第 1 回 クラスタゼミ

ゼミ担当者　：　宮部洋太
指導院生　：　中尾昌広
開催日　：　2009 年 6 月 26 日

ゼミ内容: PC クラスタについての基本的な知識と PC クラスタシステムの一つである Beowulf クラスタについて学ぶ．そして本研究室ある Beowulf クラスタ，Supernova を用いて実際に並列計算を行うことで理解を深める．

1　PC クラスタ

PC クラスタ[1]とは市販の PC をラックやケースなどに多数搭載し，それらをネットワーク結合したコンピュータシステムである．

PC クラスタはスーパーコンピュータに匹敵する性能を安価に実現したり，アプリケーションやデータの高可用性を実現したりするために利用されている．

1.1　クラスタの定義

クラスタリング技術の初期設計者の一人である Gregory Pfister [2]はクラスタを「単一で稼動するコンピュータの集まりで，一つの計算資源として使用可能な並列もしくは分散システムである」と定義している．

また，市場調査会社である Aberdeen Group の定義によれば，クラスタ・システムとは

- 複数ノードで構成されたコンピュータ・システム
- 全体として
  - 単一システムとして機能すること
  - 高可用性を有すること
  - クラスタ全体に対するシステム管理機能を有すること
  - クラスタ・ファイルシステム[3]を有すること
  - スケーラブルなプラットフォームをサポートすること
  - 柔軟なシステム構成を構築できること

としている．

1.2　PC クラスタの種類

一般的に PC クラスタは次の 3 種類のシステムに分類できる．

- 並列演算クラスタ（HPC : High Performance Computing）
  主に科学技術計算で利用される並列プログラミング・アプリケーションを使用するためのクラスタ．Beowulf クラスタは代表的な実例である．
- 負荷分散クラスタ（LB : Load Balancing）
  単一のアプリケーションを負荷分散して実行するためのクラスタ．ネットワーク・サービス・アプリケーションの実行に適している．
- 高可用性クラスタ（HA : High Availability）
  ミッション・クリティカル[4]なアプリケーションを実行するためのクラスタ．クラスタは冗長化構成を持ち障害発生時には正常なコンポーネントが処理を引き継ぐ．

1.3　クラスタの有用性

本研究室では多くの人が，最適化問題を解くための手法や最適化問題の計算モデルについて研究している．最適化問題は計算量が大きいため対象となるデータが大きいと，最適化問題を解くために膨大な計算を行う必要がある．そのため計算を一台の PC で行っていては多くの時間がかかる．

このような大きな計算量を必要とする問題に対しては並列計算を行うことで高速に解くことができる．しかしながら超高性能なコンピュータから構成されるスーパーコンピュータでは開発コスト，運用コストが非常に高価である．一方で PC クラスタならば同等の計算能力を非常に安く構築できる．

2　PC クラスタの構成要素と技術

2.1　ハードウェア

PC クラスタを構成するノードは一般に市販されている PC であるため，用いられているパーツも一般的なものである．具体的には以下のものが挙げられる．

- プロセッサ

[1] クラスタ（cluster）は英語で「ブドウの房または星団のような集合」を意味する．
[2] 元 IBM 社技術職最高位 Distinguished Engineer
[3] クラスタ/グリッドなどの大規模分散処理環境を対象とした分散ファイルシステム
[4] 24 時間 365 日，止まらないことを要求される基幹業務

- マザーボード
- メモリ
- ハードディスクドライブ
- Ethernet
- GPU(Graphics Processing Unit)

2.2 ネットワーク

2.2.1 インターコネクト

インターコネクトとは複数のノード間を接続するネットワークである．PC クラスタを構築する上でインターコネクトは，処理速度，コストという観点から，ノード自体に次いで重要な要素である．

PC クラスタではユーザが実行させたい処理を各ノードに分割して実行し，MPI 等を使ったノード間通信で同期や計算結果の集約などを行う．そのため，高い性能を引き出すためには広帯域かつ低遅延なインターコネクトが要求される．

現在 HPC 向けクラスタのインターコネクトとしては Gigabit Ethernet と Infiniband が主に使われており，Gigabit Ethernet が最も多い．InfiniBand の方が遅延が小さいが，コスト面等において Gigabit Ethernet の方が優れているためより利用されている．

