第 3 回 並列ゼミ

ゼミ担当者　：　輪湖　純也，澤田　淳二，降幡　建太郎
指導院生　　：　児玉　憲造，下坂　久司
開催日　　：　2002 年 5 月 13 日

ゼミ内容: 本ゼミでは，グローバルコンピューティングとはどのようなものであるかについて学ぶ．次に，グローバルコンピューティングを実現するためのインフラとなる Grid について，Grid とは何か，Grid を使って何ができるかを説明する．その後，グローバルコンピューティングシステムを構成している階層モデルについて説明する．

1　はじめに

本ゼミでは，グローバルコンピューティングとはどのようなものかについて説明する．そして，グローバルコンピューティングを実現するためのインフラとなる Grid について説明し，Grid を用いて何ができるか，従来のクラスタとは何が違うのか，Grid の問題点，グローバルコンピューティングシステムを構成する階層モデルについて説明する．

2　グローバルコンピューティングとは

グローバルコンピューティングとは，世界各地に存在している計算資源を広域ネットワークで結びつけて並列分散計算を行う環境を構築するものである．

こうすることで，今まででは解くことができなかった大規模な計算パワーを必要とする問題でも解くことが可能となる．そのような問題としては，次のようなものがある．

- 遺伝子解析
- 流体の計算
- 飛行機の翼の形の計算
- 材料設計
- 新薬の開発

このような試みが可能となった背景には，ネットワーク技術の進歩，コンピュータの「所有」から「利用」への考え方の変化がある．

ネットワーク技術の進歩により，LAN のみならず WAN においても高速通信が可能になり，LAN を用いた Cluster システムから WAN Cluster の実現が可能になってきた．

また,「所有」から「利用」へという流れにより，企業は数十億のスーパコンピュータへ投資をすることから HPC（High Performance Computing）技術のアウトソーシングを利用するということが今後予想される．つまり，ASP（Application Service Provider）の HPC 版に近いイメージである．

グローバルには，地球規模的という意味のグローバルだけでなく，家庭から学校，職場といった社会的機能の拡大や，パソコン，PDA，スーパコンピュータといった情報機器的拡大といった様々なグローバリゼーションという意味も含まれている．

また，コンピューティングは, 計算能力や直接関連したソフトウェア技術だけではなく，データベースや観測データへのアクセス，CG などによる可視化などを含めて，エンドユーザへのインターフェースなども対象にしている．

3　Grid とは

Grid とは，コンピュータネットワークを電力網にたとえたものである．通常，我々はスイッチを入れれば電灯が点くということがわかっていればよいのであり，その裏側にある電気がどこの発電所で作られ，どのように運ばれてきたか，ということは意識しない．一方，現在のコンピューティングはどうであるかというと，サービスが提供されている場所を自分で指定する必要があり，自分がどのようなサービスを受けたいか，ということを記述するだけではサービスを受けることはできない．そこで，そういったことを意識せずに，自分が受けたいサービスを記述するだけで，そのサービスが受けられるようにする，というのが，Grid の考え方である．

Grid では，計算資源だけでなく，あらゆる情報資源を対象にしている．たとえば，大規模データベースの利用やネットワーク上に仮想組織を構築するといったことも可能となる．

Grid は現在，Global Grid Forum と呼ばれる団体により，様々な技術の標準化が進められている．

4　Grid で何ができるのか

Grid で何ができるのかを考える前に，Grid で何がしたいのかを考える．**Virtual Organization（仮想組織）**という考え方がある．これは，世界中のさまざまな分野の研究者が，ネットワークを経由することにより，国を

超えてダイナミックに協調作業を行おうというものである．Grid はこの Virtual Organization の考え方をベースに実際何ができるのか追求するものである．ここでは，Grid 技術を次の 5 つに分類することにする（Fig. 1 参照）．

- 手元にある計算機でユーザインタフェースを提供し，遠隔の計算機を利用するシステム
- Grid 上で共同作業を支援するシステム
- 各 PC 同士の結びつきから，何らかのコラボレーションを行うシステム
- 遠隔の機器を制御したり，機器から出力されたデータを迅速に処理するシステム
- 1 つの所では格納できないような大量のデータや分散配置されたデータを処理するシステム

