RCSS ディスカッションペーパーシリーズ ISSN-1347-636X
第 79 号　2009 年 2 月
Discussion Paper Series
No.79　February, 2009

仮想計算機を用いたＰＣグリッドの開発

榎原博之

森川浩明

RCSS

文部科学大臣認定　共同利用・共同研究拠点
関西大学ソシオネットワーク戦略研究機構
関西大学ソシオネットワーク戦略研究センター
（文部科学省私立大学学術フロンティア推進拠点）

Research Center of Socionetwork Strategies,
“Academic Frontier” Project for Private Universities, 2003-2009
Supported by Ministry of Education, Culture, Sports, Science and Technology
The Research Institute for Socionetwork Strategies,
Joint Usage / Research Center, MEXT, Japan
Kansai University
Suita, Osaka, 564-8680 Japan
URL: http://www.rcss.kansai-u.ac.jp
http://www.socionetwork.jp
e-mail: rcss@jm.kansai-u.ac.jp
tel: 06-6368-1228
fax. 06-6330-3304

仮想計算機を用いたＰＣグリッドの開発

榎原 博之* 森川 浩明†

2009年2月

概 要

本研究は、仮想計算機の異質なハードウェア環境で動作可能である特徴に着目し、並列計算においてユーザが計算機を利用する時には、別の計算機に仮想計算機ごと計算内容を移行できるマイグレーション機能を実装したグリッドシステムの開発をおこなう。また、仮想計算機を利用したマイグレーションにかかる時間は、投入ジョブが利用するメモリサイズで決定されるため、メモリサイズを考慮したジョブ投入を検討する。性能評価により、仮想計算機を用いてもホストOSとほぼ同じ計算能力であることを示す。

Keyword: グリッド，PCクラスタ，仮想化，システム構築，最適化

*関西大学 システム理工学部、関西大学 ソシオネットワークソシオネットワーク戦略研究センター研究員
†大阪市立大学 創造都市研究科

Development of PC Grid with Virtual Machine

Hiroyuki Ebara[1] and Hiroaki Morikawa[2]

February, 2009

Abstract

In this paper, we propose an implementation of a grid system with migration function. If a person use the PC running a grid job, the migration function moves it to another idle PC. Furthermore, the time of migration with virtual machine depends on the memory size. So, we propose a method to carry out jobs in considering of the memory size. By performance evaluation, even if it is carried out on a virtual machine, it is shown that it is the almost same ability as on a host OS.

Keyword: Grid Computing, PC Cluster, Vertualization, System Construction, Optimization

[1]Faculty of Engineering Science, Kansai University, Researcher, The Research Center of Socionetowrk Strategies
[2]Graduate School for Creative Cities, Osaka City University

1　はじめに

本稿執筆時の 2009 年 3 月現在、計算機の演算能力・通信機能の発展がめざましく、家庭用計算機複数台で 10 年前のスーパーコンピュータに匹敵する計算能力を発揮できる。また、遺伝子解析や素粒子物理学などの研究分野で大規模な計算を必要とする問題が増大している。このような背景から高速なネットワークを介して LAN や WAN 内に散在する家庭用計算機をつなぎ、PC クラスタシステムやグリッドシステムを構築することに注目が集まっている。

グリッドシステムは、広域ネットワーク上に存在する CPU、メモリ、ストレージ、センサなどの資源を仮想化・統合するインフラシステムである。このなかで、注目されているもののひとつに PC グリッドがある。PC グリッドでは、家庭用計算機が高性能であるにもかかわらずその性能をフルに発揮していない点を考慮し、これらの計算機の有効活用を目的として、大規模なマシンパワーを発揮するシステムを構築することを目指している。PC グリッドシステムとして考えられている SETI@home[16] プロジェクトに代表されるようなプロジェクトでは、計算機があまり利用されていない遊休時間に計算ジョブを起動するシステムであり、ジョブ間の通信が必要ない問題にしか適用できないため計算できる問題が限定されている。加えて、これらのシステムでは、あらかじめ計算機が起動しているものと考えられており、シャットダウン状態の計算機にジョブを投入するシステムはない。また、キャンパスグリッドという大学などの計算機室の計算機を夜間に利用するシステムも存在するが、夜間に限定されるため長時間の実行に向かない。

一方、仮想計算機はサーバなどの計算資源の有効活用を図る方法として知られている。サーバなどの計算機では繁忙期であっても CPU 使用率やメモリ利用率が 100%に達することは無い。こういった状況から計算機の CPU やメモリなど利用する計算資源を分割し、仮想的に指定した範囲の計算資源を利用する計算機を複数構築できる仮想計算機技術が注目されてきている。仮想計算機には、1 台の計算機に複数の仮想計算機を設置できる特徴のほかに、プログラムが計算機のハードウエアに依存しない特徴がある。このため、仮想計算機を別計算機に移行させるマイグレーション機能を実装できる。

本研究では、キャンパスグリッドを想定し、計算機室には常にユーザが存在する環境で長時間計算ジョブを実行するため仮想計算機を用いたマイグレーション機能を実装し、グリッドシステムの構築をおこなう。マイグレーション機能によって計算機室内でユーザが利用を開始すると計算をおこなっている仮想計算機をサスペンドさせ、計算内容を他の計算機にマイグレーションすることで演算を途切れることなく実行でき、長時間のジョブ実行ができる。さらに、休止している計算機を見つけ、ネットワークを介して起動させる機能も備わっている。また、計算機室はおおまかな使用状況がわかりやすくスケジュールを立てやすい。マイグレーションにかかる時間は演算にかかる時間よりも小さいものであるが、効率的なジョブ投入のためには、計算機室内の利用状況から計算ジョブのスケジューラが各計算機への投入ジョブの演算時間を決定しなければならない。そのため、本研究ではマイグレーションを考慮してジョブ投入プログラムのサイズを変更することで各計算機への投入ジョブの演算時間を変更し、各計算機の予想されるログオフ期間（休止時間）に収め、効率的なジョブ投入を行うように考慮する。

以下、2 章では本研究と関連する研究を紹介し、準備として 3 章で仮想計算機を、4 章で PC グリッドシステムを説明する。5 章では構築したグリッドシステムについて説明を加える。6 章では構築したグリッドシステムの性能評価をおこない、7 章では提案したメモリ分割法に対して実験をおこない、その結果を考察する。実験・評価結果から得られた考察について 8 章でまとめる。

2　関連研究

仮想計算機のマイグレーション機能をグリッドシステムに応用した研究として、立薗らの研究 [19] がある。立薗らの研究では、仮想計算機のライブマイグレーション機能を利用し、投入ジョブ実行中の計算用 PC のジョブキュー内に複数のジョブが存在するとき、他の遊休状態にある計算機に投入ジョブの一部をマイグレーションさせ負荷分散をおこなう。この研究では、キャンパスグリッドを想定し、仮想計算機のサス

