

# 計算機リソースを有効活用するグリッド ミドルウェア：CyberGRIP

## Grid Middleware for Effectively Using Computing Resources: CyberGRIP

### あらまし

近年、地理的に分散した異質な計算機リソースをネットワークで接続し、あたかも一つの計算機システムのように利用するグリッドコンピューティング技術の研究開発が活発に行われている。これらの研究プロジェクトのほとんどがサイエンス系国家プロジェクトであり、まだ現時点で企業におけるグリッドの実用化事例や成功事例はほとんど報告されていない。

そこで著者らは、グリッドコンピューティング技術を企業における実際のシミュレーション業務に適用し、その有効性を確かめることを目的に、グリッドミドルウェア CyberGRIPを開発した。

本稿では、企業におけるシミュレーションのニーズと問題点を明らかにした後、CyberGRIPの概要について、CyberGRIPがこの問題点をどのように解決しているかに焦点を当てて説明する。また、社内で行ったCAD-Gridシステムでの検証についても触れる。

### Abstract

Various research and development activities regarding grid computing technology have recently been promoted. This technology integrates heterogeneous computing resources that are geographically dispersed and virtualizes them as a single computer system. However, there have been very few reports in the business world about grid computing becoming successful or practical because most grid computing research is carried out in government science projects. Fujitsu has developed CyberGRIP, which is a grid middleware system for applying grid computing to practical corporate simulations and verifying the effectiveness of grid computing. This paper clarifies the needs and problems of these corporate simulations and describes how CyberGRIP is solving the problems. This paper also describes in-house verification using the CAD-Grid system.



安里 彰（あさと あきら）  
ITコア研究所ITアーキテクチャ研究  
部 所属  
現在、オーガニックコンピューティ  
ング技術の研究開発に従事。



門岡良昌（かどおか よしまさ）  
ITコア研究所 所属  
現在、グリッドコンピューティング  
技術の研究開発に従事。

## まえがき

近年、計算機の高性能化およびネットワークの広帯域化を背景に、地理的に分散した異質な計算機リソースをネットワークで接続し、あたかも一つの計算機システムのように利用可能とするグリッドコンピューティング技術の研究開発が活発に行われている<sup>1)</sup>。これらの研究プロジェクトのほとんどがサイエンス系の国家プロジェクトであり、まだ現時点では企業におけるグリッドの実用化事例や成功事例といったものはほとんど報告されていない。

そこで著者らは、幅広いグリッドのアプリケーション分野の中でも、とくに原点とも言うべき計算機パワーの仮想化に焦点を当ててグリッドの実用化研究を行い、主としてシミュレーション業務を対象に、企業内における高度な計算機利用を可能にするためのグリッドミドルウェアCyberGRIP（CyberGRid Innovation Platform）を開発した。

本稿では、研究の背景となる企業におけるシミュレーションのニーズと問題点について述べた後、CyberGRIPの構成および主要なコンポーネントの動作概要を説明する。最後に、実業務への適用例として、次世代移動通信システム開発のために構築したCAD-Gridシステムについても述べる。

## シミュレーション現場の現状と問題点

シミュレーション技術の発達に伴い、大学や各種研究機関だけではなく、企業内でも実際のビジネス活動に役立てるために様々なシミュレーションが実施されている。例えばプロセッサ開発の現場では、論理シミュレーション、セルの配置配線の最適化、遅延計算、消費電力計算といった多くのシミュレーションを行う必要がある。デバイスの進化に伴って計算が複雑化する一方で、開発期間の短縮やコスト改善というマーケットの要求もますます厳しくなる状況から、これらのシミュレーションを高速かつ効率的に行うニーズは増加の一途をたどっている。

このような状況は製造業だけでなく金融分野においても同様で、取引の決済期間の短縮による顧客サービスの向上や、各種資産や先物取引における時価評価、さらにはグローバルリスク管理など、至る所で大量のシミュレーションが必要とされている。

さらに流通業においてもデータマイニングによる

マーケティング戦略立案のために大量計算が行われているなど、シミュレーションが必要な分野は広範であり、今後もビジネスの様々な分野で計算機を用いたシミュレーションの利用が拡大することは明らかである。