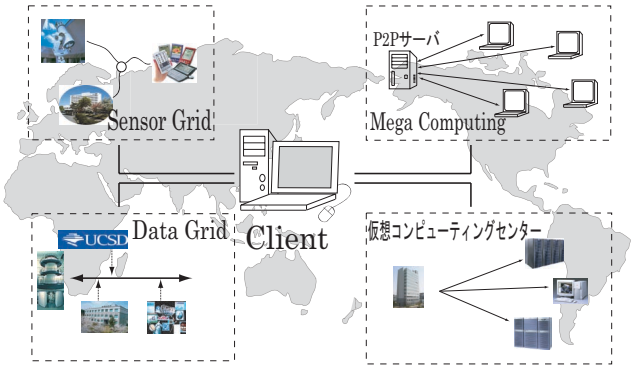

Fig. 1　Grid の応用例

4.1　手元にある計算機でユーザインタフェースを提供し，遠隔の計算機を利用するシステム

Fig. 1 の右下の例は仮想計算センターの例である．これは，手元にある計算資源（PC やワークステーション）上でユーザ・インタフェースを提供し，遠隔地にある高性能計算機を利用するシステムで，**Datorr（Desktop Access TO Remote Resources）**と呼ばれている．このシステムを用いれば，ユーザは，仮想計算センターに対して計算処理を依頼し，実行結果を受け取る．仮想計算センターは，ユーザの要求（1 時間以内に計算結果を返してほしい，1000 円以下で計算してほしい等）を満足できるように，適切なコンピュータを自動的に選択してくれる．以下で紹介する NetSolve や Ninf はこの Datorr システムのひとつである．

・NetSolve

テネシー大学で開発され，LAN あるいは広域のネットワーク上の数値計算ライブラリや科学技術計算に必要な数値情報データベースを通じて，主に科学技術計算分野の情報ならびに計算資源を提供・共有する仕組みを提供することを目的としている．

NetSolve を用いれば，手元のコンピュータに数値ライブラリをインストールする手間を省け，常に最新のライブラリで計算することができ，また計算センターのスーパコンピュータに login する必要がなくなる．

NetSolve は，NetSolve クライアント，NetSolve エージェント，NetSolve リソース（サーバ）で構成される．実行の流れは次のようになる（Fig. 2 参照）．

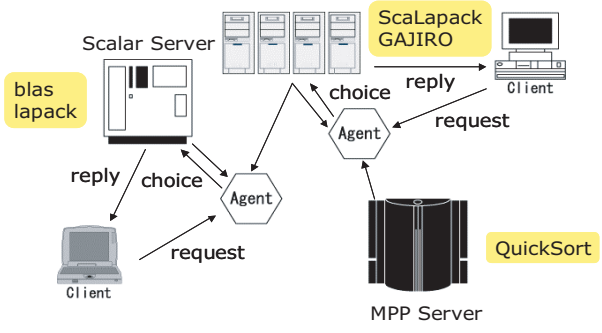

Fig. 2　NetSolve のアーキテクチャ

1. クライアントが実行の要求をエージェントに対して行う．
2. エージェントは要求を受け取り，適切な計算リソースを選ぶ．
3. クライアントは選ばれた計算リソースにデータを送る．
4. 計算リソースは，クライアントからデータを受け取り，数値ライブラリを実行し，結果をクライアントに返す．

・Ninf

A Network based Information Library for Global World-Wide Computing Infrastructure の略で，ネットワーク数値情報ライブラリという意味である．Ninf は，NetSolve と非常に類似しており，実際これらのプロジェクトはお互いに協力している．

4.2　Grid 上で共同作業を支援するシステム

・Access Grid

Grid を通して人間の協調作業をサポートする資源の全体である．Access Grid を用いれば，将来的には大規模な会議やセミナー，講義，個別指導などをサポートすることができるようになるであろう．

4.3　各 PC 同士の結びつきから，何らかのコラボレーションを行うシステム

私たちが普段日常生活で使っている PC は，ほとんどの時間イベントを待っているだけで，その計算能力の 9 割が活かされていないと言われている．このような PC の余剰計算能力をかき集めて，大きな計算能力

