7 章 クラスタコンピュータ

（執筆者：合田憲人）［2010 年 5 月 受領］

■概要■

クラスタコンピュータは，複数の計算機をネットワークで接続することにより構成される分散コンピューティングシステムである．クラスタコンピュータを構成する計算機は PC やワークステーションからスーパーコンピュータまで多岐にわたり，高性能計算機システムの構成方法として，現在最も多く採用されている方式である．本章では，クラスタコンピュータを構成するアーキテクチャやクラスタコンピュータ上で利用されるソフトウェア，プログラミング技術について説明する．

【本章の構成】

7-1 節では，クラスタコンピュータの代表例である PC クラスタを構成する計算ノードとネットワークのアーキテクチャについて説明する．クラスタコンピュータ上では，クラスタを構成する複数の計算ノードや計算ノード内の複数の CPU コアを利用した並列計算が行われる．7-2 節では，クラスタコンピュータ上での並列プログラミング手法として普及している MPI や OpenMP について説明する．クラスタコンピュータのシステムソフトウェアでは，複数の独立した計算機を一つの計算機としてユーザに見せるための技術が必要とされる．7-3 節では，これを実現するためのシステムソフトウェアを紹介する．現在のスーパーコンピュータのほとんど全ては，クラスタコンピュータと言ってよい．最後に 7-4 節では，多数のスカラ型プロセッサを高性能ネットワークにより接続することにより構成されるスカラ並列スーパーコンピュータを紹介する．

■6 群 － 5 編 － 7 章

7-1 クラスタの構成方法

（執筆者：鯉渕道紘）［2009 年 12 月 受領］

PC クラスタのハードウェアは通常，特注品ではなく，パーソナルコンピュータ（PC）に代表される大衆市場で扱われている商用の既製品（Commercial Off-The-Shelf : COTS）を組み合わせることで構築する. PC クラスタの構成要素である計算ノードとネットワークについて，順に説明する.

7-1-1 計算ノード

PC クラスタの計算ノードは，通常，汎用のパーソナルコンピュータ，サーバである．ただし，PC クラスタのなかには，特定の演算処理を高速に行うことができる複数のプロセッサを用いたアクセラレータボードを PCI Express などのインタフェースを介してホストに挿入して計算能力を増強しているものもある．この場合，構成がヘテロジニアスになるため，アクセラレータボードの性能を引き出すためには，並列プログラムを工夫することが必要となる．アクセラレータボードの例としては，ソニー，ソニー・コンピュータエンタテインメント，東芝，IBM によって開発された Cell ブロードバンド・エンジン，ClearSpeed Advance アクセラレータカードなどがあげられる．

PC クラスタは PC やサーバを多数並べることで構成するため，設置スペースが問題となる．そこで，一般の家庭で使用されるデスクトップ PC とは異なり，PC クラスタでは，**図 7･1** のように，1 台のラックに多数のラックマウント型 PC を収めることが多い．このラックは，幅が 19 インチ，高さが 1 U 単位の EIA（米国電子工業会）によって標準化されており，現在ではほとんどのサーバラックがこの規格に準拠している．

図 7･1　PC クラスタ

更に，多数の計算ノードを高密度に設置するために，ブレードサーバが開発されている．1 台のラックマウント型 PC，サーバは，1 台の PC の構成要素すべてを含むが，ブレードサーバは外部インタフェース，冷却装置などを各ブレードマシンで共有する. ブレードサーバは，

ラックの 1 U 当たり，複数のブレードを装着することが可能であるため，スペース，消費電力の両面で優れている．いずれの計算ノードも，遠隔から電力制御をするために独立したユニットをもち，故障などに対処できるようになっている場合が多い．

