超並列クラスタPACS-CSの開発・製作

- Parallel Array Computer System for Computational Sciences -

朴　泰祐

筑波大学　計算科学研究センター

(システム情報工学研究科コンピュータサイエンス専攻)

taisuke@cs.tsukuba.ac.jp

http://www.hpcs.is.tsukuba.ac.jp/~taisuke/

Outline

- CP-PACSの限界
- 大規模計算科学のプラットフォームに必要なもの
- PCクラスタによる post CP-PACS計画
- プロセッサ＆ネットワークの要件
- PACS-CSのコンセプト
- PACS-CSのアーキテクチャ
- PACS-CSのソフトウェア
- まとめ

CP-PACSはどんなマシンだったか…

- 1990年代前半～中盤：
  超並列「真っ盛り」の時代
- プロセッサ性能に対するメモリ／ネットワーク性能がかなり高かった
  - Memory: 300 MFLOPS : 1.2 Gbyte/sec
  - Network: 300 MFLOPS : 300 Mbyte/sec/link
- ベクトルとスカラーの融合が可能な性能バランス
  ＋それを実現できるプロセッサ実装
- MPP (Massively Parallel Processor)として、様々な部分に専用ハードウェアを投入できた
  - プロセッサ、ネットワーク、バンクメモリ

かつての世界最高速計算機も…

1996年11月のTOP500
第一位
ピーク性能 614 GFLOPS
Linpack性能
368 GFLOPS
(地球シミュレータの前
に日本が一位を取った
最後の計算機)

2003年11月のTOP500
ついに drop off !!

現在のHPCの状況

- ベクトル計算機の棲家が狭まりつつある
  - 性能向上に対する規模･価格･電力の増大
  - 商業的な成り立ちの難しさ
- なかなか広がらない裾野
  - 重要性は認識されてきている
  - 世界規模でも市場がまだ狭い
- クラスタの台頭
  - コモディティ部品の圧倒的な低価格性
  - メーカーのサーバー向け商品の充実
  - ネットワーク性能の向上

大規模計算科学に必要なプラットフォーム

- 演算性能
  - もちろん必要だが、
  - 性能自体を見れば現在のマイクロプロセッサでも十分
    単体プロセッサでピーク 6Gflops、1000台で6 Tflops
- ネットワーク性能
  - お金をかければ一昔前のスパコンを凌駕する性能
    Infiniband (x4): 1 Gbyte/s
    MyrinetXP (dual): 500 Mbyte/s
- 結局はメモリバンド幅
  - ベクトル計算機のお金＝メモリ（バンド幅）
  - CP-PACSは擬似ベクトル処理（ソフトウェアによる）だったが、メモリは16 bank持っていた

センターのリソースの今後の展望

- 中・長期的
  - CP-PACSの代替となる大型計算機の導入を目指す
  - センターの特色を生かした効率的なシステムを
    ⇒単に「できあいのスパコンを買う」のではなく
- 短期的
  - 現在のCPU性能とネットワーク性能、全体的な価格も含めたバランスを考えると、この数年間ではクラスタが有利
  - ベクトル化率99%の大型ベクトルプロセッサと、実効処理効率2～3割程度の多数の汎用スカラープロセッサの最終的な総合実効効率の比較

クラスタによる10～20Tflopsクラスのシステム

センターのHPCリソースの考え方

- 計算科学研究センターの特徴
  - 対象とする分野の重要課題の認識
  - ある程度アプリケーション（あるいは手法）を絞り込める
    - 素粒子（full QCD：近接通信＋collective）
    - ナノ物性　（実空間DFT、実時間DFT：近接通信＋collective）
    - その他の今後の拡張分野
- MPPからの移行
  - 格差の大きい演算／通信性能比に対する最適な構成・技術をちゃんと考える

PCクラスタによる post CP-PACS システム計画

- この数年間はPCクラスタの持つ圧倒的な対価格性能比に頼ったソリューションを取るのが得策
- とはいえ、一般的なPCクラスタでは我々のアプリケーションにおける実効性能対ピーク性能比が低すぎる
  - 例：3 GHz, dual Xeon SMP server
  - メモリバンド幅： 12 Gflops : 6.4 Gbyte/sec
  - ネットワークバンド幅： 12 Gflops : 1 Gbyte/sec
- バンド幅でできるだけ妥協せず、さらに問題の特性を活かした(準) 専用アーキテクチャで新しいPCクラスタが作れないか

一般的なHPC向けクラスタ

- プロセッサはIntel互換 (Xeon, Opteron, Itanium2)
- Dual CPU 構成が一般的
  - 全体ピーク性能を保ちつつネットワークインタフェースの数と設定スペースを減らす
  - Network boundなアプリケーションは不得意
- ネットワークは SAN (System Area Network)
  - MyrinetXP: 現在dual connection に対応
  - Infiniband: 次世代の期待、x4 まで利用可能
  - ネットワーク性能が必要ない分野ではGbEthernetが中心

我々が目指すクラスタ

- ネットワークバンド幅対策：
  - 「ネットワークにはお金をかけるのは当たり前」ではつまらない
    ⇒我々独自の工夫で乗り切れないか
  - クラスタにおけるネットワークコストの増大
    - 「クラスタは大きくするとコスト効率が悪くなる」
  - 高性能ネットワーク (ex. InfiniBand, Quadrix)
    vs コモディティネットワーク・トランク
    (ex. GbE × n)
  - 対象問題の特性を生かしたクラスタを！

