社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－HPC－107 (13)
2006／8／1

仮想計算機と仮想ネットワークを用いた仮想クラスタの構築

西村豪生[†] 中田秀基[††,†] 松岡 聡[†,†††]

グリッド環境の不均質性を隠蔽するために，分散した資源上に仮想計算機と仮想ネットワークを用いて仮想クラスタを構築する手法が注目されている．仮想クラスタ構築のためには，必要な環境構成を備えた仮想計算機 (VM) イメージを実計算資源に配布する必要がある．しかし，一般的に VM イメージのサイズは数 100MBytes から数 GBytes に及ぶため，その転送時間は無視できない．既存研究ではある程度高速な仮想クラスタ構築システムを提供しているものの，実行環境に制限がある．そこで我々は，ユーザが望む環境を備えた VM イメージを動的に高速に作成する仮想クラスタ構築システムを提案する．本システムでは利用頻度の高いパッケージ構成を含んだキャッシュイメージを自動的に生成する．また，事前に構築時間を見積もってキャッシュを用いることにより，構築に 103 秒程度要していた仮想クラスタを，75 秒程度に短縮できることを確認し，高速化への指針を得た．

Virtual Cluster with Virtual Machines and Virtual Network

HIDEO NISHIMURA ,[†] HIDEMOTO NAKADA [††,†]
and SATOSHI MATSUOKA [†,†††]

Recently, a virtual cluster constructed by using a virtual machine and a virtual network attracts attention as the technique of hiding heterogeneous of the grid environment. It is necessary to distribute a VM image which has requested environment to the real computing resources for constructing proper virtual cluster. However, the transfer time of the VM image cannot be generally disregarded,since those sizes have several GBytes from 100MBytes. In an existing research, there is a limitation in the execution environment though a comparatively high-speed virtual cluster construction system is advocated. Then, we propose the virtual cluster construction system that makes the environment for which the user hopes at dynamically and high speed. This system automatically generates cache images that contain packages composition frequently used. Moreover, due to estimating the construction time beforehand and using cache, we confirmed the construction time was shortened from about 100 seconds at about 75 seconds, and obtained the indicator to speed-up.

1. はじめに

近年，グリッド環境の不均質性を仮想化技術で隠蔽し，その上に仮想クラスタを構築して計算資源として用いる手法が注目されている[1)~4)]．仮想クラスタによって実資源の分散が隠蔽されるため，ユーザは既存のプログラムを改変することなくそのまま仮想クラスタ上で実行することが可能である．また，管理者にとっては各ユーザの要求毎に実計算環境を変更する必要がないため，プロビジョニングの負担が軽減される．

仮想クラスタシステムの多くは仮想計算機 (VM)[5)~7)] を計算基盤として用いており，構築のためには VM イメージファイルを用意して各実計算資源に配布する必要がある．そのような手法の問題点の一つとして，所望の計算環境を備えた VM イメージ準備の煩雑さが挙げられる．既存システムの多くは，OS や必要なライブラリなどをあらかじめインストールした VM イメージを事前に用意する必要がある[2),3)]．VM イメージの作成は容易であるとは言えないので，一般ユーザが個別に用意するのは負担が大きい．また，管理者が用意した VM イメージを用いる方法もあるが，それはユーザにとって柔軟な環境構成であるとはいえない．

また，VM イメージを各実計算資源に転送するため，仮想クラスタの構築に時間がかかるという問題点がある[1),3),8)]．多くの場合，VM イメージは数 100MBytes から数 Gbytes に及ぶため，グリッドを構成する各サイトへの転送時間は無視できない．

そこで我々は，グリッド上に仮想クラスタを高速かつ簡便に構築するシステムを提案する．本システムでは，VM イメージをユーザの要求に応じて動的に作成するため，ユーザや管理者が事前に VM イメージを用

† 東京工業大学
Tokyo Institute of Technology
†† 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
††† 国立情報学研究所
National Institute of Informatics

