論　文

仮想計算機を適用したPCグリッドの開発と性能評価*

森川　浩明†　　榎原　博之††　　大西　克実†　　中野　秀男†

Development and Performance Evaluation of PC Grid Which Applied Virtual Machine*

Hiroaki MORIKAWA†, Hiroyuki EBARA††, Katsumi ONISHI†, and Hideo NAKANO†

あらまし　近年，家庭用計算機の性能が向上し，家庭用計算機複数台で一昔前のスーパコンピュータ1台分程度の計算能力を発揮している．これら家庭用計算機は，ユーザが求める計算能力よりもオーバスペックであるため，計算資源の有効活用が求められている．一方，仮想化技術の発展により，計算資源を仮想的に分割・統合し，サーバなどの計算資源を有効活用する仮想計算機技術が注目されている．本研究では，仮想計算機のハードウェアに依存しない特徴に着目し，並列計算においてユーザが計算機を利用するとき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行う．更に，実験により提案システムの性能評価を行う．

キーワード　グリッドシステム，マイグレーション，仮想計算機，遊休PC，並列化

1. まえがき

現在，計算機の演算能力・通信機能の発展が目覚ましく，家庭用計算機複数台で一昔前のスーパ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近年の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，グリッドシステムやクラウ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．この中で，注目されているものの一つに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．SETI@home [17] に代表されるようなPCグリッドのプロジェクトでは，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存在しない．別の例として，大学などの演習室の計算機を夜間に利用するキャンパスグリッドというシステムも存在するが，計算機の利用が夜間に限定されるため長時間の実行に向かない．また，近年活発に利用され始めたクラウドシステムはあらかじめ企業などが提供する計算機群を活用するものなので，遊休計算機の有効活用とはかけ離れている．

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%に達することはない．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

† 大阪市立大学大学院創造都市研究科，大阪市
Graduate School for Creative Cities, Osaka City University, Osaka-shi, 558–8585 Japan
†† 関西大学システム理工学部，吹田市
Faculty of Engineering Science, Kansai University, Suita-shi, 564–8680 Japan
* 本論文はシステム開発論文である．

設置できる特徴のほかに計算機のハードウェアに依存しない特徴がある．これにより，一部の仮想計算ソフトでは実行中の仮想計算機の計算内容を別計算機に移行させるマイグレーション機能を実装している．

本研究では，キャンパスグリッドを想定し，演習室に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に仮想計算機によ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構築を行う．マイグレーション機能によって演習室内でユーザが計算機の利用を開始すると計算を行っている計算機の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．更に，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，広域グリッドシステムの SETI@home 方式では，ジョブ中断対策のため複数の同一ジョブを投入する．しかし，この手法では効率の低下が大きいため，本研究ではジョブをマイグレーションにより中断することなくユーザが計算機の利用を開始した時点の実行状態を別計算機で再開させる手法により効率の低下を小さくする．本論文は論文誌に投稿する前に執筆する性格をもつ RCSS ディスカッションペーパ [7] をもとにしている．

以下，**2.** では本研究と関連する研究を紹介し，準備として **3.** で仮想計算機を，**4.** で PC グリッドシステムを説明する．**5.** 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.** では構築したグリッドシステムのパフォーマンスの評価を行い，実験・評価結果から得られた考察について述べる．

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 [20] 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行う．この研究では，キャンパスグリッドを想定し，負荷分散のために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途切れることなくシステムが運用でき，夜間のみ計算機を利用するシステムなどにおける時間的制約を排除できる．

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 [18] がある．曽山らが行った研究では，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行った．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．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．更に，ジョブ間の通信ができないため，互いに独立したジョブしか扱えない．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，マイグレーション機能を実装することで，ユーザに負荷をかけず，長時間のジョブ実行を可能にしている．更に，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．

大規模な遊休計算機の利用を考慮したシミュレーションを行った研究として，グルノーブル HP 研究所の Richard らの研究 [16] がある．この研究では，計算機を利用していない遊休期間にソフトウェアベースのサンドボックスにジョブを投入し，ユーザが利用する OS と独立させることで安全なグリッドシステムの構築を行った．この研究では，ソフトウェアベースの並列計算用 OS を構築し，利用者の不在をマウスやキーボードの利用期間で認識する機能を備えている．しかし，ジョブマイグレーション機能がないため，ユーザが計算機の利用を開始すると並列計算の有無にかかわらずサンドボックスを終了してしまう．また，広域グリッドシステムを想定しているが，ユーザが利用していない期間の定義がマウス・キーボードを利用していない期間とユーザが利用を許可した期間であるため，得られる計算能力が限られている．本研究では，マイグレーション機能を利用し，ユーザが計算機の利用を開始すると別の計算機に計算内容を移行する機能や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機能などにより，ユーザの利用と計算能力を両立させている．