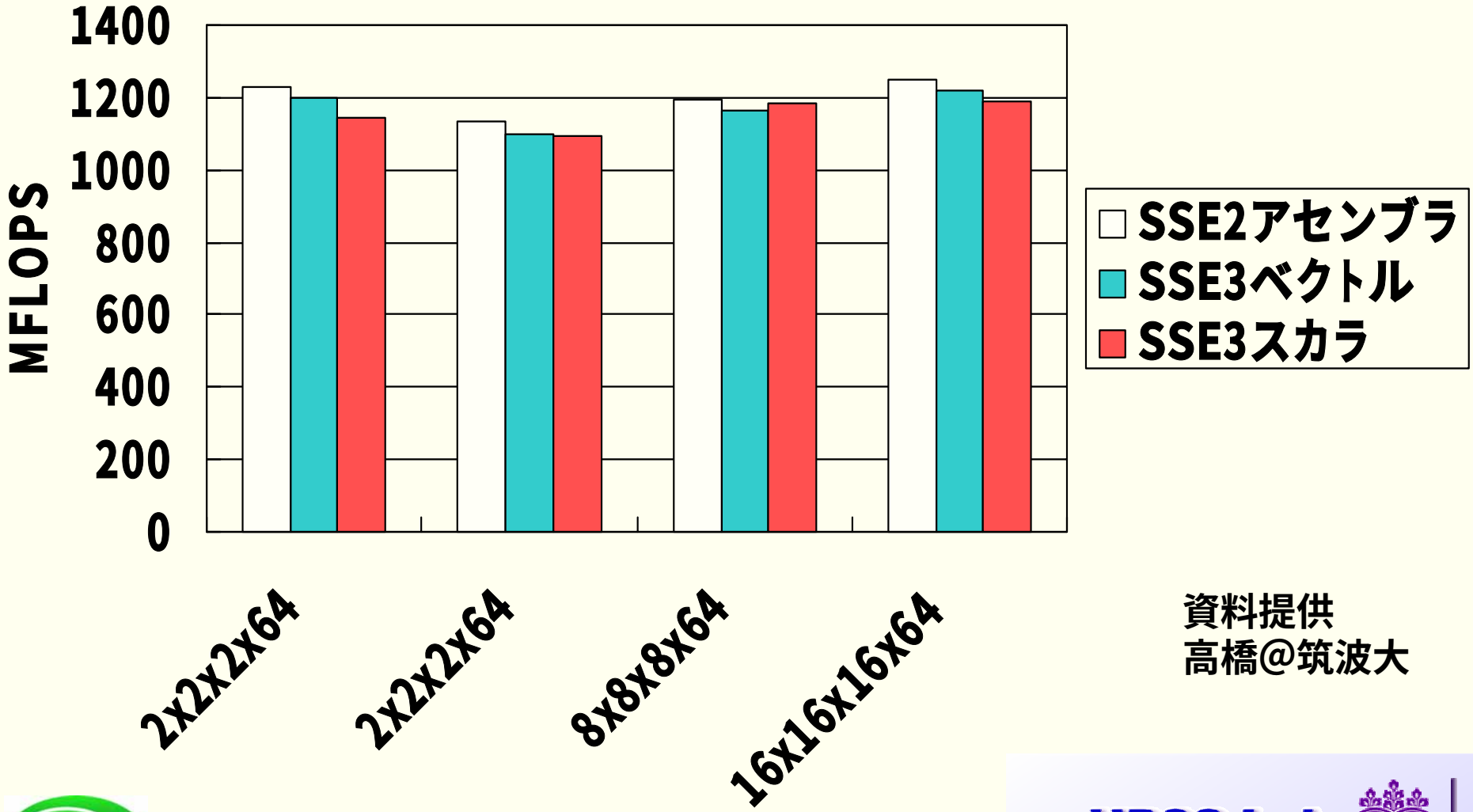
CPUに関する考察

- Intel IA-32 アーキテクチャの代表⇒Xeon
  - 最も標準的に使われている(枯れている)
  - OS (Linux, Score) が安定に動作している
  - 周波数の伸びは収束しつつあるがメモリバンド幅が向上してきている(FSB800, DDR400 interleaved 対応)
  - SSE, SSE2, SSE3 等の short vector 命令により、従来の x87 命令より効率の高い浮動小数点演算
  - EM64TによるSSE対応レジスタ数の増強(16個)
    (アドレス空間も64bit)
  - Low Voltage版の登場により消費電力的に有利

CPU性能の予備評価

- QCD – Mult Benchmark (by 石川@広島大)
- Pentium Xeon EM64T (64bit, SSE3) 3.4 GHz
- Memory: DDR400, FSB800 = 6.4 Gbyte/sec
- SSE2, SSE3 を利用し、プログラム方法を変えた場合の単体CPU性能

並列版（通信を考慮したデータ扱い）の結果



資料提供
高橋@筑波大

他のシステムとの比較

- 単体CPUにおける性能 [Mflops]

(＊P4, Xeonに関しては、SSE2により浮動小数点処理を最適化)

