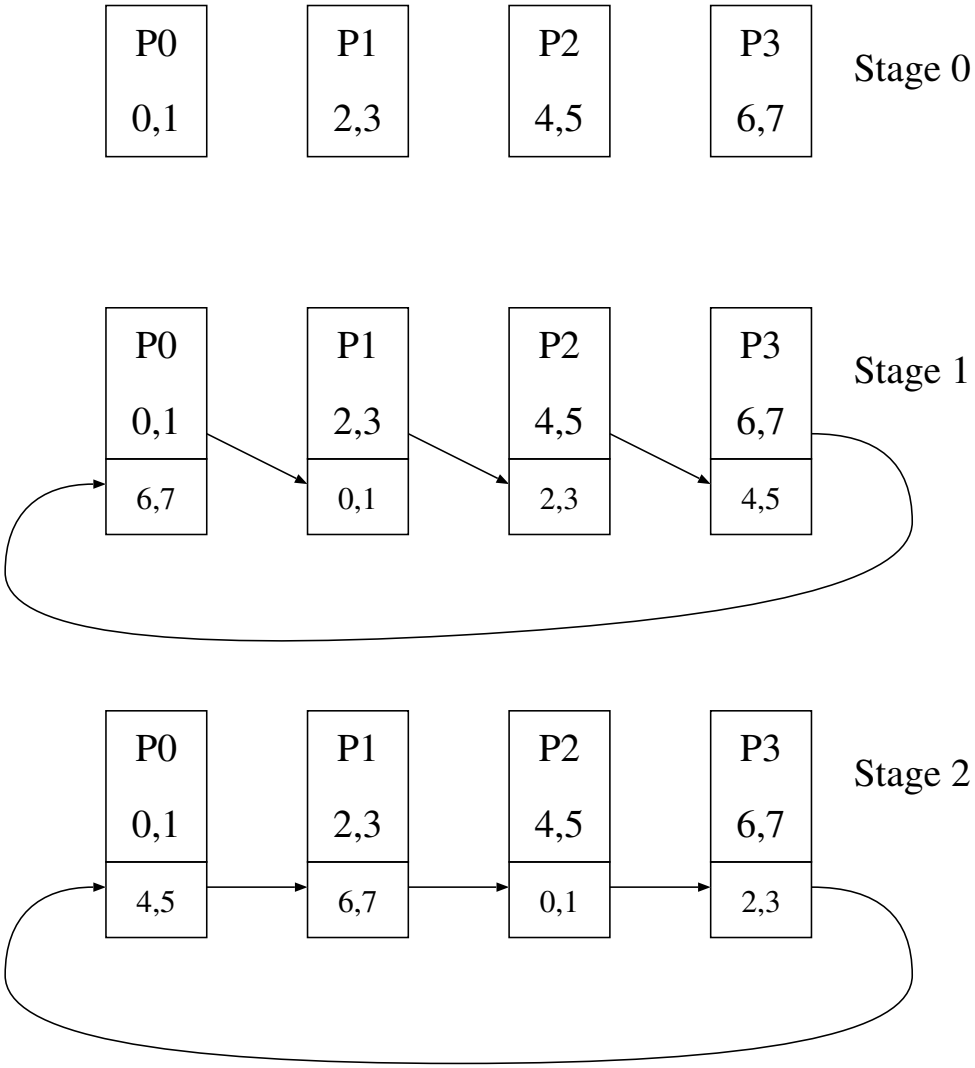
ノード一つの通信バンド幅でスケーラビリティが制限される。

The copy algorithm





The ring algorithm



計算時間のスケーリング

N 粒子、p ノード

$$T_{\mathrm{ring}} = C_f N^2/p + C_c N + C_s p. \qquad (1)$$

C_f 力の計算にかかる時間
C_c 1粒子を転送するのにかかる時間
C_s 通信ライブラリのオーバーヘッド

通信時間は p によらない $\rightarrow$ $O(N)$ ノードしか使えない。

