筑波大学計算科学研究センター
CCS HPCサマーセミナー
「並列システム」

朴　泰祐
taisuke@cs.tsukuba.ac.jp
筑波大学大学院システム情報工学研究科
計算科学研究センター

「並列システム」内容

- 並列処理システムの構成
- 並列計算機アーキテクチャ
- 並列処理ネットワーク
- 実システムと計算機アーキテクチャ
- 性能に影響を与える構成上の要素

並列処理システム

- 並列計算機（並列処理システム）は計算機であるため、プロセッサ（CPU）、メモリ（memory）、入出力装置（I/O）等の構成要素を持つ点は逐次計算機と同じ
- 複数のプロセッサ間を結合する何らかの仕組み
- システム規模は2プロセッサから数万プロセッサまで
  - 2～4プロセッサ：現在ではsingle chipで実現（multi-core CPU）
  - 10～数十プロセッサ：研究室レベルのPCクラスタ、共有メモリシステム
  - 数百プロセッサ：センター運用PCクラスタ、小型MPP (Massively Parallel Processor)
  - 数千プロセッサ～：MPP
    (multi-core CPUの台頭によりPCクラスタでも数万プロセッサが実現可能に)
- プロセッサ台数として現在世界最大のものはIBM Roadrunner（122400プロセッサ）

並列処理システムの要素

- 逐次計算機と異なる部分
  - 何らかのプロセッサ間結合ネットワーク（相互結合網：Interconnection Network）
  - プロセッサ間通信の結果のデータをメモリに保持する機構
    - 分散メモリ型マシン：相互結合網からのデータ（メッセージ）内容をメモリに保持
    - 共有メモリ型マシン：メモリ自体が逐次システムと異なり、並列プロセッサ間で共有される
      ⇒特殊なハードウェアが必要
  - 特別な同期機構を持つ場合もある
  - その他の周辺装置
    - システム全体を1つにまとめる管理機構
    - 並列プロセッサから共有可能なファイルシステム

並列計算機アーキテクチャ

- 大きく分けて
  - 分散メモリ型システム(distributed memory system)
    各プロセッサは独自のメモリ(他のプロセッサからは直接アクセス不可能)を持ち、相互結合網を用いたメッセージパッシングによってデータ交換を行う
  - 共有メモリ型システム(shared memory system)
    並列プロセッサ間で物理的に共有される共有メモリ(shared memory)を持ち、各プロセッサが普通のload/store命令を発行してデータの読み書きを行う

- さらに、共有メモリシステムを分散メモリ型に結合したhybrid型システム(constellation型)もある
  ⇒ 最近の multi-core CPU の影響から、これが普通になってきた

計算機の進歩

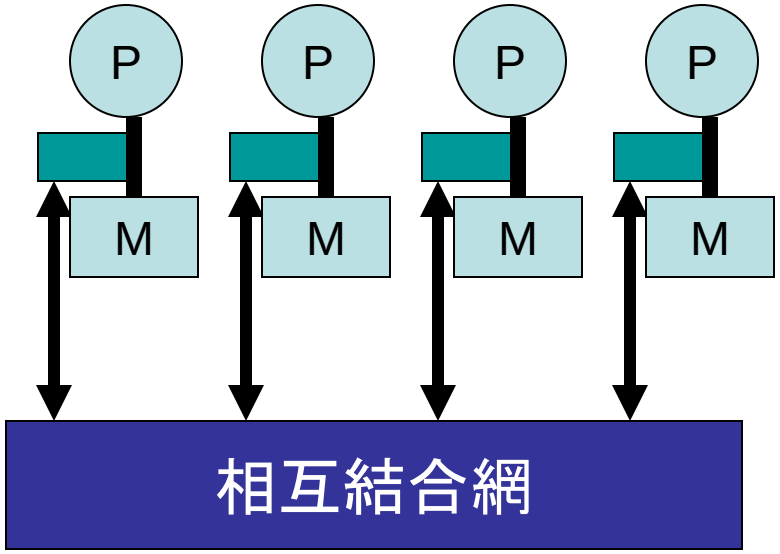
- 個々のプロセッサ
  - ベクトルプロセッサ　⇒　10年前はこのタイプが多かった
    - 一つのプロセッサで行列演算等を高速処理可能
  - スカラープロセッサ:
    x86 (IA32), Power, Itanium (IA64), Sparc

- 最近のプロセッサの動向
  - multi-coreが標準になってきた
  - さらにmany-core（数個～十数コア）プロセッサが登場しつつある
    - Intel & AMD ⇒ 4～6 core IA-32
    - IBM Cell Broadband Engine (8 core)
    - ClearSpeed (96 core×2)
    - Intel conceptual many-core chip (80 core)
  - ALU（算術演算機構）を多数持つプロセッサも
    - GRAPE-DR (512 ALU)

並列計算機の変遷

- 科学技術計算向けベクトルプロセッサ
  - 一種の並列計算機とみなせる（パイプライン並列）
  - ベクトルプロセッサを複数持つ並列ベクトルが登場
- スカラプロセッサをベースにした並列計算機
  - ～100プロセッサ程度の共有メモリマシン（SGI等）
  - quad-core CPUの登場により、8 core程度であればデスクトップPCでも共有メモリ並列システムとなる
  - MPP (Massively Parallel Processor)：1980年代後半から多数登場 ⇒一部を除き消滅しつつある
- クラスタ型計算機の登場
  - 以前は NOW (Network of Workstation), COW (Cluster of Workstation) 等と呼ばれていた
  - Linux PC を用いたものが現在の主流（Linuxがオープンシステムであるため、MPI等の並列化ツールも充実）
  - 数千プロセッサ規模のものが多数構築されている

分散メモリ型並列計算機



任意のプロセッサ間で
メッセージを送受信

P ... Processor
M ... Memory
NIC (network interface controller)