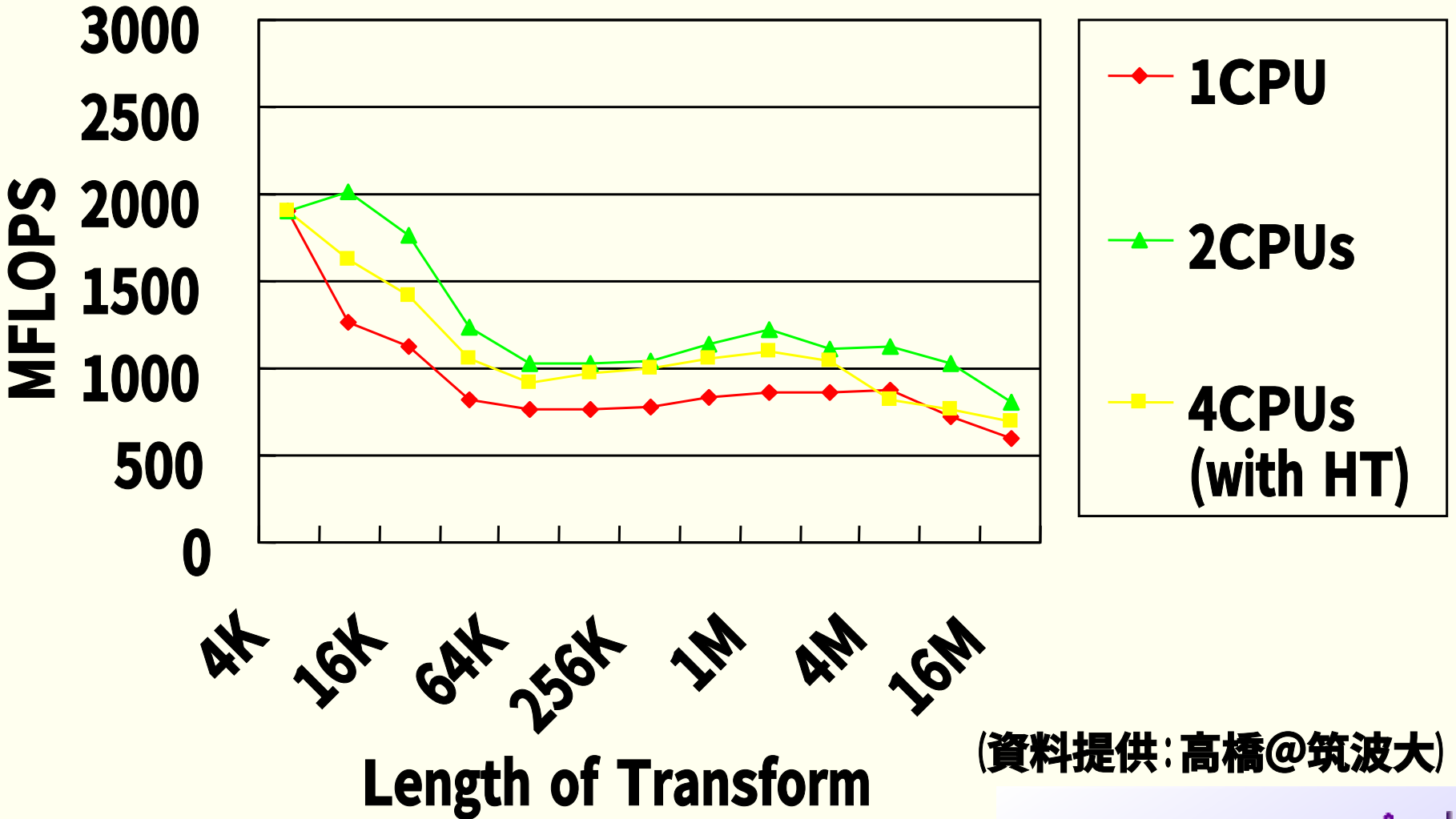
格子サイズ	P4		Xeon		EV7	
	No copy	Copy	No copy	Copy	No copy	Copy
2*2*2*64	1251	957	811	598	1190	949
4*4*4*64	1020	878	633	536	1144	1034
8*8*8*64	1045	958	686	625	1140	1082
16*16*16*64	N/A	N/A	604	573	1122	1101

No Copy: 単体CPUだけで全演算を行う場合を想定
Copy: 並列化により、通信部分の「のりしろ」データをバッファにコピーする処理を含む

メモリバンド幅が重要！！

データ提供:
石川@広島大

FFTE 3.2 (SSE2)の性能 (Xeon 3.06GHz)



(資料提供:高橋@筑波大)

CPU性能の見通し

- メモリバンド幅の拡充
  - PC3200 (3.2 Gbyte/s) x 2等
- 大容量オンチップキャッシュ
  - 3MB L2等
- short vector 機能の充実
  - SSE2
  - SSE3
- 設置スペース、熱・消費電力等を考えると現在のコモディティでかなりいける

ネットワークに関する考察

- 現在の高性能クラスタ向けネットワークはコモディティとは言えない
  (私の) コモディティの定義:
  「我々は開発費を払わなくて良い」
- CPU及び周辺の対価格性能比の延びに対し、現在の対応するネットワークの対価格性能比は不利
  (数年後はわからない)
- CPUの対価格性能比の延びとネットワークのそれはうまく釣り合っていない
- 大規模化すればするほど、特にスイッチのコストが大きくなる