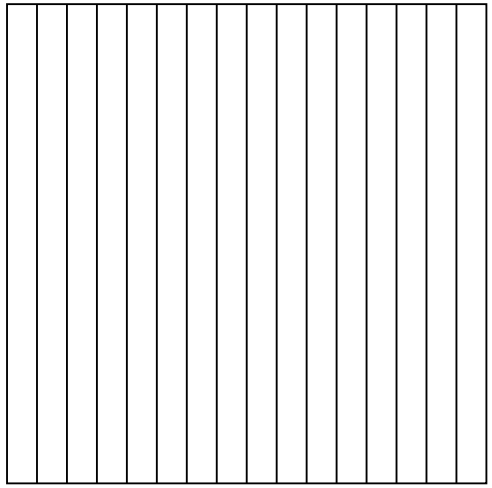
独立時間刻みの場合

- １ステップで動かす粒子数が減る
- 通信ライブラリのオーバーヘッドが無視できなくなる

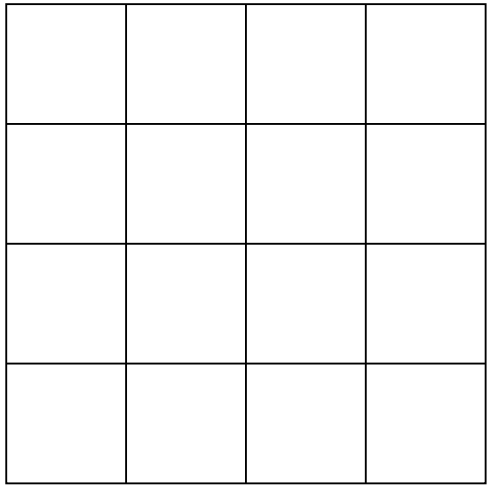
スケーラビリティが一層悪くなる

もうすこし違う考え方はないか？

近接相互作用の時（２次元の例）



1-D decomposition

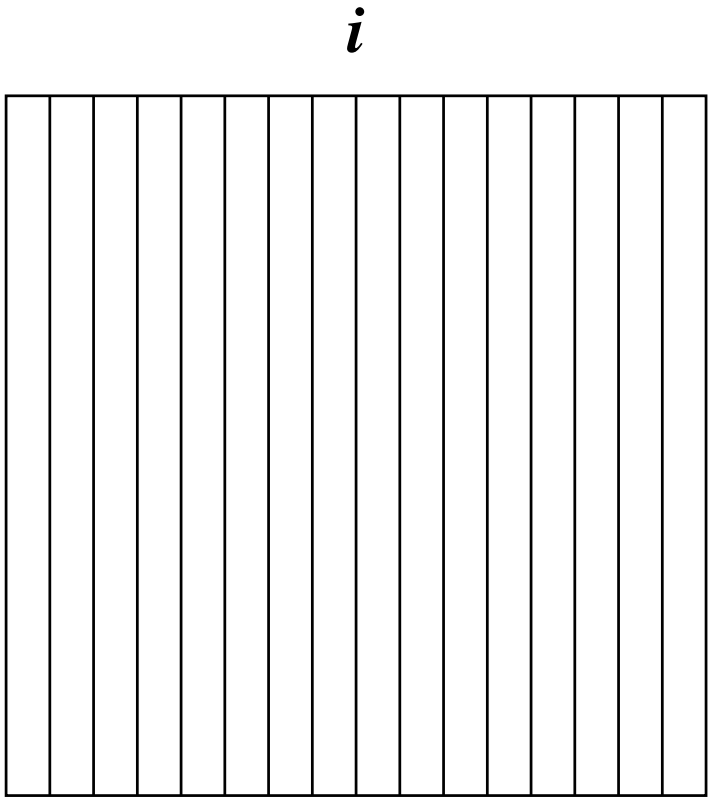


2-D decomposition

１次元空間分割：スケーラブルでない
ノード数 ∝ １次元方向のメッシュ数

