

## グリッドネットワークのエミュレーション手法

笠井 武史<sup>†</sup> 西村 元一<sup>†</sup> 前田 高宏<sup>†</sup>  
大澤 清<sup>†</sup> 合田 憲人<sup>†</sup>

グリッドソフトウェア開発者に定量的なソフトウェアの評価を可能にする、グリッドエミュレーションシステムが開発されている。しかし、現状のシステムでは時系列に変化する動的なネットワーク性能の挙動の再現に問題がある。本稿では、グリッドにおける遅延時間やスループット、パケットロスといったネットワーク性能の時系列な変化を簡単にエミュレーションするシステムを提案する。提案システムでは、ネットワークシナリオファイルに従ってエミュレーションを行い、さらにそのシナリオファイル作成を支援するツールの開発を行った。システムの性能評価を行った結果、ユーザによる効率的なソフトウェアの評価が可能になることが示された。

### Emulation Technique of Network on the Grid

TAKEFUMI KASAI,<sup>†</sup> MOTOKAZU NISHIMURA,<sup>†</sup> TAKAHIRO MAEDA,<sup>†</sup>  
KIYOSHI OSAWA<sup>†</sup> and KENTO AIDA<sup>†</sup>

Grid emulation systems, which enable grid software developers to conduct quantitative evaluation of their software, have been developed. However, the existing systems have a problem of difficulty in emulating dynamic and time series behavior of network performance. This paper proposes a network emulation system that easily enables time series emulation of network behavior e.g. latency, throughput and packet losses, on the Grid. The proposed system emulates the network behavior following the network scenario file, which is provided by the user with file generation support tools. The experimental results show the performance of the proposed tool to help users conducting effective evaluation of their software.

#### 1. はじめに

近年、その利用が期待されているグリッドコンピューティングの分野では、グリッド上で動作するさまざまなアプリケーションやミドルウェアが開発されており、基盤技術が整備されつつある。このようなグリッド上で動作するソフトウェアを効率的に開発するには、ソフトウェアを複数回実行して性能を評価する必要があるが、ソフトウェア開発者が実グリッド実験環境で評価を行うには以下に挙げるような問題点がある。

- (1) 実グリッド実験環境を構築するためには、計算機やネットワークに関する専門的な知識が要求される。
- (2) 実グリッド実験環境は複数の研究機関が計算資源を互いに供与している場合がほとんどであり、個々の実験に許されるグリッド実験環境の利用機会は制限されることがある。

- (3) 実グリッド実験環境では計算機の負荷やネットワークの性能が時々刻々と変化するため、再現性のある評価を行うことが困難である。
- (4) ソフトウェア開発者が意図的にネットワークや計算機の障害を引き起こしソフトウェアの挙動を調べたい場合、実グリッド実験環境においてそのような障害を引き起こすことは困難である。

特に多くのグリッド環境に見られるように、グリッドを構築するネットワークとしてインターネットが用いられている場合は(3)、(4)の問題は顕著である。

これらの問題を解決するために、グリッド環境のシミュレーションを行う Bricks<sup>1)</sup> や SimGrid<sup>2)</sup> 等のシミュレータに関する研究が進められている。これらのシミュレーションでは、ソフトウェアの挙動を簡略化したモデルを用いて評価が行われ、スケジューリング等のアルゴリズムの評価には適している。しかし、ソフトウェアの開発では、プログラムを実行できる評価環境が必要であり、シミュレーションでは不十分である。

グリッド環境のエミュレーションについては、MicroGrid<sup>3)</sup>、Panda WAN Emulator<sup>4)</sup> や、主にネッ

<sup>†</sup> 東京工業大学  
Tokyo Institute of Technology

トワーク技術の評価を目的とした StarBed<sup>5)</sup>, EM-POWER<sup>6)</sup>, Lenet<sup>7)</sup> 等のエミュレーションシステムが開発されている。MicroGrid は、グリッド環境エミュレーションソフトウェアであり、グリッド環境を再現した PC クラスタ上で実際のソフトウェアを動作させることができる。ネットワークエミュレーションに関しては、下層レイヤーに至るまでネットワーク構成のより詳細な再現が可能である。しかし MicroGrid では、ソフトウェア実行中の計算機やネットワークの時系列なふるまいを再現することが困難であり、また設定ファイルの記述が非常に複雑で、ユーザの負荷が大きいといった問題点がある。

