

1

グリッド・コンピューティングの商用システムへの適応性

Adaptability of grid computing to commercial systems



日本アイ・ビー・エム システムズ・エンジニアリング株式会社
第一システム・センター
サーバー・システム部
部長

鹿嶋 浩

Hiroshi Kashima

Manager of Server Systems
System Center No.1
IBM Japan Systems Engineering Co.,Ltd.

現代の企業経営における最大のテーマは、ビジネス・インテグレーションによる全体の最適化です。企業は提携を前提とした企業間での組織統合によって、プロセス変革やコスト削減など、全体の競争力強化を目指しています。この潮流に対するソリューションの基盤技術として期待されているのがグリッド・コンピューティングです。以前から学術分野では利用されてきましたが、エンタープライズ分野でもいよいよ本格的に利用される時代が始まろうとしています。

グリッドは、コンピューターに対する考え方を根本的に変えるインフラストラクチャーです。私たちはこの新しいコンピューティング環境に対しては予備的な知識しか持っていません。今後の普及は、これが何に使用され、どのように見えて、適用や運用のポイントが何であるのかといった明確なビジョンと理解が欠かせません。本論文は、これらに対する一つの有効な議論のステップとなると考えています。グリッド・コンピューティング実現に向けてのアプリケーションやテクノロジーの評価の指針となれば幸いです。

One of the main topics facing corporate management today is that of general optimization by means of business integration. Companies aim to increase their overall competitive strength through measures such as process reform and cost reduction effected through organizational integration between companies premised upon cooperation. Grid computing is regarded as a core technology in solutions intended to respond to this development. Although grid computing has been used for some time in academic fields, it is now at last coming into serious use in fields of business enterprise.

Grids constitute an infrastructure that will make fundamental changes in the way we think about computers. We still only possess rudimentary knowledge of this new computing environment. The future diffusion of this environment is dependent upon a clear vision and understanding of what it is going to be used for, of how it appears, and of the main points regarding its application and operation. This paper is intended to provide a step toward an effective debate on this subject. I hope it will provide guidelines for assessment of applications and technology with a view to realization of grid computing.

1.はじめに

私たちはインターネットによりネットワーク資源を相互に共有するようになりました。その後WWWが登場して相互に情報を共有するようになり、現在はインターネット上で商行為も行われています。インターネットの普及によりオンライン上に接続されているコンピューター資源は年々増加しており、今後は携帯電話やゲーム機、情報家電の接続も増えていくでしょう。

一般的なコンピューターの使用率は10～30%程度といわれていますが、ピーク時は10倍以上の負荷がかかることも珍しくありません。また、一部のサーバーに処理が集中するという現象もよく見られます。このピーク負荷に対応するためにサーバー資源を潤沢に用意するのはコスト的に困難です。そこで、いまままでネットワークおよび情報を共有してきたことをさらに拡張して、ネットワーク上に接続されたコンピューター資源を共有できれば、ピーク負荷対応の解決方法の一つになり得ます。

一方、企業間のビジネス・インテグレーションの流れはオンライン上でのアウトソーシングであるe-ソーシングを加速しています。また、オンデマンドで「使いたいときに使いたいだけ」コンピューターを利用する、公共サービスのe-ユーティリティの考え方も次世代e-ビジネス基盤の登場を後押ししています。

グリッド・コンピューティングは、こうした潮流に対してソリューションを提供できる技術として脚光を浴びています。

2. グリッド・コンピューティングとは

2.1. グリッド・コンピューティングの概要

グリッド・コンピューティングは、インターネット上の複数のコンピューターを接続し、プロセッシング・パワーやデータ・ストレージなどのコンピューティング資源を共有し、協調処理するための分散コンピューティング技術です。大量の処理能力が必要なときには複数のコンピューターが持つプロセッシング・パワーを統合するなど、柔軟で効率的な管理・運営を目的としています。

グリッドで共有する資源にはプロセッサ、ストレージ、I/O、データ、ソフトウェアなどがあります。共有すべきエンティティは物理的エンティティである必要はありません。具体的には分散ファイル・システムや資源プールなども共有可能です。

グリッドが実現するのは、企業や団体を越えた大規模な資源共有環境であり、従来のクライアント/サーバー・システムなどの分散技術と比較して、以下のような特性を持ちます。

- 個人・組織・団体のダイナミックな集まりにおける共有異なる組織・管理ドメイン

ダイナミックな参加 / 脱退

- セキュアでコーディネートされたアクセス
 - プロセッシングやストレージ資源、データ、サービスへのアクセス
 - 資源提供者による高度な制御
 - だれが何をどのような条件で共有可能かを明確に定義
- 地球レベルの大規模コンピューティング
 - ヘテロジニアス
 - WANによる遅延・スケール
 - ダイナミック性・予測不能

グリッドを利用する企業や団体のダイナミックな集合をバーチャル組織と呼びます。グリッドにおけるバーチャル組織は、管理ドメインやセキュリティ・ポリシーが異なるほかの企業や団体をいつでも必要に応じて追加登録 / 削除ができることに特徴があります。グリッドの本質は、バーチャル組織をあらかじめ構築し、認可された組織がいつでも追加参加可能なダイナミズムと、そのためのセキュリティを実現していることです。

グリッドのシステム形態と、従来のマルチプロセッサやクラスターとの違いについて以下に説明しましょう(図1参照)。

マルチプロセッサ(MP)は、プロセッサが高速なスイッチや共有メモリーで接続された密結合のアーキテクチャーです。通常は一つのオペレーティング・システムが動作し、メッセージングの高速性を前提にするような並列処理に使用されます。