ほかに，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 [9]，マイグレーションにかかわる研

究 [12], [4]，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 [3]，グリッドシステムのセキュリティにかかわる研究 [5], [8] 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 [23] や Xen [24]，Jail [10] 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），ハードウェ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ェアを操作している．

仮想計算機では，仮想化層によってハードウェアから切り離され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ェ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．

- ハードウェアに依存しない

ハードウェアの操作は仮想化層で行われるため，仮想計算機にはハードウェアの影響が少ない．このため，異なるハードウェア環境に VM イメージ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，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，若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内で保存した状態を展開する手法

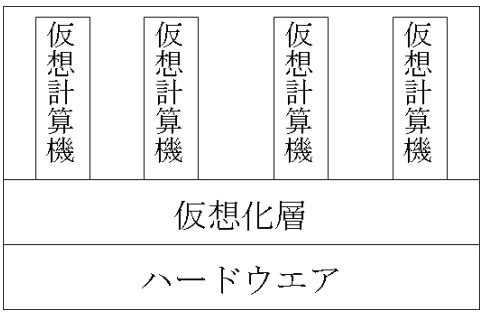

図 1 仮想計算機の構造
Fig. 1 Structure of virtual machine.

（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実行中の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．また，本システムではユーザへの配慮と障害対策のためにチェックポイントマイグレーションを実装する．これは，チェックポイントマイグレーションではライブマイグレーションで利用されるユーザの計算機利用開始の把握から別計算機にメモリ内容を送信する際にライブマイグレーション特有の処理やメモリ内容を分割して送信するなどの余分な処理が発生しないため，障害対策として優れた機能を示すと判断したためである．

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 [11] 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，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．また，『グリッド』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，主に LAN 内で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行っている．特に，演習室などの計算機をオープン利用時に利用することを想定している．この環境では，提供される各計算機にユーザが存在するので，遊休状態にある計算機（遊休 PC）を探索し，発見した遊休 PC にジョブを投入する必要がある．

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．

本研究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．ある計算機にファイル転送した後，計算資源を提供する計算機が計算途中で終了し，結果を返さないことがあるので，同じジョブを複数の計算機に投入している．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行うためには，一定時間ごとに計算機の状態を把握し，ジョブ投入時に予測される各計算機の遊休時間内に収まるようジョブ分割を行う必要がある．

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ヘッドが高くなり，通信オーバ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．PC グリッドにおいて発生する障害は，計算機のハードウェアの破損，ユーザによる計算機のシャットダウンやトラヒックの急激な増加などの通信による障害がある．これらの障害のうち本論文では，100 秒程度の比較的時間に余裕のある障害やトラヒックの増加などによる通信障害を考慮している．この対策としてマイグレーション機能の実装がある．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や障害の発生を検知し，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．ただし，本研究で想定していない通信障害などに関しては，ジョブが実行されなかったと判断してジョブの再投入を行っている．

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 種類のキューをもっている．

（1） 仮想キュー

計算用 PC を仮想的に一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．

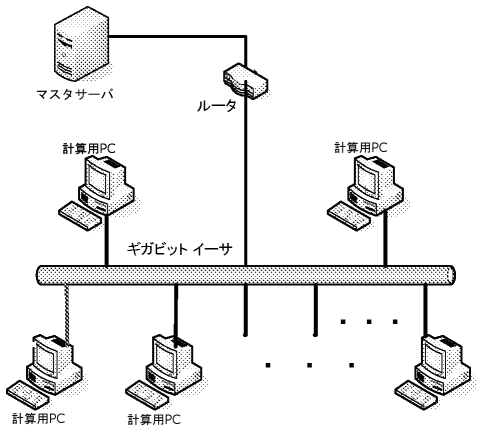

図 2　構築システムの全体図
Fig. 2　Overview of this grid system.