7-1-2 ネットワーク

PC クラスタでは，通常，並列計算に使うメインネットワークのほかに，システム管理ために独立した複数のネットワークをもつが，ここでは前者について詳しく述べる．

PC クラスタで使用されるメインネットワークは，ローカルエリアネットワーク（LAN）をそのまま流用したものと，システムエリアネットワーク（SAN）に大きく分けられる．前者の代表は広域網としても使われているイーサネット，後者の例は InfiniBand，Myrinet があげられる．

(1) イーサネット

ローカルエリアネットワーク，広域ネットワークで幅広く使用されているイーサネットは PC クラスタのネットワークとしても使用することができる．

通信プロトコルに TCP/IP，Linux などの汎用の OS を用いた PC クラスタはベオウルフ（Beowulf）クラスタと呼ばれ，並列分散環境の構築に専門の知識をあまり要さないため，広く普及している．ただし，並列計算機のネットワークが，3 次元格子状にルータスイッチを配置するトーラス（torus）などの高性能なトポロジーを採用している一方，イーサネットは，基本的にループ構造を含むトポロジーを許していないためツリートポロジーを採用している．ツリートポロジーは，トラフィックがツリーのルート付近に偏りやすい問題を解決するために，リンク集約化（IEEE 802.3ad）を用いてルート付近のリンクを強化するのが一般的である．

ただし，VLAN 技術を応用することで，具体的には**図 7･2** のように異なるツリートポロジーの VLAN を組み合わせることで任意のトポロジーと同一スイッチ間に複数経路を実現することができる [1]．そのため，PC クラスタ構築において，イーサネットがツリートポロジーに限定される問題は解決されつつある．

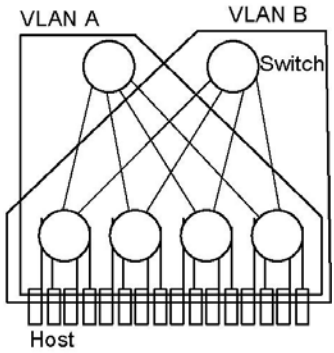

図 7･2　イーサネットにおける VLAN を組み合わせたトポロジー例

(2) システムエリアネットワーク（SAN）

イーサネットは，一般的にバンド幅，遅延の点で並列計算機のネットワークに比べて劣っていることが多い．そのため，ベオウルフクラスタの性能を専門の並列計算機に匹敵するまでに向上させるためには，ネットワークの増強が必要となる．

そこで，専用の並列計算機のネットワークと同等のトポロジー，ルーティング，スイッチング技術を採用したネットワークであるシステムエリアネットワーク（SAN）が開発されている．例えば，InfiniBand ではツリーを多重化したトポロジーである Fat ツリーなどの様々なトポロジー，ルーティングアルゴリズムを採用することができる．Myrinet，InfiniBand などの複数のシステムエリアネットワークが広く普及しているが，これらはルーティングの実装やフロー制御などのアーキテクチャでは各々異なる．システムエリアネットワークは同一の高速なスイッチ群と大容量 point-to-point リンクを用いて構築され，専用の通信ライブラリにより，高バンド幅，低レイテンシを実現する．

また，システムエリアネットワークではトポロジー，リンク長に対する制限が並列計算機の専用ネットワークに比べて緩い．これは，PC クラスタではマシンルーム内に集中配線する並列計算機と異なり，より広い範囲に分散している計算ノードを接続することがあるため，結合網の物理的配置にある程度の自由度が必要となることに起因する．

■参考文献

1) 工藤知宏, 松田元彦, 手塚宏史, 児玉祐悦, 建部修見, 関口智嗣, “VLAN を用いた複数パスを持つクラスタ向き L2 Ethernet ネットワーク,” 情報処理学会論文誌コンピューティングシステム, SIG 6 (ACS6) (45), pp.35-43, May 2004.