このように各分野で様々なシミュレーションが実施されているが、詳しく調べてみると、現場ではいろいろな問題を抱えていることが分かってきた。まず製造業においては、シミュレーションを行う計算機の空き状況や投入したシミュレーションの進捗状況を人が管理しながら作業を進めているため、十分な計算機の利用効率が得られないケースが多い。また、大量シミュレーションの一部の入力データに不適切な値を入れてしまった場合に、すべての計算が終了するまでその誤りに気付かなかつたり、ある段階で気付いたとしても、その誤入力に関連するジョブ（一般的には複数存在）を特定することが難しいため、すべてのパラメタを再度入力して同じ時間を掛けてシミュレーションを再実行させたりすることが起こっている。結果として、シミュレーションに十分な時間が割けず、エラーを残したまま後工程に進んだため、LSIのリメイクにつながるなど、大きな損失を招く場合もある。このように、いかにして大量のシミュレーションを効率的に実施するかということが重要な課題となっている。

また金融業においては、最新の金融工学を活用すれば、リスク管理などのシミュレーションの精度をもう一桁向上できることが分かっているが、必要な計算機リソースを確保できないため、要求された精度を得ることができずにビジネスチャンスを失ってしまうことが起こっている。

一方で、企業内の計算機リソースは必ずしも効率良く利用されていないことも分かってきた。一つにはサーバを部門ごとに購入して管理するため、ある部門のマシンが比較的空いていても、他部門からは利用することが難しいといった状況がある。また、業務用パソコンの能力は年々高まっているが、主として資料作成などCPUをあまり使わない用途に利用される上に、夜は眠ってしまうことが多い。

まとめると、大量のシミュレーションを効率的に実施することが課題となっている一方で、企業内の計算機リソースが十分に活用されていない現状がある。ということは、これらのリソースをうまく使う

ことができれば、業務効率が大きく改善し、問題点の多くを解決あるいは緩和できる可能性がある。著者らはここを出発点としてCyberGRIPの研究開発に取り組んだ。

### CyberGRIPの開発

前述のように大量のシミュレーション業務を効率的に実施するためには、計算機リソースの状態やジョブの実行状況を人が監視することなく大量のジョブを効率的に投入する仕組みと、企業内の可能な限り多くの計算機リソースを最大限に有効活用する仕組みを作ることが課題である。この課題を解決するために著者らはグリッドミドルウェアCyberGRIPを開発した。以下にCyberGRIPについて述べる。

#### 全体アーキテクチャ

CyberGRIPの全体アーキテクチャを図-1に示す。

CyberGRIPは、UNIX系のポータルサーバ上で動作し、主に以下に述べる三つのコンポーネントから構成されている。まず利用者が投入したジョブの実行制御を行うOJC (Organic Job Controller)につぎにOJCから渡されたジョブに対して最適な計算機リソースを決定しジョブを投入するGRM (Grid Resource Manager),そして計算機リソースの状態監視や通信を管理するSRM (Site Resource

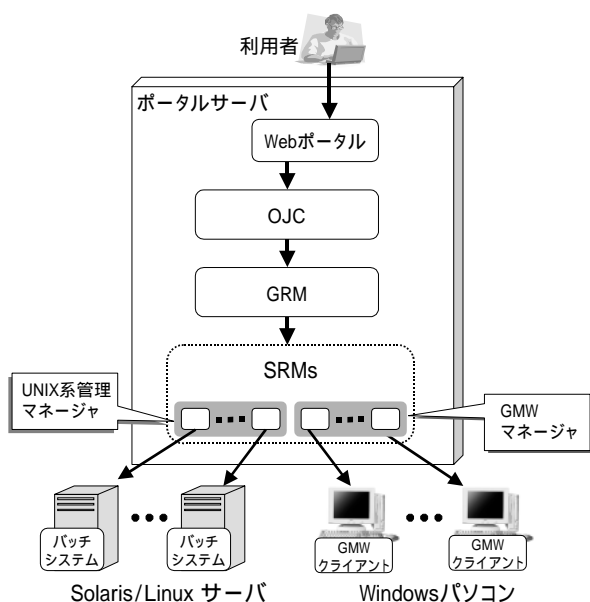


図-1 CyberGRIPのアーキテクチャ  
Fig.1-Architecture of CyberGRIP.

