オーバレイネットワークを用いたマルチサイト仮想クラスタ構築システム

多田大輝[†] 市川昊平[††] 伊達進[†††]
阿部洋丈[†††] 下條真司[†††]

近年，仮想計算基盤が注目を集めている．とりわけ，科学研究においては，計算基盤上から柔軟にリソースを切り出し，独自の計算環境を構築することが可能な仮想クラスタへの期待は高まっている．今日では，多くの研究者らによって，単一サイト内のリソースに限定されることなく，柔軟にリソースを割り当てることが可能な計算環境が求められており，複数サイトの計算リソースを効率的に集約し，単一のクラスタ環境を提供できるマルチ仮想クラスタ構築システムが期待されている．しかし，現状では実用的なマルチサイトクラスタ構築システムは提案されていない．我々は本論文で，クラスタ管理ツールである Rocks とオーバレイネットワーク技術をシームレスに統合することで物理的な計算資源に対して透過的にリソース割り当てが可能なマルチサイト仮想クラスタ構築システムを提案し，プロトタイプ実装を行った．実装したシステムの有用性を測るため，WAN 環境をエミュレートし，構築されたマルチサイト仮想クラスタ上でのアプリケーション実行性能を評価した．その結果，構築した仮想クラスタが実用上問題ない性能を示すことが確認できた．

A multi-site virtual cluster depolyment system using overlay network

Taiki Tada,[†] Kohei Ichikawa,[††] Susumu Date,[†††]
Hirotake Abe[†††] and Shinji Shimojo[†††]

Recently, virtualized computational infrastructure has attracted many attentions. In scientific research, specifically, virtual cluster has expected to allow researchers to build their own computational environments on a computational infrastructure with flexibility. However, today, practical multi-site virtual cluster system has not proposed. We propose multi-site vitrtual cluster system, which can transeparently assign recources to the cluster by seamlessly integrating Rocks cluster management tool with overlay network technology. To evaluate our prototype system, we have measured the overhead of network throughput. And then, we found that our system works well in practical for some applications.

1. はじめに

近年，クラウドなどのサービスにより，仮想計算基盤が注目を集めている．仮想化技術 [10,17] を用いることで，OS を含めた計算環境がハードウェアから完全に切り離される．これにより，CPU やメモリなどの計算リソースを切り出し，それらを柔軟にユーザへ割り当てることが可能になる．また，仮想化された計算環境は，個々が完全に独立しているので，互いに影響を及ぼさない．したがって，ユーザは自身の目的に合わせた計算環境を構築できる．

これらの特徴から，今日では HPC などの研究分野においても，計算リソースをユーザへの柔軟に割り当てられる点，およびユーザ間でリソースの共有を容易に実現できる点から，仮想化技術への期待が高まっている．とりわけ，計算基盤上で仮想化された計算機で構成する仮想クラスタが注目されており，Rocks Cluster Toolkit [13] など一部のクラスタ構築システムでは仮想クラスタの構築にも対応している．

仮想クラスタは，ユーザに対して，オンデマンドなクラスタ環境を提供する．クラスタを構成する計算リソースは，計算基盤から柔軟に切り出され，それらは個々に独立した仮想ネットワークで接続される．その結果，仮想クラスタは計算基盤上で互いに独立して構成され，ユーザはそのクラスタ上で実行したいアプリケーションに応じた独自の計算環境を構築できる．さらに，ユーザは用途に応じて，計算基盤からリソース

† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
†† 大阪大学情報基盤本部
Central Office for Information Infrastructure, Osaka University
††† 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University

を追加することで仮想クラスタを柔軟にスケールアウトできる．

一方で，現状のクラスタ構築システムで構築できる仮想クラスタは，単一サイト内の計算基盤上での利用に限定され，複数サイトの計算リソースを統合することはできない．そのため，単一サイト内の計算リソース量を超えるスケールアウトができないという問題がある．しかしながら，今日の科学研究においては，大量のデータを高速に処理できる計算環境への需要が大きく，できる限り多くの計算リソースを望むユーザも多い．実際，パラメータスタディ型の分散計算アプリケーションを用いてデータ処理を行うユーザの多くは，たとえ遠隔に位置するリソースであっても，それらを用いて柔軟に計算リソースを追加できる仮想クラスタシステムを望んでいる．なぜなら，そのようなアプリケーションの場合は，各サイトのリソース間で生じるネットワークの通信遅延は実行性能に大きく影響しないからである．例えば，バイオサイエンス分野で用いられる創薬シミュレーション DOCK [3] の場合，個々のリソースに分散したプロセスは，独立性が高く，プロセス間が相互に行う通信回数や通信データ量が小さい．故に，異なるサイトに位置するリソース間で生じるデータの通信遅延が，アプリケーションの実行性能に与える影響は小さい．