クラスターはプロセッサを緩やかに接続したシステムであり、通常、スイッチの速度はMPより遅くなります。複数のプロセッサに異なるジョブを並列処理させて、結果を集めて一つの結果を出したり、大量のスループットが要求される作業を分割処理するときなどに利用します。

グリッド・コンピューティングは複数のシステムおよびクラスターを接続し、これらを一つの大きなコンピューターのようにとらえたものです。MPやクラスターでは一般的に単独のシステムは同じ管理ドメインに属しますが、グリッドは一般に異なる管理ドメインに所属します。複数の管理ドメインを超えてコンピューター資源の管理・利用を行うので、セキュリティも含めた複雑なミドルウェア処理と、それらをまたがって効率的に処理するアプリケーション

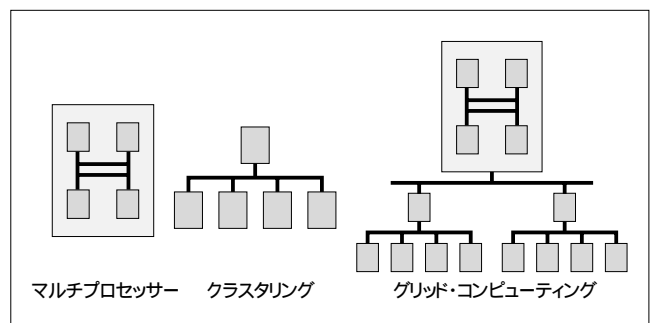


図1. システムの概念図比較

ション設計が必要になります。

グリッド・コンピューティングが優れている点の一つは、既に多くの実績があり、統一されたアクセス方式が提供されていることです。過去において、TCP/IPが十分な実績を積んだ後に商用に適用されたのと同様に、グリッドも多くの実績を背景に商用システムへの適用が行われようとしています。

2.2. 適用業務

グリッド・コンピューティングには学術分野を中心に既に多くの適用事例があります。プロセッシング能力を共有する例としては、複数のクラスターやスーパーコンピューターなどを接続し、互いのCPU空き時間を利用して遠隔ジョブ投入を行うものが代表的です。例えばUS NSF(全米科学財団)のTeraGridでは39台のスーパーコンピューターを接続し、600^{テラ}Tバイト、13.6T FLOPSの処理能力を40Gバイト/秒のネットワーク上に実現し、高速演算を行っています。こうした応用は科学技術計算を中心に行われてきました。

空きCPUを利用するアイデアは、資源を共有する分散コンピューティングに発展します。この背景にはネットワークとコンピューターの著しい高速化があります。大量のデータを複数の場所に配置して、ユーザーが位置を意識しないアクセス(位置透過性)を可能にするのがデータ・グリッドです。事例としては1^{ペタ}Pバイト/年のデータを共有する仕組みを構築するCERN(欧州原子核共同研究機関)や、TeraGrid、Euro DataGrid(高エネルギー物理学、環境科学、バイオなど)、ペンシルベニア大学の胸部X線写真を複数サイトで共有するプロジェクトなどが著名です。これは数千サイトを接続可能にすることを目標に設計されています。

より緩やかな接続の例としては、デスクトップPCの空きCPUを収集するものがあります。各家庭のPCを利用するSETI@home(地球外知的生物探査研究所)は私たち一般ユーザーでも参加できます。これはインターネット上のユーザーがコンピューターを使用していないときに、スクリーン・セーバーとして動作して、センターに準備されたジョブを分担して処理するという形態を取ります。宇宙から発せられる電波信号の中に知的生命体の兆候を探そうというプロジェクトです。

このほかにも、グリッド・コンピューティングはバイオ、ケミカル、宇宙、気象など科学技術計算分野において積極的に利用されてきました。さらに、現在研究中のものにはサービスをグリッドで提供したり、空きCPU資源を売買するなど、商用分野にも適用できるものが増えてきています。

これらをまとめると、現在のグリッド実現へのアプローチには幾つかのパターンがあることが分かります。

(1) プロセッシング・グリッド

スーパーコンピューターやクラスター・システムを接続し、アプリケーション投入を主体にプロセッシング能力を共有するタイプ。

(2) デスクトップ・グリッド

主としてWindows®デスクトップのユーザーの空き時間を利用してジョブを実行する未使用コンピューター・サイクルの収集タイプ。

(3) データ・グリッド

大量のデータを地理的に分散して配置し、ユーザーからは過剰的にアクセスする機能を提供するデータ共有のグリッド。

(4) その他

上記以外の資源を共有するグリッド。

長期的なビジョンは、ダイナミックなバーチャル組織における巨大なバーチャル・コンピューターの実現です。将来的には、ネットワークをオペレーティング・システムとしてサポートするC言語ランタイム(libc.a)のようなものが登場することも考えられます。グリッドを代表するミドルウェアであるGlobus Toolkitでは、ファイル/IOについてこのコンセプトに近いものが既に利用可能であり、ファイル名にURLを指定できるfopen関数などが提供されています。

2.3. 今後の適用分野

“The Grid [参考文献2]では、グリッドのアプリケーションとして以下のパターンを挙げて解説しています。

- 分散スーパーコンピューティング
- 高スループット
- オンデマンド
- データ・インテンシブ
- コラボレーティブ

筆者はグリッドの適応範囲はこれだけにはとどまらなと考えています。その高度な接続性やセキュリティ、パフォーマンスは、Webサービスが目指しているビジネスや、e-ソーシング、e-ユーティリティーなどの基盤として十分に普及するものと考えます。その適応能力はアイデア次第で無限の可能性を持ち、まさに次世代e-ビジネスの基盤技術といってもよい存在です(表1)。