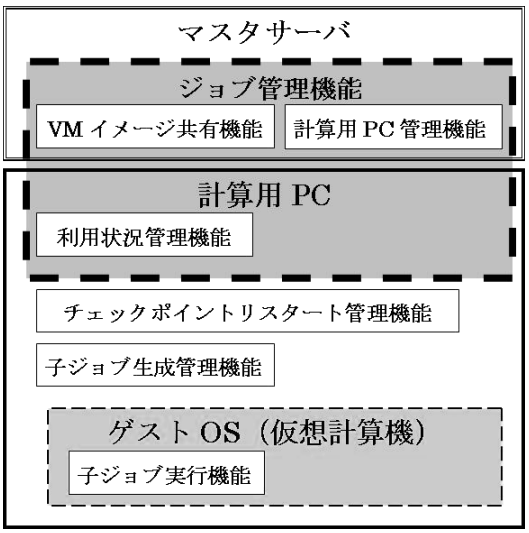

図 3　構築システムの構成
Fig. 3　Struction of our grid system.

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 PC7 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行うため，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．

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 イメージを各計算用 PC に管理させるのではなく，マスタサーバが保持している VM イメージを各計算用 PC がネットワークで共有して直接操作する．

構築したグリッドシステムは図 3 の構成になっており，以下の機能をもつ．

- 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，計算資源管理を行う．

- 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・停止を行う．

  - 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行う．

- 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．

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ない状態が一定時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とる．

（1） マスタサーバ上でシステム利用者がジョブ投入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） 利用する計算機に必要なファイル（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 イメージを結び付け，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 通信機能

本システムの通信は，将来他施設や他大学との連携を想定しているため専用の LAN やネットワークではなく，一般に用いられている TCP/IP 通信を行う設計である．また，高速・大容量の通信でトラヒックの増大に対応するためマスタサーバ・計算用 PC 間は 1000BASE-T の通信網であり，以下四つの通信機能を実装している．

- ファイル転送機能

ジョブ投入時の入力ファイルの転送及びジョブ終了時の出力ファイルの転送にかかわる CyberGRIP の機能

- ファイル共有機能

各計算用 PC がマスタサーバに存在する VM イメージを操作するためにマスタサーバが提供する機能

ファイル共有では Samba を利用しており，各計算用 PC はストレージに VM イメージをコピーせずマスタサーバにある VM イメージを直接操作する．

- ジョブ間非同期通信機能

並列計算実行中に計算用 PC の子ジョブ間で非同期通信が可能にする．ただし，子ジョブを実行する計算用 PC はマイグレーションにより動的に変化するため，必ずマスタサーバを経由しての通信となる．また，通信先の子ジョブが待機中である場合の通信を保証していない．

- 計算用 PC 状態監視機能

計算用 PC が利用可能であるかの情報を収集し，各計算機の状態把握する機能

これらの通信はギガビットイーサを使用しているため，ほとんど遅延なく通信できる．計算用 PC の数が増えれば，遅延が生じる可能性がある．

5.5 マイグレーション機能

4.2 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行う．本システムのマイグレーション機能は，障害発生時のマイグレーション速度とマイグレーションにかかる処理削減のために，チェックポイントマイグレーションを用いている．

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．

- ログオンする

ユーザが計算用 PC にログオンするとバックグラウンドで仮想計算機をサスペンドさせる．ログオン中はユーザの計算用 PC 利用状況が監視され，ユーザがマウス・キーボードを使用しない状態が 30 分経過すると計算用 PC を再び利用可能にする．ただし，この時間は設定により変更可能である．

- 障害などによるシャットダウン

仮想計算機が正常にシャットダウンできる障害であれば，仮想計算ソフトの設定によって仮想計算機をサスペンドさせる．ただし，仮想計算機のサスペンドには 100 秒程度の時間が必要になる．また，ネットワークの寸断や急なハードウェアの破損には対応できないため，このような状況では計算を最初からやり直す必要がある．

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すべて終了した後ジョブの再投入が行われる．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仮想計算機のスワップを含めた消費メ

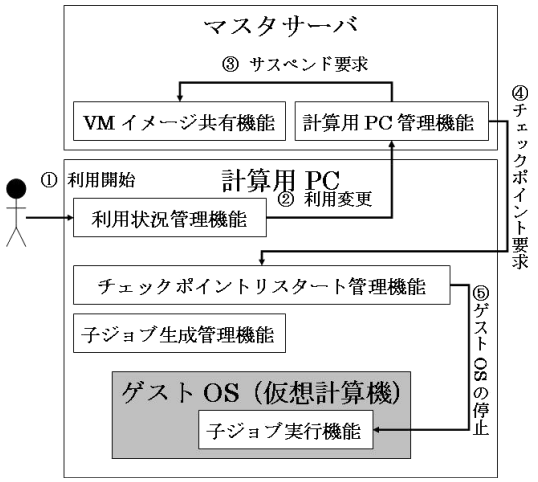

図 4　マイグレーション機能の処理の流れ
Fig. 4　Workflow of the migration function.

