並列分散システム特論

イントロダクション

佐藤

もくじ

- 並列処理とは
  - 並列プログラミング
- クラスタコンピューティング
  - PCクラスタ
- グリッドコンピューティング

- 担当：佐藤 三久
- 内容(暫定的授業案)
  PCクラスタなどの並列システム、OpenMPなどの並列プログラミング、グリッドコンピューティングを Java 分散プログラミングなど、高性能コンピューティングのための並列分散システムについて解説する。
- 目次
  具体的な並列分散システムを取り上げて解説する。次の項目について、講義する予定：
  - PCクラスタの最新動向
  - OpenMPと共有メモリプログラミング
  - MPIによるプログラミング
  - 分散プログラミング入門 (RMI)
  - Java分散オブジェクト指向プログラミング: J2EE, Jini
  - グリッドコンピューティング： Globus & grid PRC
  - P2Pコンピューティング： Jxta
- 毎回、講義の始めにその回の講義の内容に関する資料を配付する。その資料は、このページのあるURL http://www.hpcs.is.tsukuba.ac.jp/~msato/lecture-note にて、公開する。
- 参考書、文献等については、その都度、紹介する。
- 質問に関しては、 msato@is.tsukuba.ac.jp で、随時、受け付ける。
- 成績評価については、全講義講義終了後、レポートの提出により評価。

講義予定

- 第１回目 (2003/12/3) イントロダクション
- 第２回目 (2003/12/10) 並列プログラミング(MPIとOpenMP)
- 第３回目 (2003/12/17) 分散プログラミング入門
- 第４回目 (2003/12/24) Java分散オブジェクト指向プログラミング(１)
- 第５回目 (2004/1/14) Java分散オブジェクト指向プログラミング(２)
- 第６回目 (2004/1/21) グリッドコンピューティング
- 第７回目 (2004/1/28) Web ServiceとGT3
- 第８回目 (2003/2/8) P2Pコンピューティング(１)
- 第９回目 (2003/2/12、振替) P2Pコンピューティング(２)
- 第１０回目 (2003/2/18) 予備

並列処理と分散処理

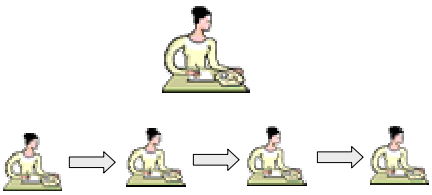
- 並列処理(parallel processing)とは、複数のプロセッサを用いて、処理を高速化する技術
  - HPC(High Performance Computing)
    - 数値シミュレーションなど
  - HTC (High Throughput Computing)
    - 大量のデータ処理
- 分散処理(distributed processing)は、もちろん、複数のプロセッサを用いるため処理を高速化することもあるが、本質的にはいろいろな場所で行う様々な処理、あるいは機能を結合し、機能分担させることが目的であり、必ずしも高速化だけが目的ではない
  - 分散オブジェクト技術
  - RMI , J2EE, Jini…

レーテンシとスループット（バンド幅）

- レーテンシは１つの仕事が始まり、終わるまでの時間
- スループットとは、単位時間当たり処理できる仕事量のこと。
  - 通信では、バンド幅となる。
- パイプラインはスループットは向上させるが、レーテンシは短くならない。