●CPUとメモリという一つの計算機システムが、ネットワークで結合されているシステム

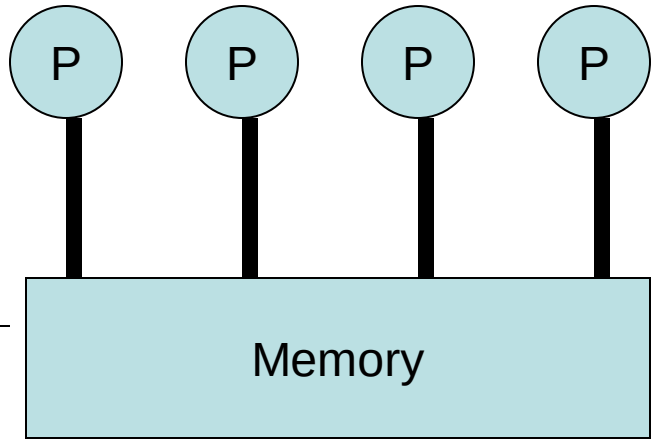
●それぞれの計算機で実行されているプログラムはネットワークを通じて、データ(メッセージ)を交換し、動作する

●比較的簡単に構築可能・拡張性(scalability) が高い
- ◆超並列計算機 (MPP: Massively Parallel Processing
- ◆クラスタ型計算機

分散メモリ型並列計算機の特徴

- 基本的にCPU+memory(+I/O)という逐次計算機構成を何らかのネットワーク（専用 or 汎用）で結合しているため、ハードウェア的にシンプル
- プログラム上からの明示的なmessage passingで通信を行うためユーザプログラミングは面倒
  - MPI (Message Passing Interface)のような標準的なツールが提供されている
  - ソフトウェア分散共有メモリによる簡便なアプリケーション記述の試みも
  - domain decompositionのような単純なデータ並列や、master/worker型の処理は比較的容易に記述可能
- システム性能は個々のプロセッサ／メモリの他、相互結合網の性能によって大きく左右される
- 1980年代後半からMPPの典型的な実装として登場、現在はPCクラスタの基本的アーキテクチャとなっている

共有メモリ型計算機



複数のプロセッサからの同時アクセスを整理することが必要

- 複数のCPUが一つのメモリにアクセスするシステム
- それぞれのCPUで実行されているプログラム（スレッド）は、メモリ上のデータにお互いにアクセスすることで、データを交換し、動作する
- 大規模サーバ
- 最近はプロセッサ１台が複数のプロセッサコアの共有メモリシステムになっている
- アーキテクチャ的にはさらにSMPとNUMAに分かれる（後述）

共有メモリ型並列計算機の特徴

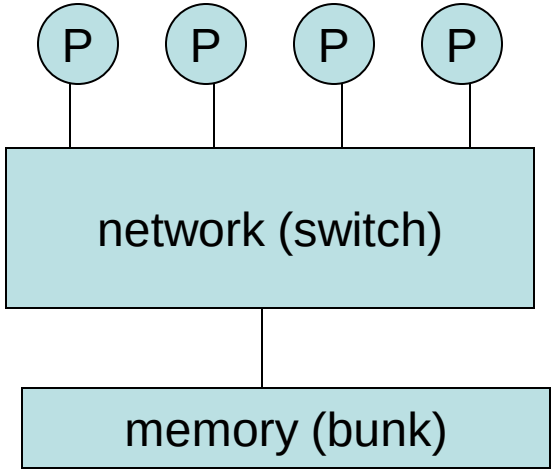
- ハードウェアによる共有メモリの提供により、ユーザにとってアプリケーションが非常に書き易い
  - multithreadプログラミング環境（POSIX thread等）
  - 共有メモリを前提とした簡易並列記述システム（標準的なものはOpenMP）
- 「メモリ」という極めてprimitiveな構成要素を共有化しているため、性能を上げるには非常に多くのハードウェア的、アーキテクチャ的工夫が必要
- 多数のプロセッサが1つのメモリ要素をアクセスする状況が簡単に記述でき、極端な性能ボトルネックを生じ易い
  - システムのscalabilityの確保が困難（数百プロセッサが限界）
- 概念的には前頁のような共有バスのイメージだが実際にはscalabilityを確保するためより複雑になっている

共有メモリ型計算機の構成の詳細

- system scalabilityを確保するため、単純バス構造の共有メモリシステムはもはや存在しない
  - bus bottleneck（busは一時には1つのtransactionで占有されてしまう）
  - 複数busを持つシステムもかつてはあった
- 共有メモリへのアクセス衝突を避けるための工夫
  - memory bank分け：適当なアドレスブロック毎に別のmemory moduleに分散して振り分け
  - crossbar networkの導入：プロセッサとメモリの結合が実際にはスイッチ結合になっている
  - coherent cache：各プロセッサは固有のキャッシュを持ち、普段はそのデータを参照する。他のプロセッサによるデータ更新をキャッチし、うまく自分のキャッシュに反映する。
  - NUMA (Non-Uniformed Memory Access)：物理的にはmemory moduleが分散していて、アドレスに寄るメモリへの距離の差が存在する。coherent cacheと共に用いられるのが普通。

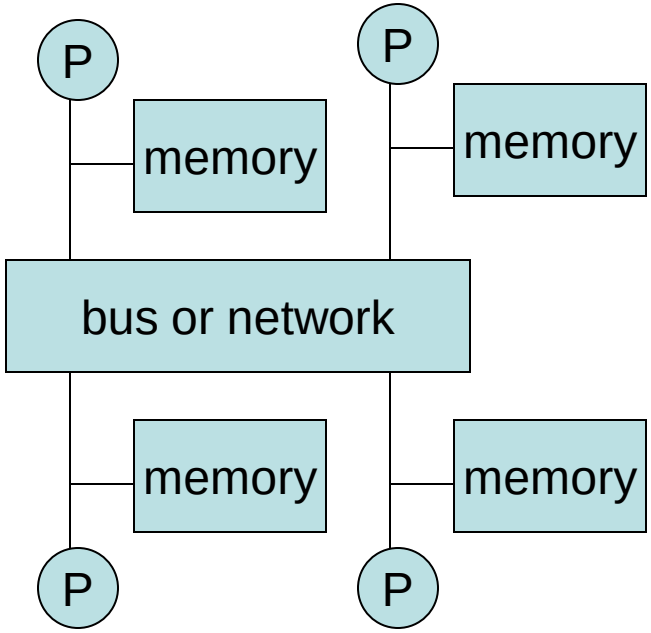
共有メモリアーキテクチャ:SMP

- SMP (Symmetric Multi-Processor)
  - 各プロセッサから見てどのmemory moduleへの距離も等しい
  - 構成としては、複数のプロセッサが共通のバスまたはスイッチを経由して、等しくmemory module(群)に接続されている
  - コモディティスカラプロセッサとしては、Intelプロセッサがこの方式
  - 大規模システムとしては富士通のHPC2500シリーズ、日立SR16000シリーズ等が該当する
  - coherent cacheとの併用が一般的
  - どのプロセッサからもデータが等距離にあるので偏りを心配しなくてよい
  - トラフィックが集中した場合に性能低下を防げない