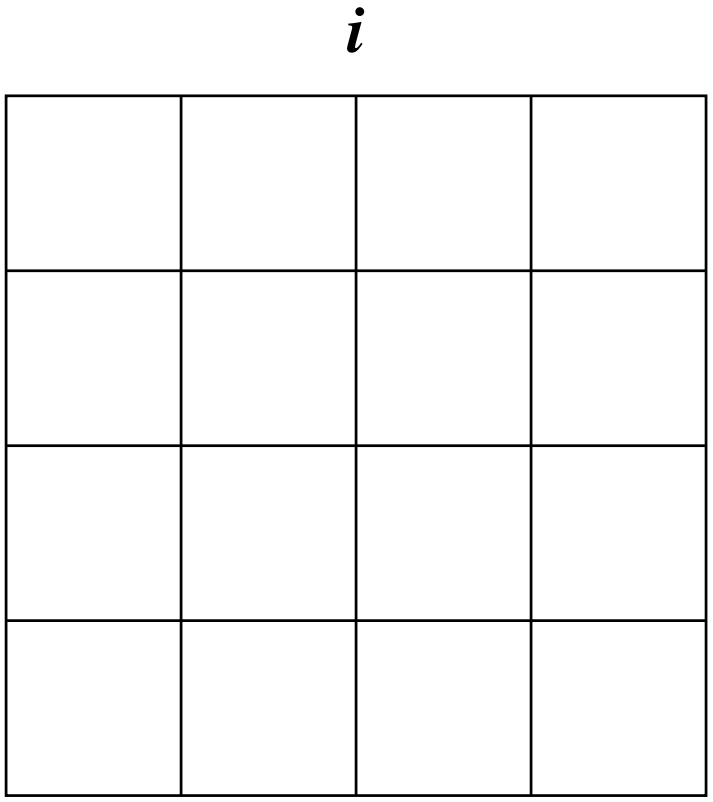
２次元空間分割：スケーラブル
ノード数 ∝ 全体のメッシュ数

遠距離相互作用 ── 相互作用行列を考える



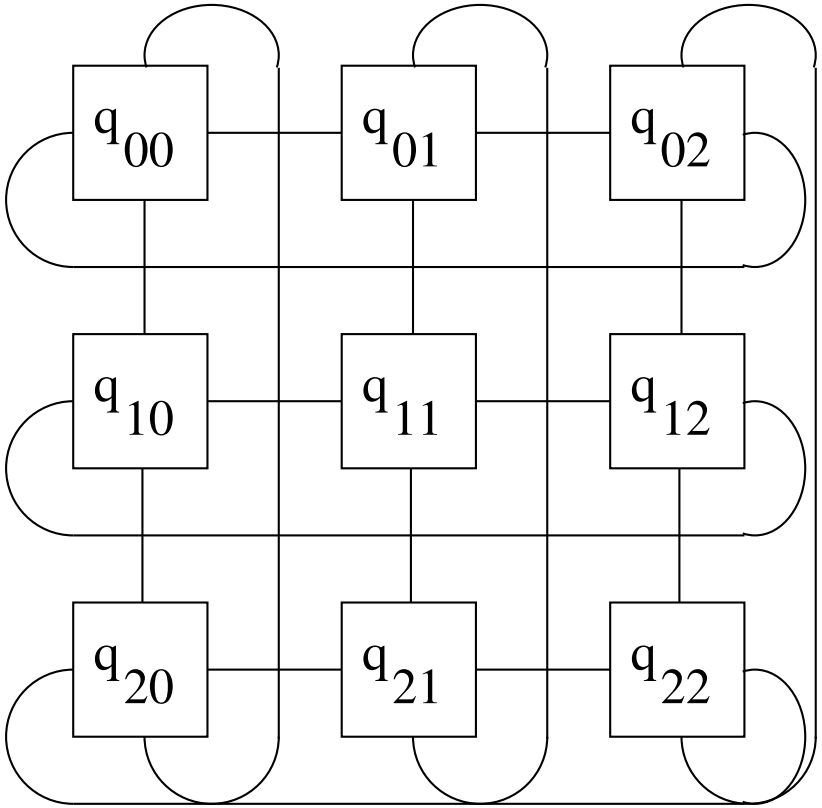
- 相互作用の行列を見ると、普通のアルゴリズムは1次元分割になっている。
- これがスケーラブルでない理由
- 2次元分割は可能か？

相互作用の2次元分割



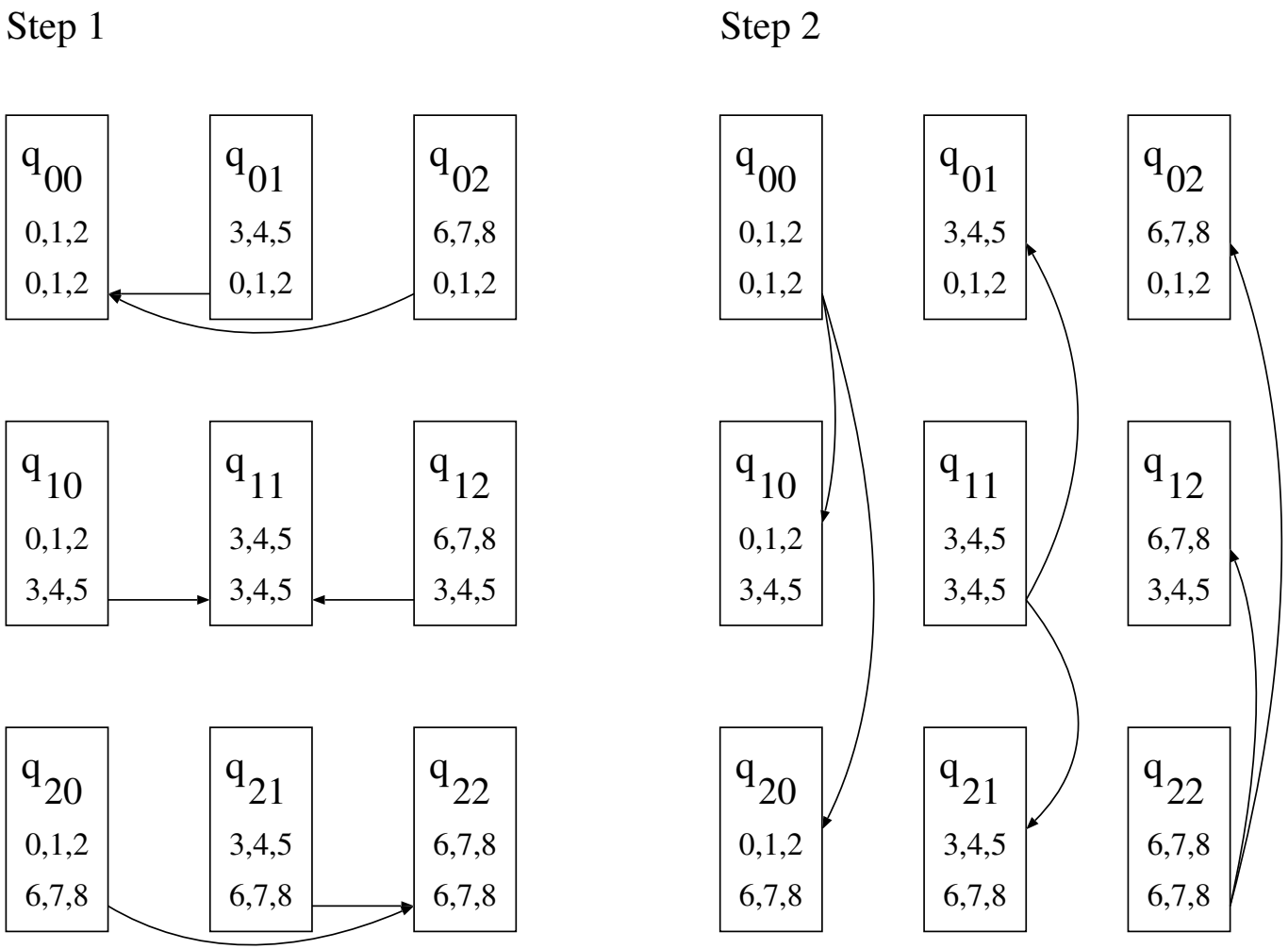
- 形式的には左のような絵を書ける。
- 実際にはこれはなにを意味しているのか？