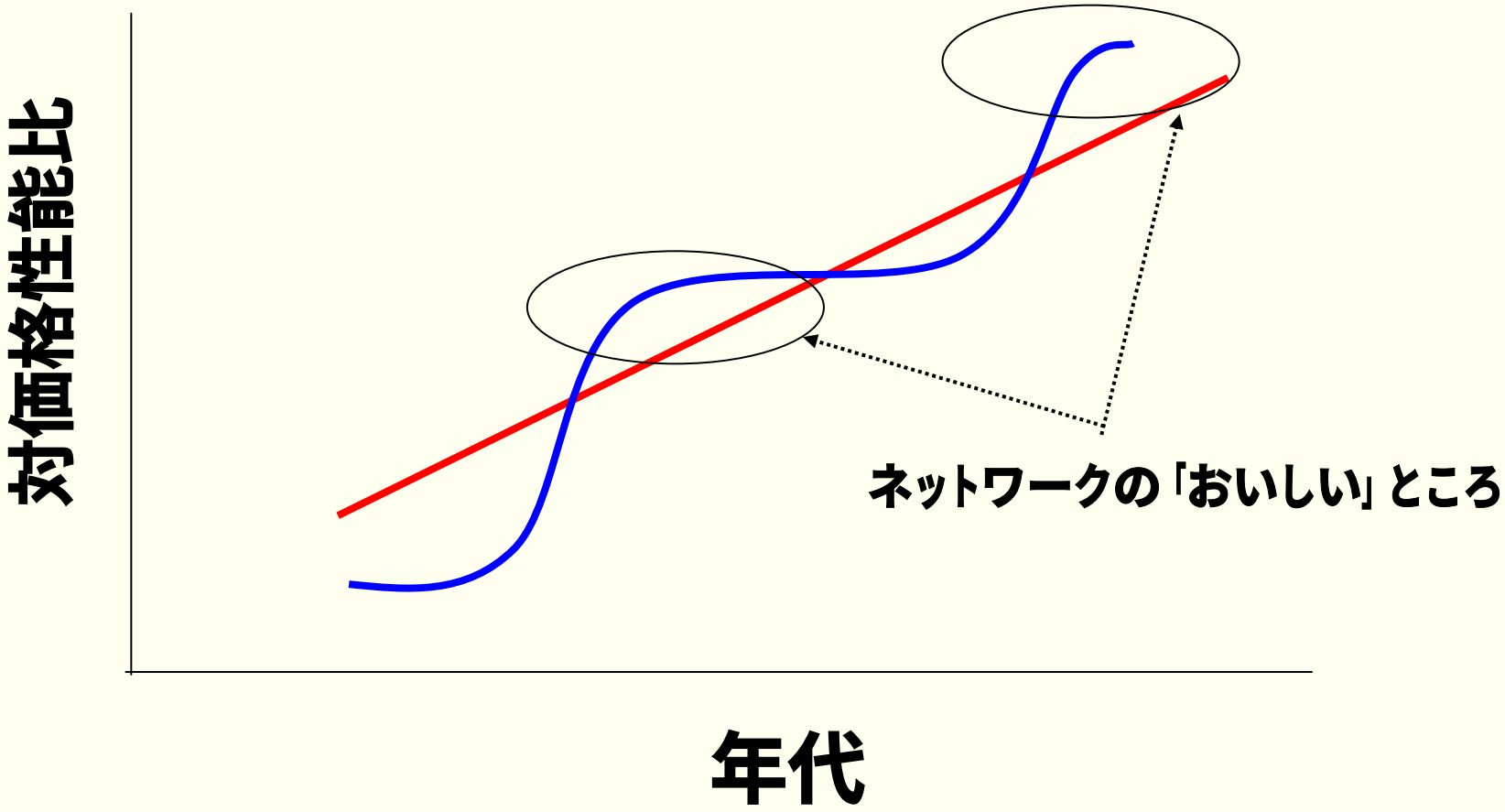
QCDにおける必要通信性能

- 実効プロセッサ性能が1Gflops、実効通信性能が0.6Gbyte/sだとした場合の、32*32*32*64の問題サイズの5000 trajectory当たりの計算時間

CPU数	時間 [日]	通信時間の割合 [%]
512 (ns=8)	2768	17
4096 (ns=16)	405	29

通信性能は非常に重要
(レイテンシよりもバンド幅)

CPUとネットワークの対価各性能比の変化



ネットワークの見通し

- CPU性能はもちろん，ネットワークバンド幅が必要
- センターの性質から，アプリケーションをある程度絞り込める
- 「どことでもランダムに速く通信できる」ネットワークは実はいらない

⇒現在のツリー中心のネットワークは、一般的な科学技術計算でそのpermutationのほとんどを使っていない
⇒ネットワークコストを大きく抑える工夫ができるのでは？
⇒例えば近接通信だけができればよい？
⇒高価なNIC+Switch 必ずしも必要ではない