Manager)である。SRMは個々の計算機リソースごとに存在し、とくにWindowsパソコンに対応するものをGMW (Grid Mediator for Windows) マネージャと呼んでいる。

一方、計算機リソース側にもSRMと通信してジョブを実行するためのミドルウェアが必要である。SolarisやLinuxをベースとするUNIX系マシンの場合は、このミドルウェアとしてCondor<sup>®</sup>などの一般的なバッチシステムを使用することができる。それに対して、Windowsパソコンを利用する場合は、GMWマネージャに対応するGMWクライアントをパソコンにインストールする必要がある。

このように、CyberGRIPにより、SolarisやLinuxマシンのみならず通常利用しているWindowsパソコンを含めて仮想化された計算機リソースに対して、個々の計算機の性能や特性を意識することなく、利用者がポータルサーバのWebポータルを通じてジョブを投入可能な環境が構築される。

#### ジョブ実行制御：Organic Job Controller

大量にジョブを実行する場合、個々のジョブ間には、例えば先行ジョブの結果が後続ジョブのパラメータに反映されるなどの依存関係がある場合が多い。これらの依存関係に従ってジョブの実行をスケジューリングすることにより、全体的な効率が向上する可能性がある。とくにグリッドコンピューティング環境では、各ジョブが独立したリソース上で非同期的に終了することが多いので、この効果は大きい。また前章でも触れたように、大量シミュレーションにおいてはある程度の入力パラメータ設定ミスがどうしても避けられないが、ジョブ間の依存関係が明確になっていれば、設定ミスしたジョブに依存するものだけを再実行すればよいので、再実行するジョブ数を最小限に抑えることができる。

これらの考察に基づき、ジョブ間の依存関係を適切かつ柔軟にハンドリングする仕組みとして、著者らはOJCを開発した。OJCの基本構造を図-2に示す。OJCの利用者は、OJCスクリプト { 図-2に例示したような構文でジョブ間の依存関係を簡潔に表現したもの。詳しい文法については参考文献(2)を参照されたい } を作成してOJCインタプリタに与えればよい。OJCインタプリタはOJCスクリプトを解釈し、ジョブ間の依存関係を表現するツリー構造を構築する。ジョブ実行時にはツリー構造に

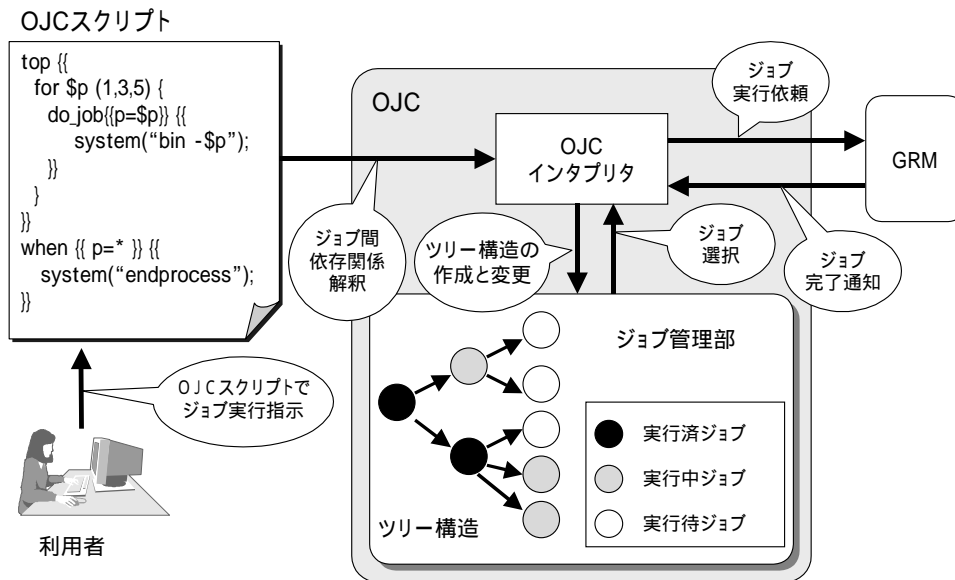


図-2 OJCの基本構造  
Fig.2-Basic structure of OJC (Organic Job Controller).

従ってOJC内部で同期待合せが行われ、実行可能になったジョブが適宜ジョブ実行部 (GRM) に渡されていく。この仕組みにより、ジョブが非同期に完了するグリッドシステムにおいても、最適なジョブ実行制御が可能となった。

また、OJCにはパラメタ設定ミスなどが発生した場合にジョブを再実行するためのインタフェースも用意されている。これを利用して当該ジョブの再実行を指示すれば、依存関係のあるジョブ群もツリー構造に従って自動的に再実行されるので、利用者は個々のジョブについて再実行を指示しなくてよい。これらの機能により、前章で挙げた大量シミュレーションにおける課題の多くが解決または軽減されることが分かる。

上述の機能に加えてOJCが備えている動的ジョブ制御機能を図-3に示す。これは、実行すべきジョブの個数が初期投入時に静的に決められない場合に効果を発揮する。具体的な例としては、不定サイズの入力ファイルを分割処理する場合や、パラメタの最適化を行う場合など、ファイルのサイズや最適化の状況によって実行すべきジョブ数が増える場合が挙げられる。従来は、このような制御を自動的に行うことが困難だったため、実行する可能性のあるジョブを無駄を承知ですべて投入する、または先行ジョブの結果を逐次人間が見て後続ジョブを投入するか否かを判断するなど、非能率的なジョブ実行を