ペンド機能を用いたマイグレーションを実現している。しかし、オープン利用時の効率的な運用を想定していない。本研究では、ユーザの利用が多い昼間でも効率を下げず運用できるようジョブ投入を改良することを目的としている。

民間企業において、全社的に遊休計算機の有効活用を図った研究として、中部電力の曽山らの研究 [17] がある。曽山らがおこなった研究では、グリッドシステムの効率運用のため、週間の電源投下状況をジョブ投入スケジュールとし、実際の環境へジョブ投入をおこなった。この研究では、起動している計算機を監視しているため利用時間の予測が容易である。また、ヘテロ環境を想定しているため、ジョブ終了時間のばらつきが大きくなることから、最適なジョブ分割に関する考察がなされている。しかし、この方式のグリッドシステムでは、夜間にジョブ投入できない、ジョブ分割が難しい計算に向かない、負荷増大で通常業務が圧迫されるなどの問題がある。さらに、互いに独立なジョブしか扱えないため、通信が必要なジョブは計算できない問題がある。本研究では、電源が切れている状態の計算機に WakeOnLAN パケットを投げることでジョブ投入可能状態にすることに加え、ジョブ分割が難しい問題に対して消費するメモリでジョブ分割をおこない、投入ジョブの細分化をおこなっている。さらに、非同期の通信機能を備え、通信が必要なジョブも扱えるようになっている。

他に、関連先行研究としては、仮想計算機を用いたグリッドシステム構築にかかわる研究 [15]、マイグレーションにかかわる研究 [7]、[3]、複数大学間でのグリッドシステム構築にかかわるプロジェクト [4]、グリッドシステムのセキュリティにかかわる研究 [23]、[10] などがある。

3 仮想計算機

3.1 仮想計算機

仮想計算機は、計算機上にメモリや CPU、通信回線などを仮想的に構築し、単一の計算機（ホスト OS）上で仮想的に複数の計算機（ゲスト OS）が動作しているかのように見せかけることのできる技術である。代表的な仮想化ソフトとしては、VMWare[21] や Xen[22]、Jail[8] などがある。

仮想計算機は、仮想計算機イメージ (VM イメージ）と仮想化層（仮想化ソフト）、ハードウエアから構成されており（図 1)、一般的に仮想計算機は仮想化層を通して間接的にハードウエアを操作している。

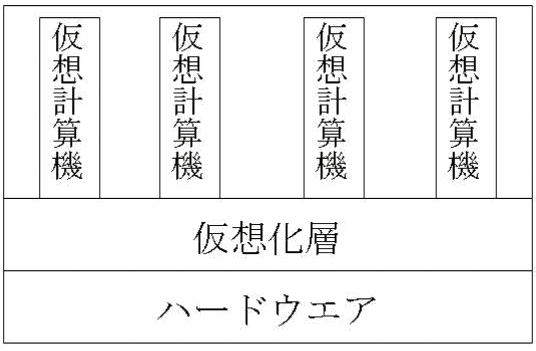

図 1: 仮想計算機の構造

仮想計算機では、仮想化層によってハードウエアから切り離させているために以下のような特徴を備えている。

- 複数の仮想計算機を起動できる

  仮想化層によるハードウエアの排他的な使用によって 1 台の計算機が複数台の計算機であるかのように見せかけられる。サーバなどでは、資源の有効活用のため導入されている。

- 異質なハードウェア環境で動作可能である

  ハードウエアの操作は仮想化層でおこなわれるため、仮想計算機にはハードウエアの影響が少ない。このため、異なるハードウエア環境にプログラムを転送しても仮想化ソフトが同じであるならば動作可能である。

- ホスト OS からの独立性

  仮想計算機はホスト OS に関係なく動作可能であり、一部の仮想計算ソフトではホスト OS が存在しない環境であってもゲスト OS が起動できる。

3.2 マイグレーション機能

代表的な仮想化ソフトに備わっているマイグレーション機能は、現在ある計算機上で実行中の計算内容や計算環境を別の計算機に移行させる機能である。マイグレーション機能は移行するデータなどによって以下のように分類できる。

1. チェックポイントマイグレーション

   一定期間、もしくは特定のアクションごとに現在実行中の状態（メモリ内容、レジスタ内容など）を保存（チェックポイント）し、障害発生時などに別計算機で保存した状態を展開する手法

2. プロセスマイグレーション

   メモリ内容などのデータを移行させるのではなくプロセス自体を別の計算機に移行させる手法

3. ライブマイグレーション

   計算機を停止させずメモリ内容などを少しずつ別計算機に移行させることで、外見上はある計算機が計算を停止したと同時にその計算機で実行していた内容を別計算機が引き継いで実行しているように見せかけることができる手法

現在、マイグレーション機能は主に仮想計算機によって実現されており、本研究でもマイグレーション機能の実装に VMware Server を利用している。

4 PC グリッドシステム

4.1 PC グリッドシステム

グリッド協議会 [9] の定義では、「グリッドは、広域ネットワーク上の計算、データ、実験装置、センサー、人間などの資源を仮想化・統合し、必要に応じて仮想計算機（Virtual Computer）や仮想組織（Virtual Organization）を動的に形成するためのインフラ」とされている。『グリッド』のもともとの意味は電力線、格子を意味しており、ネットワークにつなげばシステムに参加するすべての計算機がその計算能力の恩恵に授かることができることを目的としている。このため、大学内でのグリッドシステム（キャンパスグリッド）や家庭内でのグリッドシステム（PC グリッド）も広義の意味でのグリッドシステムと定義できる。グリッドシステムでは、一定の計算能力を発揮するクラスタシステムと異なり、ネットワーク上に存在する計算機資源を確保する。

本研究では、キャンパス内の遊休 PC を利用したコンピューティンググリッドをおこなっている。PC グリッドでは、計算機の所有者から計算機資源を貸与されている関係にあるので、スケジューリングによる効率的な資源管理や資源にアクセスするための認証などが必要になる。本研究では、提供される各計算機にユーザが存在する場合を想定し、ユーザが計算機を利用していないときのみジョブを投入する。さらに、ユーザの計算機使用頻度からスケジューラが配付する投入ジョブのサイズを変更することで、予測される使用可能期間に可能な限り処理をおこなう。