近年では，こうした需要に対して，複数サイトのリソースを利用し仮想クラスタを構築しようとする試みが始められつつある [8,12,16]．しかし，依然として複数サイトにまたがる仮想クラスタを容易に構築できる実用的な仮想クラスタ構築システムは存在しない．これは，複数サイトにまたがる仮想ネットワークを構築することは未だ困難であり，またその仮想ネットワークをクラスタ構築システムと統合する技術が未開発であるためである．

こうした背景から，本論文では，既存の仮想クラスタ構築システムに対して複数サイトにまたがるオーバレイネットワークを仮想ネットワークとしてシームレスに統合する，マルチサイト仮想クラスタ構築システムを提案する．提案するシステムはユーザに対し，単一の物理的なクラスタと同様のユーザビリティを保ちつつ，複数サイトの計算リソースを集約し利用できるクラスタ環境を提供することを目的とする．図 1 に，提案するマルチサイト仮想クラスタの概念図を示す．

以下，2 章では提案するマルチサイト仮想クラスタ構築システムに対するアプローチとシステム概要について述べる．3 章では，2 章で述べたシステムの実装について述べる．4 章ではプロトタイプ実装したシステムで実際に構築した仮想クラスタの実用性に関して議論すると共に，マルチサイト仮想クラスタ上でのアプリケーション実行性能の評価を行い，システムの有用性を示す．5 章では仮想クラスタの関連研究について述べる．最後に，6 章で本論文をまとめる．

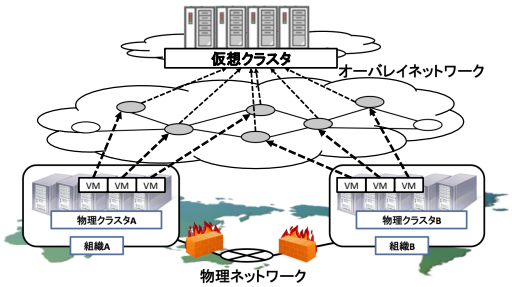

図 1 マルチサイト仮想クラスタの概念図

2. 提案するマルチサイト仮想クラスタ構築システム

2.1 アプローチ

本研究では，単一の物理的なクラスタと同様のユーザビリティを保つマルチサイト仮想クラスタを提供するため，以下に説明するクラスタ構築システムである Rocks Cluster Toolkit(以下，Rocks) とレイヤ 2 レベルのオーバレイネットワーク構築ツールである N2N [2] を統合したシステムを提案する．

Rocks は既存のクラスタ構築システムの中でも最も広く利用されているクラスタ構築システムである．ネットワーク設定などの基本設定を施すことによってほぼ全自動でクラスタが構築されるため，非常に利便性が高く，広く普及している．クラスタシステムが必要とするユーザアカウントの同期，ジョブスケジューラの設定，Home ディレクトリの共有などの設定が自動で行われるため，ユーザは迅速に利用開始できる．また，Rocks は Xen [1] をハイパーバイザとした仮想クラスタを構築することもできる．Rocks が提供する仮想クラスタはユーザごとに異なる VLAN で区切られた仮想ネットワークを提供し，個々に独立した計算機環境をユーザに提供する．

N2N は P2P によるオーバレイネットワーク技術を利用したレイヤ 2 レベルの仮想プライベートネットワークを提供するシステムである．レイヤ 2 レベルの仮想ネットワーク環境を構築するため，既存の多くのアプリケーションやツールに対して複数サイトにまたがる透過的な単一のネットワーク空間を提供することができる．そのため，複数サイトの統合環境において，既存のソフトウェア資産を最大限に有効利用すること

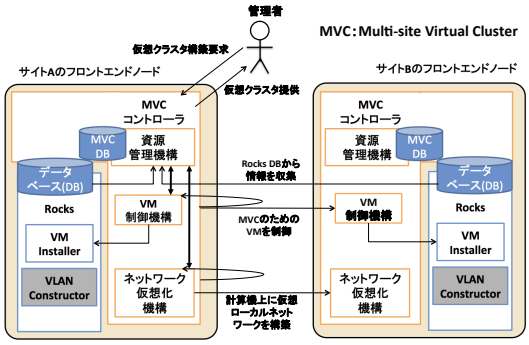

図 2 マルチサイト仮想クラスタシステム概要

を可能としている．

本研究では，Rocks の仮想クラスタにおいて利用されている VLAN による仮想ネットワーク機能を，この N2N によるレイヤ 2 レベルのオーバレイネットワークに置き換えることによって，複数サイトにまたがる仮想クラスタを構築できるように拡張する．N2N が提供する透過的な仮想ネットワーク機能を活用することによって，Rocks の仮想クラスタ構築機能を複数サイト統合環境の上で透過的に機能させることを目指す．