ハードウェアとしては、、、



- 分割した相互作用行列にプロセッサを対応させる。
- トポロジ：2D トーラスなり「ハイパークロスバー」なり、、、

計算手順



横に総和、縦に放送

スケーリング (理論)

r^2 プロセッサ

$$T_{\mathrm{2Dbcast}} = C_f N^2/r^2 + 4(C_c N/r + C_s \log_2 r). \tag{2}$$

計算時間は $(N/r)^2$、通信は N/r になる。従って、 $r \sim N$、つまりプロセッサ数を N^2 にとれる。(スタートアップが無視できれば)

使えるプロセッサ数

・効率が $50\ \%$ になるプロセッサ数

$$p_{\mathrm{half,2Dbcast}} \sim (NC_f/4C_c)^2 \tag{3}$$

ここから先でも速度は向上する (プロセッサ数の平方根に比例)

独立時間刻みの時

$$p_{\mathrm{half,ind,2D}} \sim N^{5/3} C_f / 2C_s \log_2(N^{5/3} C_f / 2C_s) \tag{4}$$

大抵の場合にスタートアップ時間がスケーラビリティを制限する。
N^2 まではいかないが、 1 次元分割に比べれば圧倒的に良い。

こんなのは昔から知られてないか？

- N^2 個の (仮想) プロセッサを使うアルゴリズム
  Barnes and Hillis (1987?) に触れられている。
- **Hyper-systolic algorithm (Lippert *et al.* 1998)**
  1 次元リング + non-unit-stride communication
  なんだか難しい数学を使って通信パターンを最適化

同様の考え方自体は知られてはいた。
スケーラビリティ等は明確には議論されていない。

GRAPE を使うと

C_f が小さくなる $\rightarrow$ 1 次元の時に状況が悪化する
(並列化がほとんどできなくなる)