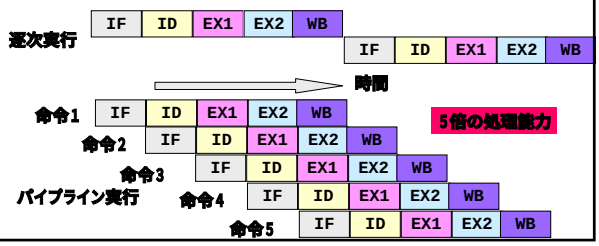
パイプライン

- いわば、流れ作業
- 一人の人がやるよりも、機能分担して、流れ作業をすればいいということ。



パイプラインアーキテクチャ

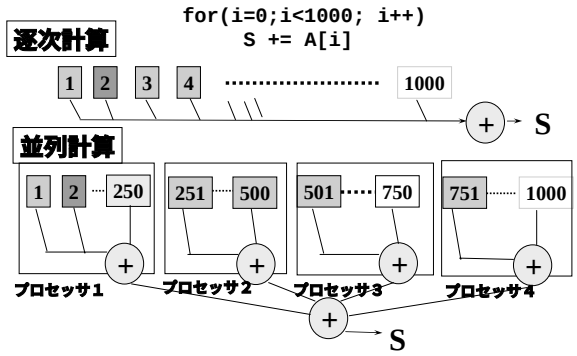
- 処理のフェーズをオーバーラップして、同時に複数の命令を時分割して実行する方式
- 同じ時刻ではそれぞれの命令は異なるフェーズを実行している。
- 細かくすればするほど、速度は向上する。
- 早いクロックのプロセッサでは各フェーズは細かくなる。



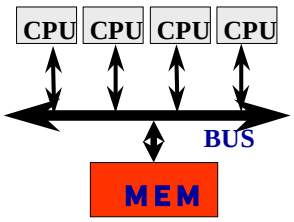
並列と並行

- 並列処理の「並列」(parallel) という用語は、「物理的に同時に動作する」ことを意味する
  - 複数のCPUで同時に処理されているものを制御する場合には「並列」
- 「並行」(concurrent) は「論理的に同時に動作する」ことを扱う場合に用いる
  - 一つのCPUの中でいろいろな処理に対応するサービスを行うオペレーティングシステムは「並行」処理の典型的な例
- スレッド(thread)：一連の命令実行の流れのことをいう
  - たとえば、１つのプログラムの実行のこと
  - ２つのプログラムを同時に実行することをマルチスレッド(multi-thread)実行という。

並列処理の簡単な例



共有メモリ型計算機

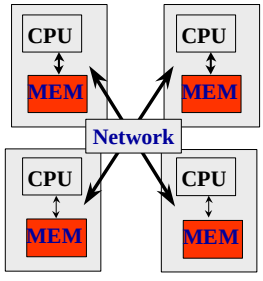


◆複数のCPUが一つのメモリにアクセスするシステム。

◆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

分散メモリ型計算機



◆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

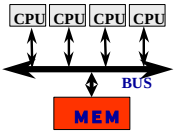
◆それぞれの計算機で実行されているプログラムはネットワークを通じて、データ（メッセージ）を交換し、動作する

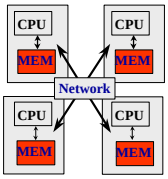
◆超並列（MPP：Massively Parallel Processing)コンピュータ

◆クラスタ計算機

並列処理の利点

- **計算能力が増える。**
  - １つのＣＰＵよりも多数のＣＰＵ。
- **メモリの読み出し能力（バンド幅）が増える。**
  - それぞれのＣＰＵがここのメモリを読み出すことができる。
- **ディスク等、入出力のバンド幅が増える。**
  - それぞれのＣＰＵが並列にディスクを読み出すことができる。
- **キャッシュメモリが効果的に利用できる。**
  - 単一のプロセッサではキャッシュに載らないデータでも、処理単位が小さくなることによって、キャッシュを効果的に使うことができる。
- **低コスト**
  - マイクロプロセッサをつかえば。





→ クラスタ技術

並列プログラミング

- **メッセージ通信** (Message Passing)
  - 分散メモリシステム（共有メモリでも、可）
  - プログラミングが面倒、難しい
  - プログラマがデータの移動を制御
  - プロセッサ数に対してスケーラブル
- **共有メモリ** (shared memory)
  - 共有メモリシステム（DSMシステムon分散メモリ）
  - プログラミングしやすい（逐次プログラムから）
  - システムがデータの移動を行ってくれる
  - プロセッサ数に対してスケーラブルではないことが多い。

並列プログラミング

- **メッセージ通信プログラミング**
  - MPI,PVM
- **共有メモリプログラミング**
  - **マルチスレッドプログラミング**
    - pthread, solaris thread, NT thread
  - **OpenMP**
    - 指示文によるannotation
    - thread制御など共有メモリ向け
  - HPF
    - 指示文によるannotation,
    - distributionなど分散メモリ向け
- **自動並列化**
  - **逐次プログラムをコンパイラで並列化**
    - コンパイラによる解析には制限がある。指示文によるhint
- Fancy parallel programming languages