■6 群 － 5 編 － 7 章

7-2 クラスタ上でのプログラミング

（執筆者：高橋大介）［2009 年 4 月 受領］

クラスタ上でのプログラミングを行う方法としては，

1. MPI（Message-Passing Interface，メッセージパッシングインタフェース）
2. OpenMP
3. RPC（Remote Procedure Call，リモートプロシージャコール）

などがあげられる．

MPI では明示的にメッセージの交換をプログラム中に記述することにより並列化を行うのに対して，OpenMP では指示文（ディレクティブ）をプログラム中に挿入することによって並列化を行う．

以下，クラスタ上でのプログラミングで使われることが多い MPI と OpenMP について説明する．

7-2-1 MPI

MPIは並列プログラミングの規格として最も広く使われている．MPIは新しいプログラミング言語ではなく，CまたはFortranから呼び出すサブプログラムのライブラリである．MPIは，業界団体や研究者らのメンバからなるMessage Passing Interface（MPI）Forum [1] によって規格が明確に定義されている．MPIの実装としては，MPICHやOpen MPI，MVAPICHなどが知られている．

(1) MPI の関数

MPI では 100 以上の関数が定義されている．大きく分けて以下の関数がある．

・1 対 1 通信関数
・派生データ型と MPI Pack/Unpack
・集団通信関数
・グループ，コンテキスト，コミュニケータ
・プロセストポロジー
・環境管理

よほど特殊な並列化を行わない限り，20 個程度の関数を知っていれば十分であり，そのなかでも特によく使用するのは 10 個程度である．

(2) コミュニケータ

コミュニケータ（communicator）は MPI においてお互いにメッセージを送ることができるプロセスの集団である．

よほど特殊な並列化を行わない限り，MPI COMM WORLD（全プロセスを含む初期コミュニケータ）を使えば十分であるが，必要であれば，別のコミュニケータを作成することも可能である．

[1] http://www.mpi-forum.org/

(3) 1 対 1 通信

1 対 1 通信関数の例としては，以下の関数がある．

・ブロッキング通信（MPI Send，MPI Recv）：一度呼び出すと，送受信が正常に完了するまで次の処理に進めない．

・非ブロッキング通信（MPI ISend，MPI IRecv）：通信と演算をオーバーラップさせることが可能．

・双方向通信（MPI ISendrecv）：安全に（デッドロックを起こさずに）双方向通信を行える．

(4) 集団通信

コミュニケータの中のすべてのプロセスを含む通信パターンを集団通信（collective communication）と呼ぶ．集団通信は通常，二つ以上のプロセスを含む．集団通信関数の例としては，以下の関数がある．

・ブロードキャスト（MPI Bcast）

・リダクション（MPI Reduce，MPI Allreduce）

・ギャザ（MPI Gather，MPI Allgather）

・スキャッタ（MPI Scatter，MPI Allscatter）

・全対全通信（MPI Alltoall）

(5) MPI を用いた並列プログラムの構成例

MPI を用いた並列プログラムの構成例は以下のようになる．

```
#include "mpi.h"
#include <stdio.h>
#define N 1000

int main(int argc, char *argv[])
{
  int myid, nprocs, sendbuf[N], recvbuf[N];
  MPI Status status;

  MPI Init(&argc, &argv);
  MPI Comm size(MPI COMM WORLD, &nprocs);
  MPI Comm rank(MPI COMM WORLD, &myid);
...
  MPI Send(sendbuf, N, MPI INTEGER, (myid + 1) % nprocs,
           0, MPI COMM WORLD);
  MPI Recv(recvbuf, N, MPI INTEGER, (myid + 1) % nprocs,
           0, MPI COMM WORLD, &status);
...
  MPI Finalize();
  return 0;
}
```

7-2-2 OpenMP

OpenMPは，マルチスレッド並列プログラミングのためのAPI（Application Programming Interface）である．OpenMPは 1997 年に発表された業界標準規格であり，多くのハードウェア及びソフトウェアベンダが参加する非営利団体OpenMP Architecture Review Board[*2] によって管理されている．