意する必要がない．また，インストール要求を統計的に分析することで，よく用いられるパッケージを事前に含んだキャッシュイメージを自動生成し，サーバに保持する．実行時にはインストール要求に応じて適切なキャッシュイメージを自動選択し，差分パッケージのみをインストールすることで高速な環境構築が可能である．

プロトタイプ実装では仮想計算機（VM）システムとして仮想計算機モニタ Xen[5]，仮想ネットワークとして仮想プライベートネットワーク（VPN）OpenVPN[9] を用いた．また，自動環境構築にはクラスタ自動インストーラツール Lucie[10] を改変して用いた．本システムにより，C/C++ 開発環境と MPI 並列計算環境を備えた 16 台の VM からなる仮想クラスタが 78 秒程度で構築できた．

本稿の以降の構成について述べる．2 章ではグリッド上の仮想クラスタ構築システムに対する要件および構想について延べる．3 章ではプロトタイプシステムの設計および全体構成について述べ，4 章ではその実装について述べる．5 章ではその評価結果，6 章では関連研究について述べる．最後に 7 章で本稿のまとめと今後の課題を挙げる．

2. グリッド上における仮想クラスタ構築システムの要件

グリッドを構成する計算資源は，一般的に広域の異なった管理ドメインに分散している．そのような分散をユーザから隠蔽して仮想クラスタを提供するシステムについては，以下の要件が存在する．

VM の高速な配備　仮想クラスタは，ユーザが望む環境を備えた VM を実資源上に配置することによって構築される．仮想クラスタ構築を高速化するためには VM の配置を高速に行わなければならない．その詳細な要件に関しては 2.1 節で述べる．

ネットワーク不均質性の隠蔽　グリッドを構成する各 PC クラスタが異なるネットワークドメインに属している場合，異なったクラスタに属する計算機同士は直接通信することができない．しかし，ユーザが仮想クラスタであることに対して透過的にアプリケーションを実行するためには，計算資源が非対称ネットワークで接続されていることを隠蔽しなくてはならない．そのためには実ネットワーク上にオーバーレイネットワークを形成し，仮想的に対称なネットワークを実現する必要がある．

効率的な資源選択　仮想クラスタを構築するためには，グリッドを構成するクラスタなどから適切な計算資源を選択する必要がある．多サイトから計算資源を調達した場合，その間の通信は WAN を介するためネットワーク性能は低下する．そのため，できる限り単一ネットワークドメイン内のみでの調達が望ましい．

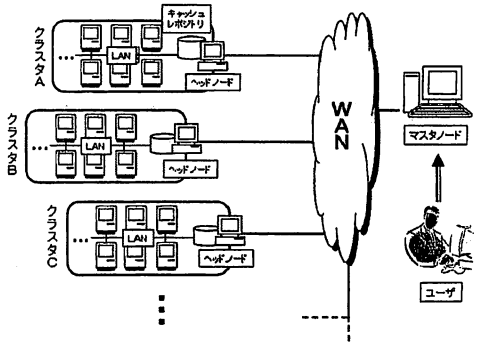

図 1　本システムの全体構成

本稿では，VM の配備を高速化することにより，仮想クラスタ全体の構築を高速化することを主眼に置いた仮想クラスタ構築システムについて議論する．

2.1 簡便にカスタマイズ可能な計算環境を高速に提供

仮想クラスタの計算環境指定は，ユーザにとって簡便に行うことができなければならない．ユーザが各々 VM イメージを用意するのは負担が大きいので，使用したい計算環境のハードウェアスペック（メモリ量，ディスクなど）とソフトウェア構成（OS やパッケージ）を指定するだけで VM 内に所望の環境が構築されて起動し，ユーザに仮想クラスタが提供されるようなシステムが望ましい．