表1. 適用アイデアの例

- | |
|--|
| <ul style="list-style-type: none"> • 大学・家庭のCPUパワーの利用 • 時差を活用した国際間での共有(昼のピーク時対策) • ピーク時のバックアップCPU • 空いているCPUをフルに使い切る • バックエンドのホストをモバイルやゲーム機のパワーとして利用 • e-ソーシングなどさまざまな通信インフラストラクチャーとして利用 |
|--|

ここまで見てきたように、グリッドは分散コンピューティングの基盤技術です。そのため今後の適用分野も、特定のアプリケーションやセグメントに限定されない広範なエリアでの活用が考えられます。企業間での共同設計におけるコラボレーションや、各種数理計算などのピーク対応、取引業務やWebホスティングのインフラストラクチャーなど、今後は商用分野での幅広い応用が期待されます。

3. グリッドに求められる機能

こうした背景を持つグリッドは、どのような機能を提供するべきでしょうか。明確な定義付けをした文献は見当たらないものの、さまざまな製品の機能やその役割を見た上で、筆者なりの考えを以下にまとめます。

3.1. 主要な機能

グリッドが企業や団体間を接続するインフラストラクチャーであることを考慮すると、物理インフラストラクチャーに加えて以下の機能を有するべきであると考えます。

3.1.1. 必須機能

(1) 資源の共有・配分

資源を共有あるいは配分する機能は本質的に必要です。グリッドは、だれが何をどのような条件下で共有できるかが明確に定義可能で、コントロールできることが重要です。

(2) 透過的なリモート・データ・アクセス

リモート・データへのアクセスは、ユーザーからは場所を意識しない透過的なものである必要があります。それにはグローバルな論理名によるアクセス機能が必要です。

(3) 性能保証

ジョブを投入する側はCPU、メモリー、ネットワークなど資源に関する条件の指定が可能で、グリッド・システムは条件に見合ったリソース・ノードを探し、割り当てる機能が必要です。そのためにディレクトリー機能がリソースの空き情報を提供する必要があるかもしれません。

(4) ヘテロジニアスな環境をサポート

インフラストラクチャーがサポートする環境には制限がない方がよいでしょう。ビジネス・インテグレーションにおいてほかの企業や団体の参加を阻害しないためです。さまざまなプログラミング言語、OS、ハードウェアをサポートし、それらの間でコラボレーションできることが望まれます。

(5) スケーラビリティ

2.2節で紹介した事例は数百～数千以上のプロセッサが接続されています。コンセプトから導出される要件としては、Webと同様に事実上は無限の接続性が必要となってきます。資源数、参加数、プログラム・コンポーネント数などに制限がないアーキテクチャーが求められます。

(6) セキュリティとプライバシー

異なる企業や団体が相互に接続して分散コンピューティングができるだけの柔軟性を持ったセキュリティ技術が求められます。ユーザーにとってはシングル・サインオンなどの使いやすい認証・認可の仕組みが必要であり、また、さまざまな管理ポリシーの組織間の接続をサポートできなければなりません。

(7) ダイナミックな環境の変化

参加者はダイナミックに登録/抹消され、提供するサービスもダイナミックに変化します。こうした環境の変化に対応しなければなりません。

以上が必須の機能です。次にオプションとして望ましい機能を列挙します。

3.1.2. オプション機能

(1) 参加が容易

優れた企業とコラボレートするためにも、参加が容易でなければなりません。それには十分に標準化された汎用的なインターフェースを採用していることが重要です。また、モジュラー化によるコンポーネントごとの選択や交換が可能で、かつそれらのインターフェースや実装がオープンになっている必要があります。

(2) 堅牢^{けんろう}・高性能

提供されているリソースは十分にロード・バランスされ、高性能である必要があります。また、フォールト・リカバリーが考慮され、参加ノード数が多くても堅牢^{けんろう}であることが求められます。

(3) 運用管理が容易

ノード数が増加しても容易に運用管理できる必要があります。そのためには、オペレーションによっては位置や数を意識させない透過性が求められることもあるでしょうし、場合によっては全体を見通せるビューの提供も必要になります。各ノードが自律的に自動運用/回復できる機能も重要になるかもしれません。また、企業間での接続を前提にする場合はモニタリングやアカウント管理機能も必要です。

これらの機能をすべて実現するのは、かつてないほど複雑です。

3.2. グリッドを支えるアーキテクチャー

前節で説明した機能を実現する製品やアーキテクチャーについて考察します。ここではデスクトップ・グリッド、データ・グ

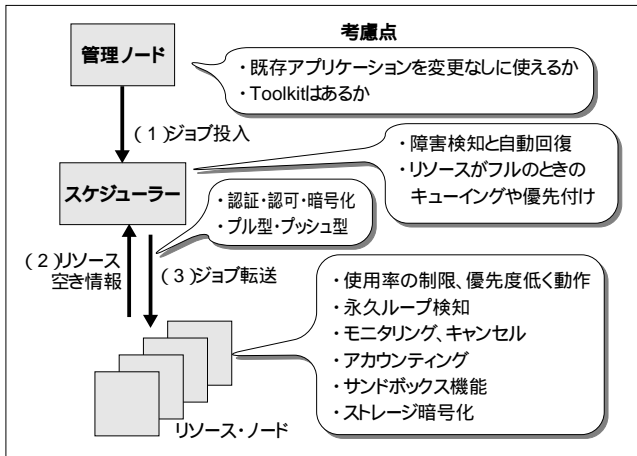


図2. デスクトップ・グリッド概念と考慮点

リッド、プロセッシング・グリッドを例にとって一般的な構成と考慮点を指摘します。

3.2.1. デスクトップ・グリッド

このタイプのグリッドは、実行したいジョブが相互に依存性の低い小さな単位に分割できる場合に利用できます。製品としてはPlatform Active Cluster、United Devices、Entropia、DataSynapseなどがありますが、おおむね図2の構成を取ります。