(1) OpenMP の特徴

OpenMP API はユーザ指示の並列化のみを対象にしている．つまり，指示文（ディレクティブ）を挿入することにより並列化を行う．指示文は OpenMP をサポートしないコンパイラでは，単にコメント行として無視される．その場合，逐次計算プログラムとしての動作が保証される．したがって，OpenMP では，逐次計算プログラムに対して指示文を挿入するという作業により，段階的に並列化を行うことが可能になる．

OpenMP は現在 Fortran と C/C++について標準化が行われている．

(2) 指示文の形式

C/C++の OpenMP 指示文は，以下のように pragma プリプロセッサ指示文で指定する．

#pragma omp *directive-name* [*clause*[[,] *clause*]

それぞれの指示文は，#pragma omp で始まる．

(3) OpenMP 構文

OpenMP では，大きく分けて以下の構文がある．

- parallel 構文
- ワークシェアリング構文
- ループ構文
- sections 構文
- single 構文
- 複合パラレル・ワークシェアリング構文
- マスター・同期構文

(4) OpenMP を用いた並列プログラムの構成例

OpenMP を用いた並列プログラムの構成例は以下のようになる．

*2 http://www.openmp.org/

```
#include <stdio.h>

int main(void)
{
...
#pragma omp parallel
  {
  ... 並列化される部分
  }

  return 0;
}
```

■参考文献

1) Peter S. Pacheco, “Parallel Programming with MPI,” Morgan Kaufmann Publishers, 1997.
2) 牛島　省, “OpenMP による並列プログラミングと数値計算法,” 丸善, 2006.

■6 群 － 5 編 － 7 章

7-3 クラスタ用基盤ソフトウェア

（執筆者：住元真司）［2008 年 10 月 受領］

クラスタ用基盤ソフトウェアとは，複数の計算サーバをあたかもひとつの高性能計算機として構成するために使われるソフトウェア群のことである．これらのソフトウェアは元々ベンダ製スーパーコンピュータが全盛であった 1990 年代には，スーパーコンピュータの提供ベンダから独自のソフトウェアがバイナリーで提供されていた．しかし，1990 年代後半に PC クラスタ上でオープンソースのクラスタ基盤ソフトウェアが開発された結果，PC クラスタは爆発的に普及することになった．現在ではオープンソースのクラスタ用基盤ソフトウェアを用いることによりハードウェアさえあれば高性能クラスタ計算機が個人でも構築可能になっている．クラスタ用基盤ソフトウェアについても，1990 年代は独自性を意識したソフトウェアが多く存在したが，現在では、相互利用性を重視したものが主流となってきている．

クラスタ用基盤ソフトウェアは，おおまかにいくつかの階層に分類することができ，ハードウェアに近いところから，InfiniBand，Myrinet，Ethernet といったハードウェア上で高い通信性能を実現する低レベル通信ライブラリ，複数のサーバ上でプログラムを実行可能にする並列プロセス実行環境，MPI，PVM などユーザプログラムの並列化をサポートする高レベル通信ライブラリ，そして、複数ユーザのプログラム実行要求を制御するジョブ制御システムとクラスタシステムの運用管理に役立つクラスタ監視システムなどのコンポーネントから構成される．なお，これらを総合的に扱うソフトウェアをクラスタオペレーティングシステム（クラスタ OS）とも呼ぶ．オープンソースで公開されているクラスタ基盤ソフトウェアはこれらの階層のうちの複数の階層をサポートしているものが多く，オープンソース間で階層間の相互互換性，利用性があるものも多い．

本節では，現在使われているクラスタ用基盤ソフトウェアを概観する．クラスタ基盤ソフトウェアには優れたものがたくさん存在するが，ページ数の都合上，大規模システムで利用実績のあるオープンソースのクラスタ基盤ソフトウェアをベースにその基盤技術について概観する．