また，仮想クラスタ上でのアプリケーション実行時間に対し，構築に大きな時間を要するようでは利便性が高いとは言えない．我々の予備評価によると，最小構成で 47 秒程度であったインストール時間が MPI 並列ライブラリ環境を備えた仮想クラスタとしてカスタマイズすることによって 73 秒程度に増加することがわかっている．このように，環境をカスタマイズすることにより全体の環境構築時間は増加する．

理想的には，数百から数千台からなる仮想クラスタが 10 秒程度で構築され，ユーザに提供されるのが望ましい．

3. 仮想クラスタ構築システムの設計

前章で仮想クラスタ構築システムの要件を大きく 3 つ挙げたが，本稿では 2.1 で述べた要件について注目した仮想クラスタの設計について述べる．

3.1 システムの全体構成

グリッド環境を構成する計算資源としては，PC クラスタが広く用いられている．そこで，本システムは図 1 のような構成の環境を前提としている．各クラスタは一つのヘッドノードを有している．ヘッドノード

にはグローバルアドレスが割り振られているが，クラスタを構成する各計算ノードにはプライベートアドレスのみが割り振られている．

マスタノードはユーザがサービスを受けるフロントエンドであり，ヘッドノードは各クラスタの計算ノードに対して VM の起動，環境構築を行う．また，各サイトにはキャッシュレポジトリが存在し，よく使われるパッケージ構成のイメージを生成，保持している．

3.2 マスタノード

マスタノードはフロントエンドノードであり，各ユーザはマスタノードにアクセスして仮想クラスタ構築の要求を行う．

ユーザは使用したいディスクやメモリ量などのハードウェアスペック，インストールしたいパッケージを指定し，仮想クラスタの環境設定を行う．これらの環境指定は GUI を通じて対話的に行うことにより，ユーザの利便性を高める．

また，仮想クラスタのディスクにアップロードしたいファイルの指定などもマスタノードで行われる．環境設定が行われると，マスタノードはどのクラスタを計算資源として用いるかのスケジューリングを行う．計算資源が決定すると，そのクラスタのヘッドノードに環境構成やユーザがアップロードしたファイルを転送し，各サイトで環境構築を開始する．

3.3 ヘッドノード

ヘッドノードはクラスタの各計算ノードで VM を立ち上げ，そのインストールを行う．

ヘッドノードはマスタノードから環境構築要請を受け取ると，そのパッケージ構成をキャッシュレポジトリへ伝え，最も適したキャッシュイメージを得る．その後クラスタ内の各計算ノードで VM を起動し，得られたキャッシュイメージからの差分についてクラスタ自動インストーラツールを用いて環境の構築を行う．

3.4 計算ノード

各計算ノードはヘッドノードからの指示で VM を立ち上げ，計算資源として用いられる．

3.5 キャッシュレポジトリ

キャッシュレポジトリは以下の二つの役割を担っている．

(1) 最適なキャッシュイメージの選択
キャッシュレポジトリはヘッドノードから要求を受けると，保持しているキャッシュイメージの中から最適なものを選択して返す．一般に，キャッシュイメージのサイズはその中に含まれるパッケージの量に比例する．最適イメージの選択は，キャッシュイメージのサイズから見積もられる各計算ノードへのコピー時間と，パッケージのインストール時間削減効果のトレードオフを考慮して行われる．

(2) キャッシュイメージの自動生成 キャッシュレポジトリはヘッドノードからのパッケージ要求を蓄積し，それを統計的に解析することで利用頻度の高いパッケージを選定する．それらのパッケージは今後も高頻度で利用されることが期待されるため，最小構成のイメージにそれらのパッケージを追加でインストールし，キャッシュイメージとして保持する．単純なアルゴリズムとして，過去 N 個の要求のうちで M% 含まれているパッケージを利用頻度の多いパッケージとして選定する．