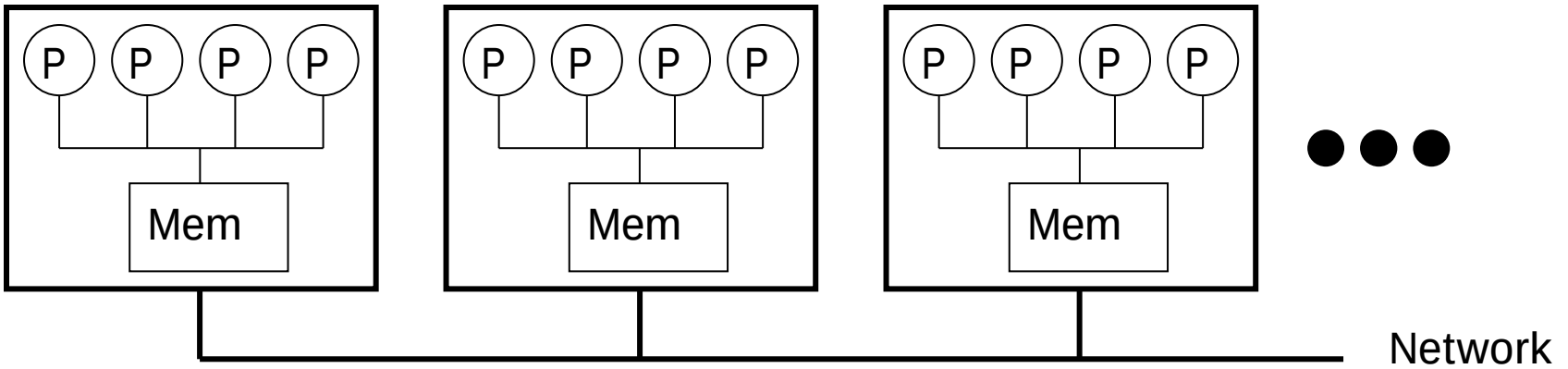


NUMA共有メモリアーキテクチャ

- NUMA (Non-Uniformed Memory Access)
  - CPUに付随して固有のmemory moduleがある
  - 共有バスまたはスイッチを介して、他のCPUのmemory moduleも直接アクセス可能
  - 遠距離memory moduleへのアクセスには時間が余計にかかる（non-symmetric）
  - コモディティスカラプロセッサとしてはAMD (Opteron)がこの方式
    ⇒ 最近、Intelも同様のアーキテクチャになった (Nehalem)
  - 大規模システムとしてはSGI Origin, Altixシリーズ等が該当
  - データをうまく分散し、参照の局所性が生かせれば性能を大幅に向上可能
  - 遠距離アクセス時の遅延時間増加に注意



分散／共有メモリ・ハイブリッド



- 共有メモリと分散メモリの組み合わせ
- 分散メモリ型システムの各ノードがそれ自身共有メモリアーキテクチャになっている (SMP or NUMA)
- マイクロプロセッサ自体が1チップで共有メモリ構成（マルチコア）となっていることが大きな要因、近年のマルチコアプロセッサ普及により急激に主流となった

アクセラレータ付並列システム

- 分散メモリ型計算機の各ノードが汎用CPUだけでなく演算性能を加速するハードウェア(アクセラレータ)を伴う
  - GPU (Graphic Processing Unit)
    最近ではGPGPU (General Purpose GPU) と呼ばれ、GPU上で汎用プログラミングも可能に
  - FPGA (Field Programmable Gate Array)
    特殊用途向けに再構成可能なハードウェア
  - 汎用アクセラレータ
    ClearSpeed等
  - プロセッサ自体がハイブリッド構成
    CBE (Cell Broadband Engine) ⇒ LANL Roadrunner

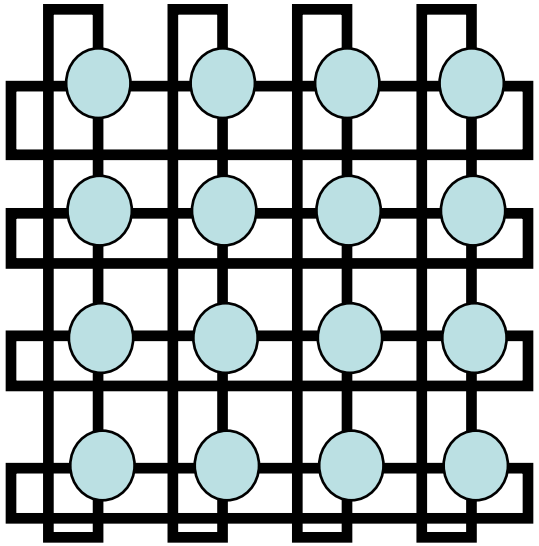
並列処理ネットワーク(相互結合網)

- 役割
  - 分散メモリアーキテクチャに基づく並列計算機における明示的なデータ交換
  - CC-NUMAアーキテクチャ (Cache Coherent NUMA)に基づく並列計算機におけるデータ及び制御メッセージの転送
- 特性／分類
  - static (direct) / dynamic (indirect)
  - diameter (distance)
  - degree (number of links)
- 性能指標
  - throughput
  - latency

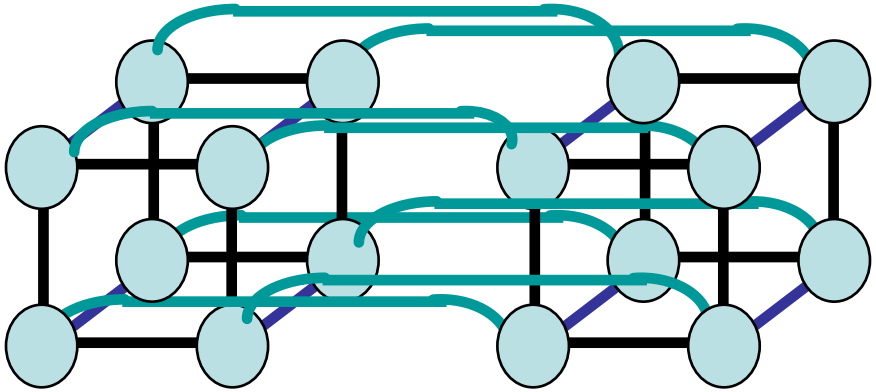
直接網(静的網)