本稿では、グリッド環境のエミュレーションにおいて特に重要であるネットワークのエミュレーション技術に着目し、これらの問題を解決するためのグリッド上のネットワークのエミュレーションシステムを提案し、その実装について述べる。

提案システムは単一の PC クラスタ上に実装されており、ネットワーク性能のエミュレーションは、NIST-Net<sup>8)</sup> を用いている。また動的なネットワーク (WAN) のふるまいを再現するために、「ネットワークシナリオ」と呼ばれるネットワーク性能が時系列表記されたファイルを用いたエミュレーションを行う。シナリオ作成負荷を軽減するため、実グリッド実験環境上のネットワーク性能を収集しシナリオファイルを自動生成するツールや、ユーザが GUI 上の描画や数学モデルを用いてシナリオファイルを生成する支援ツールが用意されている。

提案システムを実装し、性能評価を行った結果、ネットワークの遅延やスループット、パケットロスといったネットワーク性能を時系列に変化させるエミュレーションが可能であることが確認された。これにより、開発者はより現実に即した、再現性のある評価を簡単かつ繰り返し行うことができる。また実グリッド実験環境では実現が困難であるネットワーク障害やパケットロスなどもユーザが意図したとおりに引き起こすエミュレーションが可能であることも確認された。

以降、2 節では提案システムの基盤となるエミュレーション環境について概説し、3 節では、ネットワークシナリオを用いたネットワークエミュレーション手法について述べる。4 節では提案システムの実装について述べ、5 節では、性能評価結果を示す。最後に 6 節で本稿のまとめと今後の課題について述べる。

## 2. ネットワークエミュレーション環境

本節では、提案システムの基盤となるエミュレーシ

ョン環境である擬似グリッド実験環境<sup>9)</sup> について概説する。擬似グリッド実験環境では、エミュレーション対象の PC クラスタやネットワーク (以降それぞれ、仮想 PC クラスタ、仮想ネットワークと表記) の処理が PC クラスタ上の計算ノードに割り当てられ、エミュレーションが実現される。このうち仮想計算ノードの処理は、PC クラスタ上の計算ノードで実行される。また仮想ネットワーク間の通信、即ち WAN 上の通信は、PC クラスタ上の PC ルータを介して行われ、本 PC ルータ上で仮想ネットワーク上の client-gateway 間の通信エミュレーションが実行される。現在、マスタ・ワーカ方式のソフトウェアの評価を目的としたシステムが実装されている。

## 3. ネットワークシナリオを用いたグリッドネットワークエミュレーション手法

本節では、ネットワークシナリオを用いて時々刻々と性能が変化するグリッド上のネットワークのエミュレーション手法について述べる。

本システムを利用するソフトウェア開発者のニーズとして、以下の 2 点が考えられる。

- (1) ユーザが意図した仮想的なネットワークの状況を再現したい
- (2) 実グリッド実験環境におけるネットワークの状況を再現したい

ここでそれぞれの場合において想定される本システムの具体的な利用例とシナリオ作成支援の方法について述べる。

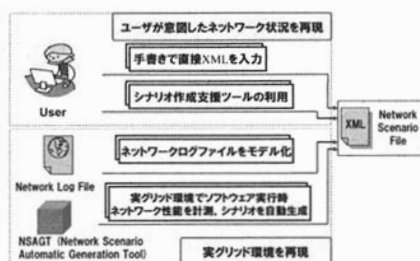


図 1 ネットワークシナリオの作成方法

### 3.1 仮想的なグリッド環境上での実験

仮想的なグリッド環境上での実験を行う利用例として、以下のような状況が想定される。

- グリッドにおける負荷分散手法を評価するため、グリッドソフトウェア実行中にネットワーク障害を意図的に引き起こして、ソフトウェアの挙動を調べ、正しく負荷分散が行われているか調べたい。

- パケットロスが恒常的に数%生じるようなネットワーク環境において、開発したソフトウェアが正しく動作するかどうか検証したい。

実グリッド実験環境でこれらの実験を行うことは困難である。しかし、擬似グリッド実験環境においては、どのくらいの期間、どの程度のネットワーク障害やパケットロスを引き起こすか、ネットワークシナリオファイルに記述するだけで実現することが可能である。

ユーザが意図したネットワーク状況を再現する場合、図1に示すようにユーザは、手書きでシナリオファイルを作成することも可能であるが、長時間にわたるシナリオになるとそのサイズが大きくなるため、ユーザの負荷が大きい。そのため本システムでシナリオ作成を容易にする「シナリオ作成支援ツール」が用意される。本ツールでは、GUI上でグラフを操作することで視覚的にネットワークシナリオを作成することが可能である。