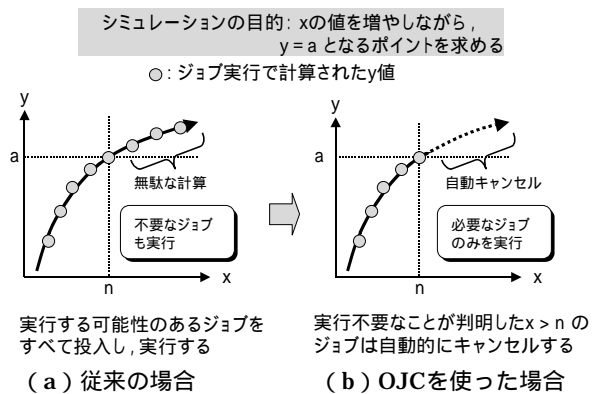


図-3 OJCの動的ジョブ制御機能  
Fig.3-Dynamic job control of OJC.

せざるを得なかった。しかし、OJCではこの判断部分が自動化されたため、ジョブ実行効率を飛躍的に高めることができた。

資源管理とジョブ管理：GRM, SRM, GMW

GRMはOJCから実行依頼されたジョブを実行するのに最適な計算機リソースを選択し、ジョブを投入する役割を持つ。OJCから渡される情報にはプログラム名や入力パラメタ以外にジョブの属性が含まれている。ジョブの属性としては、Pentium, SPARCなどのプラットフォームやWindows, LinuxなどのOSの指定の有無、予想される実行時間、リソース選択ポリシーなどがある。GRMはSRMを通じて配下にある計算機リソースの動作状

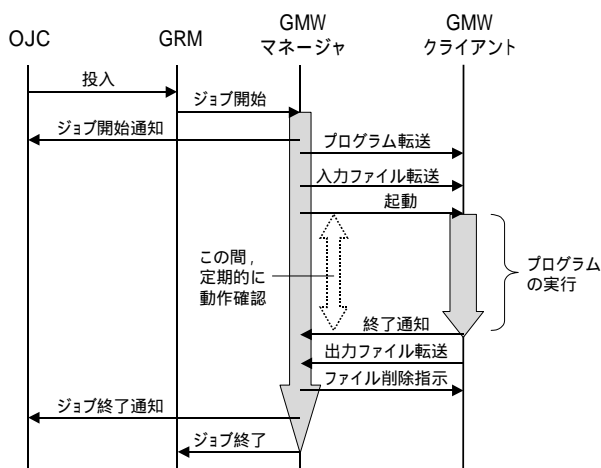


図-4 ジョブ実行シーケンス (Windows/パソコン)  
Fig.4-Job executing sequence (Windows Personal Computer).

況を常に監視し、監視結果と渡されたジョブ属性を突き合わせて、その時々々に最適なリソースを割り当てていく。リソース選択ポリシーとしては、現在の開発版ではなるべく速いマシンを選ぶなど単純なものしかサポートしていないが、より複雑な制御ができるように、今後拡張していく予定である。

なお、プログラムのバイナリは、実行する計算機リソースが割り当てられたときに、ポータルサーバから配信される。したがって、プラットフォームやOSが異なる計算機リソースで同一のプログラムを実行させる場合は、あらかじめ複数種類のバイナリをポータルサーバ上に用意しておく必要がある。

SRMは計算機リソース上でのジョブ実行を管理するミドルウェアで、個々の計算機リソースごとに存在するが、ここでは特に計算リソースがWindowsパソコン（以下、パソコン）の場合に絞って説明する。ジョブがパソコンで実行される手順を図-4に示す。パソコンにかかわるSRMは、サーバ側で動作するGMWマネージャとパソコン側で動作するGMWクライアントで構成される（図-1）。GRMがジョブをパソコンに割り当てることを決めると、直ちにGMWマネージャに通知され、GMWマネージャはGMWクライアントに対してプログラムと入力ファイルの転送を行うとともに、プログラムの起動を指示する。GMWクライアントは、この指示を受けると直ちにバックグラウンドでジョブを実行し、完了した段階でジョブの結果を出力ファイルとしてGMWマネージャへ転送する。この間、