7-3-1 MPI ランタイムシステム：MPICH 2，OpenMPI

MPICH 2 と OpenMPI は MPI フォーラムで標準化されている MPI（Message Passing Interface）規格に基づく MPI 高レベル通信ライブラリである．これらのランタイムシステムには，それだけで MPI プログラム実行が可能なように，並列プロセス実行環境とソケットを用いた低レベル通信ライブラリを含んでいる．クラスタ専用のネットワーク（InfiniBand，Myrinet など）は，それぞれのネットワーク用の低レベル通信ライブラリを導入することで利用可能になる．

MPI 高レベル通信ライブラリでの技術的な課題は，いかに大規模なクラスタシステムで安定して動作し，かつ，高い通信性能を実現するかである．高い通信性能を実現するためには，複数の計算サーバ間でより短い通信遅延と高い通信バンド幅を実現することが課題である．低遅延通信と高バンド幅通信の実現にはそれぞれのハードウェア特性に合わせた通信方式を選択することが重要である．これを，それぞれの MPI 高レベル通信ライブラリは独自の工夫により実現しているほか，全対全通信やリダクション演算処理などの集合通信と呼ばれる通

信機能の実現には，通信性能に加えクラスタのネットワーク構成に応じた通信アルゴリズムの設計と実装が重要になる．

7-3-2 バッチスケジューラ：PBS，OpenPBS，Torque

7-3-1 項で述べた MPI ランタイムシステムの利用により，クラスタシステムにおいて並列プログラムを実行する環境は構築可能である．しかし，構築したクラスタシステムを複数のユーザで共用利用するためには，クラスタシステムという計算資源を複数のユーザでうまく融通するための機能が必要になる．このための機能としては，従来の汎用機から使われてきたバッチシステムと Unix 系オペレーティングシステムで採用されている時分割で共有する TSS（Time Sharing System）がある．クラスタシステムにおいては前者のバッチシステムを用いてシステム運用がされる場合が多く，そのためのオープンソースソフトウェアも開発され利用されている．

現在，多く利用されているのは，1990 年代に NASA がオープンソースで公開した PBS（Portable Batch System）ベースのバッチシステムである．PBS は，クラスタを構成する計算サーバのリソース管理とジョブのスケジューラ機能をもっている．ジョブスケジューラはプラグイン形式でスケジューリングポリシーを変更できる機能を搭載している．このプラグインの機能を利用して Maui スケジューラなどいくつかのスケジューラが開発されている．

PBS 自体はその後 OpenPBS と名前を変更して開発が続けられたが，2002 年にオープンソースとしては開発が終了し，それ以降は PBS をベースにたくさんの改良が加えられた Torque と呼ばれるオープンソースのバッチスケジューラが開発され利用されている．

バッチスケジューラの課題は，大規模なクラスタシステムでの安定稼働のほか，いかに様々なユーザジョブの要求を満たしシステムの利用効率を高めることができるかにある．

7-3-3 管理ツール：Ganglia

大規模なクラスタシステムになると，たくさんの計算サーバのそれぞれの状態の把握，ならびに，全体のシステムの利用状況を把握することが困難になってくる．このようなクラスタシステムの運用状況を把握するためにクラスタ監視システムがある．

オープンソースのクラスタ監視システムとして Ganglia がある．Ganglia は各計算サーバから計算機資源の利用状況を獲得しサーバに収集する機能をもつほか，これを Web 経由でグラフィカルに表示する機能を搭載している．

7-3-4 InfiniBand 向けオープンソースパッケージ集：OFED

OFED（Open Fabrics Enterprise Distribution）は InfiniBand を用いたクラスタ向けに低レベル通信ライブラリ，並列プロセス実行環境，高レベル通信ライブラリをまとめて配布しているパッケージである．最近では一部の 10 G Ethernet ネットワークインタフェースに対応した低レベル通信ライブラリも含まれている．高レベル通信ライブラリとしては，7-3-1 項で紹介した OpenMPI のほか，MPICH 2 から派生して開発された MVAPICH という MPI 高レベル通信ライブラリが含まれている．また，DAPL と呼ばれる低レベル通信ライブラリも含まれており，一部のベンダ製 MPI 高レベル通信ライブラリにも対応が可能になっている．