4.2 PCグリッドが備えるべき機能

PCグリッドが備えるべき機能としてスケジューリング、ユーザビリティ・障害対策、セキュリティなどがある。

本稿ではこれらの機能のうちスケジューリングとユーザビリティ・障害対策について述べる。

・ スケジューリング機能

グリッドシステムの投入ジョブのスケジューリング機能は、各計算機の遊休時間を把握できるか否かが問題になる。

例えば、SETI@homeでは計算機の遊休状態を把握していないため、ある計算機にファイル転送した後、ユーザの資源開放要求があると現在の実行状態をジョブ管理ノードに転送するため、投入ジョブに含まれる複数のファイルが使用されずファイル転送が無駄になることもある。大規模なシステムでは、すべての計算ノードの使用状況把握は困難であるが、ドメインごとに大まかな使用状況を把握し、ジョブ投入スケジュールを作成できることが望ましい。スケジューリング機能を活用し効率的なジョブ投入をおこなうためには、一定時間ごとに計算機の状態を把握し、ジョブ投入時予測される各計算機の遊休時間に収まるようにジョブ分割をおこなう必要がある。

また、計算機環境がヘテロ環境であるかどうかも考慮しなければならない。ヘテロ環境では低スペックの計算機に終了時間が影響を受ける。低スペックの計算機が、終了時間に影響を及ぼさないためにはジョブ分割数を多くすることが必要である。しかし、ジョブ分割数が多いと通信オーバーヘッドが高くなり、通信オーバーヘッドとジョブ分割数のトレードオフになる。

・ ユーザビリティ・障害対策

PCグリッドの各計算機にユーザが存在する環境では、ユーザが資源解放を要求したとき即座に状態保存と資源解放を実行しなければならない。また、障害発生時にも高速な状態保存と資源解放が必要になる。この実現方法としてマイグレーション機能の実装がある。マイグレーション機能は、現在ある計算機上で実行中の計算内容や計算環境を別の計算機に移行させる機能である。現在、マイグレーション機能の実装は仮想計算機を活用したものが主流であり、本研究でもユーザが計算機の使用を開始したというアクションを検知し、仮想計算機のサスペンド機能を用いてチェックポイントマイグレーションを実装している。

5 グリッドシステムの開発

5.1 Systemwalker Cyber GRIPの概略

本研究では、高性能並列演算環境を提供するグリッドミドルウェアとして、富士通株式会社のグリッドミドルウェア製品Systemwalker Cyber GRIPを採用する。そして、これをベースに富士通研究所が開発したジョブマイグレーション機能を統合したシステムを利用している。

Systemwalker Cyber GRIPのジョブスクリプトは独自の記述方法を採用している。しかし、perlライクな記述であり、パラメータスイープなジョブの実行スクリプトが容易に記述できる。また、ジョブを処理する計算機の構成に柔軟に対応するために、Systemwalker Cyber GRIPは次の2種類のキューを持っている。

1. 仮想キュー

   計算用PCを仮想的にひとつの計算機に統合した際、基準となる計算機（マスタサーバ）に存在するキューで、投入されたジョブを計算用PCの実行キューに振り分ける。

2. 実行キュー

   各計算用PCに存在するキューで、投入ジョブの実行する。

ジョブの実行ファイルや入出力データファイルは、Systemwalker Cyber GRIPのファイル転送機能を使い、計算用PCに転送して実行することができる。

5.2 システム概略

構築したグリッドシステムは関西大学と株式会社富士通研究所が共同で設計し、富士通研究所が実装したシステムでマスタサーバ 1 台と計算用 PC 7 台を用いて、サーバ・クライアントからなるスター型のネットワーク構造を成している（図 2)。また、これらシステムで利用している計算用 PC にはそれぞれ使用者が存在し、研究や業務に使用している。本システムではユーザが存在する環境において効率的なジョブ投入をおこなうため、VM 管理テーブルを利用したジョブ管理機能と仮想計算機によるマイグレーション機能を実装している。

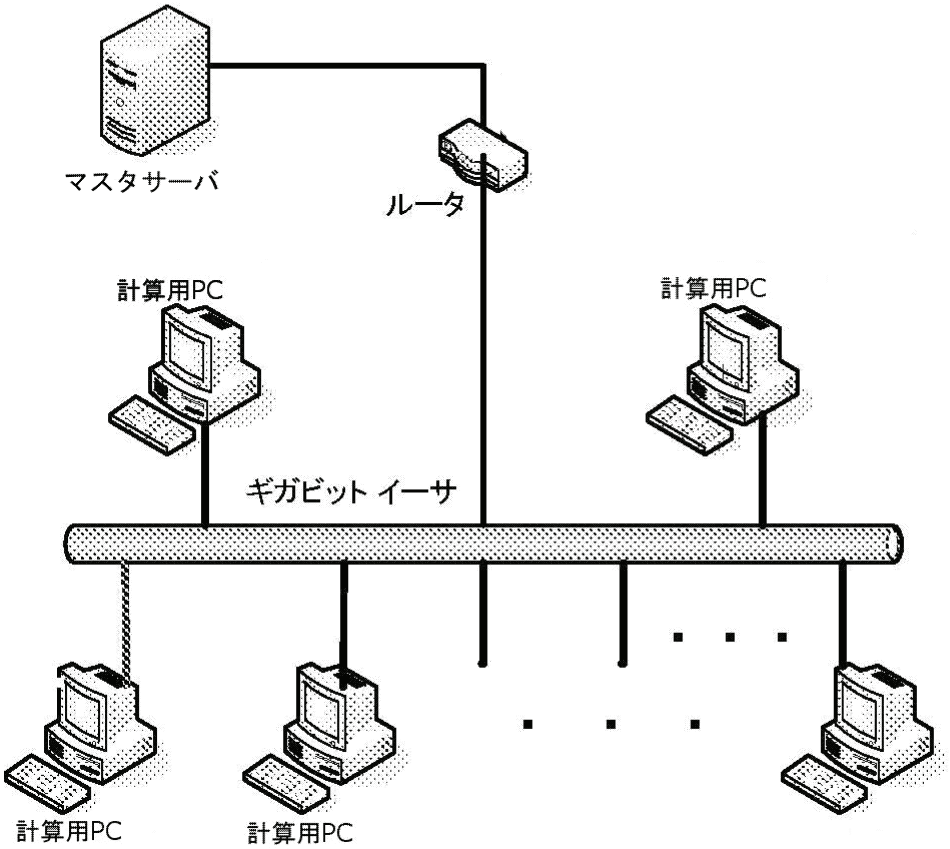

図 2: 構築システムの全体図

グリッドシステムでは高速にマイグレーション機能を利用するため、VM イメージをコピーして送信するのではなく、マスタサーバが保持している VM イメージを各計算機がネットワーク上の共有ファイルとして直接アクセスする。

構築したグリッドシステムは図 3 の構成になっており、以下の機能を持つ。

- マスタサーバ
  - 計算用 PC 管理機能

    計算用 PC の利用状況管理機能から利用状況変更通知を受け、計算機の利用状況を更新する。必要に応じてチェックポイントとリスタート命令を計算用 PC のチェックポイントリスタート管理機能に送る。
  - VM イメージ共有機能