2.2.2 外部との通信

PC クラスタは，幅広いユーザからのアクセスを許すような環境で使用されることが多い．外部から内部への接続は LAN に接続された Network Interface Card（NIC）を通して行われる．外部からの接続の様子を Fig1 に示す．外部とローカルエリア内とをつなぐノードをゲートウェイノードと呼ぶ．ゲートウェイノードは内部と外部とを結ぶために 2 つの NIC を持ち，外部に対する唯一の接触点となっている．このように外部と内部との接触を避けることで，セキュリティが高められている．

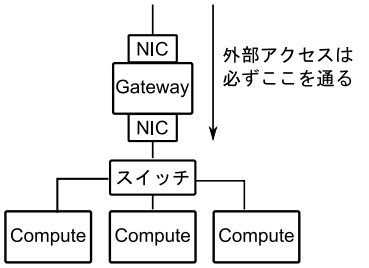

Fig. 1　外部から内部への接続の様子

3 Beowulf クラスタ

ここでは PC クラスタの代表な実装法として Beowulf クラスタを取り上げる．

3.1 Beowulf とは

Beowulf とは計算機構成法の一つであり，NASA とカルフォルニア工科大学が共同プロジェクトで開発したものである．安価な PC を大量につなげることでスーパーコンピュータ並の計算速度を実現している．

Beowulf クラスタの思想は安価で誰にでも実現可能な高性能計算である．技術情報を公開し，オープンな開発体制をとることで計算方法やソフトウェアの充実を図っている．

3.2 Beowulf システム

Beowulf システムは，複数の PC で構成され，PC を Ethernet などのネットワークで一つに接続したシステムである．Beowulf システムには PC や，NIC，スイッチングハブなどの一般的なハードウェア部品が用いられる．このため特別なハードウェアの開発費や準備期間を必要としない．

Beowulf システムには，LinuxOS や PVM（Parallel Virtual Machine），MPICH（MPI CHameleon）などの，オープンソースソフトウェアが用いられることが多い．

Beowulf は通信に伴う遅延に対して技術的な課題を抱えている．よって性能を十分に引き出すためには，並列タスクの粒度が中規模から大規模であり，通信がそれほど多く生じないようなアルゴリズムを使用する必要がある．幸い，多くの大規模問題向けのアルゴリズムがこの要求条件を満たしている．

3.3 Beowulf システムの利点

Beowulf システムには，以下の利点がある．

- 低コストで高性能が得られる

  Beowulf はコモディティハードウェア部品とオープンソースソフトウェアで構成されるため，スーパーコンピュータ並の性能が安価に引き出せる．

- 技術動向に速やかに対応できる

  Beowulf には独自のアーキテクチャがあるわけではなく，市販されている部品を使っているため，最新技術を容易に取り込むことができる．

- 拡張性に優れる

  Beowulf は一個のスイッチングハブで接続されるノードだけで構成した小規模なシステムから，何百台ものプロセッサで複雑なトポロジ[5]を組み込んだ大規模なシステムまで，幅広く対応できる．

3.4 Beowulf システムの欠点

Beowulf システムには以下の欠点がある．

[5] LAN の接続形態

- 通信性能が低い.

  Beowulf システムは PC 間の通信が低速である. なぜならば構築コストを抑えるために，専用のインターコネクトである InfiniBand 等を使用せずに Ethernet を用いるからである.

- 並列プログラムの作成が困難

  逐次に実行していくプログラムでは通信量を考慮する必要がないが，並列に実行していくプログラムでは通信量が多いと処理速度が低下するため考慮する必要がある．そのため，並列計算向けのアルゴリズムを変更する必要が生じる．このようなプログラムの作成は難しいことがある.

4 Beowulf クラスタに使われる技術

- RSH(Remote SHell)

  手元のマシン上で遠隔マシンの働きをさせるためのコマンドとして SSH や RSH がある．SSH はデータ通信の際に認証と暗号化が行われる．RSH は認証や暗号化なしにデータ通信を行えるためネットワークに負荷をかけずに通信できる.