を必要とする問題を抱えている人に提供しようとする試みは，**Volunteer Computing（ボランティアコンピューティング）**と呼ばれる（Fig. 1の右上参照）．このような，イーサネット，インターネットで接続された異種のPCを数十〜数百万台使う大規模分散処理を**Mega Computing**と呼ぶこともある．このボランティアコンピューティングを実現するのが**P2P（Peer-to-Peer）**と呼ばれる技術である．

・P2P

各個人，各PC同士を相互につないで，ファイルや演算能力などの情報資源を共有するシステムである．P2Pを実現した例として，NapsterやGnutellaがある．このNapsterやGnutellaといったソフトウェアは，インターネットを通じて個人間で音楽データの交換ができるシステムで、無料で音楽が手に入ることから爆発的に普及した．

P2Pは，ファイル交換サービスにとどまらず，Grid環境においても計算/検索/データおよび情報の共有を促進する技術として注目されている．このP2P技術を用いて，すでに百万台以上のPCの参加を集めたプロジェクトにSETI @ homeがある（Fig. 3参照）．

・SETI@home

インターネットにつながっているコンピュータを使って地球外知的生命体の探査 (SETI) を行なう科学実験ですある．無料のプログラムをダウンロードして電波望遠鏡のデータを分析することで、誰でも参加することができる．SETI @ homeのソフトウェアは，通常スクリーンセーバとして動作し，PCが日常業務に使われている間は計算を行わないので利用者の邪魔にならない．

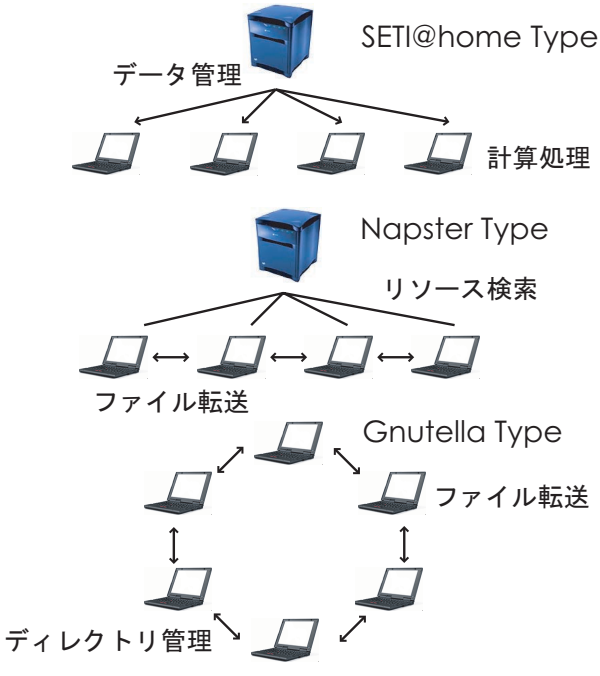

Fig. 3　P2Pの形態

4.4　遠隔の機器を制御したり，機器から出力されたデータを迅速に処理するシステム

・Sensor Grid

Fig. 1の左上がSensor Gridの例である．ユビキタスコンピューティングという社会基盤にセンサー群が配置されると，それを通じて日常の社会生活から得られるデータは膨大になる．これらの膨大なデータを記憶し，そこから有益な情報を知識として引き出すデータマイニングを行う処理を行うシステムがSensor Gridである．

4.5　1つの所では格納できないような大量のデータや分散配置されたデータを処理するシステム

・DataGrid

Fig. 1の左下がDataGridの例である．DataGridは，分散した数百TeraByteからPetaByte級の共有されたデータを格納し，世界中の研究者が分析処理を行うためのインフラである．

5　従来のクラスタと何が違うのか

Gridの従来のクラスタとの違いを以下に挙げる．

- **多数の組織から，様々なタイプのリソースが提供されて構築されるヘテロ環境**
- **動的に変化する環境**
- **外乱の多いネットワークによる接続**
- **手にできる計算資源の規模が格段に大きい**