表 1　マスタサーバ
Table 1　Structure of master server.

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

表 2　ホスト OS
Table 2　Structure of host OS.

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

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．また，マイグレーション中の計算用 PC がマイグレーション条件を満たすと仮想計算機を一度展開した後に別の計算用 PC にマイグレーションする．

マイグレーション機能の処理の流れを図 4 に示す．

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実行に大きな影響を与えないためである．

表 3　ゲスト OS
Table 3　Structure of guest OS.

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

表 4　計算用 PC の初期処理にかかる時間（単位：s）
Table 4　Time of initial processing in calculation PC.

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

(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ホスト OS とゲスト OS を保持する計算用 PC の利用開始時には，WakeOn-LAN（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すべての計算機が排他的にアクセスするファイルで，各 VM イメージを利用している計算用 PC を関連づけ，計算用 PC 管理機能に利用している．VM テーブルのロックは VM の起動と並行して行われるため，VM の FTP サーバ起動待ち（最大約 600 s）などが発生する．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．これら三つの初期処理と全体にかかる時間は表 4 のようになる．

本システムのジョブ投入には，VM テーブルのロックに関する排他処理などによって初回起動時に 150 秒程度の遅延が発生する．しかし，この遅延は初回時のみであるので，システム全体としては大きな影響を受けないと考えられる．

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，仮想化層を経由するオーバヘッド程度では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

表 5 ICMP パケット送受信にかかる時間の比較（単位：ms）
Table 5 Comparison of response time of ICMP packets.

計算機	平均応答時間
ホスト OS – マスタサーバ間	10.0
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 FTP 送信にかかる時間の比較（単位：ms）
Table 6 Comparison of time of file transfer.

FTP	100 MByte	200 MByte	300 MByte
ホスト OS（実測時間）	8.5	17.1	26.2
ホスト OS（見積時間）	8.3	16.7	25.0
ゲスト OS（実測時間）	14.7	28.5	74.3
ゲスト OS（見積時間）	8.9	17.9	26.8

信速度の計測実験を行う．

(1) 通信速度の計測

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 5 に示す．また，同一計算用 PC 内でゲスト OS からホスト OS への同様の計測を行う．

表 5 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分かる．しかし，この程度の速度低下が並列計算に与える影響は軽微である．

(2) ファイル転送の計測

FTP によるファイル転送でホスト OS とゲスト OS のスループットの比較を行う．表 5 からホスト OS–他の計算用 PC，マスタサーバへの平均通信速度は 96 Mbit/s 程度であり，ゲスト OS からマスタサーバへの転送速度は 83 Mbit/s 程度であると見積もることができる．

そこで，ホスト OS とゲスト OS からマスタサーバへ FTP で 100 M から 300 M までのファイルを転送し，通信速度からの見積時間と実測時間との比較評価を表 6 に示す．

表 6 から分かるように，ホスト OS の転送時間は，実測時間と見積時間で大差ないが，ゲスト OS の転送時間は，ファイルサイズが大きくなるにつれて，大幅に実測時間が増加している．これは，ゲスト OS からマスタサーバへファイル転送する際，マスタサーバに存在する VM イメージ内の転送ファイルを計算用 PC のメモリに取り込み，あて先ホスト（マスタサーバ）にファイルを送信するため二重にファイル転送がかかるためである．また，使用するメモリ領域の小さい 100 MByte や表 5 で用いた ping パケットなどでは，一度でメモリにファイルを取り込む．利用したシステムでは 200 MByte や 300 MByte のファイル転送の際にメモリへの取込みが 128 MByte 単位で行われたことと仮想計算機に最も負荷がかかる kernel-mode の利用が多く発生したことが，ファイルサイズの増加に比例した時間増加が発生しなかった原因として考えられる．しかし，本システムにおけるファイル転送はジョブ投入時と終了処理の 2 回に限られるため，ファイル転送による影響は軽微であるといえる．

表 7 マイグレーション発生時のスタートアッププログラム起動時間とサスペンド時間（単位：s）
Table 7 Program boot time and Virtual Machine suspend time when migration occurrs.