7-3-5 クラスタ高性能 OS：SCore

SCore は経済産業省（当時の通商産業省）が実施した RWC プロジェクト（1992～2002 年）によって開発されたオープンソースの高性能クラスタ OS である．SCore は 7-3 節で説明した低レベル通信ライブラリ，並列プロセス実行環境，高レベル通信ライブラリ，ジョブ制御システムの機能を備えており，これらを一つのパッケージとして配布している．

SCore の特徴は，当初よりクラスタシステムにおいてあるべき環境とは何かということを考えた結果，高いアプリケーション性能を実現するために低レベルの高性能通信ライブラリ PM v2 をベースに，PM v2 上でのオブジェクト指向並列処理言語 MPC++、MPC++で書かれた Score-D グローバル OS で構成された一体構造のシステムになっていることである．

SCore はネットワークが違っても，プロセッサや OS が違っても意識なく使えるというシームレスな実行環境を目指した結果，PM v2 は同じ API で Ethernet，Myrinet，InfiniBand，共有メモリ通信をサポートしている．このため PM v2 上で動作する MPC++と SCore-D グローバル OS はネットワークを意識しないで動作することが可能になっている．1990 年代は様々プロセッサ，OS が普及していたため，SPARC や Alpha プロセッサ向けの実装もサポートされていた．RWC プロジェクト終了後，開発活動を PC クラスタコンソーシアムに移した後は，PC クラスタ向けの開発にシフトしている．

SCore における低レベル通信ライブラリは PM v2 高性能通信ライブラリにおいて提供され，異種ネットワーク上での高性能通信を実現しているほか，通信ライブラリ自体でギャングスケジューラ，チェックポイント，ジョブマイグレーションのサポート API をもっている．これを用いた並列プロセス実行環境である SCore-D と連携してギャングスケジューラ，チェックポイントリスタート機能，ジョブマイグレーション機能を実現している．高レベル通信ライブラリとしては，MPICH ならびに MPICH 2 をサポートしているほか，クラスタ対応の並列実行環境として Omni OpenMP も含まれている．ジョブ制御システム機能は，PBS, OpenPBS, Sun Grid Engine のほか，SCore-D はギャングスケジューラを用いたクラスタシステム上での TSS 実行環境を実現している．

クラスタ基盤ソフトウェアの研究開発は，今後もより大規模で高性能なクラスタ上で，より使いやすく容易に高いアプリケーション性能を実現するために続けられ，開発された様々なコンポーネントは，複数のコンポーネントでの相互利用が可能になっていくであろう．これは，クラスタ基盤ソフトウェアを利用するユーザのみならず，コンポーネントを開発する研究者においても有用な基盤となり，更に高度な基盤ソフトウェア開発を可能としている．

■参考文献

1) InfiniBand: http://www.infinibandta.org/
2) Myrinet: http://www.myri.com/
3) MPI: http://www.mpi-forum.org/
4) MPICH2: http://www.mcs.anl.gov/research/projects/mpich2/
5) OpenMPI: http://www.open-mpi.org/
6) Torque: http://www.clusterresources.com/pages/products/torque-resource-manager.php
7) Gnaglia: http://ganglia.info/
8) OFED: http://www.openfabrics.org/

9) SCore: http://www.pccluster.org/
10) 石川　裕, 佐藤三久, 堀　敦史, 住元真司, 原田　浩, 長谷川篤史, 清水正明, 亀山豊久, “Linux で並列処理をしよう(第 2 版),” 共立出版, 2007.