    計算用 PC で利用する VM イメージを保持し、VM イメージを利用している計算用 PC のホスト名と投入ジョブ ID を関連付け、計算資源管理をおこなう。

- 計算用 PC
  - 利用状況管理機能
    ユーザのログオン・ログオフを監視し、その結果をマスタサーバの計算用 PC 管理機能に通知する。
  - チェックポイントリスタート管理機能
    マスタサーバの計算用 PC 管理機能からの要求に応じてジョブの起動・停止をおこなう。
  - 子ジョブ生成管理機能
    マスタサーバからジョブ実行要求を受け、仮想計算機の起動とチェックポイントをおこなう。
- ゲスト OS
  - 子ジョブ実行機能
    計算用 PC の子ジョブ生成管理機能から子ジョブ実行要求を受け取ると子ジョブを実行する。実行完了後、子ジョブ生成管理機能に結果を通知する。

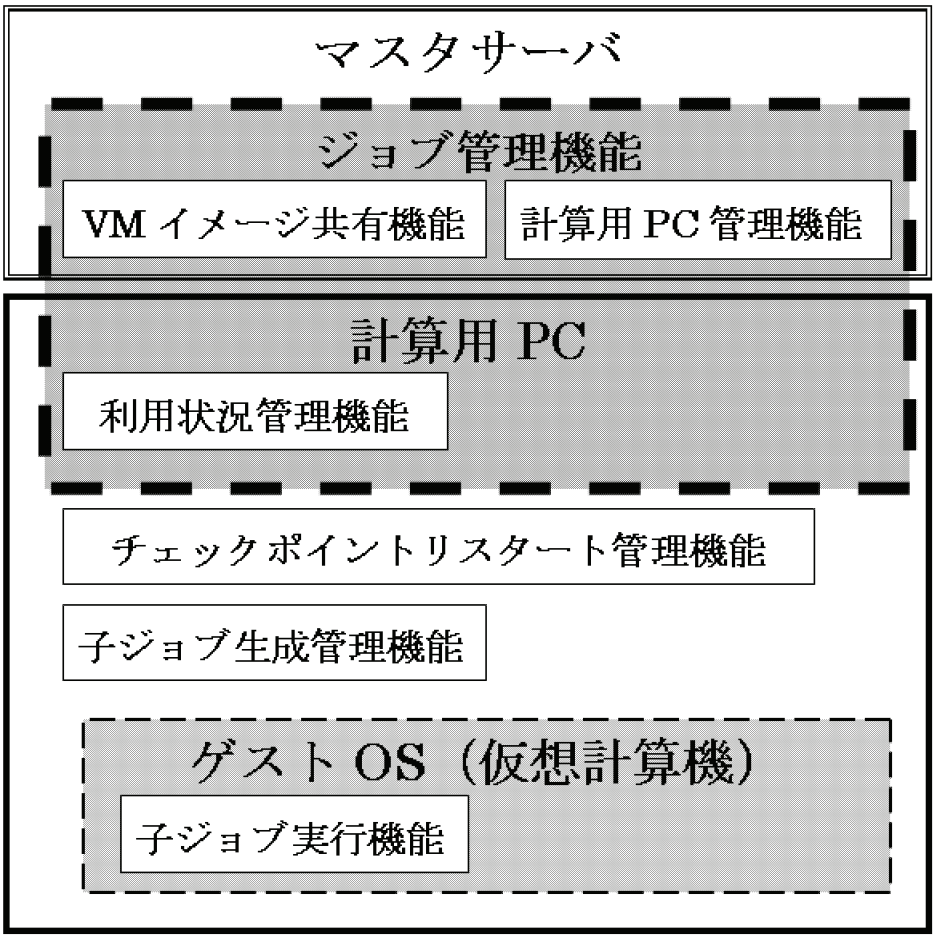

図 3: 構築システムの構成

5.3 ジョブ管理機能

ジョブ管理機能は、マスタサーバの計算用 PC 管理機能と VM イメージ共有機能、計算用 PC の利用状況管理機能から構築されている。

計算用 PC 管理機能によるジョブ投入可能計算機の把握は、特定のポートで利用状況管理機能からの利用状況変更通知を待ち受け、ログオン状態になると計算機の状態管理テーブルの登録情報を BUSY 状態に更新する。ログオフするかマウス・キーボードの利用が無い状態が一定期間過ぎると再び利用状況管理機能から利用状況変更通知が送信され IDLE 状態に更新することで計算機を利用可能にする。本機能により、マスタサーバでは常にジョブ投入可能な計算機を把握することができ、ジョブ投入スケジュールの作成や計算機の集中的な管理ができる。

システムへのジョブ投入は以下のプロセスを取る。

1. マスタサーバ上でシステム利用者がジョブスクリプトを実行する。
2. 利用する計算用 PC に WakeOnLAN パケットを投げ、計算用 PC を起動し、ジョブを投入する。
3. 利用する計算用 PC に必要なファイル（入力ファイル、実行ファイル）とパラメータを送信する。
4. ジョブ ID と VM イメージを結びつけ、各計算機で利用する VM イメージをロックし、仮想計算機を起動する。
5. ゲスト OS でジョブを実行する
6. ジョブの出力ファイルなどをマスタサーバに返す。

これらのプロセスでは、VM イメージ共有機能によって現在 VM イメージを利用している計算用 PC と実行中のジョブ ID を結び付けているため、マスタサーバでジョブの状態を逐次知ることができる。

5.4 マイグレーション機能

4.2 節で述べたユーザビリティ・障害対策を実現するためにマイグレーション機能の実装をおこなう。

本システムのマイグレーション機能は、仮想計算機のサスペンド機能を用いて行われ、以下の条件のときに発生する。

- ユーザがマウス・キーボードを使用する（ログオンする）
- 障害などによるシャットダウン[3]

上記の条件が発生すると、ジョブ実行中のゲスト OS の計算用 PC が使用状況の変化をマスタサーバへ送信する。それと同時にログイン中のホスト OS のバックグラウンドでゲスト OS をサスペンドさせる。マイグレーションしたジョブは Systemwalker Cyber GRIP のジョブキューの最後に登録されるため、VM 管理テーブルが満たされるかマイグレーションしていないジョブが全て終了した後ジョブの再投入がおこなわれる。マイグレーションによって別計算機にゲスト OS を移行させる場合、マスタサーバでサービス開始可能な計算機を探し、専用スクリプトから再び投入されるためサスペンドしたスワップを含めた消費メモリ領域を展開する時間とマスタサーバが再投入する計算ノードの探索時間がかかる。