システムは、ジョブを投入する管理ノードとスケジューラー、CPUを提供するデスクトップ(リソース・ノードと呼ぶ)から成ります。処理の流れは以下のようになります。

- (1) 管理ノードからジョブを投入します。
- (2) リソース・ノードはアイドルを検知するとジョブを受け付けできることを知らせます。
- (3) スケジューラーはリソース・ノードにジョブを転送します。入力データ転送・アプリケーション転送・結果戻しの一連の動作を行います。

通常はリソース・ノード間での通信は行えません。このタイプのグリッドで重要なのは、専用のアプリケーションを開発する必要があるかどうかという点と、必要な場合の開発環境の充実度です。また、デスクトップ・ユーザーから見ると、他人のプログラムが勝手に動作することになるので、サンドボックスなどプライバシーを守る仕組みも重要であると同時に、グリッド・ジョブの内容を盗み見されないようなストレージ暗号化などの仕組みも必要になります。

3.2.2. データ・グリッド

データ・グリッドでは、データはユニークな論理名でグローバルに識別され、物理的に異なる場所にあらかじめ配置されています。カタログは論理名とレプリカの場所を登録してあります(図3)。処理の流れは以下の通りです。

- (1) 処理の前提として、各レプリカはファイル転送などの方法によって内容の同期が取られています。
- (2) カタログは、アプリケーションから要求されると適当なレプリカを選択して、その情報を戻します。
- (3) アプリケーションはレプリカからデータを得ます。

データ転送には多くの場合、ファイル転送プロトコルが使用されます。筆者の知る限り、現時点ではNFSやAFS®のような、ローカル・ファイル・セマンティクスでのアクセス方法を提供している製品はまだ登場していません。データ・グリッドの例として有名なのはGlobusプロジェクトとLegionプロジェクト(Avaki社が製品化)です。

3.2.3. プロセッシング・グリッド

リモート・ジョブ投入やRPCなど、このタイプのグリッドは実績が多く、Platform LSF、PBS、Condor、Globus Toolkitなどのソフトウェアがあります。この中でセキュリティーなど、本来のグリッド機能を提供しているという意味では、Globus Toolkitが群を抜いた実績を誇ります。図4にGlobus Toolkitにおけるアプリケーション投入の仕組みを示します。

Globus Toolkitの中でも、リモート・ジョブ投入のコンポーネントはGRAMと呼ばれます。処理の流れは以下の通りです(コンポーネント名の略語は次節で説明します)。

- (1) クライアントはセキュリティー認証を済ませておきます。
- (2) クライアントは適当なリソースを探すためにディレクトリーMDSを検索します。MDSは検索効率を高めるために、全体の検索のためのGISと、個別リソースの情報を保持するGRISに分かれていて、DNSのように階層管理を行います。
- (3) GRISはリソースの最新情報を照会して保持しています。
- (4) GRISは各リソースの使用状況などをクライアントへ提供します。
- (5) クライアントは適当なリソースを選択し、それを担当するGRAMサーバーにリクエストします。GRAMサーバー側では、Gatekeeperデーモンがセキュリティー・チェックをした後に、ローカルIDにマッピングしてその権限で実行します。

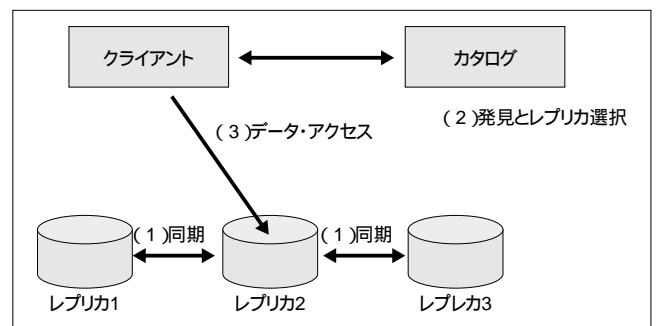


図3. データ・グリッドの概念

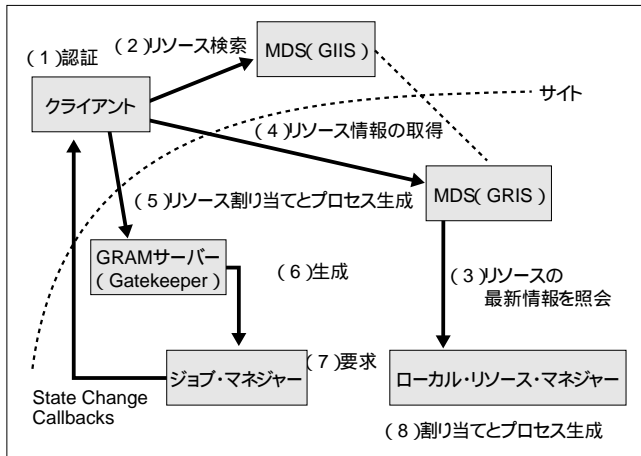


図4. アプリケーション投入の概念

- (6) Gatekeeperはジョブ・マネジャーを生成します。ジョブ・マネジャーは、ローカル・リソース・マネジャーとのインターフェースを取り、状況をクライアントへ報告します。
- (7) ローカル・リソース・マネジャーへジョブをリクエストします。ローカル・リソース・マネジャーは単純な場合は単にfork関数であることもあり、Platform社LSFのようなクラスター・ソフトウェアの場合もあります。
- (8) ローカル・リソース・マネジャーは、リソースを割り当てた上でプロセスを生成するなど、ジョブを実行します。