■6 群 － 5 編 － 7 章

7-4 スカラ並列スーパーコンピュータ

（執筆者：安里　彰）［2009 年 12 月 受領］

7-4-1 スカラ並列スーパーコンピュータの発展

高い計算性能を実現するため，多数のコンピュータを並列に動作させるというアプローチの初期のものとして Cosmic Cube [1] があげられる．Cosmic Cube は，5 MHz クロックの Intel 8086 CPU と 8087 FPU を計算ノードとして使用し，64 個の計算ノードを約 2 Mbit/s の専用のネットワークで 6 次元のハイパキューブ接続した構成であった. このシステムは 1985 年に論文発表されたが，初期のシステムは 1983 年から稼働していた.

このように商用のマイクロプロセッサを計算ノードに使用し，計算ノード間を高性能のネットワークで接続するシステムは，比較的安価に高性能を実現することが可能であり，これ以降，スカラ並列型のスーパーコンピュータの開発が進められ，Cosmic Cube に引き続いて，1980 年代後半には，nCUBE や Intel iPSC などのスカラ並列型のスーパーコンピュータが商用化された.

そして，1992 年には Intel i860 RISC プロセッサを計算ノードとして使用し，2048 ノードを 2 次元メッシュ接続した Intel Paragon システムが稼働している.

米国エネルギー省は ASCI（Accelerated Strategic Computing Initiative；現在は，Advanced Simulation and Computing（ASC）と改称）プロジェクトにおいて世界トップレベルのスーパーコンピュータの開発を推進してきており, 1996 年に納入された ASCI Red システムは 200 MHz クロックの Pentium Pro プロセッサを計算ノードとし，4510 ノードを 38×32×2 の 3 次元メッシュ接続し，史上初めて LINPACK（連立一次方程式を解くベンチマークプログラム）で 1 TFlops を達成した.

引き続き，250 MHz クロックの MIPS R1000 プロセッサを計算ノードに使用し，6144 ノードを HIPPI（High Performance Parallel Interface）ネットワークで接続した ASCI Blue Mountain（1998 年），332 MHz クロックの PowerPC 604e を計算ノードとし，5808 ノードを SP Switch と呼ぶ多段クロスバスイッチで接続した ASCI Blue Pacific システム（1999 年），375 MHz クロックの POWER 3 プロセッサを計算ノードに使用し，8192 ノードを SP Switch で接続した ASCI White システム（2000 年），1.25 GHz Alpha プロセッサを計算ノードに使用し，8192 ノードを Quadrix の多段クロスバスイッチで接続した ASCI Q システムなどが開発され，スカラ並列スーパーコンピュータは Top 500 ランキング [2] の上位を占め続けた.

7-4-2 大規模スカラ並列スーパーコンピュータ

大規模スカラ並列スーパコンピュータシステムに用いられているプロセッサとしては，汎用サーバと同じ商用マイクロプロセッサ用いるシステムと，スーパーコンピュータ向きに改良を加えた専用プロセッサを用いるシステムがある．前者は，プロセッサを開発する必要がない点と，大量生産でありコストが安いというメリットがある．一方，後者は，プロセッサを最適化して性能を上げたり，消費電力を減らしたりすることができる点が有利である.

ネットワーク（インターコネクトとも呼ぶ）についても，10 Gbit Ethernet や InfiniBand といった汎用ネットワークを用いるシステムと，専用のネットワークやスイッチを独自開発す

るアプローチがある. 汎用ネットワークでは, 小規模, 低価格のシステムでは 10 Gbit Ethernet が用いられるが，高性能の大規模システムでは機能，性能に優れる InfiniBand が採用されるケースが多い．Ethernet ではプロトコル上ループを作ることができないので，ツリー状のトポロジーが用いられ，InfiniBand では，多段のクロスバスイッチで FBB（Full Bisection Bandwidth）網や，上位のスイッチ間の接続本数を絞った Fat Tree 接続が一般的である．

しかし，これらの汎用ネットワークでは InfiniBand の場合でも，現実的には 4000 ノード程度の接続が上限であり，更に大きなシステムを構成する場合は，計算ノードとネットワークのスイッチを一体化した直接網として，3 次元メッシュや 3 次元トーラスなどの構造が用いられる．