メッセージ通信プログラミング

- **sendとreceiveでデータ交換をする**

```
int a[250]; /* それぞれ、250個づつデータを持つ */

main(){    /* それぞれのプロセッサで実行される */
   int i,s,ss;
   s=0;
   for(i=0; i<250;i++) s+= a[i]; /*各プロセッサで計算*/
   if(myid == 0){   /* プロセッサ0の場合 */
      for(proc=1;proc<4; proc++){
         recv(&s,proc);  /*各プロセッサからデータを受け取る*/
         s+=ss;   /*集計する*/
      }
   } else {  /* 0以外のプロセッサの場合 */
      send(s,0);  /* プロセッサ0にデータを送る */
   }
}
```


POSIXスレッドによるプログラミング

- **スレッドの生成**

Pthread, Solaris thread

```
for(t=1;t<n_thd;t++){
  r=pthread_create(thd_main,t)
}
thd_main(0);
for(t=1; t<n_thd;t++)
      pthread_join();
```

```
int s;  /* global */
int n_thd; /* number of threads */
int thd_main(int id)
{ int c,b,e,i,ss;
  c=1000/n_thd;
  b=c*id;
  e=s+c;
  ss=0;
  for(i=b; i<e; i++) ss += a[i];
  pthread_lock();
  s += ss;
  pthread_unlock();
  return s;
}
```

- **ループの担当部分の分割**
- **足し合わせの同期**

OpenMPによるプログラミング

これだけで、OK!

```
#pragma omp parallel for reduction(+:s)
   for(i=0; i<1000;i++) s+= a[i];
```

OpenMPとは

- 共有メモリマルチプロセッサの並列プログラミングのためのプログラミングモデル
  - ベース言語(Fortran/C/C++)をdirective（指示文）で並列プログラミングできるように拡張
    - OpenMPの指示文は並列実行モデルへのＡＰＩを提供
    - 従来の指示文は並列化コンパイラのためのヒントを与えるもの
  - 科学技術計算向け（5%のコードが95%の実行時間を実行）
- 共有メモリマルチプロセッサシステムの普及
  - SGI Origin /ASCI Blue Mountain System
  - SUN Enterprise
  - PC-based SMPシステム
- 米国コンパイラ関係のISVを中心に仕様を決定
  - Oct. 1997 Fortran ver.1.0 API、Nov 2000 ver.2.0 API
  - Oct. 1998 C/C++ ver.1.0 API
  - http://www.openmp.org/

クラスタコンピューティング

- PCやネットワークなど、汎用（コモデティ）の部品を組み合わせて、並列システムを構成する技術
  - PCが急激に進歩した！
  - ネットワークも早くなっている！
  - 安価に、個人レベルでも並列システムを作れる！

クラスタコンピューティング

- クラスタシステム：既存のワークステーションやＰＣを（既存の）ネットワークで結合して，並列計算を行うシステム
- 第１世代：既存のＬＡＮで並列計算するシステム
  - Poorman's supercomputer
  - 遊休のワークステーションを利用
  - お茶大 Sun IPC cluster （計算センタのワークステーションを利用）
  - イーサネット
- 第２世代：クラスタ専用の計算機システム
  - etlwiz Alpha cluster, 東大富士川研クラスタ
  - 100 BASE-TX SWITCH，ATM
  - beowulf class クラスタとも呼ばれる．
- 第３世代：高速のネットワークによるクラスタ
  - 高並列計算機（ＭＰＰ）なみの性能
  - RWCP PC cluster
  - Myrinet，Gigabit Ethernet, Fiber Channel, DEC Memory Channel, IBM SP2 network
- その他
  - ＳＭＰクラスタ （UCB CLUMPS, RWC COMPaS)

クラスタコンピューティングを支える技術

- ハード，ソフトのコモデティ化，高性能化，標準化
- ハードウエア
  - プロセッサテクノロジロードマップの恩恵
    - 急激な高性能化，価格性能比の向上
  - ネットワークの高性能化
    - ethernet : 10Mbps から 100Mbps そしてGigabit either
    - Myrinetなどの SAN Network
  - 高性能 I/Oインタフェースの標準化
    - ＰＣＩなど
- ソフトウエア
  - 並列通信ライブラリの発展・標準化
    - PVM, P4, TCGMSG，MPI，MPI2
  - 標準ライブラリ上に並列ソフトウエアが開発できる．
  - フリーなオペレーテングシステムの普及