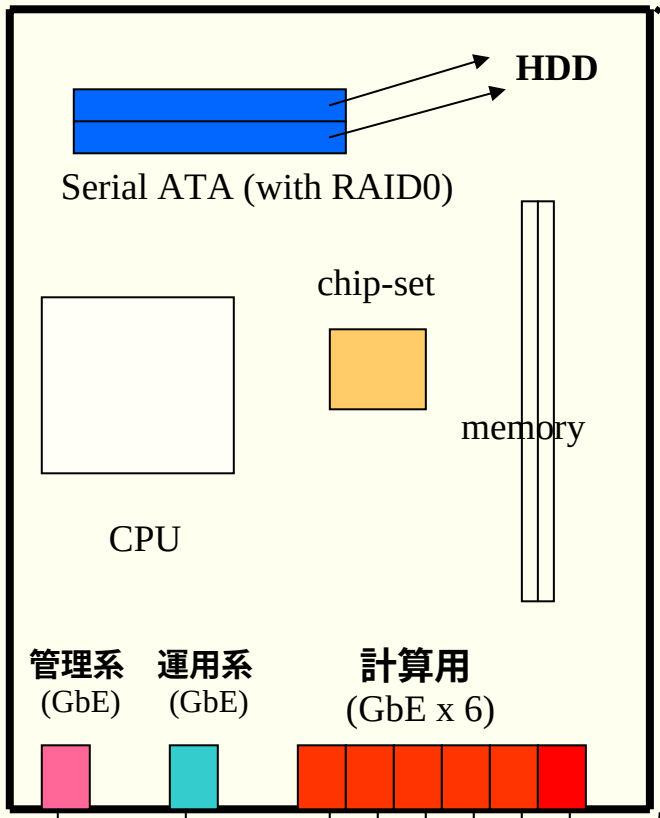
超並列クラスタ：PACS-CSのコンセプト

- Parallel Array Computer System for Computational Sciences
- コモディティ技術を用いた超並列PCクラスタ
  - 搭載する部品（IP）は全てコモディティ
  - マザーボード等は必ずしもコモディティ製品ではない
  - ソフトウェア（OS、コンパイラ、ライブラリ）もコモディティ
- SMP構成にとらわれない、バンド幅＆スペース効率追求システム
- ネットワークに独自アーキテクチャを導入
- 従来の既成PCサーバに基づくクラスタとは異なる

PACS-CSのハードウェア概要 (現時点での想定)

- ピーク性能：17.2 Tflops (2.8GHz CPUを想定)
- プロセッサ数：3072 CPU (node)
- ノード構成：single CPU とし、各ノードに独立なネットワーク
- 総メモリ容量：6.1 Tbyte
- 総ディスク容量： 614 Tbyte (RAID0 mirror)
- ネットワーク： GbEthernet trunk (dual link × 3方向)
  ホストインタフェース： PCI-X dual
  Gigabit Ethernet トランクによるバンド幅増強
- ネットワーク構成： 3-D Hyper Crossbar
  小規模コモディティスイッチ＋ソフトウェアルーティング

ボード構成概念図(0.5Uスロット)