マイグレーション機能の処理の流れを図 4 に示す。

[3]ただし、障害によるシャットダウンでは仮想計算機をサスペンドさせるため 100 秒程度時間が必要になる。

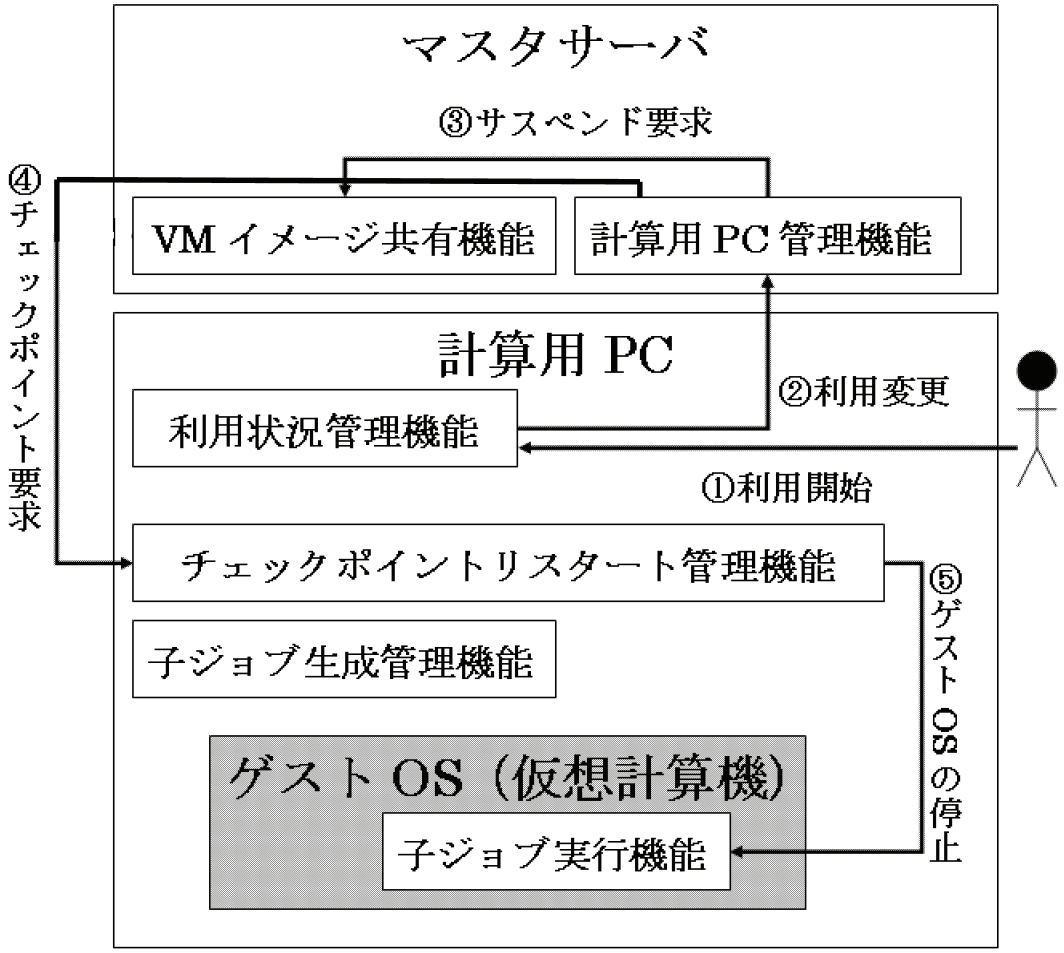

図 4: マイグレーション機能の処理の流れ

6　グリッドシステムの性能評価

6.1　実験システム

1) マスタサーバと計算用 PC の構成要素

本研究での構築システムは、表 1、表 2 に示すようなスペックのマスタサーバと計算用 PC（ホスト OS）7 台で構成している。各計算用 PC に実装するゲスト OS のスペックは表 3 のようになっている。本研究では、以下のスペックの計算用 PC(ホスト OS)7 台と各計算用 PC に構築するゲスト OS を用いてシステムの評価と実験をおこなう。

表 1: マスタサーバ

OS	RedHatEnterPriseLinux4
CPU	Intel Xeon 1.86GHz
Memory	2GB
分散環境	Systemwalker Cyber GRIP

本システムのゲスト OS では、メモリと CPU をホスト OS の半分しか使用していないが、これはユーザがログオンしてきた際に、バックグラウンドで実行するマイグレーション処理によってユーザのプログラム

表 2: ホスト OS

OS	Windows XP Professional SP2
CPU	Intel Core2 Duo 2.4GHz
Memory	2GB
仮想計算機	VMware Server1.0.5
分散環境	Systemwalker Cyber GRIP

表 3: ゲスト OS

OS	CentOS 4.4
CPU	1Unit
Memory	1GB
ネットワーク設定	NAT

実行に影響を与えないためである。

2) ジョブ投入初回時の基本要素

グリッドシステムを構成する計算用 PC の利用開始時には、WakeOnLAN(WOL) による起動と、仮想計算機（VM）の起動、VM テーブルのロックが必要である。VM テーブルとは、計算用 PC で VM イメージを利用開始する際に全ての計算機が排他的にアクセスするファイルで、VM テーブルでは各 VM イメージを利用している計算用 PC を関連付け、計算用 PC 管理機能に利用している。加えて、初回投入時 Systemwalker Cyber GRIP の専用スクリプトの翻訳と終了処理、仮想計算機の FTP の利用開始など投入にかかる遅延時間がかかる。これらの時間は一部並行しておこなわれるため、初期処理全体にかかる時間は合計よりも短くなる。なお、2 回目以降の投入にかかる遅延は、1 から 9 秒程度である。これら 4 つの初期処理時間と初期処理全体にかかる時間を表 4 に示す。

表 4: 計算用 PC の初期処理にかかる時間 (単位:sec)

WOL による起動時間	53.3
VM の起動時間	89.6
VM テーブルのロック時間	80.4
初期処理全体にかかる時間	149.8

3) マイグレーション時の基本要素

マイグレーション時には、初期処理でもおこなった仮想計算機の起動と VM テーブルの再ロックが必要になる。加えて、マイグレーション機能は、消費するメモリ領域によってそのマイグレーション時間が異なる。これは次節で詳細な実験をおこない、性能評価する。

また、マイグレーションをおこなう計算機では、マイグレーション発生時にホスト OS のバックグラウンドでゲスト OS のサスペンド処理をおこなうため、ユーザに負荷がかかる。それを調べるため、スタートアップに登録したブラウザソフト (Internet Explorer) の起動時間を測定し、表 5 に示す。

表 5 から通常時に比べ、2 倍程度の起動時間がかかるのは、バックグラウンドで走るサスペンド処理のためと考えられる。しかし、この処理は、およそ 100 秒で終了するので、その後は、ユーザに負荷を与えない。