  従来の並列処理では，同じプロセッサを用いて，比較的似通ったコンピュータを使用してネットワークを構成していた．これに対して，Gridで提供されるグローバルコンピューティングでの並列処理環境では，プロセッサやアーキテクチャが異なったコンピュータを結ぶヘテロなシステムが形成されている．また，そのネットワークが高速であっても，バンド幅やレーテンシは時間とともに変動する．

  その一方で，Gridでは，クラスタと比較して，大量の計算資源を得ることができるという特徴もある．

- **HPC[1]だけではなく，HTC[2]への要求**

  グローバルコンピューティングのユーザには，１つ１つのプログラムの実行速度にはこだわらないが，限られた時間内にできるだけ多くのプログラムを実行したいという要求がある．たとえば，数値シミュレーションの場合である．この場合，プログラムに与えるパラメータを少しずつ変えて実行することがよくあり，限られた時間内に多くのシミュレーションプログラムを実行する必要がある．こういった，

[1]High Performance Conputing
[2]High Throughput Computing

一定時間内に実行されるプログラム数（スループット）に注目して運用することを HTC という．

6 Grid の技術的，社会的問題

Grid 実現には数多くの考えなければならないことが存在する．それらのうち，主なものを以下にあげる．

- Grid では，これまでの並列処理技術がそのまま適用できない．Grid のような動的な環境に対応した並列プログラムでは，手作業でプログラミングすることは非常に困難になってくる．このため，並列の負荷分散を自動的に行うコードを生成してくれるコンパイラの支援が必要になる．

  また，複数のサイト間で効率的なスケジューリングを行うためには，従来のように単一のサイトでの独自のスケジューラを用いるだけでは不十分である．Grid で，複数のサイトで協調する並列プログラムを使用する場合，各サイトで同時にそのプログラムがスケジュールされている必要がある．このため，各サイトのポリシーで運用されているスケジューラを協調させるスーパースケジューラが必要である．

- ヘテロで動的な並列処理環境である Grid では，どうやってセキュリティを保証するか，どうやって自分の持つポリシーと相手の要望の妥協点を見つけるかという技術的問題がある．

- Grid によって，全世界の計算資源を集めて，いかに有効に使うかということも問題となる．

- 計算資源の割り当てを，現在のインターネットのようにだれもが自由に使えるようにするのか，利用度に応じて料金を負担するのかといった経済面での問題もある．ユーザの要求を超えるほどの計算パワーが提供されればうまくいくが，そうでなければ，デマンドアンドサプライの社会モデルをつくる必要がある．

- 信頼関係がない見ず知らずの他人との間におけるコンピューティングの共有をどのように促進するかという個人の問題もある．

7 Grid の実現のための技術

7.1 スケジューリング

グローバルコンピューティングシステム上でプログラムを実行させる場合，どの計算機にどの処理を行わせるかによって，実行にかかる時間が変わってくる．そこで，どの計算機にどの処理を割り当てるべきか，ということをサーバの性能や現在の負荷，サーバ・クライアント間のネットワークの混雑状況などの情報から決定するのがスケジューリングソフトと呼ばれるものである．

HTC を実現するためのスケジューリングソフトとして，Condor と呼ばれるものがある．Condor はネットワーク上にある，現在は何の処理が行われていない計算機を探し出し，クライアントの要求するプログラムを実行させるものである．

7.2 セキュリティ

グローバルコンピューティングシステムを用いる場合，セキュリティの確保が重要となる．そのための技術のひとつとして，認証技術である GSI[3]について紹介する．

GSI の特徴として，分散する様々な計算資源へのアクセスをたった 1 度だけのユーザ認証で実現する，シングル・サインオン機能があげられる．もしこの機能がないと，各計算機を利用するごとにパスワードが要求されることになり，利便性が大幅に減少してしまう．

GSI では，公開鍵暗号を用いた認証を行う．クライアントはまず認証局から証明書を発行してもらい，それからプロキシ証明書を作成し，ゲートキーパーに送る．ゲートキーパーはそれを復号化することでクライアントを確認し，同時にマップファイルを用いてアクセス制限を行う．