- ノード(プロセッサ)に数本のリンクを持ち、それらが互いに結合してネットワークを形成
- ノード上でのルーティングが行われるがノード以外のスイッチは持たない
- 代表的な直接網トポロジ
  - 2-D/3-D Mesh/Torus
  - Hypercube
  - Direct Tree

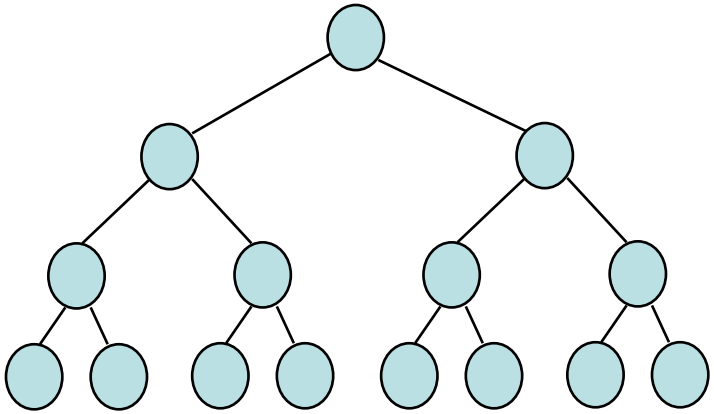
Mesh/Torus (k-ary n-cube)



Hypercube (n-cube)

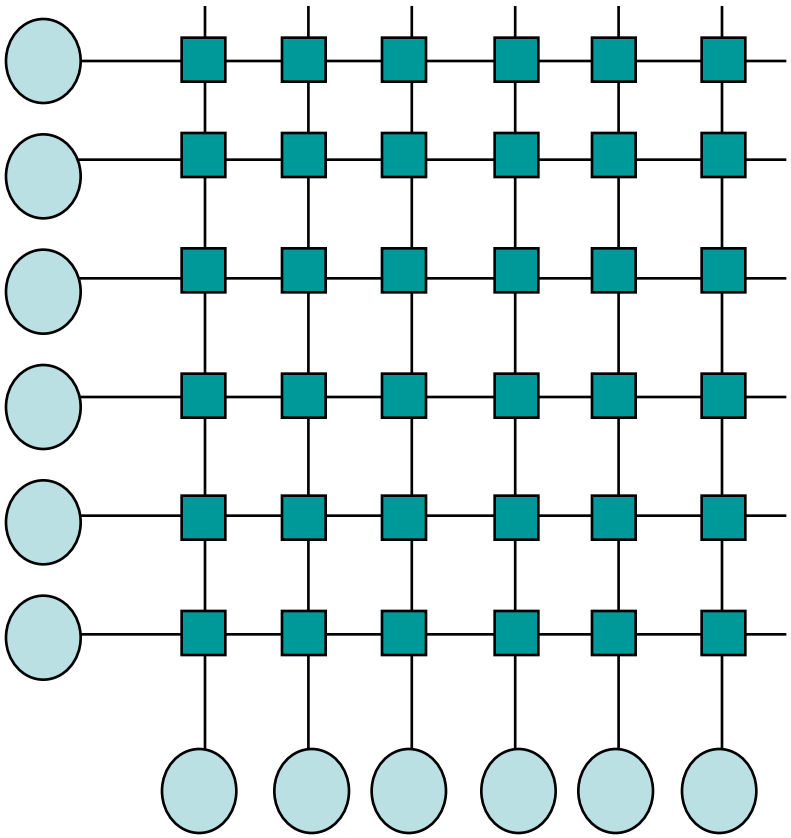


Direct Tree

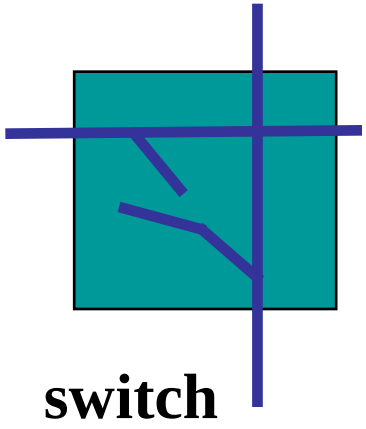


間接網(動的網)

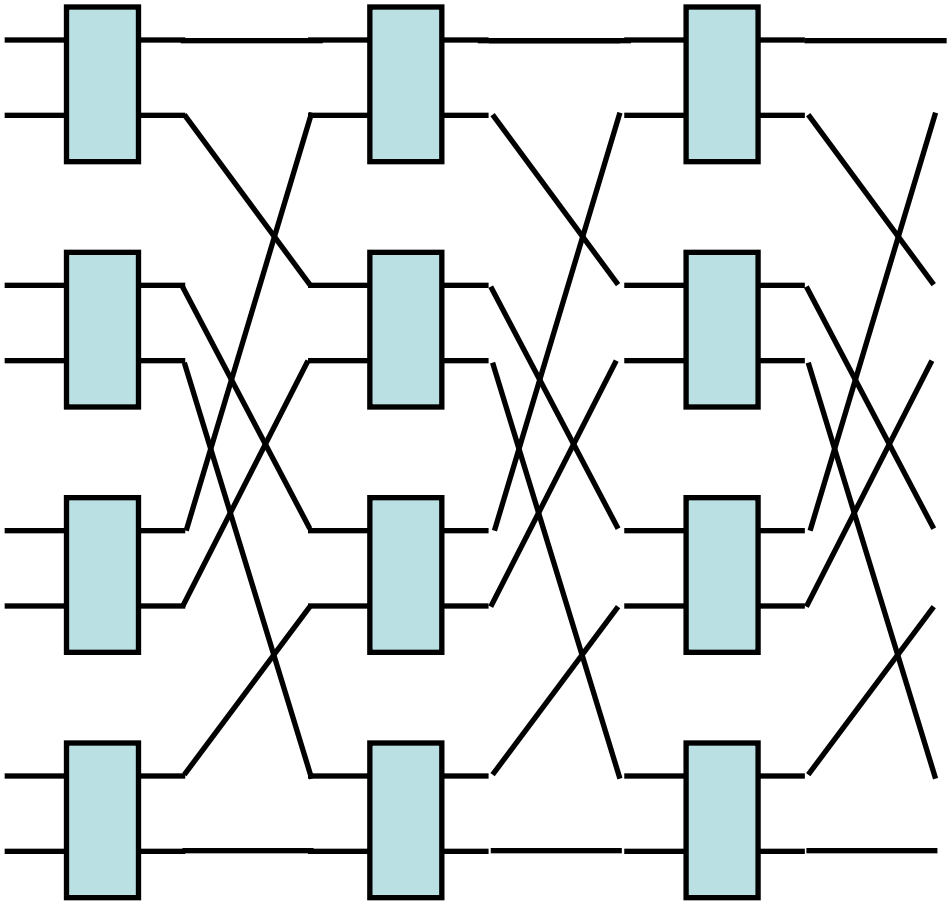
- ノードからは一般的に1本のリンクのみ(例外あり)
- 各ノードからのリンクを1つ以上のスイッチで結合してネットワークを形成
- スイッチでのルーティングが基本
- 代表的な間接網
  - Crossbar
  - MIN (Multistage Interconnection Network)
  - HXB (Hyper-Crossbar)
  - Tree (Indirect)
  - Fat Tree