3.3. Globus Toolkitのアーキテクチャー

Globus Toolkitの説明はコンポーネント名も多く、難解であるかもしれませんが、しかしながらGlobusはグリッドのデファクト・スタンダードであるため、アーキテクチャーについても理解しておく必要があります。

Globus Toolkitは、グリッドにおけるアプリケーション開発に必要なとされる要素技術を実装したソフトウェア・ツール群です。標準的なグリッドのプロトコルやAPIを「袋詰めした技術」としてモジュラー構造かつオープン・ソースで提供されています。Globusプロジェクトは米アルゴンヌ国立研究所や南カリフォルニア大学が中心になって進めているプロジェクトであり、グリッド・コンピューティングにおける指導的立場にあります。

Globus Toolkitは、インターネット・プロトコルとサービス上に構築され、概念的にレイヤー化されています(図5)。

(1) ファブリック・レイヤー

コンピューター、リソース・プール、ファイル・システム、カタログ、ネットワークなど、共有するリソースへのインターフェースです。ローカルへのインターフェースと考えることもできます。

(2) コネクティビティー・レイヤー

ネットワーク接続用のレイヤーであり、インターネット・プロトコルやセキュリティ(認証・認可・暗号化など)を提供します。

(3) リソース・レイヤー

アプリケーションが単一のリソースを共有するための要素技術を提供します。

(4) コレクティブ・レイヤー

利用可能なリソースの把握や負荷分散、障害検知などを提供し、アプリケーションが複数のリソースを使用する支援を行います。

これらのレイヤーに配置される実際のコンポーネントには次のようなものがあります。

- GSI: Grid Security Infrastructure

SSLやX.509など標準プロトコルをベースに、シングル・サインオン、デレゲーション(委譲)、ローカルへのIDマッピングなど、グリッド環境における高度なセキュリティ機能を実現します。

- GRAM: Grid Resource Allocation Management

リソース割り当て・予約・監視・制御などのリソース管理を行います。プロトコルはHTTPベースのRPCです。要求を記述するフォーマットはRSL(Resource Specification Language)として規定し、マシンの種類、ノード数、メモリーなどの多くのパラメーターを指示できます。また、従属的に動作するジョブ・マネジャーへ委譲するためのインターフェースがあり、Platform LSF、OpenPBSなどと連携可能です。

- MDS: Metacomputing Directory Service

グリッドのリソース情報を提供します。GIISとGRISで構成されますが、どちらもアクセスにはLDAPを使用します。

- GIIS: Grid Index Information Service

グリッド上でのサーチ・エンジンの役目を担い、「CPU 800MHz以上、メモリー120Mバイト以上のサーバー」などのような検索を可能にします。

- GRIS: Grid Resource Information Service

リソース情報を収集します。

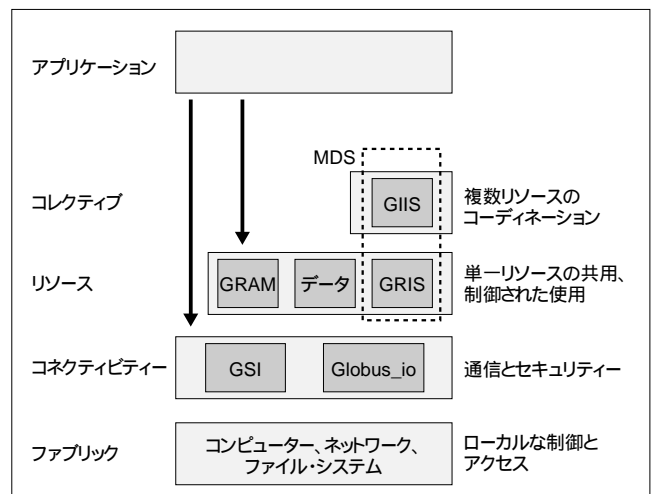


図5. レイヤー化されたグリッド・アーキテクチャー(Globus Toolkitは網掛け部を提供)

表2. 従来技術とGlobus Toolkitの比較

	クライアントサーバー	OSF DCE	CORBA	Webサービス	Globus Toolkit
通信	Socket/RPC	RPC	ORB	HTTP	Globus_io
遠隔実行	RPC	DCE RPC	ORB	SOAP	GRAM
インターフェース定義	IDL	IDL	IDL	WSDL	RSL
セキュリティ		Security			GSI
ディレクトリー	DNS	Directory		UDDI	MDS(GUIS/GRIS)
時刻同期	ntp	Time			(ntp)
ファイル共有	NFS	DFS			GridFTP, GASS

以下に挙げるGridFTP、Replication、GASSは、データ・グリッドを構成するコンポーネントです。

- GridFTP: Grid File Transfer Protocol

FTPの拡張で、ストライピング、並列データ転送、部分ファイル・アクセス、転送状況モニタリング、リスタートなどの機能を提供し、高速データ転送を実現します。

- Data Replication

ファイルやデータ・コレクションの論理名と、それに対応する一つ以上の物理ロケーションとのマッピングを管理し(レプリカ・カタログ)信頼性の高い複製を生成します(レプリカ管理)。

- GASS: Globus Access to Secondary Storage

ファイルのステーキング、I/Oリダイレクション機能を提供し、URLで指定されたファイル・システムへのアクセスを可能にします。

Globus Toolkitが提供しているコンポーネントは以上ですが、これらをサポートする並列計算ライブラリー(MPICH-Gなど)や、スケジューラー(Condor-G、Nimrod-Gなど)RPC(Ninf-Gなど)のミドルウェアやアプリケーションが多数開発されています。