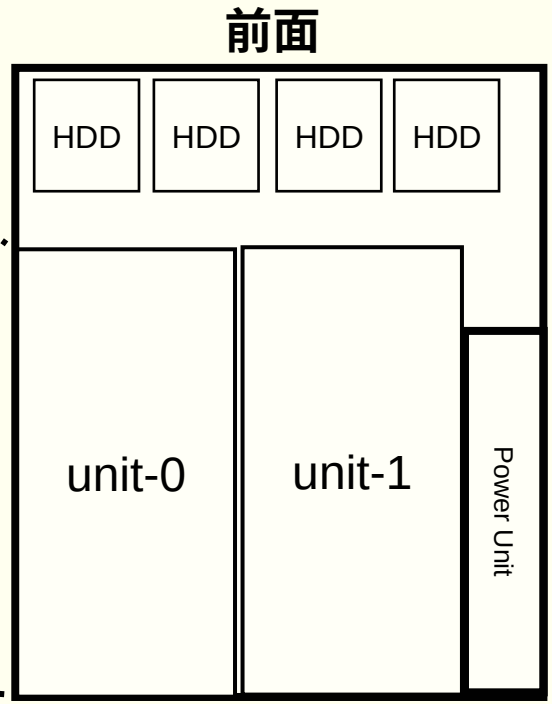


x0, x1: X次元クロスバに接続されるdual link

y0, y1: Y次元クロスバに接続されるdual link

z0, z1: Z次元クロスバに接続されるdual link

1Uシャーシ相当のノードのイメージ

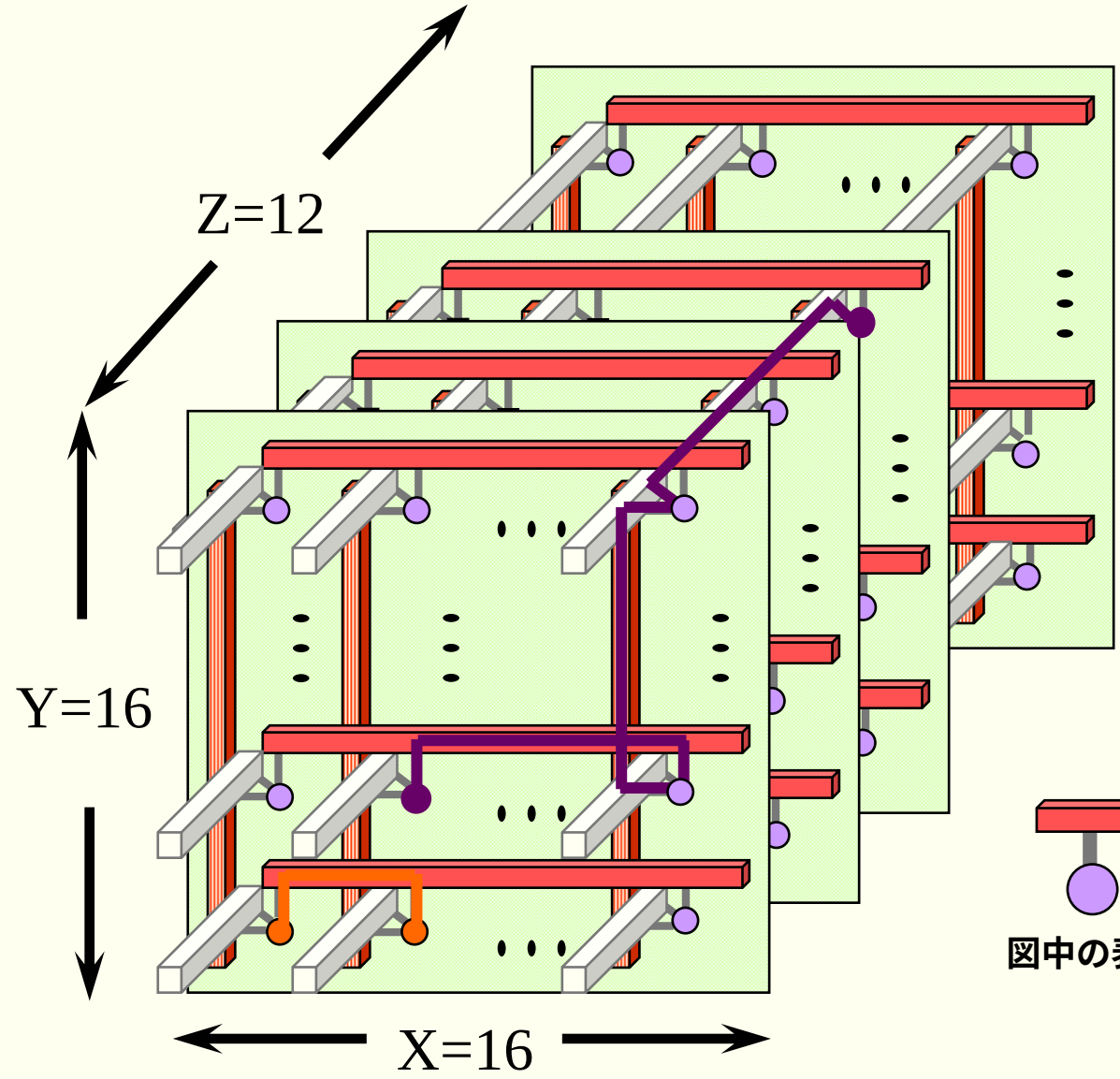


運用系ネットワークへ（ファイルサーバ等）

管理系ネットワークへ（集中コンソール）

ノード(CPU)間の論理的結合 *(3次元ハイパクロスバ網)*

3072 node 構成



X次元スイッチ

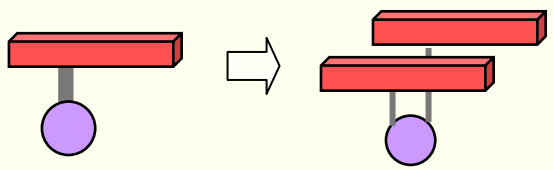
Y次元スイッチ

Z次元スイッチ

計算ノード(CPU単位)

1つのスイッチで直接結合されているCPU間の通信

複数の次元のスイッチを経由して行われるCPU間の通信(中継CPUが必要)