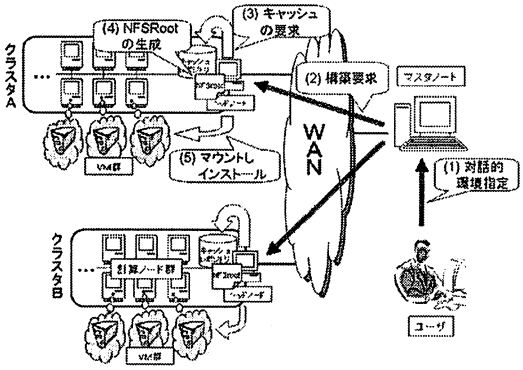

図 2 環境構築の流れ

4. プロトタイプ実装

本章では，前章で提案した仮想クラスタ構築システムのプロトタイプ実装について述べる．

4.1 計算環境とネットワークの仮想化

プロトタイプ実装では，仮想計算機システムとして仮想計算機モニタ Xen[5] を用い，ネットワークの仮想化に OpenVPN[9] を用いた．プロトタイプ実装ではマスタノードを VPN サーバ，各 VM をクライアントとして仮想ネットワークを形成した．しかし，OpenVPN を用いることによってネットワーク性能は大きく低下する[11] ため，VIOLIN[12] のような異なる仮想ネットワークの利用を今後検討する必要がある．

4.2 自動環境構築

プロトタイプ実装では環境構築にクラスタ自動インストーラツール Lucie[10] を本システムのために改変して用いた．Lucie は，事前にインストールするパッケージや構成を記述しておくことにより，クラスタを構成する計算機に対して柔軟な環境構築を行うツールである．

一連の環境構築の流れを図 2 に示す．

(1) ユーザはマスタノードで GUI による Lucie の対話的環境設定を行う．
(2) マスタノードは使用する実資源のヘッドノードへ仮想クラスタの構築を要求する．
(3) ヘッドノードはキャッシュレポジトリからキャッシュイメージを得る．
(4) ヘッドノードはインストール用の NFSRoot

表 1 評価環境

	マスタノード	ヘッドノード	計算ノード
CPU	AMD Opteron 250		AMD Opteron 280
RAM	1GB	2GB	4GB
Network	1000BASE-T		
OS	Linux-2.4.31		Linux-2.6.12.6-xen
VM	-		Xen-3.0.1
VPN	OpenVPN-2.0	-	

を生成し，各 VM がマウントできるように DHCP，NFS サーバの設定を行う．

(5) 各計算ノードで VM がヘッドノードの NFSRoot をマウントして起動し，キャッシュイメージをローカルディスクに展開してインストール処理を開始する．

(6) インストールが終了すると VM は再起動し，ユーザに仮想クラスタが提供される．

現在のシステムではイメージの転送を NFS を通じて行っているが，構築台数が数百から数千になった場合は Dolly+[13] などの高速ファイル転送機構を用いるべきである．

4.3 キャッシュの生成・利用

プロトタイプ実装ではキャッシュレポジトリはヘッドノードに存在する．また，キャッシュイメージとして Debian のブートイメージ (/bin/，/lib/ など) を圧縮したアーカイブを用いる．アーカイブは NFSRoot 内に配置され，各 VM は NFS 経由でローカルディスクへとブートイメージを展開する．

キャッシュの生成に関しては 3.5(2) で述べたアルゴリズムを用いる．すなわち，過去の要求の中で一定以上含まれていた deb パッケージを利用頻度が高いパッケージであると選定する．選定されたパッケージを最小構成ブートイメージに追加インストールし，そのブートイメージを圧縮したアーカイブをキャッシュイメージとして保持する．

キャッシュの利用に関しては，キャッシュイメージの転送時間とパッケージインストール時間削減のトレードオフを考慮したアルゴリズムを用いる．パッケージの事前インストールによって VM での動的インストール時間は減少するが，ブートイメージのサイズは増大するので，NFS 経由の転送時間は増加する．よって，パッケージの動的インストール時間減少が転送時間の増加時間を上回る場合のみキャッシュを用いる．