クラスタの“レシピ”

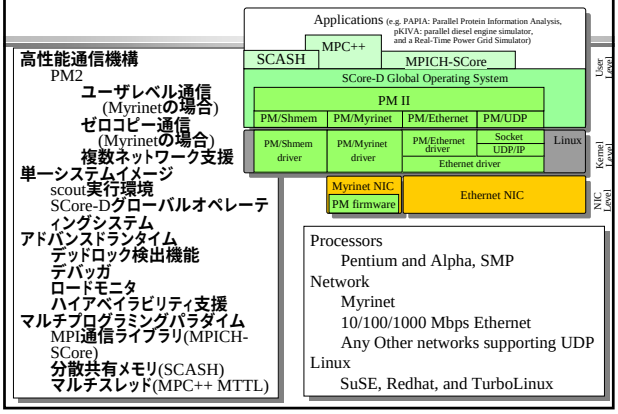
- まず，ノードプロセッサの選択
  - ＰＣならば，20万円，DEC alphaならば，50万円，SMP？
  - メモリサイズ，デスクサイズは？
- 何台？
  - （8台で，ＰＣならば160万円）
- ネットワークは何にする？
  - 100 Base Ethernetならば，ノードあたり2万円，スイッチ150万円，ハブ？
  - myrinetならば，ノードあたり30万円，スイッチ80万円
  - gigabit ether ?
- ノードのオペレーテングシステムは？
  - FreeなＯＳ(netBSD, linux)ならば，ただ
  - Solaris/x86, Digital Unix
  - Windows NT
- 通信ライブラリ
  - TCP/IPならば，MPICH (ただ！)
  - 専用のネットならば，ドライバが必要
- その他
  - ラック，ケーブル・ハブ，コンソールスイッチボックス（50万）
- あとは，組み立て！

Clusters in RWCP TRC

クラスタのソフトウエア

- SCore
  - クラスタ向けの基本ソフトウエア
    - 通常のソフトウエアを組み合わせただけでは不十分
  - 高速通信ライブラリPM
    - TPC/IPをつかわない
    - Myrinetなどの高速の専用ネットワークが使える
  - シングルシステムイメージ
    - ギャングスケジューリング：並列ジョブのマルチタスキング
  - 大規模クラスタのサポート

SCore Version 3 Software Architecture



高性能通信機構
- PM2
  - ユーザレベル通信 (Myrinetの場合)
  - ゼロコピー通信 (Myrinetの場合)
  - 複数ネットワーク支援

単一システムイメージ
- scout実行環境
- SCore-Dグローバルオペレーティングシステム

アドバンスドランタイム
- デッドロック検出機能
- デバッガ
- ロードモニタ
- ハイアベイラビリティ支援

マルチプログラミングパラダイム
- MPI通信ライブラリ(MPICH-SCore)
- 分散共有メモリ(SCASH)
- マルチスレッド(MPC++ MTTL)

Processors
- Pentium and Alpha, SMP

Network
- Myrinet
- 10/100/1000 Mbps Ethernet
- Any Other networks supporting UDP

Linux
- SuSE, Redhat, and TurboLinux

クラスタテクノロジの課題

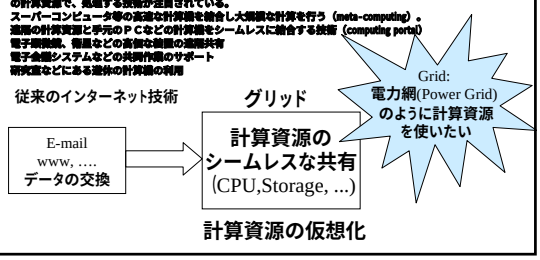
- 並列ソフトウエアは，ＭＰＰとそれほど違いはない．ソフトは依然として問題．
  - ＭＰＩはやはり、面倒！
  - OpenMP！ HPF？
- 通信インタフェースの高性能化
  - 既存のプロトコルでは通信性能を使いこなせない
    - Gigabit Ether 100MB/s以上だが，TCP/IP/Gigabitでは30MB/s
  - リモートメモリ通信
    - たとえば，VIA
  - 標準化，コモデティ化は依然として重要
- 並列プログラムのジョブ管理
  - マルチユーザ並列プログラム環境
    - たとえば，Score-D/gang scheduling(RWCP), implicit co-scheduler (UCB)
- スケーラビリティは重要か？中規模までならクラスタで十分
  - RWCP 1000CPU/512ノードのクラスタを構築
    - ネットワークが重要、ほどんどMyrinet
  - NCSAやSandia-Lab等で、大規模クラスタができている