### 3.2 実グリッド実験環境の再現

実グリッド実験環境の再現を行う例として、以下のような状況が想定される。

- 大規模な実グリッド実験環境で1度だけソフトウェアの評価を行った。ただし個人に提供されるその実験環境の利用期間は制限されている。そのため、評価を行った時と同種のネットワーク、計算機の状況を再現した上で細かいパラメータを変えつつ繰り返しソフトウェアの性能評価を行いたい。

この場合、図1に示したようにユーザは APAN(Asia-Pacific Advanced Network)<sup>10)</sup> などから提供されているネットワークログファイルをモデル化することによりネットワークシナリオを作成することができる。

また、本システムではネットワークシナリオ自動生成ツール“NSAGT: Network Scenario Automatic Generation Tool”を提供する。本ツールでは、実グリッド実験環境でグリッドソフトウェア実行時に同時にネットワーク性能を測定し、ネットワークシナリオファイルを自動的に生成することが可能である。NSAGTの機能の詳細については、4.3節で述べる。

## 4. 実装

ネットワークシナリオを用いたグリッドネットワークエミュレーションを実現するためのシステムの全体設計図を図2に示す。本システムは、「シナリオ作成部」、「ネットワークシナリオシステム部」、「擬似グリッド実験環境部」の3つの部分から構成されている。

ユーザがネットワークシナリオを用いて実際にネットワークエミュレーションを行うまでの手順は以下の

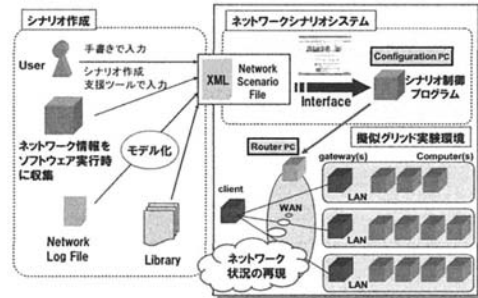


図2 グリッドネットワークエミュレーション手法の全体設計

ようになる。

### (1) ネットワークシナリオの作成

ユーザはまず「ネットワークシナリオファイル」を用意する。

ユーザは手書きで書くこともできるが、図中にあるようなネットワークシナリオ作成支援のためのツールを用いて用意することもできる。

### (2) シナリオファイルをシステムへ入力

ネットワークシナリオファイルが用意できたら、それを Web インタフェースを通してネットワークシナリオシステムへ入力する。

シナリオファイルをブラウザを通してアップロードするだけで、後の操作は、図中の“Configuration PC”上にある「シナリオ制御プログラム」がすべて自動的に行うため、ユーザは煩雑な設定ファイルを記述する必要がなく、簡単な操作性を実現している。

### (3) グリッドネットワークのエミュレーション

ユーザが Web インタフェース上でエミュレーション実行を指示することにより、ネットワークシナリオに基づくエミュレーションが実施される。この状態でユーザ評価対象のソフトウェアを2節で説明した擬似グリッド実験環境上で実行する。

次節以降は、細かい実装に関して詳説する。

#### 4.1 ネットワークシナリオファイルの記述仕様

ネットワークシナリオファイルは、ネットワークの動的な性能変化をソフトウェア実行開始時間を0とした時系列で記述する。現在のバージョン(ver.1.0.2)では以下の3つのネットワーク性能項目に関して記述することができる。

- ネットワーク遅延時間 [ms]
- スループット [Byte/sec]
- パケットロス率 [%]

シナリオファイルは XML を用いた記述仕様が定められている。XML を用いることで、ファイルの可読性を保つことができ、エディタを用いて容易に簡単なシ

```

<?xml version="1.0" encoding="UTF-8"?>
<network_scenario testbedname="Emulated Grid Testbed" location="AidaLab" version="1.0.2">
  <network_route source="g0.alab.ip.titech.ac.jp" destination="g1.alab.ip.titech.ac.jp">
    <scenario time="0">
      <delay>10</delay>
      <bandwidth>100000000</bandwidth>
      <packet_loss>0</packet_loss>
    </scenario>
    <scenario time="10">
      <delay>20</delay>
      <bandwidth>100000000</bandwidth>
      <packet_loss>0</packet_loss>
    </scenario>
    <scenario time="20">
      <delay>30</delay>
      <bandwidth>100000000</bandwidth>
      <packet_loss>0</packet_loss>
    </scenario>
  </network_route>
  <network_route source="g1.alab.ip.titech.ac.jp" destination="g0.alab.ip.titech.ac.jp">
    <scenario time="0">
      .....
    </scenario>
  </network_route>
  <network_route source="g2.alab.ip.titech.ac.jp" destination="g0.alab.ip.titech.ac.jp">
    <scenario time="0">
      .....
    </scenario>
  </network_route>
</network_scenario>