キャッシュイメージの転送時間は VM の台数に比例するため，過去の転送時間を参照することによって転送時間を見積もる．キャッシュレポジトリは，各キャッシュイメージに対して利用の度に転送に要した時間を記録し，再利用時に参照する．

パッケージのインストール時間に関しては，過去の仮想クラスタ構築の際のインストール時間を参考にする．プロトタイプ実装では予備評価 (図 3) に基づき，deb パッケージのインストール時間がその容量に比例すると仮定して見積もりを行う．

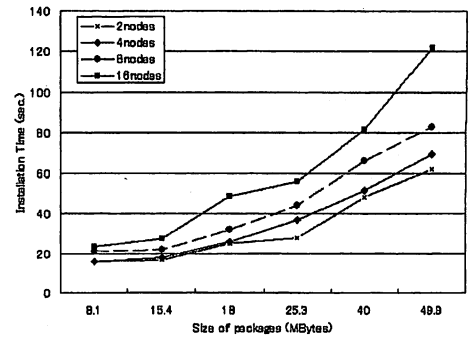

図 3 deb パッケージのサイズとインストール時間

5. 評　　価

本章では，プロトタイプ実装を用いた仮想クラスタ構築時間の評価を行う．評価環境としては東京工業大学 PrestoIII クラスタを用いた．環境構成は表 1 の通りである．

異なるパッケージ構成の仮想クラスタについて，最小構成の Debian イメージからインストールした場合と，共通したパッケージをあらかじめ含んだキャッシュイメージからインストールした場合の構築時間をそれぞれ計測した．なお，VM のスペック要求としては，1.6GB のディスク容量，320MB のメモリ容量とした．

計測に用いたパッケージ構成を以下に示す．なお，括弧内はインストールに必要なパッケージ数とその容量である．

構成 1 C/C++ と MPI の開発環境
(47packages，18.0MBytes)

構成 2 数値計算ツール Octave[14] と MPI 並列環境
(51packages，25.3MBytes)

事前に上記の構成の仮想クラスタ構築要求が複数回存在したし，共通するパッケージを含んだキャッシュイメージがキャッシュレポジトリに存在することを仮定する．上記の構成に共通するパッケージは 38 個であり，その容量は 9.9MB 程度であった．事前にパッケージをインストールすることによって，最小構成 46.0MB 程度であったブートイメージは 61.3MB 程度のキャッ

表 2 イメージ転送時間の見積もり/実測値 (秒)

	最小構成 (46.0MB)	キャッシュ(61.3MB)
4nodes	6.44 / 6.52	8.66 / 8.44
8nodes	6.76 / 6.58	9.03 / 9.11
16nodes	7.44 / 7.45	10.3 / 10.4

表 3 構成 1 のインストール時間見積もり/実測値 (秒)

	キャッシュなし	キャッシュあり
4nodes	25.7 / 26.7	12.8 / 17.1
8nodes	32.0 / 34.3	16.5 / 18.7
16nodes	41.5 / 47.0	19.1 / 23.2

表 4 構成 2 のインストール時間見積もり/実測値 (秒)

	キャッシュなし	キャッシュあり
4nodes	35.2 / 29.7	22.3 / 17.1
8nodes	43.4 / 39.6	27.9 / 21.2
16nodes	58.0 /54.8	35.6 / 24.4

シュイメージとなった.

イメージ転送時間の見積もりおよび実測値を表 2 に示す. イメージ転送時間の見積もりは, キャッシュイメージの過去の転送時間の実測値の平均である. 結果から, 過去の実測値の平均を用いることで 3% 以内の誤差で見積もりが行えていることを確認した.

図 3 の予備評価で得られた実測値を過去の履歴とし, 近似直線を用いてインストール時間の見積もりを行った. 構成 1 と構成 2 に関して, パッケージインストール時間の見積もりと実測値をそれぞれ表 3 と表 4 に示す. 構成 1 に関しては最大でも 18% 程度の誤差で見積もりを行えているが, 構成 2 では最大 46% 程度の誤差が生じてしまった. この原因として, プロトタイプ実装ではパッケージサイズとインストール時間が比例するとした仮定がある. 今後, パッケージのインストール時間に関してより詳細な調査と見積もり方法の検討をする必要がある.