参考までにGlobus Toolkitのコンポーネントと従来技術との比較を表2に示します。企業においては、グリッドに精通した技術者はまだ少ないと思われるかもしれませんが、既存のアーキテクチャーの知識を有効に活用することができます。

3.4. 商用も見据えた新世代のグリッド機能

グリッド・コンピューティングの研究や応用を標準化の面でリードしているコミュニティーにGGF(Global Grid Forum)があります。GGFはグリッド・コンピューティングの仕様を定義している標準化団体であり、European Grid(欧州)、US Grid Forum(米国)、AP Grid Forum(アジア)が集まって組織されています。

ネットワーク・プロトコルの設計や、標準インターフェースの策定およびアプリケーション開発の促進などを行い、扱うエリアはインフォメーション・サービス、セキュリティ、スケジューリングと管理、パフォーマンス、アーキテクチャー、データ、アプリケーションとモデルとなっています。

GGFが提案している次世代のグリッド標準がOGSA(Open

Grid Services Architecture)です。

OGSAはインターオペラビリティの向上とグリッド技術の容易な統合を可能にします。Webサービスの技術を採用し、Globus Toolkit上に構築されます。従来のWebサービスがステートレスであり、永続的なサービスのサポートを対象にしていたのに加え、サービス・インスタンスの作成などを含んだステートフルな環境を実現し、分散アクティビティのステートをサポートします。そのために、サービスのネーミング・発見・管理・使用などの方法について言及していることが特徴です。また、ホスティングの対象としてJ2EE、.NET、C言語などの複数の環境を考慮しています。OGSAではグリッド・サービスとして、以下のインターフェースと振る舞いを規定しています。

- 作成(ファクトリー)
- グローバル・ネーミングとリファレンス(GSH & GSR)
- ライフ・タイム管理(明示削除と、キープアライブによるタイムアウト)
- 登録と発見
- 認証
- 通知(動的発見・登録、監視エラー通知などの多様な目的)
- コンカレンシー
- 管理

2002年2月にトロントで開催されたGlobal Grid Forumにおいて、IBMとglobus.orgはOGSAを発表しました。OGSAはグリッド・コンピューティングとWebサービスの統合の標準化と仕様の制定です。特にリソースの発見、スケジューリング、セキュリティなどの分野においてグリッドの標準化にWebサービスの標準を取り入れています。

4. 商用システムへの適用を目指して

現時点でのグリッドは、商用システムへの適用という意味ではまったくの黎明期であり、未知の分野への第一歩を踏み出したに等しい状況です。IBMは過去においても新しい技術を商用システムへ適用させてきた実績があります。その経験を基に、現状の把握と問題解決への施策、そしてシステム設計上の考慮点を以下にまとめます。

4.1. 現状への対応

初めにグリッドを商用システムへ適用するために、現時点で想定される課題を洗い出し、さらに必要なスキル向上のために検証タスクを実施しました。図6はグリッドを商用システムへ適用する際に問題となる事項と原因、そして筆者の所属グループが

実施中の施策を表しています。

(1) 設計ノウハウが不足

グリッド・コンピューティングは、今までのクライアント/サーバー・システムとは異なるシステム形態であるため、設計ノウハウがないという問題があります。このため、まずは既存のミドルウェアや事例を研究して、設計に対する考え方を体得することになりました。

Platform、Avaki、DataSynapse、Entropy、United Devices、OpenPBS、Condor、Sun Grid Engine、LoadLeveler[®]などの製品群を実際に使用し、その考え方やお客様への適応性を検証しています。

また、アプリケーションを実際に試作して見識を得るとともに、不足しているスキルを認識して対応を検討することとしています。例えば、設計に当たっては各種アルゴリズムへの理解が必要であるため、各種文献を研究するとともに、学会やフォーラム、メーリング・リストへ参加して、メンバーへの必要なフィードバックを行っています。

(2) キャパシティー設計が難しい

キャパシティー設計のためには、ソフトウェアのパフォーマンス情報を取得する作業が必要です。このほかに、グリッドではシステム構成の複雑性を考慮した予測を行う必要があります。グリッドは管理ソフトウェアや設計アルゴリズムによるパフォーマンスの差が大きく、さらに評価や改良を実際に広域かつ大規模なシステム構成の上で行うことが極めて困難なシステムであるため、場合によってはシミュレーターが必要になることもあります。学会やフォーラムからの情報も積極的に取り入れてはいますが、残念ながらこのエリアについては課題も多くあります。まずは個別のパフォーマンス情報の取得から始めています。

(3) 問題判別に時間がかかる

Globus Toolkitは、ドキュメント不足などに起因するAPIの使

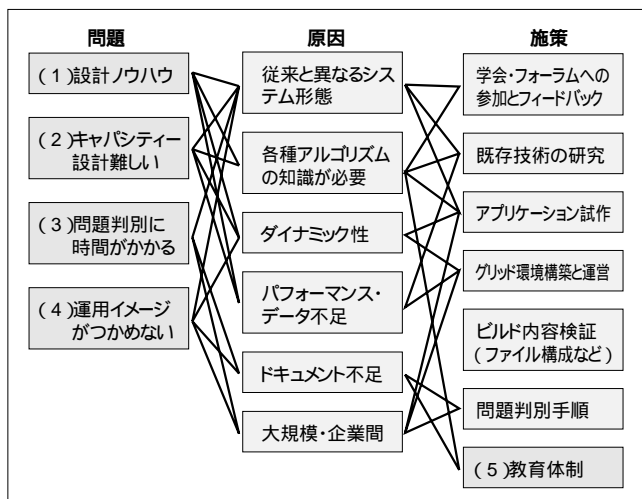


図6. 商用システムへ適用する際の問題と対策

表3. 検証環境

プラットフォーム	OS/環境	ツール・キット	備考
zSeries	Linux	Globus Toolkit2.0	ソースからビルド
iSeries	Linux	Globus Toolkit2.0	ソースからビルド
pSeries	AIX	Globus Toolkit2.0	CD/ソースからビルド
	Linux	Globus Toolkit2.0	ソースからビルド
xSeries/PC	Linux	Globus Toolkit2.0	ソースからビルド