```

図 3 ネットワークシナリオの記述例

ナリオや細かいパラメータの変更を記述することができる。また、XML でシステムとしての汎用性も増す。

ネットワークシナリオファイルの記述例を図 3 に示す。この例では、2 台のホスト間 (g0.alab.ip.titech.ac.jp → g1.alab.ip.titech.ac.jp) で、各シナリオ時間 (0, 10, 20[s]) それぞれにおける遅延、スループット、パケットロス率を記述している。この場合は遅延時間のみ 10, 20, 30[ms] と変化させている。

#### 4.2 シナリオ作成支援ツール

本システムでは、ユーザが意図したネットワークの状況を再現する際にネットワークシナリオの作成を簡単化するシナリオ作成支援ツールを提供している。本ツールでは、図 4 に示すような GUI 上でグラフを作成することで、ネットワークの性能変化を視覚的に表現しながら、シナリオを作成することができる。また、想定するシナリオが特定の数学モデルに従う場合は、数式を入力することも可能である。

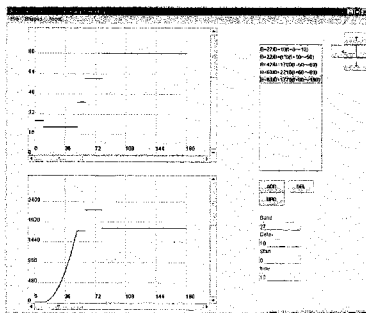


図 4 シナリオ作成支援ツールの GUI 画面

#### 4.3 ネットワークシナリオ自動生成ツール

ネットワークシナリオ自動生成ツールは、実グリッド実験環境上でグリッドソフトウェアを実行している際、同時に本ツールを動作させるだけで該当する複数ホスト間のネットワーク遅延時間を測定し、自動的にネットワークシナリオファイルを生成する。

遅延時間の測定には ping を用いている。ユーザは本ツールの設定ファイルに ping を実行する時間間隔、各 gateway のホスト名 (または IP アドレス) などを記述する。ツールが実行されると、各 gateway ごとに遅延時間測定結果を一時ファイルに保存し、遅延時間データの収集が完了後、それらをまとめて 1 つのネットワークシナリオファイルとして生成する。

#### 4.4 ネットワークシナリオシステム制御プログラム

Web インタフェースを通して入力されたネットワークシナリオファイルは図 2 における“Configuration PC”上に置かれたネットワークシナリオ制御プログラムに読み込まれる。エミュレーションを行うまでの本プログラム動作は以下ようになる。

##### (1) ネットワークシナリオファイル (XML) の解析

書かれたネットワークシナリオが XML 仕様記述に沿ったものであるかどうか、解析する。

##### (2) ネットワーク性能値の補正

ユーザは、本システムを構築した PC クラスタ間の物理的なネットワーク性能値 (遅延時間、スループット、パケットロス率、NISTNet のオーバーヘッドによるスループットロス率) をシステムの設定ファイルにあらかじめ記述しておくことができる。シナリオ制御プログラムはこれらの値を用いて、シナリオに書かれた遅延時間から物理的な遅延時間を差し引いたり、NISTNet のオーバーヘッド分、スループットの設定値を増加させるなどの補正を行う。また、シナリオに書かれたネットワーク性能値が物理的な性能値を超えている場合は、エミュレーションは行われない。

ネットワーク障害発生がシナリオに記述されていた場合、NISTNet での設定を packet loss を 100% とし、“Router PC”を通るすべてのパケットを破棄させることで実現する。

##### (3) NISTNet 制御スクリプトの生成

補正された値をもとにネットワークエミュレータ“NISTNet”の設定を動的に変更するスクリプトが生成される。

##### (4) NISTNet 制御スクリプトの実行

ユーザが Web インタフェース上でエミュレーション実行を指示すると、制御スクリプトが図 2 における“Configuration PC”から NISTNet がインストー