通常時	8.72
マイグレーション時	17.28
ゲスト OS の停止時間	97.18

本システムの通信機能では，仮想計算機による遅延と VM イメージをマスタサーバで管理することによる遅延が発生している．しかし，本実験では非同期通信を行っているため，仮想計算機による遅延が並列計算に与える影響はほぼない，マスタサーバに VM イメージを保存していることによるファイル転送にかかる遅延はジョブ投入と終了処理の 2 回だけなので，長時間のジョブでは遅延の影響は軽微である．

6.3 マイグレーション機能の性能評価

本システムにおけるマイグレーションにかかる時間は，マイグレーションの発生した計算用 PC でのゲスト OS のサスペンド時間と，マスタサーバでの他の遊休 PC 探索時間，加えて投入先の計算用 PC でのゲスト OS の起動時間と消費メモリ領域の展開時間の二つからなる処理に分類できる．ただし，マイグレーションが発生した計算用 PC でのサスペンド時間とマスタサーバでの遊休 PC の探索時間，投入先計算用 PC でのゲスト OS での起動時間は一定である．

また，マイグレーションが発生した計算用 PC では，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行うため，ユーザに負荷がかかる．このユーザが計算用 PC の利用開始する際にかかる負荷を調べるため，スタートアップに登録したブラウザソフト（Internet Explorer）の起動時間とゲスト OS のサスペンド時間を測定し，表 7 に示す．

表 8　メモリサイズごとのマイグレーションにかかる時間（単位：s）
Table 8　Time of migration per using memory size.

100 MByte	300 MByte	500 MByte	700 MByte
212	216	265	264

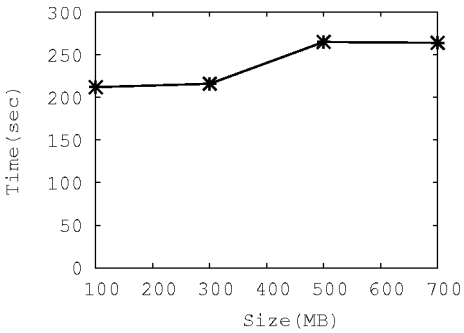

図 5　メモリサイズごとのマイグレーションにかかる時間
Fig. 5　Time of migration per using memory size.

表 7 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，サスペンド処理はプログラムが利用するメモリサイズによらず 100 秒程度で終了するので，長時間にわたり計算資源を占有することなく，その後ユーザに負荷を与えない．

次に，仮想計算機のメモリ領域の展開時間は利用しているメモリ量によって変化するため，指定したメモリ量を確保する malloc プログラムを用いて，100 MByte から 700 MByte までメモリ量を確保し，その際の投入先でのマイグレーションに要する時間を実験により求める．

表 8 と図 5 により，消費するメモリ領域が 300 MByte から 500 MByte でマイグレーション時間が急しゅんしていることが分かる．これは，ジョブ再開時にマスタサーバで共有する VM イメージに投入先の計算用 PC からアクセスする際，メモリ展開のためのファイルが約 450 MByte ごとに投入先の計算用 PC へ転送されるため，マイグレーション時間がこのように増加していると考えられる．

本システムのマイグレーションにかかる時間は表 8 より 200 秒以上で，マイグレーションに数秒しかかからないライブマイグレーションと比べると低速である．しかし，代表的なグリッドシステムである Condor-G [6] ではマイグレーションに 15 分以上かかることやライブマイグレーションが障害に弱いことを考慮すると，速度と障害対策に優れたマイグレーション機能を実装できたといえる．

6.4　計算システムの総合評価

本節では，実際の並列プログラムを実行させ提案システムの総合的な評価を行う．長時間実験のため研究や業務での使用を長時間休止できる計算用 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 [22]）をホスト OS とゲスト OS で実行する．