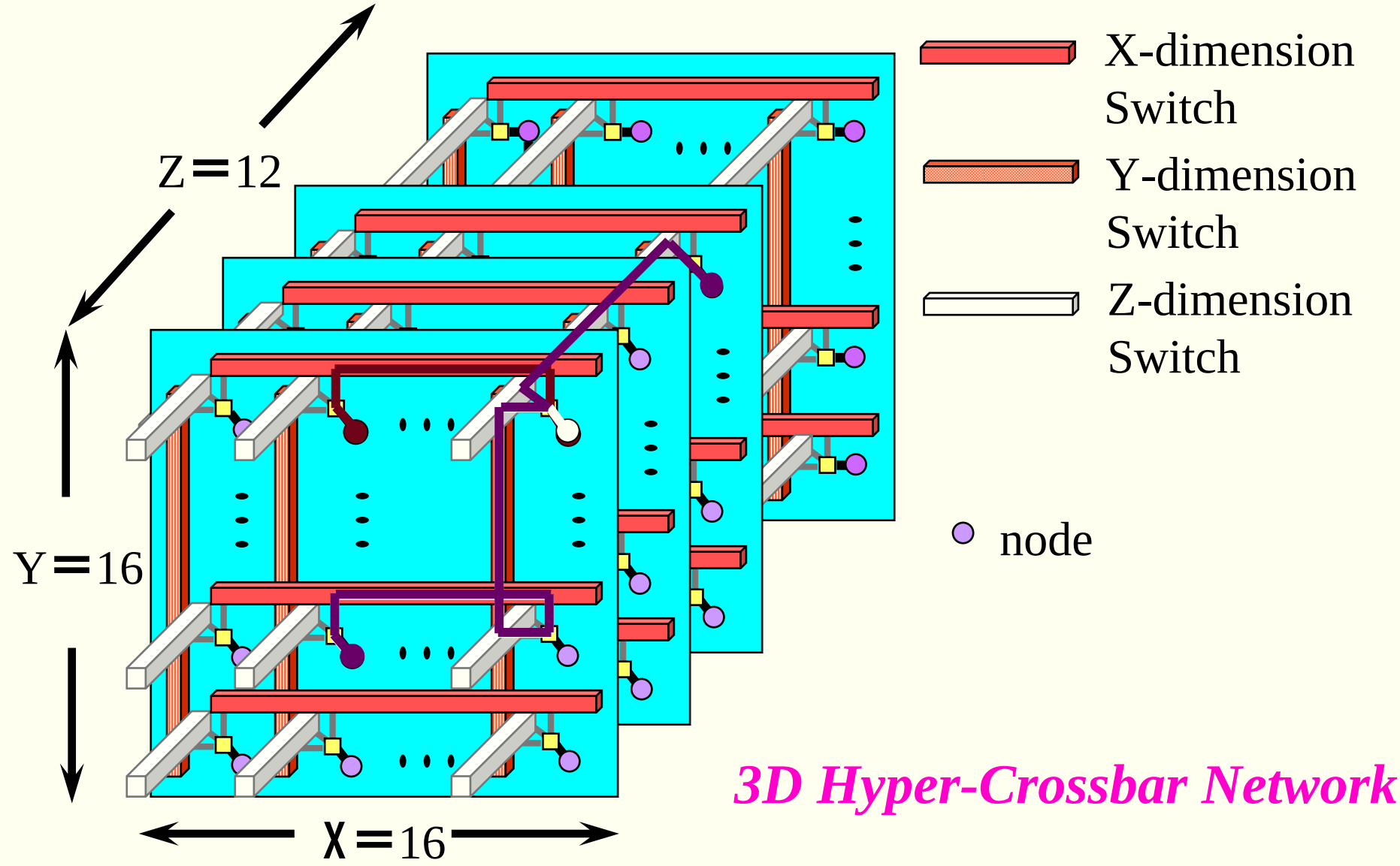
図中の表現

実際の接続形態
(dual link によるバンド幅増強)

PACS-CSハードウェアの考え方

- ボードは作成するが部品は全てコモディティ
  ⇒短期間での開発と低コスト化
- CPUの実効性能はある程度のメモリバンド幅とSSE等のショートベクトル処理で稼ぐ
- 最低限のメモリバンド幅を稼ぐために dual CPU SMP にはしない
- コモディティベースの3次元ハイパークロスバーネットワーク
  ⇒近接通信は全てダイレクト通信
  　それ以遠は少しバンド幅が落ちるが通信可能
  ⇒実質的なバンド幅を犠牲にせずコストを大幅ダウン

3-D HXBのルーティング



3D-HXBネットワーク

- 隣接転送最優先の構成
- 最大で2つの中継ノードを経由してルーティング可能
  - ルーティングによるバンド幅ダウン、遅延増加は我慢する
  - ユーザレベルのライブラリを効率的に実装すればほとんどの通信を隣接通信の繰り返しに落とせる
  - 完全SPMD的なプログラムの動作を想定（「足並みが揃っている」）
  - 計算集中区間と通信集中区間の切り分け
- GbEthernetのトランクによる性能
  - 125 Mbyte/sec/linkがピーク⇒1次元：250 Mbyte/sec
  - 3次元同時転送：750 Mbyte/secのピーク性能
  - もし実効効率がピークの7割とすると約 520 Mbyte/sec
    ⇒ Infiniband (peak 1Gbyte/sec) を dual CPU SMP につけるより速い

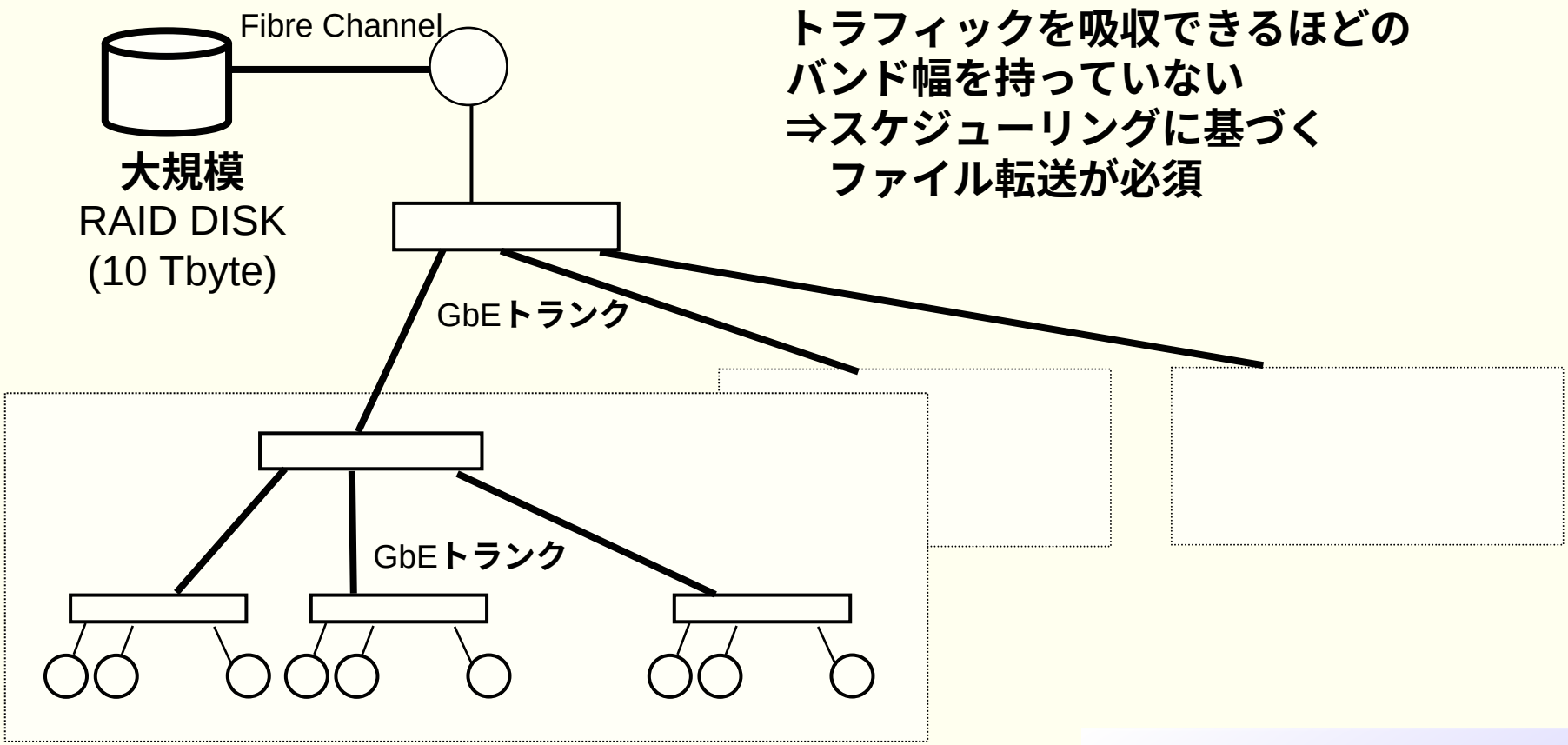
3D-HXBネットワーク (続き)