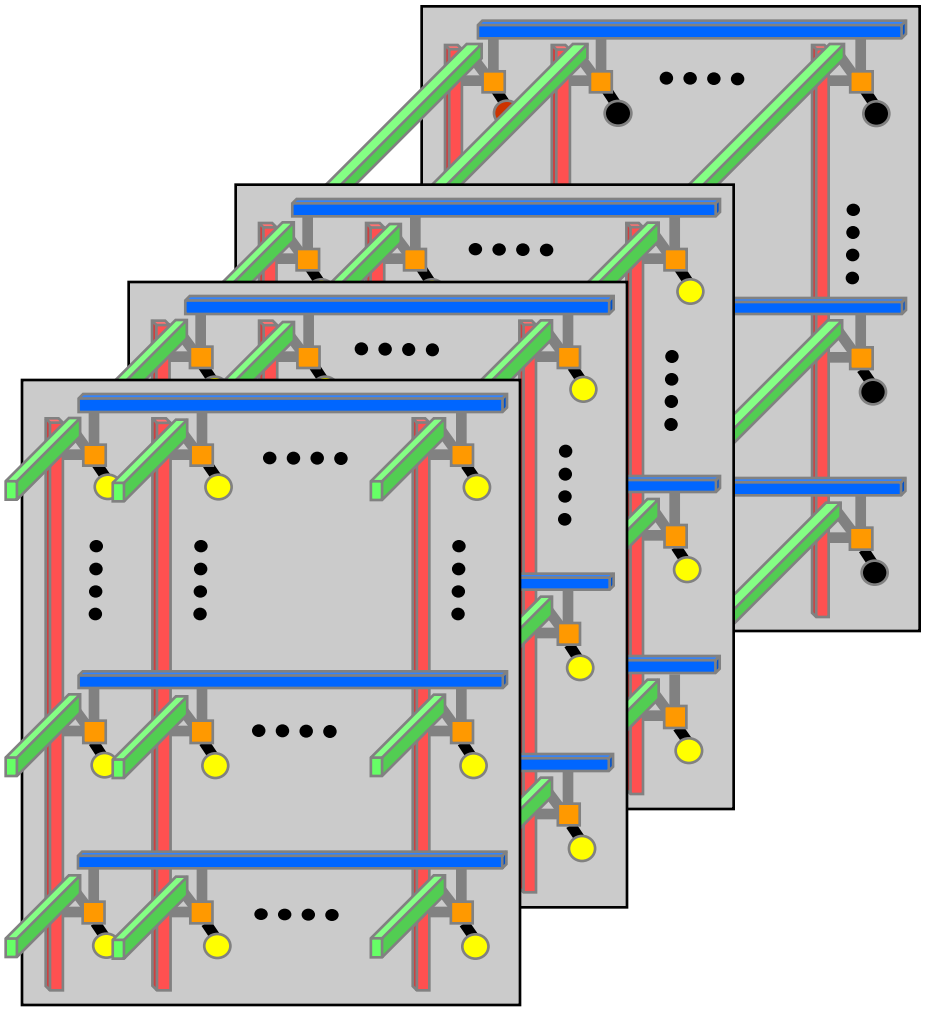
Crossbar



MIN (Multi-stage Interconnection Network)