表 5: マイグレーション時のスタートアッププログラム起動時間 (単位：sec)

通常時	8.72
マイグレーション時	17.28

6.2 通信機能の性能評価

本システムでは、実際の計算は仮想計算機で実行するため、図 1 のように通信やメモリ領域を操作するためには仮想化層を経由しなければならない。このため、通信はホスト OS でのみ計算するシステムに比べ低速になる可能性があるが、通信速度の低下による効率への影響はほぼないものと思われる。これを調査するため、通信速度の計測実験をおこなう。

通信速度の計測には、ゲスト OS とホスト OS のそれぞれからマスタサーバに 60000 バイトの ICMP パケットを 100 個投入し、応答にかかる時間の平均を求め、その結果を表 6 に示す。また、同一計算用 PC 内でゲスト OS からホスト OS への同様の計測をおこなう。

表 6: ICMP パケット送受信にかかる時間の比較（単位：msec）

計算機	平均応答時間
ホスト OS - マスタサーバ間	10
ゲスト OS - マスタサーバ間	11.6
ゲスト OS - ホスト OS 間	0.743

表 6 から、送受信に仮想化層がわずかな影響を与え通信速度の低下を引き起こしていることがわかる。しかし、この程度の速度低下は無視できるレベルである。

6.3 マイグレーション機能の性能評価

本システムでは、マイグレーション時、仮想計算機にファイル転送がおこなわれないため、マイグレーション時間はジョブ投入する計算用 PC の探索時間と、仮想計算機の起動時間、消費メモリ領域の展開時間のみっつからなる。

マイグレーション時間の取得では、n 次正方行列の乗算と n 次正方行列の乗算で消費するメモリ領域（実メモリとスワップ領域の和）と同じ実メモリ領域を確保する malloc プログラムを用いて、2048 次正方行列から 4096 次正方行列までのプロセスがマイグレーションに要した時間（表 7）を実験により求め、実際の投入ジョブで消費する実メモリ・スワップ領域からマイグレーション時間を得る。

表 7 により、スワップ領域を利用した行列演算の場合、スワップを利用しなかった malloc プログラムよりも 180 秒程度マイグレーションにかかる時間が延びていることがわかる。このことから、スワップ領域を利用するとマイグレーション時間が長くなることがわかる。

6.4 計算システムの総合評価

通信速度がほぼ同じことからゲスト OS とホスト OS の能力が同じであれば、計算速度の低下はほとんど起こらないはずである。そこで、計算用 PC3 台を用いて並列タブーサーチによる巡回セールスパーソン問題（TSP）をホスト OS とゲスト OS で実行する。並列タブーサーチプログラムには、大阪市立大学で大植ら [14] が作成したものを本システムに移植し、TSPLIB[20] の問題例である rat575 を 10 回実行する。通信

表 7: 実メモリ・スワップ領域ごとのマイグレーションにかかる時間 (単位:sec)

問題 (実メモリ) (スワップ)	2048 (80MB) (100MB)	2986 (145MB) (199MB)	3547 (212MB) (297MB)	4096 (280MB) (395MB)
malloc	209	218.5	265	264.8
行列演算	392.2	416.8	420.2	436.0
差	183.2	198.3	155.2	171.2

有では、1 回の実行における通信回数と 10000 回の探索にかかる探索時間を測定する。ただし、ホスト OS とゲスト OS の性能を一致させる目的でゲスト OS で利用する CPU 数を 2Unit とする。また、ホスト OS とゲスト OS のそれぞれで、通信機能による遅延を測定するために通信をおこなわないものと並列化による実行時間短縮を測定するために 1 台に 3 台分のジョブを投入した場合の探索時間を計測する。

表 8: rat575（都市数 575）の探索回数 10000 回の通信回数と探索時間

問題	探索中の通信回数	平均探索時間 (sec)
ホスト OS（通信有）	635.4	341.1
ゲスト OS （通信有）	653.4	439.6
ホスト OS（通信無）	-	311.0
ゲスト OS（通信無）	-	393.4
ホスト OS(1 台で)	-	905.1
ゲスト OS(1 台で)	-	1047.1

表 8 の結果より、並列タブーサーチのような通信回数が多い問題であっても、ジョブ投入のための遅延が必要になるだけで通信の遅延によって探索時間に影響がほとんど無いことがわかる。この結果より、仮想計算機を用いて構築した並列計算システムでも十分な計算能力を示す。

6.5 考察

- マイグレーション機能の評価

  本システムのマイグレーション時間は表 7 より実システム上で 200 秒から 450 秒で、高速なマイグレーションといえる。加えて、障害によって計算機がシャットダウンするときでも仮想計算機のサスペンド状態の遷移の報告には 100 秒程度で済むため、障害対策としても十分機能する。しかし、ネットワークドライブによって VM イメージを共有しているため、別ドメインの計算機からアクセスがあった場合には非常に時間がかかるという問題が発生する。また、互いに独立に走るジョブならば問題は無いが、他の計算機からの情報に依存するジョブでは、マイグレーションによって一部のデータが反映されないという問題がある。

- ジョブ投入システムの評価

  ジョブ投入システムでは、VM イメージのロックに関する排他処理によってジョブ初回起動時約 80 秒程度の遅延が発生する。しかしながら、通信機能を持つグリッドシステムとしては、全体的なジョブ投入速度は高速である。VM イメージのロックも初回とマイグレーション時のみなので、全体に対す

る影響は、多くのジョブ投入をおこなう場合とマイグレーション頻度の高い環境にジョブ投入する場合に限られる。

- 通信機能の評価

  仮想計算機による通信遅延が発生していたが、ほぼホストOSと変わらない速度での通信および実行能力を発揮することがわかる。マイグレーション機能を考慮し、通信機能を最適化したジョブであれば通常のジョブと変わらない能力を発揮できる。

  現在実行しているシステムではLAN環境でおこなっているため、通信遅延の影響が少なかったが、通信範囲が広域ネットワークである場合に仮想計算環境による遅延がどの程度の影響があるかが今後の課題である。

7 メモリ分割法

7.1 メモリ分割

マイグレーション機能にかかる時間は、ファイル転送が仮想計算機にあらかじめ転送されているため、仮想計算機のサスペンド時間と、VMイメージ取得時間、仮想計算機の再開時間（マイグレーション時間）によって決定される。サスペンド時間と再開時間については、利用するスワップ領域で決定されるため、マイグレーション時間は表7で示されるとおり利用するメモリ領域が多いほど増加する。

ただし、1つのジョブが利用するメモリ領域を削減する目的でジョブ分割数を増やした場合、ジョブの投入回数が多いほどジョブ投入や最終処理などに遅延が要求されるため、実行時間が延びる可能性がある。