$$p_{half} \propto NC_f \tag{5}$$

なので、同じ N なら (値段はともかく) 実現できる最高速度は GRAPE を使っても使わなくても同じ

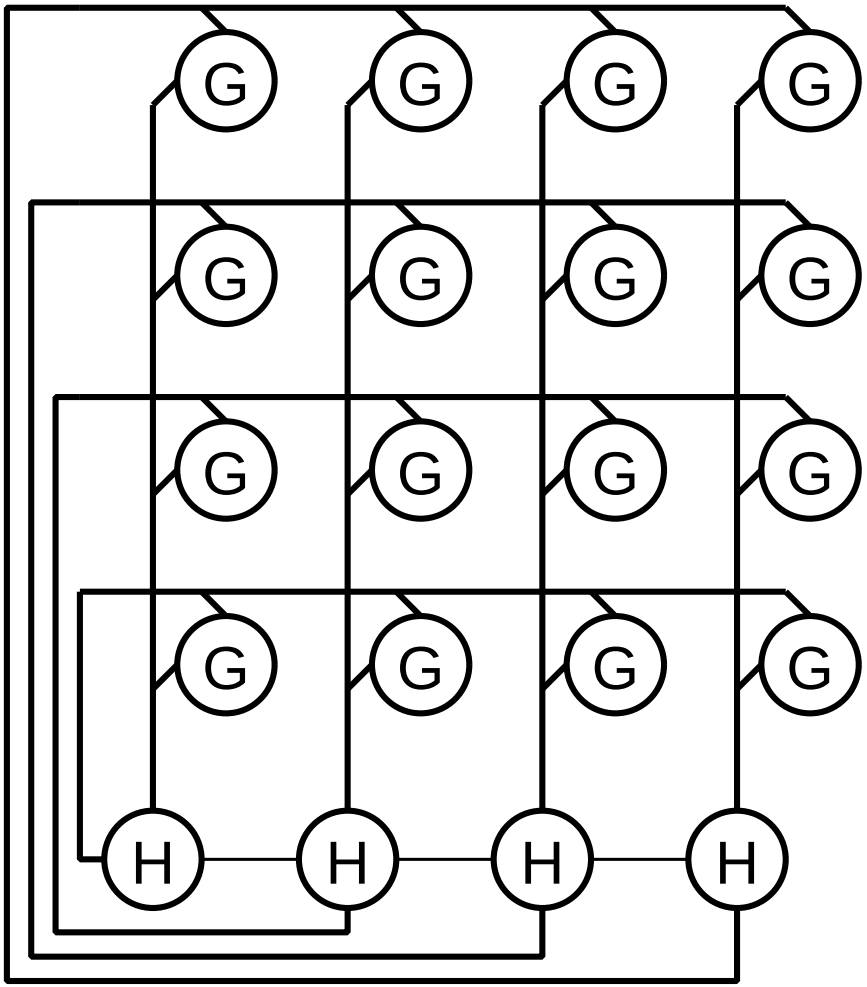
2次元の場合

2次元アルゴリズムだと一応スケーラブル

$$p_{half} \propto N^2 C_f^2 \tag{6}$$

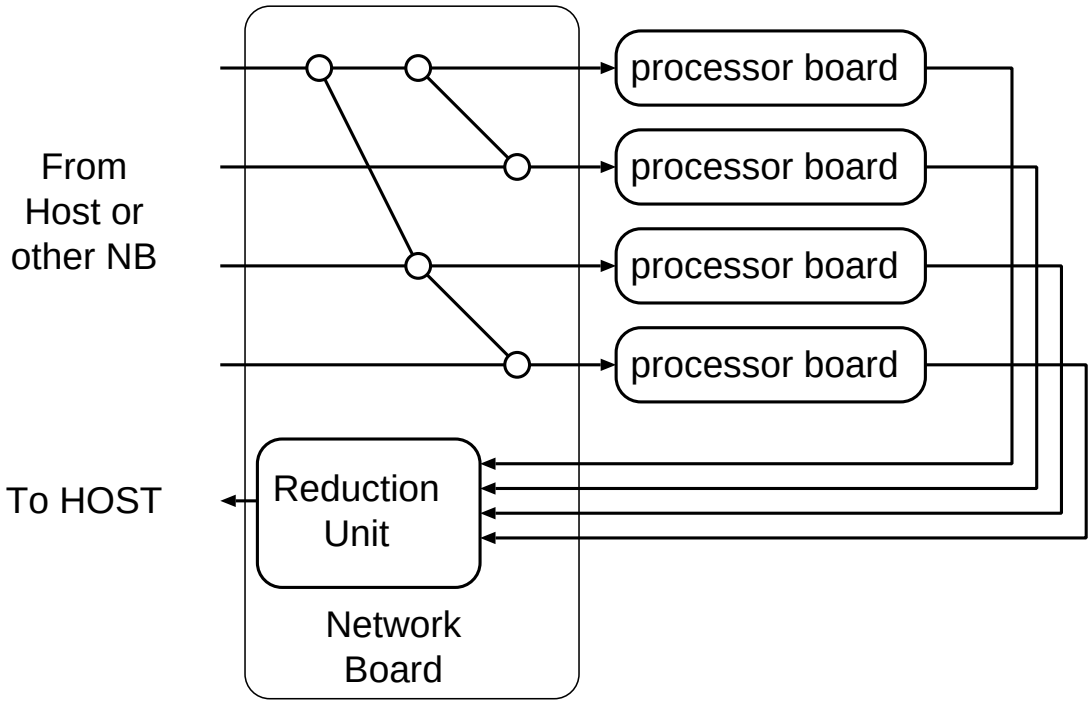
なので、同じ N の時に達成できる性能は実は GRAPE を使うと下がってしまうが、もともと非現実的に大きなプロセッサ数まで使えるようになっているので実質的には OK

実際の GRAPE-6



ホストは 1 次元
ホスト間は通信なし (同期はある)