8 グローバルコンピューティングシステムの紹介

8.1 グローバルコンピューティングシステムの階層モデル

グローバルコンピューティングは，階層構造を持っている．それは，アプリケーション層，ミドルウェア層，ツール群層，ネットワーク・テストベッドの 4 層である．これを Fig. 4 に示す．グローバルコンピューティングに関する研究が盛んになり始めたころは，主にミドルウェア層に位置するシステムが多く研究，開発されてきた．現在，代表的なグローバルコンピューティングシステムの多くは，ミドルウェア層に位置する．これらは，グローバルコンピューティング環境で動くアプリケーションの作成や，実際に動かす際の手助けをするシステムである．また，プログラマが直接触れることになるシステムでもある．

8.2 ミドルウェア層

- NetSolve
  ネットワークを介して遠隔地にある科学技術計算ライブラリを利用できるシステム．

- Ninf
  Netsolve と類似しており，遠隔地にあるハードウェア，ソフトウェア，データベース等を利用できるシステム．

[3]Grid Security Infrastructure

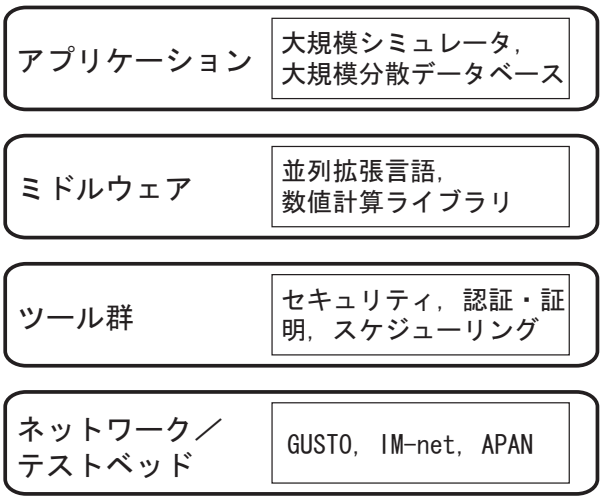

Fig. 4　グローバルコンピューティングシステムの階層モデル

- Condor
  分散した資源を使って高スループットな計算（HTC）を行うシステム．Condor の基本思想は，アイドル状態にある計算機を集めてユーザのプログラムを実行させるというものである．オフィスや学校等にある計算機の集まりを Condor pool と呼び，Condor pool に含まれるアイドル状態の計算機を探し出してユーザのプログラムを実行する．この実行手順を以下に示す．

  1. ユーザは，自分のプログラム実行に必要な条件（必要メモリ量，OS の種類など）を提示する．
  2. Central Manager(CM) は，ユーザの提示する条件を満たし，かつアイドル状態の計算機を Condor pool から探し出す．ここで，CM とは，Condor を構成する計算機のひとつで，Condor poor に含まれる各計算機の状態を監視するものである．
  3. 条件に合う計算機で実行する．

  ユーザのプログラムの実行中である計算機が，持ち主による利用を再開すると，実行中のプログラムを停止して，実行を他のアイドル状態にある計算機に移す必要がある．このために，Condor は，常にプログラムの実行の途中結果を保存している．

- PACX-MPI
  グローバルコンピューティング環境用に拡張した MPI 通信ライブラリ．MPI とは，メッセージ通信による並列プログラミングにおいて，最も標準的に利用されているメッセージ通信ライブラリである．しかし，既存の MPI をグローバルコンピューティング環境で利用するのはアーキテクチャの異なる計算機間での利用や遠隔システム上でのジョブプロセスを生成する方法に関して問題があり，性能面でも問題がある．

  広い地域に分散配置された複数の並列計算機をネットワークで接続したものを巨大なクラスタとみなし，その上で MPI で書かれたプログラムを動かす場合，性能が問題となる．すなわち，異なる計算機上で動くプロセス間では TCP/IP を用いて通信することになる．通常，並列計算機は，計算機に依存した方法で高速にメッセージ通信を行うことができる MPI が実装されているが，通信相手が自分と同じ計算機上で動いているのか，もしくは，異なる計算機上で動いているのかを判断するのは困難であるため，すべてのプロセスを TCP/IP を用いて通信せざるを得なくなる．TCP/IP は MPI にくらべて，通信性能が大きく劣っている．PACX-MPI は，この問題を解消した．それは，PACX-MPI が，同じ計算機上で動作するプロセス間ではその計算機固有な MPI を使って通信し，異なる計算機上で動作するプロセス間では TCP/IP を使って通信することを可能としたからである．