- 3072ノード構成では16x16x12等
  ⇒各次元のスイッチは高々16ポート
- 16ポートのGbEthernetのL2スイッチの価格
  ⇒約 3000円／port
- 16x16x12 の 3D-HXB with dual link GbEthernetの
  スイッチ価格
  ⇒ 3000 x (16 x 16 x 12 x 2 x 3) = 約5500万円 (！)
- その分、ケーブリングは大変
  ケーブル総本数＝約18000本

**現在のコモディティ技術によるネットワークを想定すると
この数年で最も対価格性能比の良いソリューション**

ファイルシステムのネットワーク構成

運用系ネットワークを利用



ネットワークとスイッチの構成は
下流のスイッチ及びノードからの
トラフィックを吸収できるほどの
バンド幅を持っていない
⇒スケジューリングに基づく
　ファイル転送が必須

集中コンソール機能

- 各ノードの VGA/keyboard/power-switch 等を集約し、Ethernet 経由で集中制御する集中コンソールシステムによってBIOSレベルでのノード管理
- 少数の集中コンソール用計算機から全システムを一括して管理 (power-on/off, reset等)
- ノード個別の対応を取らなければならない場合のために、VGA/keyboard/USB等を個別に引き出せるようにはなっている

PACS-CSのソフトウェア

- OS
  - Linux (free Linux: ex) Fedora Core 等)
  - SCore (大規模クラスタに対応した管理用ミドルウェア)
  - 3D-HXB用特殊ドライバ (新規開発)
- プログラミング環境
  - MPI並列プログラミング
  - 3D-HXBを直接使った高性能通信ライブラリ
  - Fortran, C, C++ with MPI
- ジョブ実行環境
  - PBS等によるバッチキュー管理
  - SCore制御下のノードグループ (パーティション) 管理
  - ファイル入出力制御スクリプト

PACS-CSシステムの運用イメージ

- 512ノード程度を単位とするシステム運用
  (必要に応じ、1024、2048、3072ノード等も)
- バッチ・キューに基づくScore上のパーティション単位でのジョブ実行
- ノードの free pool 等の扱いはしない
  - 1ジョブの単位が大きい (固定パーティションでデメリットはない)
  - 3D-HXBを基本としているので物理パーティションとの一致が重要 (不一致でも動くが通信性能が落ちる)
- ファイルシステムへの各ノードからのランダムアクセスは行わせない
  - 必要なデータは計算前に各ノードのローカルHDDにコピー
  - 計算終了後、結果データをローカルHDDからファイルサーバにコピー

ファイル入出力

- 全ノードからアクセス可能なファイルサーバを提供
- ただし、全ノードから「NFSマウント」されているわけではない
- ファイルサーバをNFS上で直接参照できるノードは限られている（ノードグループ当たり数台？）
- 計算に必要なファイルは計算実行前にファイルサーバーから各ノードのローカルディスクにコピー
- 特定ノードのみがアプリケーションプログラム中からNFS経由でファイルを参照可能
  ⇒必要な場合はプログラム中で通信を用いてデータ転送
- 計算結果もローカルディスクに出力
- 計算終了時に各ノードのローカルディスクからファイルサーバーにコピー

開発スケジュール

- 3年計画だが、基本的に最初の1年＋αで動くもの（product-runに適用可能なサイズのシステム）を作り、計算を進める
- スパコン政府調達に則ったスケジュール
  - 2004/1 資料招請
  - 2005/2 仕様書原案提示
  - 2005/4 仕様書提示
  - 2006/6 全システムの2/3を導入・運用
  - 2007/6 残り1/3を導入・フルシステム稼動
- 3D-HXB特殊ドライバ開発は試作機に合わせて行う

まとめ

- PACS-CS:計算科学研究センターにおける post CP-PACS システム (約30倍の性能)
- コモディティ技術をベースとしつつ、バンド幅に拘ったシステム構成＋対価格性能比の良い独自ネットワーク
- アプリケーションを睨んだ「無闇にバンド幅・性能を追求しない」効率的なシステム
- 20 Tflops弱の性能で、今後数年間の計算科学研究センターのアプリケーション需要に対応
- SMP構成を取らないがそれと同様の実装密度・消費電力