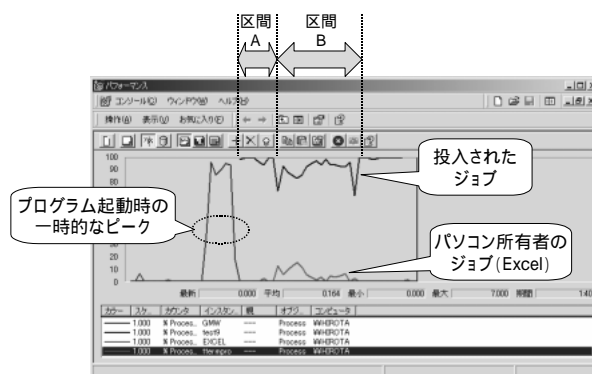


図-5 ジョブ実行中のパフォーマンスモニタ  
Fig.5-Performance monitor under executing jobs.

ジョブが確実に実行されていることを確認するためにGMWマネージャからGMWクライアントに対して、一定周期でジョブ実行の確認メッセージが送信される。このメッセージに対する応答が返らなければ、GMWマネージャはパソコンに何らかの異常が発生して処理が途中で放棄されたとみなし、新たなパソコンを選んでジョブを再度実行させる。

CyberGRIP配下のパソコンでジョブが実行される場合、投入されたジョブは最も優先度が低く設定されており、パソコンの所有者の作業に影響を及ぼさない仕組みとなっている。計算機リソースとして用いられているパソコンのパフォーマンスモニタを図-5に示す。最初のピークはGMWマネージャからプログラムが投入されてパソコン側で起動されたことによりCPU使用率が一瞬高くなっていることを示す。その後も所有者がパソコンを利用していない間は、投入されたジョブがほぼ100%CPUを利用している（区間A）。ところがその直後に所有者が表計算プログラムExcelを利用し始めたため、その処理に必要なCPU能力を明け渡していることが図-5から分かる（区間B）。

#### CAD-Gridシステムへの適用

冒頭でも述べたように、著者らは企業におけるグリッドの実用化事例、成功事例を示すことが重要と考えている。そこで、そのための第一歩としてCyberGRIPを社内業務に実際に適用し、効果の検証を試みた。この目的で構築したのがCAD-Gridシステム<sup>(4)</sup>である。

CAD-Gridシステムでは、社内にあるPCクラスターやUNIXサーバ、さらに社内ネットワークに接続

された事務用デスクトップパソコンを束ね、仮想的な計算機リソースとして使用した。

詳細は参考文献(4)に譲るが、CAD-Gridシステムを基地局、移動機の開発部門の移動通信システムシミュレーション業務に適用した結果、シミュレーション期間を約1/4に短縮するとともに、作業工数を約1/3に削減するという、非常に大きな成果が得られた。この成功により、CyberGRIPのアプローチが、前に述べた大量シミュレーションにおける課題の解決に有効であることが示されたと考えている。また、グリッドの実用化事例、成功事例を作り上げるといったプロジェクトの目的も、予想以上の結果を伴って達成することができた。

### む す び

本稿で述べたように、CyberGRIPは第一歩としては予想以上の成功を収めることができた。しかし、CAD-Gridシステムでの検証を通じて、多くの改善すべき点が明らかになってきている。例えば、計算機リソースを割り当てる仕組みが現状では単純すぎるため、緊急のジョブを優先的にさばいて欲しいといった要求に対応できない。また、各処理が必ずしも最適化されていないため、扱うジョブ数が増えるにつれてオーバヘッドが無視できなくなることも分

かってきた。さらに、これらの機能的、性能的な問題に加えて、CyberGRIPのセキュリティ強化も大きな課題となっている。

今後はこれらのポイントを改善し、CyberGRIPの品質強化に取り組んでいく予定である。社内では移動通信システムシミュレーションだけでなく、より多くの業務アプリケーションに適用し、業務効率の一層の改善を図る。また、関連部門と協力してCyberGRIPのビジネス化も推進していきたいと考えている。

### 参 考 文 献

- (1) I. Foster and C. Kesselman : The GRID: Blueprint for a New Computing Infrastructure . Morgan Kaufmann , SC-18 , 1999 .
- (2) 上田晴康ほか：ジョブ投入と待ち合わせを記述できるスクリプト言語オーガニックジョブコントローラ - CAD-Gridへの適用 - . 第6回問題解決環境論文集 , 2003 , p.37-42 .
- (3) Condorホームページ . <http://www.cs.wisc.edu/condor/>
- (4) 中村武雄ほか：製品開発を加速するCAD-Gridシステム . FUJITSU , Vol.55 , No.2 , p.121-126 ( 2004 ) .