用や問題判別ノウハウ蓄積の難しさがあります。これに対して、ソース・コードを研究した上で、メッセージやコードに基づく問題判別手順の確立を目指しています。これにはアプリケーション試作や実運用ノウハウも反映します。このほかにビルド内容を検証してファイル構成や依存関係を把握することも行っています。

(4) 運用イメージがつかめない

グリッドが今までにない規模のコンピューティングであるために、運用イメージがつかめないという問題もあります。企業をまたがるダイナミック性も考慮しなければなりません。この点については、実際にグリッド環境を構築・運営して、導入と構成から運用まで、大規模構成を意識したノウハウを蓄積することになりました。当グループでは表3の環境を中心に検証テストを実施しています。

正式にモジュールが提供されているのはPC用とAIX[®]用ですが、そのほかの環境でもソースからビルドすることにより稼働します。AIX用においてもCPU利用率などを取得するモジュールは提供されており、現在では遜色なく使用できています。なお、自分たちの環境で単独のグリッド環境を運営するだけでなく、IBM社内の大規模グリッド環境に参加して、規模の大きさに起因する運用ノウハウも共有するようにしています。

(5) 教育体制の整備

最後に、これらを効果的に社内スキルとして広めるために教育体制を整備しつつあります。

4.2. 商用システムへの適用の利点

商用システム適用において、利点として認識した機能を列挙します。

- 異なる組織・管理ドメインを超えたバーチャル組織を構成できます。
- ダイナミックに登録 / 抹消できます。
- セキュアでコーディネートされた環境です。
- ヘテロジニアスな環境(当方ではzSeries[™]、iSeries[™]、pSeries[™]、xSeries[™]/PCで検証)をサポートします。
- よく標準化されています。
- アーキテクチャー的に過去のUNIX[®]環境との親和性が良く、従来持ち合わせてきたライブラリーやツールが活用可能であ

り、クライアント/サーバー・システムからのシームレスな移行が可能で

上記については、既に説明を行ってきたものですから詳細な説明は行いません。世の中の変化、イベントの発生に応じて、必要なものを必要なだけ手に入れることのできるこのモデルの重要性は、今後ますます高まっていくでしょう。

4.3. システム設計における考慮点

次に商用システムへの適用に先立って解決しなければならない問題や、システム設計上の考慮点について考察します。

(1) セキュリティー

Globus Toolkitの提供するセキュリティーは認証・認可・暗号化といった面で非常に強力です。ただし、異なる組織間を接続することを考えると、今後はよりきめ細かな設定やチェックが必要になると思われます。それにより難易性が増すことはセキュリティー・ホールを生み出しかねないので、ポリシー・ベースの安全な仕組みが求められます。この分野でIBMには多くの要素技術や製品、サービスがあるのでうまく統合するべきでしょう。

(2) プライバシーの保護

自分のCPUを他人が使いにくるということに対する精神的な不安は、想像以上に高いものがあります。通常のセキュリティーとは別に、信頼性のあるサンドボックスの必要性が高いでしょう。例えばEntropia社の製品は、デスクトップ上でのジョブ実行環境として、独自のサンドボックス機能を提供しています。しかし、こうした製品が提供しているような機能については、信頼性をどれだけ上げても、最終的には実績が多くなければエンド・ユーザーから信用してもらえないかもしれません。これは一つの提言ですが、サンドボックスとしてLPAR(論理パーティション)やVM(仮想マシン)を利用することは非常に有効です。筆者自身も自分のデスクトップPCを開放するに当たっては、VMwareを導入してその環境を開放しています。この場合、LPAR/VM機能で重要なことは、ゲスト・オペレーティング・システムへの

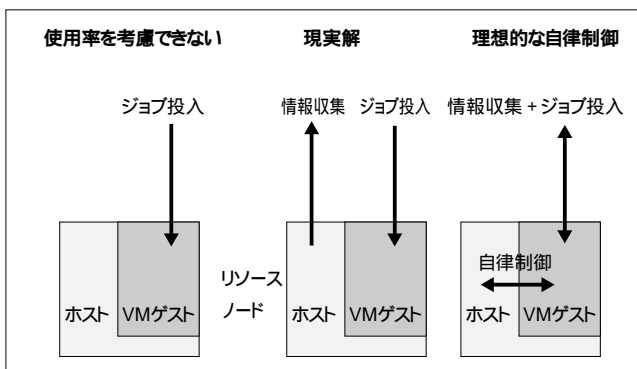


図7. VMによるサンドボックスの実現レベル

CPU割り当てを、いかに効果的に動的に行うかということです。グリッド上でのポリシーに合わせて動作できることが重要です(図7)。

こうした点を考慮すると、ジョブ実行環境としては実はzSeriesが最も優位性が高いことが分かります。その高度なLPAR、VM環境を活用すれば、セキュリティーとしては別のマシンで実行しているのと同様に高く保持でき、リソースを必要なところに有効に配分できる利点があります。

(3) サーバーの管理容易性・可用性

グリッド・コンピューティングのマシン数は非常に多く、システムも広域にわたるため、個々のサーバーの運用容易性と可用性は重要な考慮点になります。グリッドのダイナミックな性格を前提にすると、それらを事前に想定することは困難です。また、マシンの管理も台数に応じて複雑になると思われますが、これを解決せずに運用に入ることではできません。また、他組織の環境をどのように管理するかということも考慮する必要があります。管理に当たってはGUIなどを効果的に使用した一括ビューや仕組みづくりが有効です。