仮想クラスタ全体の構築時間の内訳を図 4 に示す. VM の台数が増えるにつれて増加する部分はパッケージインストール時間と NFS 経由のイメージ展開である. この結果より, 表 234 で示した通り, 予測通りに適切なキャッシュを使用することによってイメージ展開部分はやや増大するが, それ以上にパッケージインストール時間が削減されることが確認された. 今後, パッケージインストール時間とイメージ展開時間について考慮した改良や, その他の処理を改善することでさらなる高速化が期待できる.

6. 関連研究

VMPlants[1] は, 本システムと同様にグリッド上に仮想クラスタを構築するシステムである. ユーザは最小公倍数的な Golden Image からの環境の差分を DAG を用いて手続き的に記述することにより, 高速な環境構築を行う. しかし, ゴールデンイメージを管理者が事前に用意する必要がある. 本研究は, そのような共

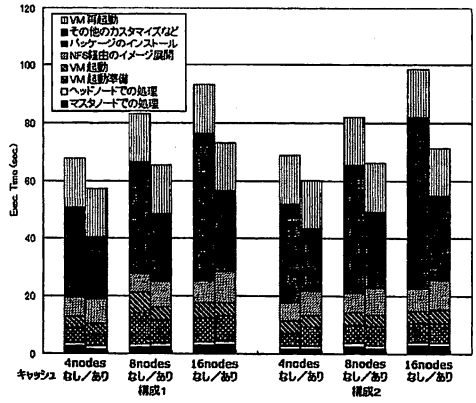

図 4 構築時間

通イメージを解析的に分析・判断し, 自動的にキャッシュイメージを生成する. また, DAG による構築手順の記述は, OS やソフトウェアの設定方法に熟知している必要がある. 我々のシステムは Lucie のメタパッケージを用いることにより, GUI によって機能単位に抽象化されたセットアップが可能である.

ORE Grid[4] は, Globus Toolkit[15] のジョブ実行機構 GRAM と連動した仮想計算環境提供システムである. 投入されたジョブに対して, ユーザが指定した環境を VM 内に Lucie を用いて動的に構築することで実行する. また, 使用後のイメージをキャッシュサーバ[16] に保持して再利用することで構築時間を短縮している. 我々のシステムが用いるキャッシュは, 使用されたイメージそのものではなく, 高頻度に用いられている共通パッケージを事前にインストールしたイメージである. また, ORE Grid はユーザが投入した個々のジョブに対して VM を構築するのに対し, 我々のシステムは, ユーザに構築した仮想クラスタそのものを計算資源として提供する.

Virtual Workspaces[3] は, あらかじめ作成しておいた VM イメージを複製し, 仮想クラスタを構築するシステムである. しかし, VM イメージのサイズが大きくなるにつれて計算資源へのコピー時間が増加し, 仮想クラスタ全体の構築時間も増加する. 我々のシステムは静的に用意された VM イメージを計算資源へ転送するのではなく, ユーザの要求に合わせて動的に VM イメージを構築するため高速である.

The Virtuoso Model[17] は, グリッド上の VM を仮想ネットワークでユーザのローカルネットワークに接続し, 計算資源として提供するシステムである. しかし, 基本的なソフトウェア構成のみが事前にインストールされているだけであるので, ユーザが独自に使用したいアプリケーションは個々にインストールする必要がある. そのため, 仮想クラスタの規模が数百から数千台の規模になったときの負担が大きい. 我々の

システムは事前に環境構成を指定することにより，環境性は全て自動的に行われる．

VioCluster[2] は仮想ネットワークとして VIOLIN[12] を用いた，多サイトにまたがる仮想クラスタ構築システムである．しかし，VM イメージは事前に準備され，各サイトに転送されていることを仮定している．