MDX (Multi-Dimensional Crossbar)... HXB



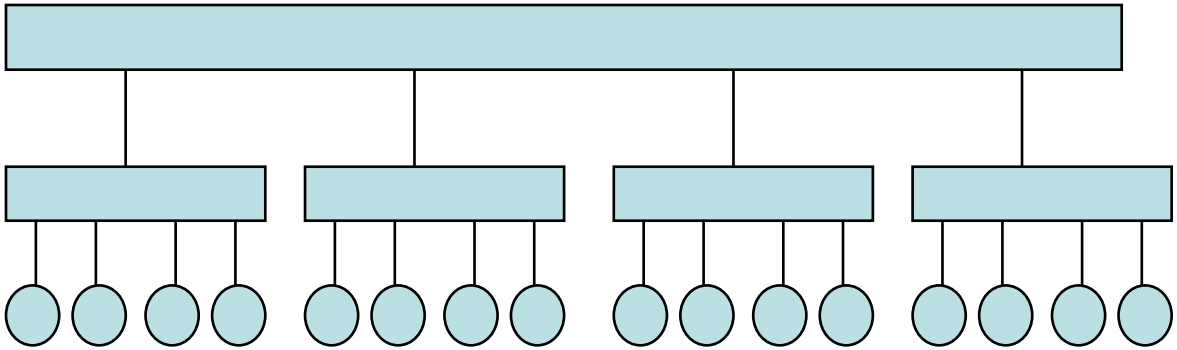
X-dimension Crossbar Switch

Y-dimension Crossbar Switch

Z-dimension Crossbar Switch

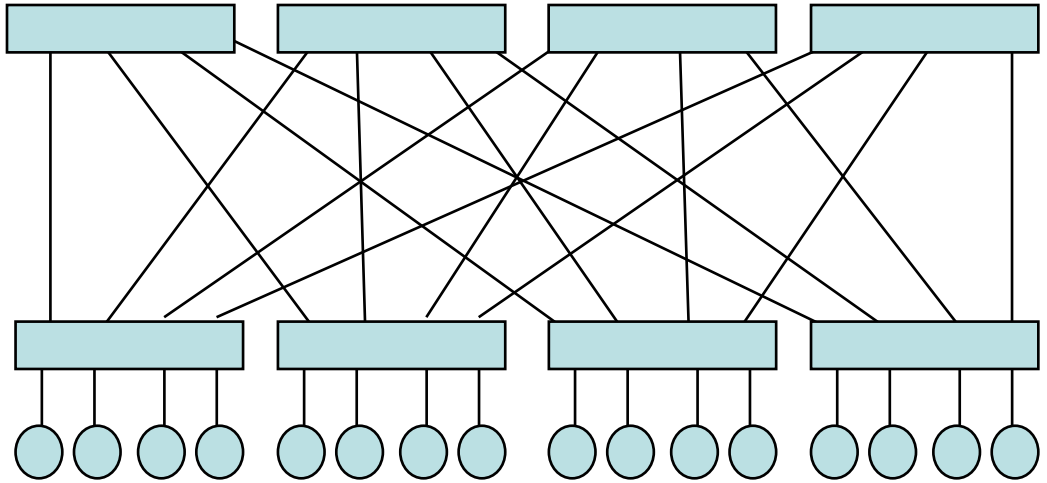
EX (Exchanger)

PU (Processing Unit)



Tree

Fat Tree



並列処理ネットワークの性能メトリック

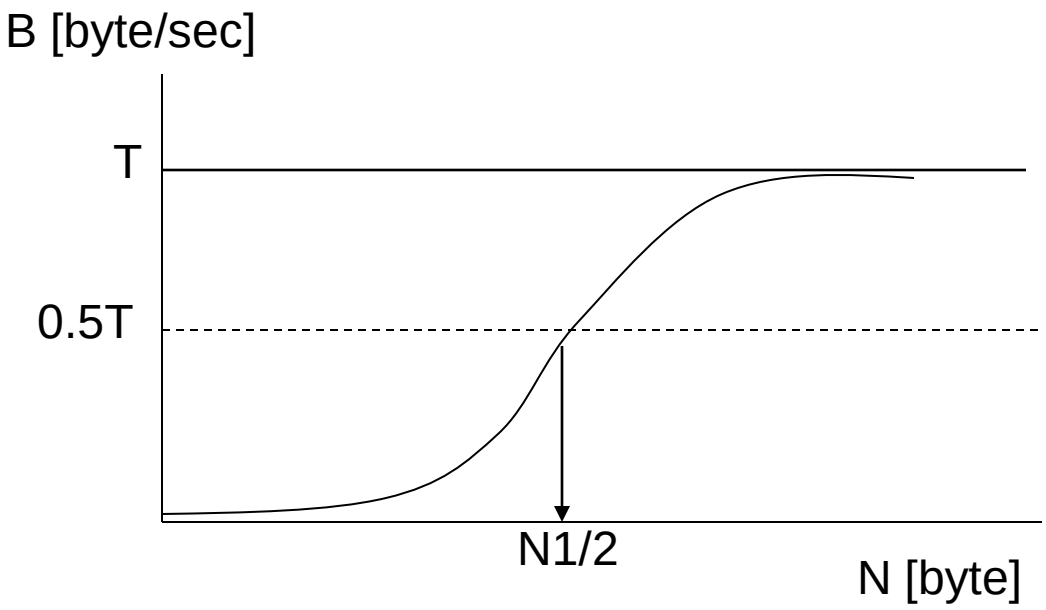
- Throughput（スループット）
  - リンクあるいはネットワーク全体の単位時間当たりのデータ転送性能
  - 単位：[byte/sec]
    （あるいは [bit/sec]、8bit=1byteとは限らないので注意が必要）

- Latency（遅延時間）
  - 狭義：転送すべきデータの先頭がsourceを出発してからdestinationに到着するまでの時間（ここではこれに従う）
  - 広義：転送すべきデータ全体がsourceを出発してからdestinationに到着するまでの時間
  - 単位：[sec]

ネットワーク転送性能とメッセージ粒度

- ネットワークリンク上で他のメッセージとの衝突がないとする。
  T [byte/sec]のスループットとL [sec]の遅延時間を持つネットワーク上で、*N* [byte]のメッセージを完全に転送し終わるまでの時間*t* [sec]と、有効バンド幅*B* [byte/sec]は以下のようになる

  $t = L + N/T \qquad B = N / t$



ここで、理論ピークバンド幅(T)の半分の0.5Tの性能が出るメッセージ長を$N_{1/2}$ (N-half「半性能長」)と表す。
理論的には

$N_{1/2}[byte] = L \times T$

となる。
$N_{1/2}$は「この長さ以下ではLがdominantで、この長さ以上ではTがdominantである」ことを表し、これが小さい程、短いメッセージの通信に強いネットワークということになる。