マイグレーションの発生頻度の低い環境では、メモリ領域によるマイグレーション時間の増加を考慮する必要はないが、ジョブの分割数による影響が大きく、マイグレーションの発生頻度が高い環境では、マイグレーション時間の増加によってジョブ実行に悪影響を与える可能性がある。そのため、投入ジョブで利用するスワップ領域を含むメモリ領域のサイズをマイグレーション発生頻度から自動変更し、ジョブ実行時間の最適化を試みる。

本研究では、ジョブが利用するメモリ領域を変化させることで、マイグレーション頻度に対するジョブ実行時間の最適化でき、効率的なジョブ投入が可能であると考え、投入プログラムの作成・評価をおこなう。

7.2 メモリ分割法のための準備実験の手法

本実験では8192次正方行列の乗算を4台の計算用PCを用いて並列計算する。行列演算は$O(n^3)$の問題であり、対象となる行列をx分割すると$O(x^3)$*$O(\left(\frac{n}{x}\right)^3) + O(n^2)$となり、行列演算の最終処理に$O(n^2)$の演算がかかる。並列化では、8192次正方行列を6.3節で利用したn次正方行列の4096、2048、1024次正方行列に分割し、メモリ分割法の指標となる各マイグレーション発生頻度0%から40%での実行時間を比較する。マイグレーションの発生には、一定時間ごとにマイグレーション頻度の確率で各計算用PCにログインするかの判定をおこない、仮想計算機のサスペンドが終了するとすぐにログオフするプログラムを利用する。

本実験では、各分割数による評価を目的とし、実行時間の最小のものをメモリ分割法に用いる。

7.3 実験結果

グリッドシステムを利用し、メモリ分割とマイグレーションを考慮しないジョブ分割でマイグレーション頻度を変更させるシミュレーションをおこない、実験結果を図5に示す。

図5より、マイグレーション頻度が増加するに従い、実行時間も増加していることがわかる。特に、最大の分割数である4096次正方行列では実行時間が2000秒も増えるなど大きな影響がある。一方、1024次

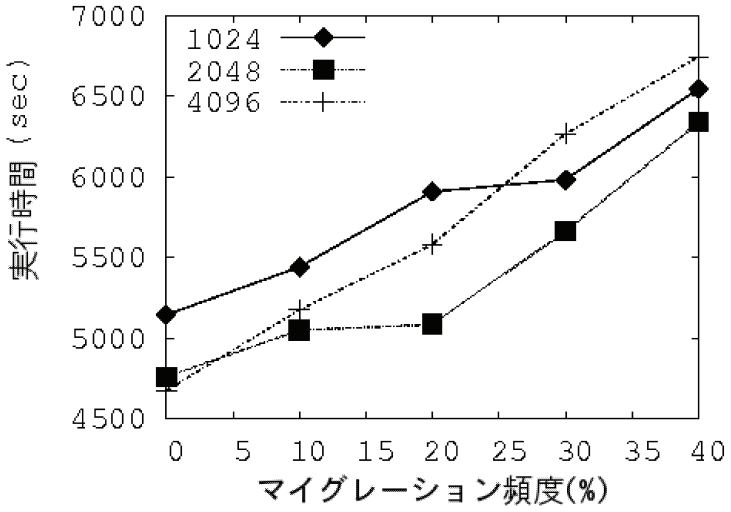

図 5: マイグレーション発生時の 8192 次正方行列並列演算の実行時間の比較

正方行列は実行時間の変化が他の分割数と比べてなだらかであったが、マイグレーションが発生しない環境で 400 秒程度他の分割数よりも実行時間が多くかかる。ジョブの分割数を多くすればするほど、マイグレーション頻度による影響を受けにくくなり、傾きがなだらかになる。これは、ジョブサイズが大きいとマイグレーション時間が長くなるからである。

これらの結果から、マイグレーション頻度が 0 から 20%未満までは 4096 次正方行列に分割し、20 から 40%までは 2048 次正方行列に分割することが最も効率的なジョブ投入となる。

7.4 メモリ分割法の実験・評価

利用した 4 台の計算用 PC に 8192 次正方行列を 4 分割 (2048 × 8192 行列と 8192 正方行列）したジョブを投入し、メモリ分割法の結果と比較する（図 6)。

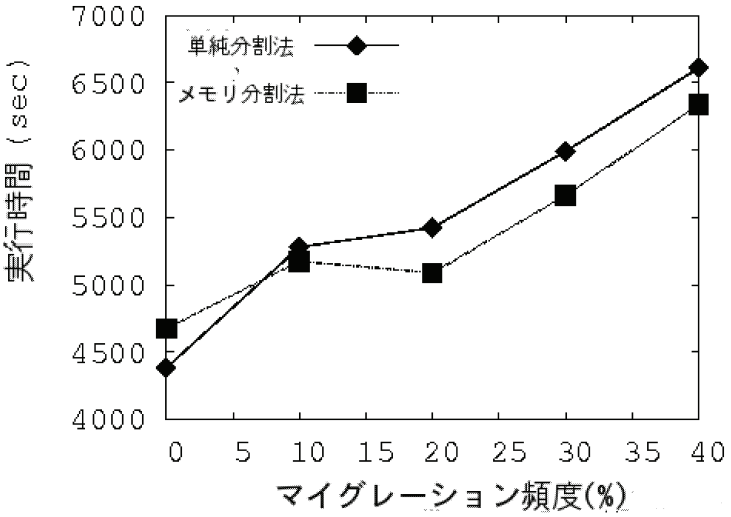

図 6: マイグレーションを想定したプログラムと単純分割の実行時間の比較

図 6 より、マイグレーションが発生しない環境では、単純分割法より 300 秒程度時間がかかるが、マイ

グレーション頻度が 20%付近から逆転し、単純な 4 分割よりも良い結果を返す。この理由として、低いマイグレーション頻度では、分割したジョブの統合処理（終了処理）やファイル転送にかかるオーバーヘッドにより実行時間が単純な分割よりも上回るが、マイグレーション頻度が高い環境では、マイグレーションにかかるオーバーヘッドがメモリ分割法の最終処理を含めた実行時間を上回ったためと考えられる。

この結果より、あらかじめ投入ジョブが消費するメモリサイズを把握できるのであれは、ジョブ投入をより効率的におこなうことができると推察できる。

8　社会科学への応用例

6.4 節の総合評価で使用した並列プログラムは、巡回セールスパーソン問題（TSP）だけでなく、一般的な最適化問題に利用できるタブーサーチである。本システムは、最適化問題に関わらず、通信が非同期であればすべての並列アルゴリズムに適用可能である。最適化問題に絞って考えても、TSP だけでなく、多くの最適化問題に利用できる。たとえば、資産運用などの金融工学の問題 [11] [18] や、確率論的費用フロンティアの推定 [6]、高等教育政策 [5] [13] などに適用可能である。