グリッドにおいて、インターネットをコンピューティング・プラットフォームとして活用するには、自律的で知的な機能を持ったインフラストラクチャーとして構築しないと、管理コストが膨大になり、結局は運用不能に陥ってしまいます。IBMはオートノミック・コンピューティングと呼ぶ自己管理機能の提供が必要であると考えています。あたかも人間の自律神経系のように、周囲の状況を自律的に判断して自己を防衛するさまざまな機能を、オープンなソフトウェア上で提供していくことを目指しています。これはノード数が多いグリッドにおいては極めて重要な機能になります。

(4) 現在の製品は相互運用できない

既存のグリッド製品群はそれぞれの環境ではよく動作しますが、相互運用性がありません。グリッドが接続先を広げていく性質のものであることを考えると、この状況は容認できません。乱暴な言い方をすれば、デファクトであるGlobusに対応していない製品は選定対象に入れないなど、ある程度の割り切りが必要だとも考えます。

(5) 高速ネットワークと伝送遅延を前提にした技術

ネットワークが高速化されたとしても、その伝送遅延は内部バスに比較して大きいものです。アプリケーション設計上はその遅延を意識する必要があります。単純な対応としては、ネットワーク通信をまたがる部分の通信を疎に保つことです。第一に考えられるのは、現在主流であるリモート・ジョブ投入のレベルを保つことです。あるいはメッセージ・キューイング手法を主体として、遅延を前提とした非同期モデルで設計することです。こうした方法によらずに密結合の設計をする場合は、遅延を克服する

特殊な技術が必要になるでしょう。例えばButterfly.net社では、ネットワーク上でゲーム環境を構築できるような推論技術を用いたミドルウェアを提供しています。

(6) 資源の発見のための仕組み、資源の市場

実際にグリッドを商用に適用するには、WebサービスにおけるUDDI(Universal Description ,Discovery and Integration)のようなディレクトリー機能の提供をはじめとするサービスの取引市場が必要になります。この際のジョブ投入の概念は、現在よりもユーザー・ニーズに合ったものが登場する必要性も感じます。例えばメルボルン大学では、演算を終えたい時刻や予算を入力すると、ミドルウェアが条件を満たすリソースをグリッド環境の中から発見し、処理を依頼する経済ベースのシステムであるNimrod-Gを研究しています。

(7) 異なるアーキテクチャー間での資源共有

グリッドは異なるアーキテクチャー間でも動作しますが、アプリケーション・モジュールは必要な環境の分だけそろえておく必要があります。このため、環境ごとのモジュール作成を省略できるJava™環境を利用して構築できるかどうかは、初期段階に検討したい事項です。

(8) パフォーマンスとスケーラビリティ

設計ノウハウ不足ということもあるでしょうが、グリッドでパフォーマンスとスケーラビリティを実現するのはそれほど容易ではありません。セキュリティやディレクトリーのサーバーがボトルネックになってしまうことも多く、より一層の設計と運営のノウハウが必要であると考えています。

5. まとめと今後の課題

本論文ではグリッドが脚光を浴びている背景と動向、そして機能とアーキテクチャーについてまとめました。その上で商用システムへの適用を前提とした筆者の所属グループの施策と、その結果として得ることができた知見についてまとめました。

これらはまだ十分な内容に結実しておらず、特に以下の内容については今後の課題と考えています。

- パフォーマンス検証とキャパシティー設計の手法確立。
- 業務プログラマーが容易に扱える開発環境。
- OGSA対応版のソフトウェア検証と、XMLベースでのチューニング手法。

6. おわりに

グリッド・コンピューティングは今後、必要不可欠なインフラストラクチャーとして進化するでしょう。WWWが今日なくてはならないインフラストラクチャーに進化したのと同じように発展すると思われます。適用分野も科学技術計算分野から商用分野まですべてのマーケットで使用されていくでしょう。

グリッド上でサービスを提供するASPや、オンデマンドに資源を提供するデータ・センターが登場して、新しいネットワーク環境が構築されます。キャンパス・研究機関のグリッドから企業内・企業間・パブリックなグリッドまでさまざまなコミュニティーがダイナミックに接続・利用されていくでしょう。

グリッド・コンピューティングがますます進化し、ネットワークがサービスや資源を提供する媒体となれば、サーバーはいつでも利用でき安全で信頼性あることが期待され、アプリケーションはさまざまなサービスを広く提供するために、再利用可能なモジュールとして部品化することが期待されます。

e-ビジネスの技術はあまりに変化が激しく、利用者がすべての技術に対応することは難しくなっています。私たちは技術力を向上する努力を行い、優れたサービスとソリューションを提供することにより、利用者がアプリケーションの実現に集中することができる環境を早期に構築できることを願っています。

(ページ数および表記上の観点から、著者の了解を得て編集部にて手を入れてあります)

[参考文献]

- [1] "The Triumph of the Light - Extensions to fiber optics will supply network capacity that borders on the infinite"/Gary Stix/Scientific American/Jan. 2001
- [2] "The Grid: Blueprint for a New Computing Infrastructure"/Ian Foster and Carl Kesselman/Morgan Kaufmann Publishers ,Inc/1999
- [3] 財団法人情報処理学会『情報処理学会研究報告2002-HPC-91』2002年
- [4] Globus Project: <http://www.globus.org>
- [5] Data Grid Project: <http://eu-datagrid.web.cern.ch/eu-datagrid/>
- [6] Global Grid Forum: <http://www.gridforum.org/>