- MPI(Message Passing Interface)

  MPI とはメッセージ通信を行うためのインタフェースの仕様である．MPI はメッセージパッシング方式[6]に基づいており MPI Forum によって標準化されている．MPI はあくまで仕様であるので，「MPI_Send 関数はデータを送信するための関数で，引数はこれ」ということが決められているだけである．実際にメッセージパッシング方式をプログラムで実装するためのライブラリが，メッセージパッシングライブラリであり，その代表例として MPICH がある.

- NFS

  Beowulf クラスタでは，多くの場合 NFS（Network File System）というファイル共有システムが利用される．NFS を利用することで，遠隔ホストにあるファイルシステム[7]をローカルにマウント[8]でき，遠隔ホストとローカル上でファイルを共有できる．NFS はクライアント/サーバモデルを使用しており，共有するディレクトリをサーバが提供し，クライアントはそのディレクトリをマウントすることでディレクトリ内のファイルにアクセスできる.

Fig. 2 を例にとって説明する．Fig2 では，NFS サーバが/home を提供し，Client1〜3 が/home にマウントしている．提供された/home は NFS サーバのものである．Client からは/home がローカルにあるファイルシステムであるかのように見える．/home へのファイルの書き込み，/home への読み込みはすべて NFS サーバの/home に対して行われる.

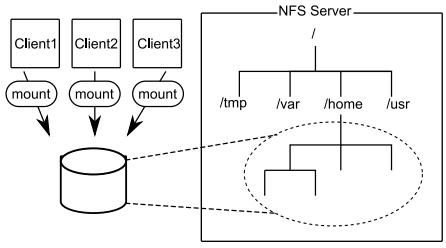

Fig. 2　NFS の流れ

5 Supernova

Supernova は Beowulf 型の PC クラスタであり，AMD 社の 64 ビット CPU(Opteron)，512 プロセッサ (現在は 192 プロセッサ) により構成されており，インターコネクトには Gigabit Ethernet が採用されている．実行性能は 1TFlops を超え，構築当時は日本一の計算速度を誇る PC クラスタであった.

Fig.3 に Supernova の簡単な構成を示す．Supernova はゲートウェイノード (snova) と Xen 用サーバと 96 台の物理ノード (nova001〜nova096) によって構成されている．snova ノードは NFS サーバとしての役割も兼ねている．物理ノードの上では Xen が動いており，2 つの仮想 OS が計算ノードとして動作している．具体的には Table 1 のような仮想 OS が動作している．IP アドレスは物理ノード自体と 2 つの仮想 OS に対して割り当てられている．つまり物理ノードごとに 3 つの IP アドレスが割り当てられている.

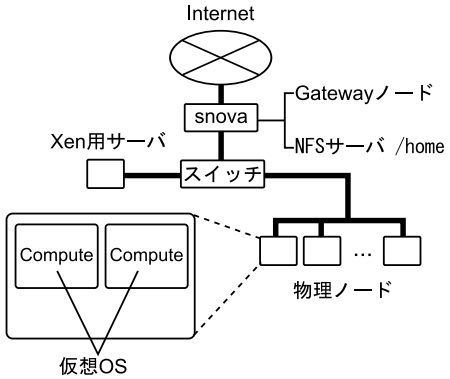

Fig. 3　Supernova

[6]プロセス間でメッセージを交信しながら並列処理を実現する方法

[7]Unix ではすべてのファイル/ディレクトリ/入出力装置は一つの木構造になっている．ドライブごとにディレクトリの木が存在し，それを張り合わせる（マウントする）ことで一つの木を構成している.

[8]UNIX のカーネルにファイルシステムの木があることを認識させること.

Table 1　Supernova 上の仮想 OS

仮想 OS	所有
isdl001～isdl036	同志社大学 知的システムデザイン研究室
ene001～ene008	同志社大学 エネルギー変換センター
pro001～pro100	名古屋大学
flo001～flo010	関西大学

6　MPICH によるの並列処理の実行

6.1　MPICH(MPI CHameleon)

MPICH とは，MPI の実装ライブラリの一つである．MPICH は，アメリカのアルゴンヌ国立研究所が模範実装として開発し，無償でソースコードを配布している．移植しやすさを重視した設計となっているため，盛んに移植が行われ，世界中のベンダの並列計算機上で利用できる．

6.2　MPICH による制御

6.2.1　mpirun

MPICH では，ファイルの実行時にオプションをつけることで，動作を制御できる．具体的には以下のように行う．

```
mpirun （オプション）　（実行プログラム）
```