文献 [11] は、ポートフォリオ最適化に関しての著作である。ポートフォリオ最適化とは、いろいろな資産に対してどのように投資していけばよいかを数理計画法を用いて解く手法である。文献 [18] では、政策資産配分を策定するための最適化モデルを構築し、それに対する近似解放を提案している。政策資産配分とは、厚生年金基金が資産運用を行う際に作成する資産配分のことである。

文献 [6] では、イギリスにおいて公庫から多くの資金を提供されている産業での複数の財・サービスを提供している組織に関してクロスセクションデータを使って確率論的費用フロンティアを推定している。この推定は、規模と範囲の経済性の指標を提供するために使われる。フロンティア推定モデルに現れる効率性の指標は、DEA（包絡分析法）によって得られたものと比較される。各々の組織に関しての費用関数と生産物の情報は、タブーサーチに基づいたヒューリスティック法を使って、産業構造を最小化する全体的な費用を推定するために使われる。全体の生産物ベクトルが与えられたとして、各々の生産者に関する生産物ベクトルを構築することができる。最適システムは生産者が同質でなく、科学教育への提供をより多くすれば、全体的な費用を減少させることができることがわかる。

文献 [5] は、イギリスの高等教育政策に対しての答えを与えている。タブーサーチに基づいたヒューリスティック法を使って、複数の財・サービスを提供している組織に関する費用関数を推定している。文献 [13] もイギリスの高等教育政策に関する論文である。DEA に基づいたベストプラクティス効率性指標を使って、現在の政策目標がどれほど実証的な支援になっているかを調べている。

9　おわりに

本研究では、遊休計算機を有効活用するために、計算途中のジョブを他の計算機にマイグレーションできる機能を有した PC グリッドシステムを構築し、その評価をおこなった。特に、マイグレーション機能とマイグレーション機能の実現手段である仮想計算機に注目して実験をおこなった。その結果、仮想計算機の仮想化層がハードウエア利用に与える影響は軽微で、仮想計算機を用いてもホスト OS とほぼ同じ計算能力を示した。また、マイグレーション機能を考慮してジョブ分割をおこなうメモリ分割法を提案し、実験結果から効率の良いジョブ投入ができることがわかった。しかし、今回のシステムは計算用 PC が 10 台未満の小規模なものだったので、より大規模で広域ネットワークに及んだ場合の影響を今後検討する必要がある。

謝辞

本研究をおこなうにあたり、多くの人々にご協力をいただいた。研究のシステム構築に携わってくださった富士通研究所の皆様、研究の場を提供してくださった関西大学ソシオネットワーク戦略研究センターの皆

様に感謝の意を表す。
また、本研究は、平成 20 年度関西大学重点領域研究助成金において、研究課題「休止中のコンピュータを有効利用するグリッドシステムの構築とその応用」として研究費を受け、その成果を公表するものである。

参考文献

[1] 合田憲人・関口智嗣 編著: グリッド技術入門, コロナ社, (2008).

[2] 秋岡明香, 村岡洋一: グリッド環境での CPU 負荷予測に基づくネットワーク負荷中期予測電子情報通信学会論文誌，VolJ87-D-I(2004).

[3] Eun-Kyu Byun and Jin-Soo Kim: DynaGrid: A dynamic service deployment and resource migration framework for WSRF-compliant applications, Parallel Computing，Vol33(2007).

[4] F. Berman, H. Casanova, A Chien, K. Cooper, H. Dail, A. Dasgupta, W. Deng, J. Dongarra, L. Johnsson, K. Kennedy, C. Koelbel, B. Liu, X. Liu, A. Mandal, G. Marin, M. Mazina, J. Mellor-Crummey, C. Mendes, A. Olugbile, M. Patel, D. Reed, Z. Shi, O. Sievert, H. Xia and A. YarKhan: New Grid Scheduling and Rescheduling Methods in the GrADS Project, International Journal of Parallel Programming，Vol33,(2005).

[5] Geraint Johnes: Costs and Industrial Structure in Contemporary British Higher Education，The Economic Journal, Vol.107, No.442, pp.727-737 (1997).

[6] Geraint Johnes: The Costs of Multi-product Organizations and the Heuristic Evaluation of Industrial Structure，Socio-Economic Planning Sciences, Vol.32, No.3, pp.199-209 (1998).

[7] F. Heine, M. Hovestadt, O. Kao and A. Keller: SLA-aware job migration in grid environments, Advances in Parallel Computing，Vol14, (2005).

[8] FreeBSD jail: http://www.onlamp.com/pub/a/bsd/2003/09/04/jails.html.

[9] グリッド協議会: http://www.jpgrid.org/index.html.

[10] Haibo Chen, Jieyun Chen, Wenbo Mao and Fei Yan: Daonity – Grid security from two levels of virtualization，Information Security Technical Report，Volume 12, Issue 3,(2007).

[11] 枇々木規雄 著: 金融工学と最適化, 朝倉書店, (2001).

[12] ITpro 編: すべてわかる仮想化大全 2009, 日経 BP, (2008).

[13] J. Colin Glassa, Gillian McCallionb, Donal G. McKillopc, Syamarlah Rasaratnama, Karl S. Stringer: Implications of variant efficiency measures for policy evaluations in UK higher education，Socio-Economic Planning Sciences, Vol.40, pp.119-142 (2006).

[14] 大植裕之、大西克実、中野秀男、榎原博之: 巡回セールスマン問題を対象とした並列タブーサーチにおけるプロセス間通信の効果について, 情報処理学会研究報告 (MPS),2005-MPS-54(2005).

[15] Renato J. Figueiredo, Peter A. Dinda and Jose A. B. Fortes: A Case For Grid Computing On Virtual Machines，Distributed Computing Systems, Vol23 (2003).

[16] SETI@home: http://setiathome.berkeley.edu/.

[17] 曽山豊: 企業におけるグリッド・コンピューティングの活用とその成果, グリッド協議会セッション,Grid World 2006(2006).

[18] 玉之内直, 猿渡康文: 政策資産配分策定モデル，日本経営工学会論文誌, Vol.54, No.6, pp.382-389 (2004).

[19] 立薗真樹, 中田秀基, 松岡聡: 仮想計算機を用いたグリッド上での MPI 実行環境，SACSIS 2006，(2005).

[20] TSPLIB: http://www.iwr.uni-heidelberg.de/
groups/comopt/software/TSPLIB95/.

[21] VMware: http://www.vmware.com/.

[22] Xen: http://www.xen.org/.

[23] Yan Fei, Zhang Huanguo, Sun Qi, Shen Zhidong, Zhang Liqiang and Qiang Weizhong: An improved grid security infrastructure by trusted computing, Wuhan University Journal of Natural Sciences, Volume 11, Number 6,(2006).