TSP とは有限個の都市の 1 地点から存在するすべての都市を一度だけ通り，その最短の経路（最適解）を求める NP 困難な最適化問題の一つである．TSP の都市数 n の問題例では解の数が $(n-1)!$ 個あるため，n = 数百 程度で膨大な計算量になる．このため，TSP の解法には時間などの制約条件のもとになるべく最適解に近似した解（近似解）を求め，最適解に近づけていく方法（近似解法）をとる．本実験でも近似解法の一つであるタブーサーチを用いる．

タブーサーチ [19] は，一部の移動を制限することで最適解でない解（局所最適）で停滞することを防ぐため，一定の改悪を許可し，改悪・改善の情報を移動のためのタブー情報とする手法である．これにより，ユーザが定めた期間内になるべく広域の解情報に到達できる．TSP では生成した都市間の経路を組み換えた情報をタブーとし，よりよい解（暫定解）を求めていく手法である．本実験では，タブーサーチの経路組換えにある四つの地点の二つの経路を組み換える 2-opt 法と複数計算機によるタブー情報の共有という方法をとることで，1 探索当りの解の改善を効果的に進めていく手法をとる．本システムでは大阪市立大学の大植ら [15] が作成した並列タブーサーチプログラムを移植する．

本実験では，TSPLIB [21] の問題例である pr1002（表 10）と rat575（表 9）を 10 回実行する．この並列プログラムでは，タブーリストの更新ごとにタブーに設定される都市間の経路情報（16 BYTE）と，解の遷移ごとに暫定解（都市数*4 BYTE）を他の計算用 PC に送信している．また，実行プログラム及び入出力ファイル転送にかかる総時間は rat575 で 1 秒未満，pr1002 で 1 秒程度であり，実験に影響しないレベルの転送時間である．本実験では，1 回の実行におけるタブーリスト及び暫定解更新のための 1 ジョブ当りの通信回数と 100000 回の探索にかかる探索時間を測定する．ただし，ホスト OS の実験環境とゲスト OS の実験環境を一致させる目的でゲスト OS が利用する

表 9　rat575（都市数 575）の探索回数 100000 回の通信回数と探索時間
Table 9　Number of communication and search time in rat575.

問題	探索中の通信回数	平均探索時間 (s)
ホスト OS（通信有）	2231	4372
ゲスト OS（通信有）	2109	5182
ホスト OS（通信無）	—	3979
ゲスト OS（通信無）	—	4564
ホスト OS（1 台で）	—	10844
ゲスト OS（1 台で）	—	11745

表 10　pr1002（都市数 1002）の探索回数 100000 回の通信回数と探索時間
Table 10　Number of communication and search time in pr1002.

問題	探索中の通信回数	平均探索時間 (s)
ホスト OS（通信有）	1627	10480
ゲスト OS（通信有）	1563	11864
ホスト OS（通信無）	—	9557
ゲスト OS（通信無）	—	10738
ホスト OS（1 台で）	—	28097
ゲスト OS（1 台で）	—	30624

CPU 数を 2Unit とする．また，ホスト OS の実験環境とゲスト OS の実験環境のそれぞれで，通信機能による遅延を測定するために通信を行わないものも計測する．更に，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実験する．この実験では通信を行わない．

表 9，表 10 の結果より，ホスト OS とゲスト OS の実行時間は，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ほぼ影響がないことが分かる．また，pr1002 の方が rat575 と比べ並列化による速度低下の比率が少ないことが分かる．これは，問題例 pr1002 よりも問題例 rat575 の方が実行ごとの通信回数が多いため，通信にかかるオーバヘッドによって実行速度の低下を引き起こしていると考えられる．

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．また，実行中の計算機のバックグラウンドで実行する並列プログラムと比べ，実メモリの有効活用が図れ，安定した計算能力を発揮できるといえる．

6.5　実環境のシミュレーション

実際の計算用 PC の利用環境を想定し，効率的な運用を行うために関西大学 IT センターの演習室の環境を再現し，実験を行う．関西大学での演習室内の使用状況 [14] は，年度初めや学期末を除くと表 11 のようになる．

表 11　演習室の 1 日の利用状況
Table 11　Using state of PC room.

総計算機数	149
1 日の利用回数	984
1 日の利用時間（時間）	875.5
1 台当りの利用回数	6.6
1 回当りの利用時間（分）	53

表 12　ユーザ利用がある環境での長時間のジョブ実行（単位：分）
Table 12　Long term job processing in which someone uses PCs.