分割可能なシステム



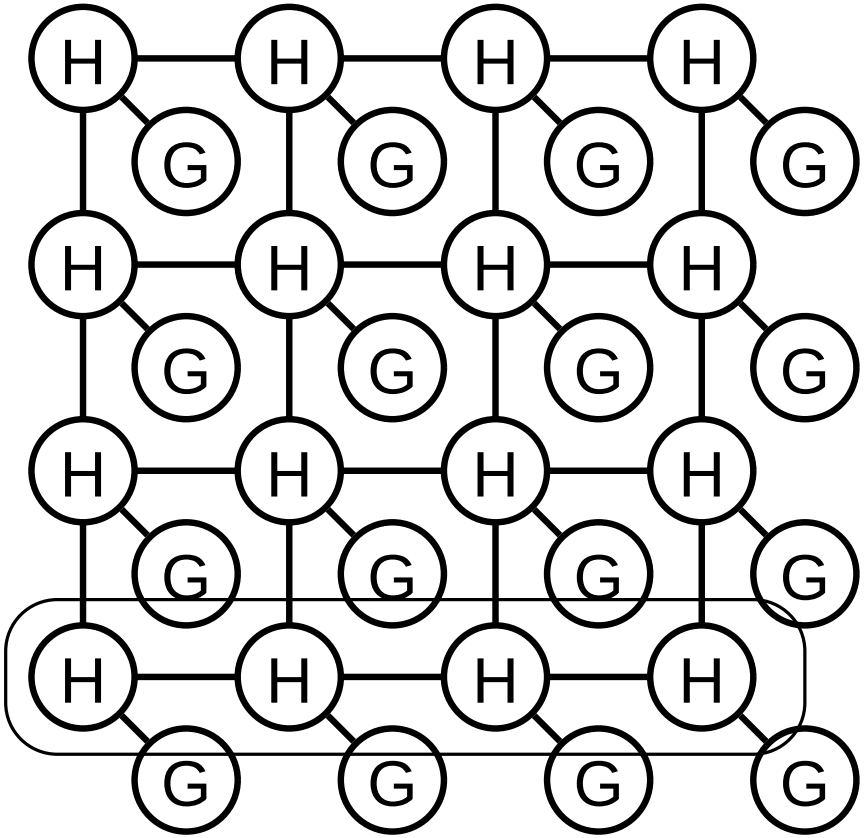
1台のホストからの放送、n 台からの直接接続、その中間を切替えられるようにする。

バタフライネットワークでももちろんOKだが、規模が大きい。

これは必要ハードが $O(n)$ で済むので実現が楽。

もうひとつのありえた (ありえる)