グリッドとは

- 世界中のPCをつないで計算　seti@home,...
  - これだけではない！
- スーパーコンピュータをつないで大規模計算
  - これだけではない！

グリッドとは

- グリッド技術とは広域の高速ネットワーク上において大量のデータ、計算資源、貴重な装置等を共有し、協調作業、資源の有効活用するネットワーク基盤技術（ソフトウエア、ネットワーク、ハードウエア）と、これを活用する応用技術
  - 大量のＰＣ（クラスタ）などの計算資源の相互に共有し、大量かつ大規模な計算を行う。
  - 大規模なデータの処理。例えば、地球規模の観測データ、電波望遠鏡の観測データなどを大量のＰＣなどの計算資源で、処理する技術が注目されている。
  - スーパーコンピュータ等の高速な計算機を結合し大規模な計算を行う（meta-computing）。
  - 遠隔の計算資源と手元のＰＣなどの計算機をシームレスに結合する技術（computing portal）
  - 電子顕微鏡、衛星などの高価な装置の遠隔共有
  - 電子会議システムなどの共同作業のサポート
  - 研究室などにある遊休の計算機の利用



グリッドとは

- なにが変わったのか？
  - 以前から、分散コンピューティングの研究はあった。
  - インターネットの急激な進歩、普及
  - 全世界で共通な基盤を作ろうとしている！(Grid Forum, …,ApGrid)
- Grid:（高速）広域ネットワーク上で、データ、計算資源、装置を共有し協調作業をサポートするネットワーク基盤
- 何につかえるのか？
  - 計算資源（スーパーコンピュータ）を共有し、大規模計算を行う(meta-computing, MPI-G, …, ITBL?)
  - 遠隔の計算資源と手元のPC/WSをシームレスに結合(computing portal, GridPRC)
  - 大量のデータの処理(data intensive computing, gfarm for ATLAS)
  - 高価な装置の遠隔共有 (電子顕微鏡、衛星データ,加速器？)
  - 共同作業のサポート(電子会議システム、AccessGrid)
  - 遊休の計算機の利用、SETI@home, P2P, …, XtermWeb

グリッドのソフトウエア

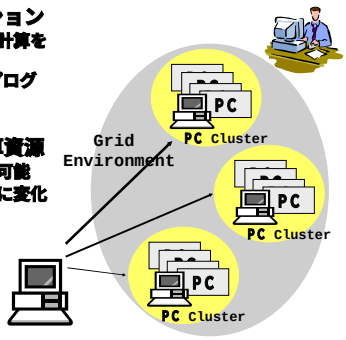
- Globus：最もmajorになりつつあるgridコンピューティングのツールキット(ミドルウエア)
  - ジョブ起動、セキュリティ、通信のためのライブラリ
  - globusrun : globus 版のrsh
  - MPICH-G: globus版のMPICH
  - gftp: globus版のftp
- Nimrod-G(enfusion): クラスタ、Grid向けのjob dispatcher. jobレベルのパラメータサーチを行うツール（GUI 付き）
- Condor-G: 遊休の計算機に対するジョブスケジューラ。プロセスの移送もサポート
- GridRPC: Grid上の遠隔手続き呼び出し
  - Ninf-G, NetSolve, …. OmniPRC

Globus

- Resource management ; GRAM
  - 計算資源の割り当て・ジョブ起動実行制御
- Communication Infrastructure Globus IO
  - 様々なプロトコルをサポートする通信レイヤ
- Metacomputing Directory Service MDS
  - Grid上のGlobusで利用できる計算資源の情報提供
  - LDAPを使った情報提供
- Globus Security Interface GSI
  - 認証などのセキュリティ機構
  - X509 Ceriticate-base の認証
  - Single-Sign-ON
- Hearbeat Monitor HBM
  - システムの状況モニタ
- Remote data access GASS
  - ファイルへのリモートアクセスサービス
- executable management GEM
  - 実行ファイルの構築、転送