理想平均実行時間：t_{avg}	112
ジョブ投入数：num_{job}	72
実測平均実行時間	169
実測最大実行時間	231
実測最小実行時間	110
総実行時間：$total$	4043
平均マイグレーション回数	0.85
効率低下率：d(%)	50.4

表 11 より，各計算用 PC の利用回数が 1 台当りの利用回数になるようランダムにログインし，一様乱数で使用時間を 1 回当りの利用時間± 50%（使用時間 27 分から 80 分の間）とすれば，単調でない利用状況を設定でき，利用時間の平均が 53 分付近に収束する実際の計算用 PC 利用環境を再現できる．この環境でジョブ（約 2 時間）72 セットを 3 台の計算用 PC に投入し，その結果を表 12 に示す．ここで，平均マイグレーション回数とは，投入 1 回当りに発生するマイグレーションの平均回数である．また，効率低下率 d は総実行時間 $total$ と理想総実行時間 $total_{img}$ から以下の式で定義される．

$$d = \frac{total - total_{img}}{total_{img}}$$

理想総実行時間 $total_{img}$ は理想平均実行時間 t_{avg} とジョブ投入数 num_{job}，実行ノード数 num_{node}（3 台）より以下の式で求められる．

$$total_{img} = \frac{t_{avg} * num_{job}}{num_{node}}$$

表 12 より，外部からの影響がない環境よりも実際のユーザ利用環境では，50%程度効率が低下していることが分かる．この効率低下は，使用時間と使用回数に比例して変化していることが分かる．この結果より，演習室がオープン利用時であってもジョブが実行できるため，長時間のジョブを実行できる．また，ユーザ

表 13　他方式との効率低下率の比較（単位：%）
Table 13　Degree of delay in 24 hours.

夜間利用	100.0
二重化	100.0
三重化	200.0
マイグレーション方式	50.4

表 14　実際環境での長時間のジョブ実行（単位：s）
Table 14　Long term job processing in real situation.

理想平均実行時間：t_{avg}	8031
ジョブ投入数：num_{job}	120
実測平均実行時間	9765
実測最大実行時間	21903
実測最小実行時間	7740
総実行時間：$total$	234368
平均マイグレーション回数	0.79
効率低下率：d(%)	21.6

に与える影響も計算用 PC の利用開始時に 100 秒程度の間負荷をかけるだけであり，ユーザが利用停止した後も自動で計算用 PC を起動できるため，ユーザは並列計算を気にせず計算用 PC を利用できる．

次に，夜間のみの利用やジョブの多重化を行った場合において，効率低下率を提案方式であるマイグレーション方式と比較し，その評価を表 13 に示す．なお，演習室のオープン利用時間は 1 日 12 時間とする．

表 13 より，マイグレーション方式が他の投入方法より効率的であることが分かる．また，投入 1 回当りのマイグレーション頻度が表 12 より 0.85 回であり，演習室でユーザの利用が集中した場合，マイグレーションが起こりやすく，多重化したジョブすべてが途中終了する可能性もある．本システムでは，多重化と異なりジョブが途中終了する危険がなく，長時間の実行にも対応できる優れたシステムであるといえる．

6.6　実際環境での実験

本実験では利用者が実際に使用している 5 台の計算用 PC(注1)にジョブ投入を行い，その結果を評価する．研究棟内での利用法は常時利用するものや週 1 回の利用にとどまるものなどまちまちである．この環境は，**6.5** で考察した演習室の環境と比べ使用頻度が少ない．この環境に都市数 1002(pr1002) のタブーサーチプログラムを投入し，その結果を表 14 に示す．

表 14 より，表 12 との平均マイグレーション回数の差が 0.06 であるにもかかわらず，効率低下率の差が 28.4%である．この理由は，実環境ではある一定期間に利用が集中しており，逆に長い遊休時間にジョブを実行することができたためと思われる．この利用が集中している期間では，すべての計算用 PC が使用中となったので，ジョブの多重化ではすべてのジョブが途中終了してしまう．このため，途中終了の起こらないマイグレーション方式が有効であるといえる．

(注1)：7 台中，1 台故障，1 台移動のため 5 台で実験する．

7. むすび

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，その評価を行った．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行った．その結果，仮想計算機の仮想化層がハードウェ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，実際の計算機の使用状況をもとにしたシミュレーションを行った．その結果，夜間のみの利用やジョブの中断に備えたジョブ多重化に比べて，効率の良いジョブ投入ができることが分かった．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　本研究を行うにあたり，多くの人々に御協力を頂いた．研究のシステム構築に携わって下さった富士通研究所の皆様，研究の場を提供して下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．

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，” 信学論（D-I），vol.J87-D-I, no.9, pp.845–854, Sept. 2004.