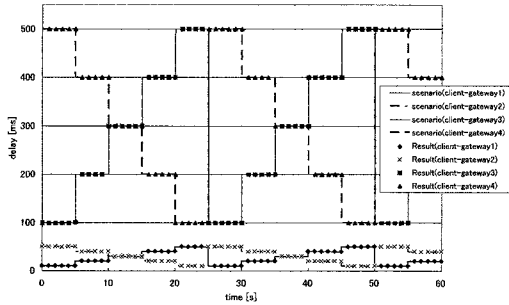


図5 client(1台)-gateway(4台)間の遅延時間の再現性評価結果

ルされた“Router PC”に向けて実行され、時系列に沿ったネットワークのエミュレーションが行われる。

## 5. 性能評価

本節では、シナリオの再現性評価を遅延時間、スループット、パケットロス率に対して行った。なお、パケットロス率については他と同様に再現性が確認できたので、紙面の都合上、結果の掲載を省略する。

提案システムの実装を行ったエミュレーション環境の計算機構成の詳細を表1に示す。

### 5.1 遅延時間シナリオの再現性評価

遅延時間の測定は、pingを用いて行った。

まず、client(1台)-gateway(1台)間において、シナリオ時間間隔を1秒として遅延時間を変動させた時、1秒間隔で変動するシナリオを十分再現できることが確認された。

次に、gatewayホストを複数台にした場合、シナリオ時間間隔を5秒にして同様の実験を行った。図5にclient(1台)-gateway(4台)間の用いたシナリオと遅延時間測定結果を示す。gatewayが複数台になっても、シナリオに記述されたとおりに遅延時間が動的に変動していることが確認できる。

### 5.2 スループットシナリオの再現性評価

スループットシナリオの再現性評価は、Netperf<sup>[11]</sup>を用いて該当ホスト間のスループットを測定することで行った。図6に評価に用いたシナリオと測定結果を示す。

実験に先立ち、提案システムを実装したホスト間の最大スループットを測定したところ、およそ38.5[MB/sec]であったため、それを超えるスループットは再現することができない。図中、0~40[s]時は、設定スループットが最大スループット以上であるため、NISTNetによるエミュレーションは行われていない。40[s]以降、最大スループットに近い値をスループットとしてシナリオに記述すると、設定したスループットと測定されるスループットの間には差異が見られ、最大で20%程度測定値の方が小さくなった。これは、NISTNetを用いてエミュレーションを行う時のオーバーヘッドであると考えられる。

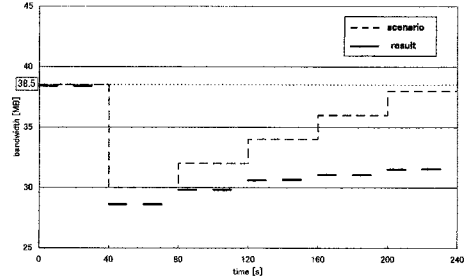


図6 スループット再現性評価

また、本実装での最大スループット38.5[MB/sec]に対し、およそ30[MB/sec]以下の値をシナリオ中にスループットとして設定した場合には、エミュレーションの精度は高まるが、この場合でも設定値と実測値との間に4~5%程度の誤差が生じる。しかし、30[MB/sec]以下の設定値の時には、恒常的に4~5%であるため、シナリオ制御プログラムがその恒常的な誤差を勘案して補正することによってより精度の高い再現性が実現できる。

### 5.3 ベンチマークプログラムを用いたスループットシナリオ再現性評価実験

本節では、行列積計算を行う実際のソフトウェアを用いたスループットシナリオの再現性評価実験について述べる。本ベンチマークプログラムは、Ninf-G<sup>[12]</sup>のAPIから得られる計算に用いる入力行列データをclientから各gatewayへ送信する時と計算結果の出力行列データをgatewayからclientへ返す時のそれぞれにおいて転送データサイズと転送時間からスループットを算出した。

本実験に用いたシナリオと結果を図7に示す。各計算ノードで計算を行っている最中に各ホスト間のスループットが変化するシナリオを想定している。5.2節で述べたスループットの誤差分(5%)は、シナリオ制御プログラムにおいてNISTNet制御スクリプトを生成する際、自動的に補正されている。図よりシナリオどおりのスループットが本提案システムを用いて再現できていることが確認された。