専用ネットワークの例としては，Cray の XT(-3/4/5) システムでは SeaStar と呼ぶ専用のスイッチ LSI を開発し，このスイッチ経由で 6 方向の隣接ノードに接続している．また，IBM の BlueGene は 6 方向の隣接ノードに接続するスイッチをプロセッサチップに内蔵し, コンパクトな計算ノードを実現している．

汎用マイクロプロセッサと汎用ネットワークを用いる典型的な大規模クラスタシステムとして，テキサス大学に設置された Ranger システムがある．このシステムは，2.3 GHz クロックの 4 コア Opteron プロセッサを計算ノードに使用し，4 個のプロセッサチップを HyperTransport で接続した 16 コアの計算ノードを InfiniBand で接続したシステムである．

また，2009 年 6 月の Top 500 ランキングで 1 位となった Roadruner システム [3] は，1.8 GHz クロックの 2 コア Opteron プロセッサを 2 個搭載するブレードと，当初，Playstation 3 向けに開発された CELL プロセッサに倍精度浮動小数点演算性能と接続メモリ量を増加する改良を加えた PowerXCell 8i プロセッサ（3.2 GHz クロック 8 計算コア）を 2 個搭載するブレード 2 枚を組とした TriBlade を計算ノードとし，3240 ノードを InfiniBand で接続している．

同ランキングで 2 位の Jaguar システムは, 2.4 GHz クロックの 4 コア Opteron プロセッサ 2 個を計算ノードとし，150152 コアを SeaStar2+と呼ぶ専用のスイッチを経由して 3 次元トーラス接続したシステムである．

同 3 位の JUGENE システムは IBM BlueGene [4] の 2 世代目の BlueGene/P システムであり，850 MHz クロックの PowerPC 450 プロセッサに倍精度浮動小数点演算能力を強化し，計算ノード間を接続するスイッチなどの機能を集積した専用プロセッサチップを使用している．計算ノードはこの専用チップ 1 個と DRAM メモリからなる小規模の計算ノードであり, 72 K（73728）計算ノードを 3 次元メッシュで接続している．スイッチ機能をプロセッサ LSI に一体化した高機能 LSI の開発と，クロック周波数を低めにした低消費電力設計により高密度実装を行っている．

本来は，グラフィックス処理用に開発された GPU（Graphics Processing Unit）は，グラフィックス処理の多様化から，汎用的な計算処理が可能な GPGPU（Genral Purpose GPU）に進化してきた．グラフィックス処理で中心となる単精度浮動小数点演算であるが，1 TFlops を超える演算性能をもつ LSI が市販されている. この GPU を計算アクセラレータとして使用するという試みが行われており, 東工大の TSUBAME 1.2 システムでは Opteron 16 コアの Sun X4600 計算ノードに nVIDIA の GPU を使用する Tesla S1070 を接続している. システム全体では 680 GPU ボードを増設し, 単精度浮動小数点演算ピーク性能では約 700 TFlops の能力を追加している．

この S1070 システムは倍精度浮動小数点演算では単精度の 1/12 の性能であるが，GPU を科学技術計算に使用する傾向に鑑み，倍精度演算の能力を強化する方向であるので，今後，このような計算アクセラレータを装備するクラスタが増加すると予想される．

■参考文献

1) Charles L. Seitz, “The cosmic cube,” Communications of the ACM, vol.28, Issue 1, pp22-33, Jan. 1985.
2) Top500 Supercomputing Sites, http://www.top500.org/
3) Kevin J. Barker et al., “Entering the petaflop era: The architecture and performance of Roadrunner,” High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008, 15-21, pp.1-11, Nov. 2008,
4) A. Gara et al., “Overview of the Blue Gene/L system architecture,” IBM J. RES. & DEV. vol.49, no.2/3, pp.195-212, Mar./May 2005.