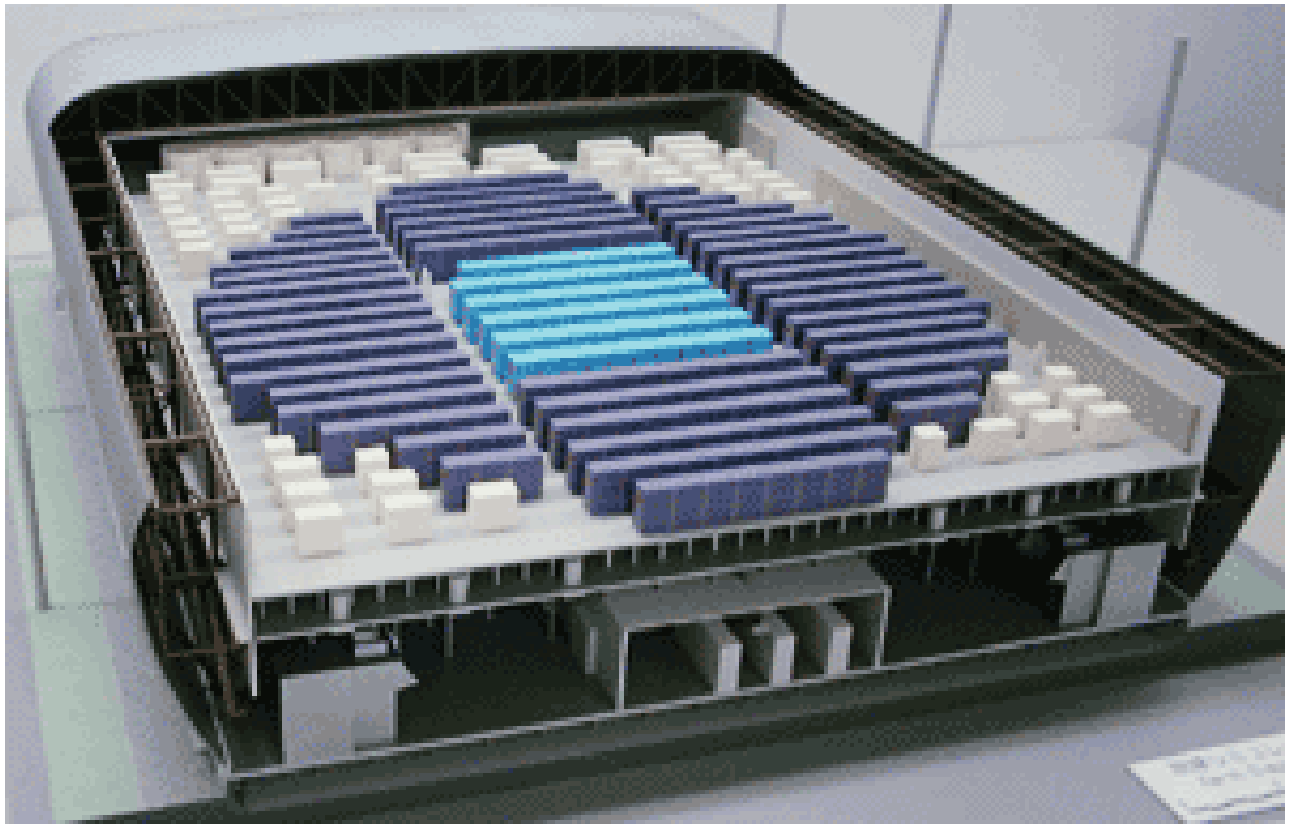
実際の並列計算機概観

- システムの分類
  - MPP (超並列計算機)
    - IBM BlueGene/L
    - Cray RedStorm, XT-series
    - 筑波大／日立 CP-PACS (SR2201)
  - 大規模並列ベクトル計算機
    - NEC 地球シミュレータ
    - 富士通 VPP-5000
  - スカラ並列計算機（クラスタを含む）
    - LLNL/IBM ASCI Purple
    - 筑波大/日立/富士通 PACS-CS
    - 筑波大・東大・京大/Appro・日立・富士通 T2K
  - アクセラレータ付ハイブリッド計算機
    - 筑波大/HP FIRST
    - 東工大/SUN TSUBAME
    - LANL/IBM Roadrunner

CP-PACS

- 筑波大学計算物理学研究センター
- 筑波大学＋日立
- 1996年完成
- 大学主導計算機として世界最高速となった貴重な例
- 計算物理学のための計算機
- ソフトウェアベクトル処理のために強化されたプロセッサ
- 2048 CPU 614GFLOPS

地球シミュレータ



- 海洋技術研究所・地球シミュレータセンター
- NEC
- 2002年完成
- 国産ベクトル計算機として世界最高速
- 大規模気象シミュレーション等様々な分野で応用
- 共有メモリ結合されたベクトルプロセッサ
- 5120 CPU
  40 TFLOPS

ASCI Purple

- Lawrence Livermore National Lab.
- IBM
- 2006年完成
- 共有メモリスカラプロセッサをネットワーク結合
- 応用分野は特になし
- 12208 CPU
  92 TFLOPS

Blue Gene/L

- Lawrence Livermore National Lab.
- IBM
- 2005年完成
- 組み込み用低性能プロセッサを非常に多数ネットワーク結合
- 素粒子計算、流体計算等
- 65536 CPU
  360 TFLOPS

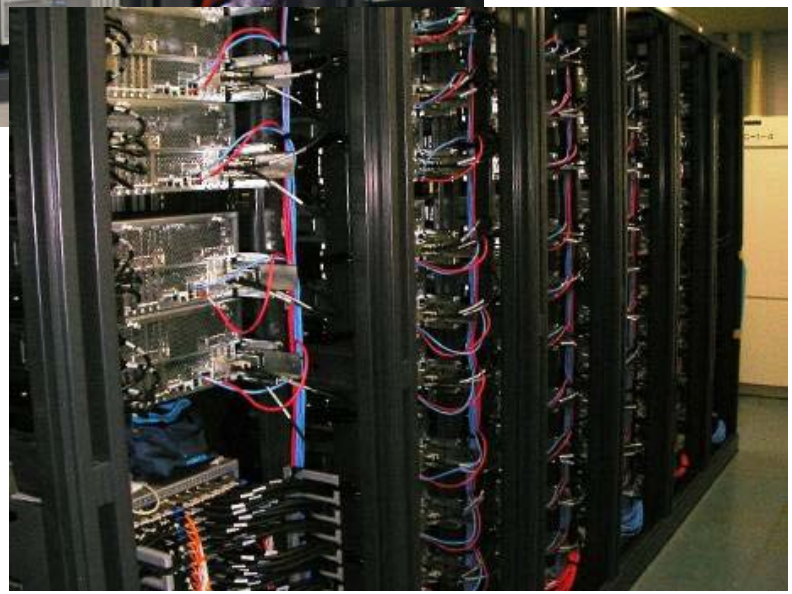
FIRST

- 筑波大学計算科学研究センター
- HP + 専用アクセラレータ
- 2005年完成
- 各ノードをdual socket Xeon + GRAPE-6 board (Blade-GRAPE) で構成したヘテロクラスタ
- 計算宇宙物理学用
- 256 nodes/512 cores + 1024 GRAPE-6 chip
- Host: 3.1 TFLOPS
  Blade-GRAPE: 33TFLOPS

PACS-CS

- 筑波大学計算科学研究センター.
- 筑波大＋日立＋富士通
- 2006年完成
- PCクラスタ技術と汎用Ethernetの多重結合を使ったユニークなネットワーク
- 計算科学全般
- 2560 CPU
  14.4 TFLOPS