6.2.2　-np

オプションとして -np （数値）を用いることで，プログラムの実行に使用するプロセスの数を設定できる．

6.2.3　machinefile

MPI プログラムでは並列プロセスをどのノードに対して実行させるかは，machinefile によって設定できる．例えば，isdl001，isdl002，isdl003 という計算ノードを利用したい場合，以下のようなファイルを用意する．

```
isdl001
isdl002
isdl003
```

また MPICH における machinefile では，1 つのノード名が複数回現れた場合，対応する複数のプロセスを同一ノードに割り当てる．

6.2.4　nolocal

nolocal は，ローカルマシンをノードとして動かさないようにするためのオプションである．つまり，各ノードにプログラムを分配するマシンではプログラムが実行されない．

6.3　並列処理の流れ

ユーザの PC からゲートウェイノードに接続し，計算ノードに並列処理を実行させるまでの流れを Fig. 4 に示す．

1. SSH(Secure SHell) でマスタノードに接続する．
2. RSH(Remote SHell) を使用して計算ノードにログインする．
3. 計算ノードでコンパイルを行い，実行ファイルを生成する．
4. MPI プログラムを実行し並列処理を行う．

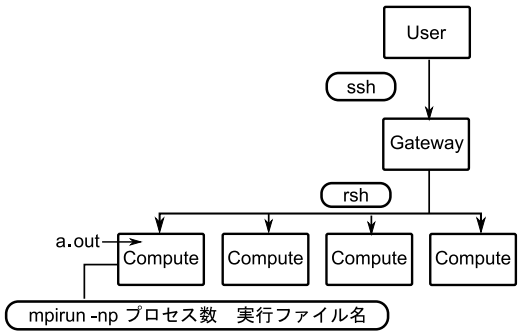

Fig. 4　並列処理の流れ

6.4　具体例

MPICH を用いて C 言語で書かれた MPI プログラムを実行する手順を示していく．

以下のようにコマンドを入力し，コンパイルする．

```
mpicc プログラム名.c -o 実行ファイル名
```

利用するマシンを設定するため，vi や emacs といったエディタで，利用するマシン名を一行ずつ縦に入力し，保存する．

```
isdl001
isdl002
   :
```

並列計算を実行する．この例では-np の後ろに指定されているプロセス数で，machinefile で指定したマシンでプログラムが実行される．

```
mpirun -np (プロセス数) -machinefile (設定した machinefile 名) (実行ファイル名)
```

参考文献

1) 日本 HP - HP Serviceguard for Linux ProLiant クラスター
.http://h50146.www5.hp.com/products/servers/proliant/svglinux/
qa.html

2) ADVenture Cluster
.http://www.clubsse.com/advc/information/
mpiinit/mpiinit.html

3) ADVenture Cluster
.http://www.clubsse.com/advc/information/
mpich/debug.html

4) Gregor User's Manual
.http://202.23.147.51/gregor-mpi.htm

5) MPI による並列プログラミングの基礎
.http://mikilab.doshisha.ac.jp/dia/smpp/
cluster2000/PDF/chapter02.pdf

6) トーマス・L・スターリング他.PC クラスタ構築法 − Linux によるベオウルフ・システム− , 産業図書株式会社,2001.

7) 日本 HP パートナーアライアンス「HP and Oracle : 製品・ソリューション情報:Oracle9i Real Application Clusters (RAC)」
.http://h50146.www5.hp.com/partners/alliance/oracle/prodserv/rac/

8) LIAISON DOSHISHA UNIVERSITY LIAISON OFFICE
NEW LETTER 2004.Vol.6
.http://liaison.doshisha.ac.jp/doc/research/newsletter/files/111.pdf

9) IT 用語辞典 e-Words
.http://e-words.jp/

10) MPI による並列プログラミングの基礎
.http://mikilab.doshisha.ac.jp/dia/smpp/cluster2000

11) 並列処理入門
.http://mikilab.doshisha.c.jp/dia/smpp/cluster2000

12) 中村敦司 他. 新 The UNIX Super Text【上】, 技術評論社, 2003 年.

13) 久野靖. UNIX による計算機科学入門, 丸善株式会社, 2004.