グリッドと並列アプリケーション

- 典型的なアプリケーション
  - パラメータ検索：同じ計算を膨大な計算資源で実行
  - master-slave型の並列プログラム
- 典型的なグリッド計算資源
  - 複数のクラスタが利用可能
  - 計算資源の状況が動的に変化



Gridの並列プログラミング

- Globus shellベースのjobのsubmit
- MPICH-G
  - Grid上のMPI (メッセージ通信プログラム)
  - 汎用だが、すべて書かなくてはならない。
  - 資源のリソースの増減に対応できない。
- Grid RPC: Ninf, NetSolve
  - 直感的でわかりやすい プログラミングインタフェース
  - 並列プログラミングは 非同期なRPC
    - activeなrequestをユーザが管理

```
Ninf_call_async(A,…)
Ninf_call_async(B,…)
…
Ninf_call_wait()
```

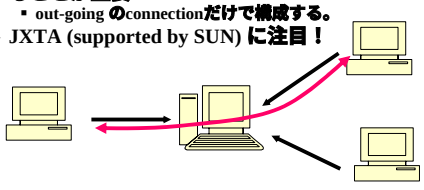
⇨ OmniRPC

OmniRPC

- グリッド並列プログラミングミドルウエアOmniRPCの開発
  - グリッドでの Master-workers 型の並列プログラムをサポート
  - Gridのend-pointの計算資源として、クラスタを対象とする
    - プライベートアドレスのクラスタについてもサポート
    - Firewall内のクラスタにもアクセス可能
  - 既存のプログラムに対し、簡単な並列プログラミング環境を提供
    - 並列プログラミングの記述、制御はOpenMPで行うことができる
  - クラスタでは “rsh”, globus環境では “GRAM” 、”ssh”環境でも使用できる。
  - 各計算資源の管理ポリシーを考慮したジョブの起動をサポート
    - サイトのスケジューラ (PBS, SGE,…)
  - 大規模なグリッド環境 (upto 1000 hosts!)のサポート
  - http://www.omni.hpcc.jp/OmniRPCで公開中
- グリッドアプリケーション
  - HMCS-G (Grid-enabled Heterogeneous Multi-computer System)
    - 広域ネットワーク環境において、複数な計算リソースである重力専用計算機GRAPE-6を共有し、汎用計算機と融合計算を行う。
    - OmniRPCを用いて、セキュリティ、認証、プログラミングインタフェースを提供
  - 細胞的分子動力学プログラムのグリッド並列化
    - グリッド上に分散している大規模なクラスタ計算資源を利用して、計算が可能
    - OmniRPCで複数のクラスタで実行、有効性を検証

P2Pコンピューティング

- P2P（pear to pear)コンピューティング
  - ファイル交換ソフト(Gnudella,…)で注目
  - 各PCが自立的につながる形態
    - 各PCは対等の関係（？）
  - 各PCは、ファイヤウオールの中でもつなぐことができることが重要
    - out-going のconnectionだけで構成する。
  - JXTA (supported by SUN) に注目！



グリッドコンピューティングの動向

- 標準化はGGF (Global Grid Forum)で議論されている。
  - IETF、W3Cのような役割
- 現在のホットな話題は　OGSA　(Open Grid Service Architecture)
  - GridにWeb Serviceのコンセプトを導入
  - Globus Toolkit 3に試験的に導入
  - コマーシャルでの適用、利用拡大を目指している
- 各メーカとも次世代のインタネット技術として注目している。
  - IBMがleading…
- 日本ではNational Research Grid Project (NaReGI)が進行中
- P2Pはグリッドコンピューティングとはちょっと方向が違うが、新しいアイデアとして注目
  - JXTA by SUN

終わりに

- 並列処理は必然の技術
  - プロセッサの高速化も本質は並列処理
- クラスタ技術
  - 並列システムが身近に
- グリッド技術
  - ネットがつながっていれば、計算資源を統合して利用できる
- では、何につかうか？
  - １０００台のPCでも使える時代に
  - イマジネーションが必要