TSUBAME

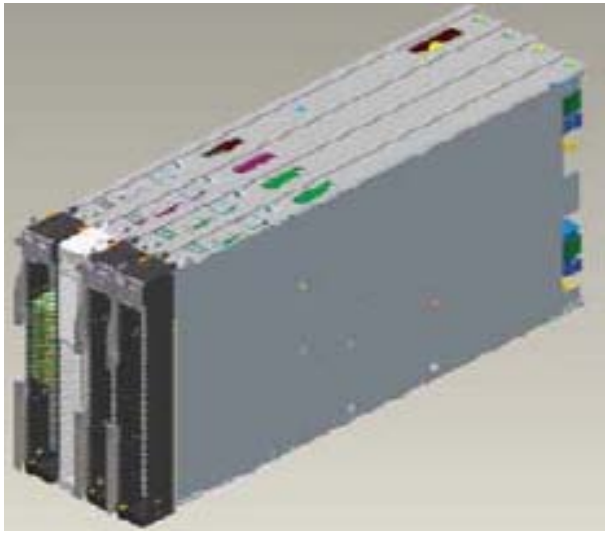


- 東京工業大学
- SUN Microsystems + NEC
- 2006年完成
- 各計算ノードをmulti-core CPU (dual-core Opteron) とアクセラレータ(ClearSpeed) によって構成したハイブリッドクラスタ
- 計算科学全般
- 655 nodes/10480 cores
- 109.7TFLOPS(アクセラレータ込み)

T2K筑波

- 筑波大学計算科学研究センター.
- Appro International + Cray Japan
- 2008年完成（6月稼動開始）
- ノード性能とネットワーク性能をコモディティとして最高レベルに上げたPCクラスタ
- 計算科学全般
- 2592 CPU chip = 10368 CPU core
  95 TFLOPS

Roadrunner



- Los Alamos National Lab.
- IBM
- 2008年完成
- 各ノードにOpteronプロセッサとIBM Cell Broadband Engineを搭載したハイブリッド型クラスタ
- 世界初のPFLOPSコンピュータ ⇒ 2008/6-11, 2009/6の3期連続 TOP500#1
- 129600プロセッサ
  1.46 PFLOPS
  (Linpack: 1.11 PFLOPS)

性能に影響を与える要因

- 通信・同期のコスト、通信ネットワークの特性（トポロジ等）
  - 並列処理効率の観点から、システム規模を大きくするに連れて並列プロセスの粒度が小さくなるため通信ボトルネックが目立つ
  - システム規模を大きくすることにより通信・同期コストが一般に大きくなるため、「相乗効果」として効いてきてしまう
- 単体プロセッサ上の性能
  - データが問題の並列化に適合してマップされているか（特に、NUMA共有メモリの場合）
  - クラスタ等のスカラプロセッサではcacheがどれくらい有効活用されているか（演算部分でのさらなる最適化）
    ⇒並列化によって好ましい方向に向かう場合がある
    - 並列プロセスの処理粒度が小さくなると、working set（ある範囲の計算上で必要なデータ集合）も小さくなる、一定の粒度以下では全working setが完全にcacheに乗る
      ⇒ Super-Linear現象が起きることがある（e(p) > 1)

分散メモリシステム上での通信の注意

- point-to-pointの通信とcollective通信
  - point-to-point: 各通信ペア上の一対一通信。domain decompositionされた偏微分方程式の差分解等（隣接通信）
  - collective: 全プロセスが関与する「集合通信」。
    - barrier: 全プロセスの同期待ち
    - broadcast: 1プロセスのデータを全プロセスに「放送」
    - reduction: 全プロセスの値を「縮約」し、1プロセスに集める（例：特定の変数の総和を求める）
- point-to-point通信は問題の持つ特性とネットワークの形状が合致することが望ましい（が、それをユーザが知りえる場合とそうでない場合がある）
- collective通信はシステムのMPIライブラリ等に依存するが、プロセス数の増加に従って*log(p)*あるいはそれ以上のオーダーで増えるのが一般的
- 並列化によるプロセスの細粒度化とシステム並列度増大による間接的な通信コスト増加に注意！

並列処理システムの動向

- MPPは徐々に衰退（特定マシンのみ躍進）
- コモディティ化が進む（クラスタの台頭）
  - コモディティなスカラープロセッサ（IA32=x86）
  - コモディティなネットワークとスイッチ
    - Ethernet (1Gbps ⇒ 10G ??)
    - Infiniband (2GByte/sec、高級機器だったが徐々に価格低下）
- 全体的に、演算性能：メモリ性能：通信性能のバランスが悪くなっている
  - 演算性能はプロセッサのmulti-core化等により順調に向上
  - メモリ性能（バンド幅）は相対的に低くなっている（プロセッサが速すぎる）
  - 通信性能は段階的に上がっていく（Ethernet等）
  - プロセッサコストは*O(N)*だがネットワークコストは*O(N log N)*程度なので相対的にシステム価格を圧迫
  - 結果的に並列処理効率を上げるのが難しくなってきている。より一層のアルゴリズム、ソフトウェア上の工夫が必要。

まとめ

- 並列処理システム／アーキテクチャ
  - 逐次プロセッサ（コア）性能の限界より、全体性能は並列処理に頼らざるを得ない
  - 性能を維持しつつ拡張性（scalability）を確保
  - 分散メモリ vs 共有メモリ
- 並列処理ネットワーク
  - scalabilityが最も重要
  - 以前はMPP向け、現在はcommodity networkの充実によりfat-treeでかなりの規模が可能
  - 2つの性能メトリック：throughput & latency
- 並列処理システムの実際
  - 数十万コア規模まで拡大、最高性能約1PFLOPS
  - 基本は分散メモリシステムだがmulti-coreの一般化によりハイブリッドが基本
  - アクセラレータが注目されている

課題

1. 実際に構築された大規模並列計算機（講義資料で取り上げたもの、あるいはその他）のうち１台を選び、規模・性能・アーキテクチャ的特徴について詳細を調べてまとめよ。調査にはweb等を活用することを薦めるが、全て出展（URL等）を明らかにすること。

2. IntelのXeonシリーズCPUと、AMDのOpteronシリーズCPUは、現在共にx86アーキテクチャに基づくマルチコアプロセッサとして広く用いられている。しかし、（Nehalem以前の）XeonはSMPアーキテクチャ、OpteronはNUMAアーキテクチャを持ち、その性質は異なる。メモリバンド幅、プログラミング上の注意等の観点から、両者を比較せよ。

3. 並列処理ネットワークの理論ピークスループットは年々急激に増加している。しかし、一般的に、レイテンシの削減はそれほど急激には進まない。この結果として、プログラミングやアプリケーション上でどのような影響が現れるか論ぜよ。