## 6. まとめと今後の課題

本稿では、グリッドソフトウェア開発者により効率

表 1 擬似グリッド実験環境の計算機構成

	CPU	Memory	OS	ノード数	ネットワーク仕様
client	Pentium4(2.4GHz)	1GB	Linux2.4.18	1	1000BASE-TX
Router PC	Pentium4(2.4GHz)	1GB	Linux2.4.7	1	1000BASE-TX
Configuration PC	Pentium4(2.4GHz)	512KB	Linux	1	1000BASE-TX
gateway	Pentium4(2.4GHz)	512KB	Linux2.4.7	4	1000BASE-TX
Worker	Pentium3(1.4GHz)Dual	512KB	Linux2.4.10	36	1000BASE-TX

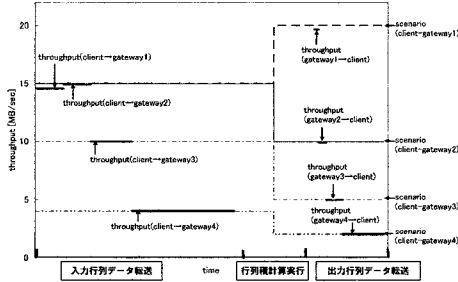


図 7 ベンチマークプログラムを用いた再現性評価実験結果

的な評価環境を提供するため、グリッド上のネットワークのエミュレーションシステムを提案し、その実装について述べた。ネットワーク性能が時系列に記述されたネットワークシナリオを用いて、ネットワークエミュレータ NISTNet の設定を動的に変化させることで、グリッドネットワークの時系列なふるまいを再現できることが確認され、本提案システムが有効であることが確認された。また、本システムでは、ユーザのニーズごとにより簡単なシナリオ作成手法とその支援ツールを提供した。

現状のネットワークシナリオに加え、計算機の動的な性能変化を実現するための計算機シナリオとそのシステム実装を進めている。また、現状では物理的な性能を超えるシナリオが記述された場合、それを再現することができないが、今後は VM と仮想時間を用いて例えば物理的な計算機数よりも多い計算機を想定した実験環境でのソフトウェアの評価を行えるよう検討中である。

謝辞 本研究の一部は、科学研究費補助金基盤研究(B)(課題番号 18300018)による。

### 参考文献

- 1) Takefusa, A., Matsuoka, S., Nakada, H., Aida, K. and Nagashima, U.: Overview of a Performance Evaluation System for Global Computing Scheduling Algorithms, *In Proceedings of 8th IEEE International Symposium on High Performance Distributed Computing (HPDC8)*, pp. 97–104 (1999).
- 2) Casanova, H.: Simgrid: A Toolkit for the

Simulation of Application Scheduling, *Proceeding of the First IEEE/ACM International Symposium on Cluster Computing and the Grid(CCGrid2001)*, pp. 430–437 (2001).

- 3) Song, H. J., Liu, X., Jakobsen, D., Bhagwan, R., Zhang, X., Taura, K. and Chien, A. A.: The MicroGrid: a Scientific Tool for Modeling Computational Grids, *Supercomputing* (2000).
- 4) Kielmann, T., Bal, H., Maassen, J., van Nieuwpoort, R., Eyraud, L., Hofman, R. and Verstoep, K.: Programming environments for high-performance grid computing: the albatross project (2002).
- 5) Miyachi, T., Chinen, K. and Shinoda, Y.: Automatic Configuration and Execution of Internet Experiments on an Actual Node-based Testbed, *Tridentcom 2005*, pp. 274–282 (2005).
- 6) Zheng, P. and Ni, L. M.: EMPOWER: A Cluster Architecture Supporting Network Emulation, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 7, pp. 617–629 (2004).
- 7) 石野正英, 前田香織, 河野英太郎, 石田賢治: ネットワークエミュレータ Lenet の RTCP を用いたキャプチャ・リプレイ機能の設計, *IC2006*, pp. 59–65 (2006).
- 8) Carson, M. and Santay, D.: Tools: NIST-Net: a Linux-based network emulation tool, *ACM SIGCOMM Computer Communication Review*, Vol. 33 (2003).
- 9) 大角知孝, 合田憲人: ネットワークエミュレーションを用いたグリッドアプリケーション負荷分散手法の評価, *シミュレーション*, pp. 104–111 (2005).
- 10) APAN: Asia-Pacific Advanced Network. ([www.apan.net](http://www.apan.net)).
- 11) NetPerf: NetPerf WWW pages and NetPerf manual included in archive. ([www.netperf.org/netperf/NetperfPage.html](http://www.netperf.org/netperf/NetperfPage.html)).
- 12) Takemiya, H., Shudo, K., Tanaka, Y. and Sekiguchi, S.: Development of Grid Applications on Standard Grid Middleware (2003).