7. おわりに

本稿では，キャッシュを用いた高速な仮想クラスタ構築システムを提案し，プロトタイプを実装した．過去のイメージ転送時間とパッケージインストール時間を用いて事前にキャッシュ使用による構築時間を見積もり，最適なキャッシュ選択を行う．キャッシュの使用により，構築に 100 秒程度かかっていた仮想クラスタを 75 秒程度に短縮した．

本稿で挙げた設計およびプロトタイプ実装はまだ途上であった．提案したキャッシュを用いた高速な構築システムに対して，以下の課題が存在する．

- 誤差が目立ったパッケージインストール時間見積もりアルゴリズムの改善
- キャッシュの生成・利用アルゴリズムの評価，改良
- 評価を元にした構築時間削減のためのチューニング
- 大規模環境での評価
- NFS ではなく Dolly+[13] などのファイル転送ツールの利用

また，仮想クラスタ構築システムとしては仮想ネットワークの充実や資源調達アルゴリズムの考案なども今後の課題である．

謝辞 本研究の一部は，科学技術振興機構・戦略的創造研究「低消費電力化とモデリング技術によるメガスケールコンピューティング」による．

参考文献

1) Krsul, I., Ganguly, A., Zhang, J., Fortes, J. and Figueiredo, R.: VMPlants: Providing and Managing Virtual Machine Execution Environments for Grid Computing, IEEE/ACM SC'04 (2004).
2) Ruth, P., McGachey, P. and Xu, D.: VioCluster: Virtualization for Dynamic Computational Domains, *Proceedings of the IEEE International Conference on Cluster Computing* (2005).
3) Foster, I., Freeman, F., Keahey, K., Scheftner, D., Sotomayor, B. and Zhang, X.: Virtual Clusters for Grid Communities, *CCGRID 2006* (2006).
4) 高宮安仁, 山形育平, 青木孝文, 中田秀基, 松岡聡: ORE Grid：仮想計算機を用いたグリッド実行環境の高速な配置ツール, 先進的計算基盤システムシンポジウム SACSIS2006, pp. 541–550 (2006).
5) Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A.: Xen and the Art of Virtualization, *Proceedings of the ACM Symposium on Operating Systems Principles* (2003).
6) VMware. http://www.vmware.com.
7) Dike, J.: A User-Mode Port of the Linux Kernel, *USENIX Annual Linux Showcased and Conference* (2000).
8) Zhao, Ming, Zhang, J. and Figueiredo, R.: Distributed File System Virtualization Techniques Supporting On-Demand Virtual Machine Environments for Grid Computing, *Cluster Computing Journal 9(1)* (2006).
9) OpenVPN. http://openvpn.net.
10) 高宮安仁, 真鍋篤, 松岡聡: Lucie: 大規模クラスタに適した高速セットアップ・管理ツール, 先進的計算基盤システムシンポジウム SACSIS2003, pp. 365–372 (2003).
11) 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, 先進的計算基盤システムシンポジウム SACSIS2006 論文集, pp. 525–532 (2006).
12) Jiang, X. and XU, D.: VIOLIN: Virtual internetworking on overlay infrastructure, *Technical Report, Department of Computer Sciences, Purdue University* (2003).
13) Manabe, A.: Disk cloning program 'dolly+' for system management of pc linux cluster, *Computing in High Energy Physics and Nuclear Physics* (2001).
14) Octave. http://www.octave.org.
15) Globus Project. http://www.globus.org.
16) 山形育平, 高宮安仁, 中田秀基, 松岡聡: グリッド上における仮想計算機を用いたジョブ実行環境構築システムの高速化, 情報処理学会研究報告 2006-HPC-105(HOKKE2006), pp. 127–132 (2006).
17) Shoykhet, A., Lange, J. and Dinda, P.: Virtuoso: A system for virtual machine marketplaces, *Technical Report NWUCS-04-39, Department of Computer Science, Northwestern University* (2004).