GRAPE-6

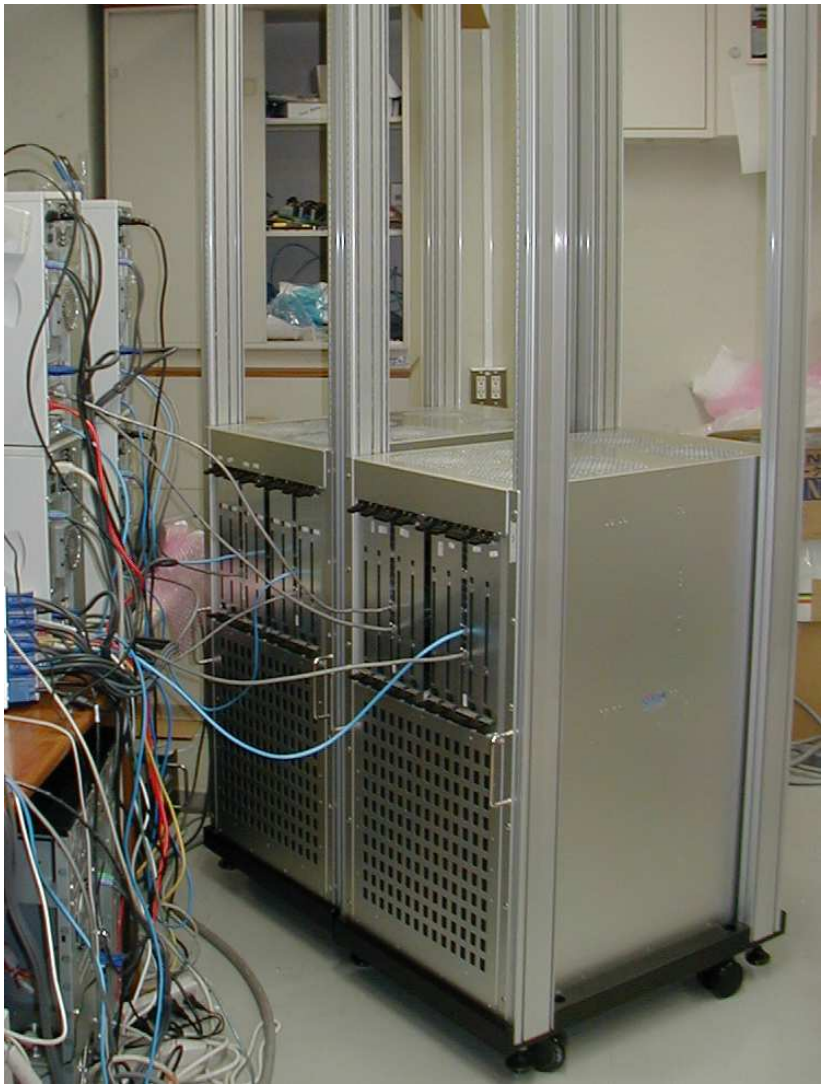


ホストも 2 次元

ホスト間に高速 (バンド幅、レイテンシとも) のネットワークが必要。

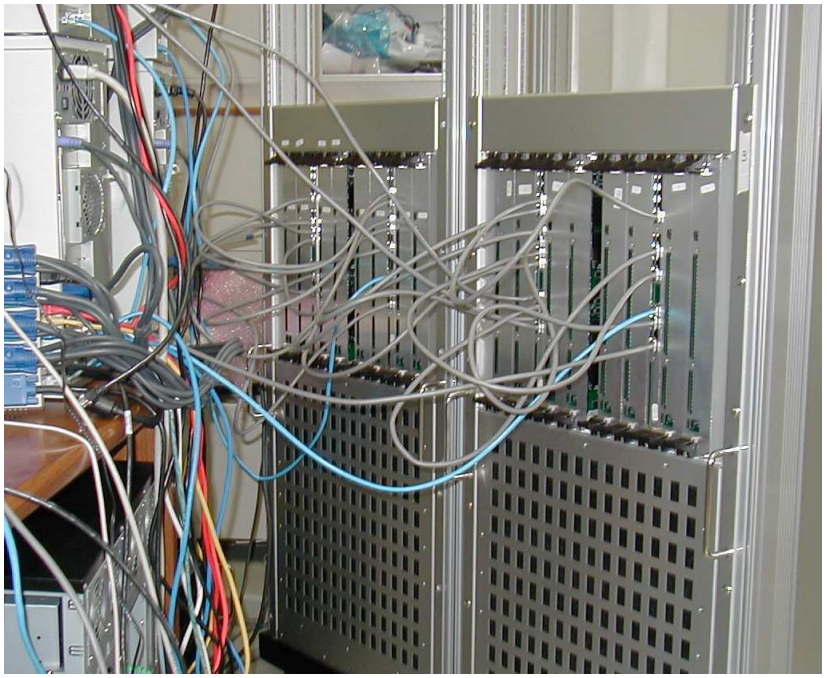
Myrinet/PM とか **MPI/GAMMA** とかなら使えなくはない

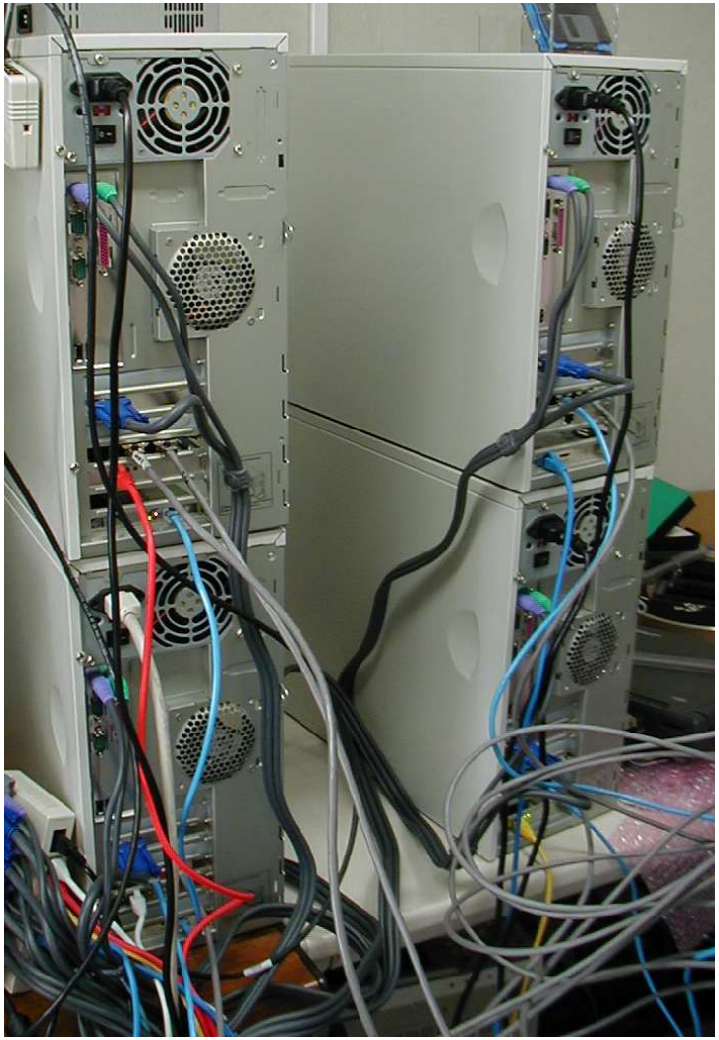
GRAPE-6 現状



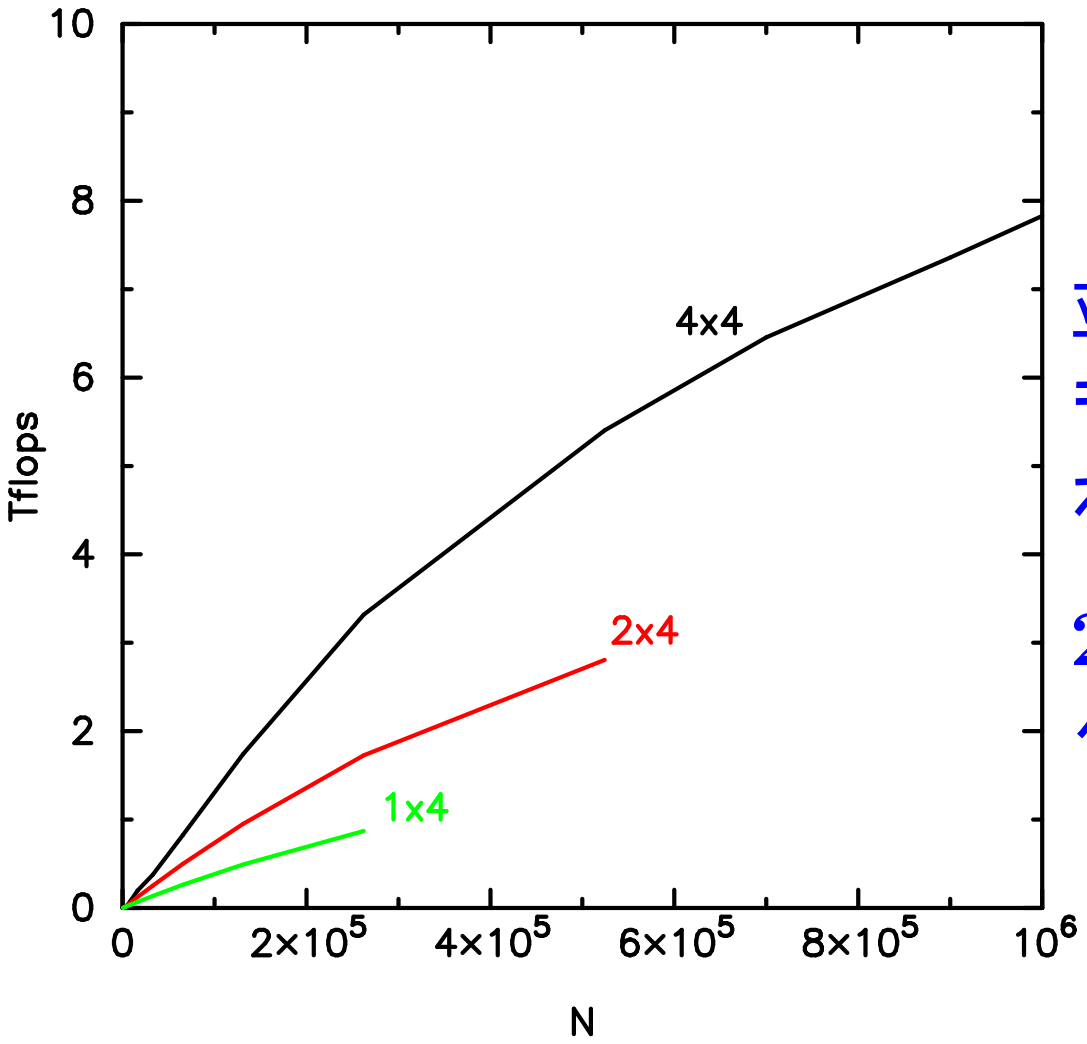
- ホスト 4 台 + カード 16 枚の基本構成が組上がった。
- ホストは Intel P4 1.7GHz (Linux)

配線、ホスト





実効速度



並列度が上がっても、半性能長は変わらない。

2 次元ネットワークの効果。

まとめ

- 多階層性といってもいろいろある。
- $O(N^2)$ の「単純な」アルゴリズムで、 $O(N^2)$ のプロセッサ数までスケールできる新しい並列化の方法を提案した。
- **GRAPE-6** での並列化はもともとそうなっている、、、

講演会のご案内

世界最高速重力多体問題専用計算機 GRAPE-6 完成記念講演会
GRAPE で探る宇宙–月の形成から宇宙の大規模構造まで–

日時: 2001 年 7 月 13 日 (金)18 時-19 時 30 分
会場: 東京大学山上会館大会議室

プログラム:

18:00-18:10 GRAPE-6 完成報告
牧野淳一郎 (東京大学)
18:10-18:30 ”The GRAPE Project–Worldwide Perspective”
Piet Hut(プリンストン高等研究所)
18:30-19:20 「理論天文学の望遠鏡–超高速多体問題専用計算機 GRAPE–」
杉本大一郎 (放送大学)
19:20-19:30 質疑応答

URL: http://grape.astron.s.u-tokyo.ac.jp/