[3] 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,” Int. J. Parallel Program., vol.33, pp.209–229, 2005.

[4] E.-K. Byun and J.-S. Kim, “DynaGrid: A dynamic service deployment and resource migration frame-

work for WSRF-compliant applications," Parallel Comput., vol.33, pp.328–338, 2007.
[5] H. Chen, J. Chen, W. Mao, and F. Yan, "Daonity – grid security from two levels of virtualization," Information Security Technical Report, vol.12, no.3, pp.123–138, 2007.
[6] Condor Project: http://www.cs.wisc.edu/condor/
[7] 榎原博之，森川浩明，"仮想計算機を用いた PC グリッドの開発," RCSS Discussion Paper Series, no.79, 2009.
[8] Y. Fei, Z. Huanguo, S. Qi, S. Zhidong, Z. Liqiang, and Q. Weizhong, "An improved grid security infrastructure by trusted computing," Wuhan University J. Natural Sciences, vol.11, no.6, pp.1805–1808, 2006.
[9] R.J. Figueiredo, P.A. Dinda, and J.A.B. Fortes, "A case for grid computing on virtual machines," Distributed Computing Systems, vol.23, pp.550–560, 2003.
[10] FreeBSD jail, http://www.onlamp.com/pub/a/bsd/2003/09/04/jails.html
[11] グリッド協議会，http://www.jpgrid.org/index.html
[12] F. Heine, M. Hovestadt, O. Kao, and A. Keller, "SLA-aware job migration in grid environments," Advances in Parallel Computing, vol.14, pp.185–201, 2005.
[13] ITpro（編），すべてわかる仮想化大全 2009，日経 BP, 2008.
[14] 関西大学インフォメーションテクノロジーセンター（編），関西大学 IT センターフォーラム 2008, 関西大学インフォメーションテクノロジーセンター，no.23, 2008.
[15] 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，" 情処学研報，2005-MPS-54, 2005.
[16] B. Richard, N. Maillard, C.A.F. De Rose, and R. Novaes, "The I-cluster cloud: Distributed management of idle resources for intense computing," Parallel Comput., vol.31, pp.813–838, 2005.
[17] SETI@home: http://setiathome.berkeley.edu/
[18] 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.
[19] E.G. Talbi, Z. Hafidi, and J.-M. Geib, A parallel adaptive tabu search approach, 1998.
[20] 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.
[21] TSPLIB: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
[22] 山本芳嗣，久保幹雄，巡回セールスマン問題への招待，朝倉書店，1997.
[23] VMware: http://www.vmware.com/
[24] Xen: http://www.xen.org/

（平成 21 年 12 月 7 日受付，22 年 3 月 8 日再受付）

森川　浩明

2007 龍谷大・理工・電子卒．2009 阪市大大学院修士（創造都市）課程了．現在，阪市大大学院博士（創造都市）課程．組合せ最適化問題，並列計算，地理情報システム等の研究に従事．創造都市修．

榎原　博之　（正員）

1982 阪大・工・通信卒．1987 同大大学院博士（通信）課程了．同年阪大工学部助手．1994 関西大工学部専任講師となり，現在，准教授．組合せ最適化問題，計算幾何学，並列アルゴリズム等の研究に従事．工博．情報処理学会，IEEE，ACM 各会員．

大西　克実　（正員）

1992 阪大・工・通信卒．1994 同大大学院修士（通信）課程了．1996 阪市大学術情報総合センター助手．2005 阪市大創造都市研究科准教授となる．組合せ最適化問題，通信ネットワーク工学等の研究に従事．工修．情報処理学会，日本 OR 学会各会員．

中野　秀男　（正員）

1970 阪大・工・通信卒．1975 同大大学院博士（通信）課程了．同年阪大工学部助手．1991 阪大工学部助教授．1996 阪市大学術総合センター教授．2003 阪市大創造都市研究科教授となる．組合せ最適化問題，通信ネットワーク工学，暗号理論等の研究に従事．工博．情報処理学会，応用数理学会，日本 OR 学会，ソフトウエア科学会各会員．