8.3　ツール群層

- Globus Toolkit
  ミドルウェア開発のためのツールキットでツール群層にあたる．ユーザ認証システム，通信ライブラリ，計算資源管理機構といったグローバルコンピューティングシステムの構築に必要になる基本的なサービスの集まりである．Grid のツール，サービス，アプリケーションを開発する上で標準的な Grid プロトコル・API を定め，実装する．このようにして，Grid ノード間のプロトコルの違いを吸収する．

  Ninf や PACX-MPI などミドルウェア層に位置するシステムを構築するためには，ユーザ認証，通信，遠隔計算機上でのプロセス生成など，様々な要素技術（ツール）が必要になる．各システムは，それぞれ独自にこれらのツールを実装してきたために，各システム間に互換性がなくなってしまった．

  しかし，96 年頃に発足した Globus プロジェクトによって，状況が変わってきた．これは，並列・分散計算，ネットワーク，セキュリティなど様々な分野の研究者たちが参加するプロジェクトである．このプロジェクトで Globus Toolkit が開発された．Globus Toolkit によって，それまで問題であったミドルウェアのシステム間の互換性を保つことができるようになった．Fig. 5 に Globus Toolkit が提供するサービスの一覧を示す．

サービス	名前	概要
資源管理	GRAM	リソースの割り当ておよびプロセス生成
通信	Nexus	Unicast/Multicast 通信サービス
情報	MDS	システムの構造および状態に関する情報へのアクセス
セキュリティ	GSI	authentication などのセキュリティサービス
状態管理	HBM	システムの状況サービス
遠隔データアクセス	GASS	データへのリモートアクセスサービス
実行ファイル管理	GEM	実行ファイルの構築，キャッシングおよび配置

Fig. 5　Globus Toolkit のコアサービス

- Legion
  デスク上のパソコン・ワークステーションから世界中の資源にアクセスできるオブジェクトベースのメタシステム．すなわち，広域における，プロセス生成，ファイルシステム，セキュリティ，資源管理などの OS のサービスを提供するものである．Legion のファイルシステムは，実際にはファイルは世界中のどこかのコンピュータの記憶装置に格納されているにもかかわらず，世界中のどのコンピュータからでもまったく同じようにそのファイルにアクセスできる．

  ただし，ファイルのアクセススピード，プログラムの実行速度，ファイル・ディレクトリの作成時間などの短縮が難しいという問題もある．

Globus と Legion はともにツール群層にあたるが，Globus がサービスの集合であるのに対して，Legion はひとまとまりの OS であるといえる．また，今後は，ツール群層の標準化にとどまらず，ミドルウェアの標準化を目指すことになる．これにより，アプリケーション層以外は，互換性を保つことができるようになる．

8.4　テストベッド

グリッドのソフトウェアを開発すると，その研究成果を実際の環境で試すことが必要となる．そのとき用いられるのがテストベッドである．テストベッドとは，グリッドのソフトウェアをテストするために使用する，ネットワーク，サーバ，クライアントといったグローバルコンピューティングの物理的環境のことである．テストベッドは，アプリケーションの作成，ユーザの拡大などの点においても非常に重要である．

Globus プロジェクトでは GUSTO(Globus Ubiquitous Supercomputing Testbed Organization) というグリッドのテストベッドを構築している．これは，現在 23 カ国，125 サイト（大学，研究所など）で構成され，グリッドのテストベッドとしては最大のものである．

なお，Legion を用いたテストベッドでは，カリフォルニア大学サンディエゴ校，Caltech，NASA Ames などによる National Legion Net があり，共通の計算環境を提供している．

参考文献

1) 関口智嗣 他.『Computer Today』 2000.1 月号〜2001.11 月号．サイエンス社

2) 谷村勇輔，下坂久司.『Grid Tutorial』