2.2 提案システムの概要

本研究では，マルチサイト仮想クラスタを構築することを目的に MVC(Multi-site Virtual Cluster) コントローラを設計した．提案する MVC コントローラの概要を図 2 に示す．MVC コントローラは，1) マルチサイト仮想クラスタを構成する仮想計算機の情報および利用可能な計算リソース情報を管理する資源管理機構，2) 複数サイトにまたがる仮想ネットワークを構築・維持するネットワーク仮想化機構，3) 仮想計算機の起動及び破棄処理を実施する VM 制御機構から構成される．この MVC コントローラを用いてマルチサイト仮想クラスタを構築する手順は，次の通りである．まず，最初に資源管理機構が各サイトにおける利用可能な計算リソース情報に基づき，仮想クラスタに使用する計算リソースを選択する．次に，ネットワーク仮想化機構が，選択されたリソース間で N2N による仮想ネットワークを構築する．最後に，VM 制御機構が，N2N の構築した仮想ネットワークを各仮想計算機に接続し，起動する．以上により，マルチサイト仮想クラスタは構築される．

3. 提案するマルチサイト仮想クラスタ構築システムの実装

本節では，前節までに概要を説明したマルチサイト仮想クラスタ構築システムに関して，その実装の詳細を述べる．具体的な実装の説明に先立って，本節ではまず，従来の Rocks における単一サイト内の仮想クラスタ構築について説明する．そして，Rocks の仮想ネットワーク部分を置き換え，マルチサイト仮想化へ対応させる N2N のアーキテクチャについて述べる．その後，前節で説明した MVC コントローラを構成する 3 つの機構によってどのように Rocks と N2N が統合されるか説明する．

3.1 従来の Rocks による仮想クラスタの構築

Rocks は，クラスタ構築を自動化するために，PXE ブートによるネットワーク経由でのインストール方式を採用している．ユーザはフロントエンドのみ手動でインストールし，計算機ノード群は PXE ブートにより自動的にフロントエンドノードを探し出し，インストールが始まる．PXE ブートによる自動インストールの仕組みは次の通りである．1) 計算機ノードは起動時にブロードキャスト通信を行なって，フロントエンドノードの発見を行う．2) フロントエンドノードを発見すると DHCP により IP アドレスの自動割り当てを受ける．3) そして，インストーラ OS のイメージをネットワーク経由で取得し，インストールを自動的に始める．

Rocks による仮想クラスタ構築においても，この PXE ブートによる仮想計算機のインストールが行われる．ただし，仮想クラスタ構築の場合は一つの物理クラスタ上に複数の仮想クラスタが混在することになるので，ネットワークの独立性の確保が必須となる．特に PXE ブートは仕組み上，レイヤ 2 レベルのブロードキャスト通信が必須であるため，レイヤ 2 レベルでの独立性の確保が必要となる．

Rocks は各仮想クラスタ間のネットワークの分離に，タグ VLAN を用いる．Rocks は仮想クラスタの構築ごとに，ユニークな VLAN ID を仮想クラスタに割り当て，VLAN 機能によってネットワークの独立性を確保する．タグ付き VLAN ポートを作成すると，新しい仮想的なネットワークデバイスが OS 上からは認識される．例えば，eth0 デバイスに対して，VLAN ID が 2 のポートを作成すると，新たに eth0.2 というデバイスが生成される．Rocks はこの VLAN に対応したネットワークデバイスを，仮想クラスタの計算ノードに提供するネットワークデバイスにブリッジデバイスを経由して繋ぎこむ．したがって，仮想計算機の視点からは背後の VLAN 設定を意識することなく独立した通信を行える．

Rocks による仮想クラスタ構築の具体的な手順を，図 3 に示す．まず，最初に前提条件として，1) クラス

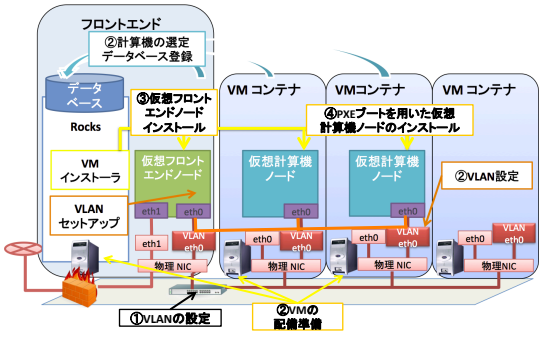

図 3 従来の Rocks による仮想クラスタの構築

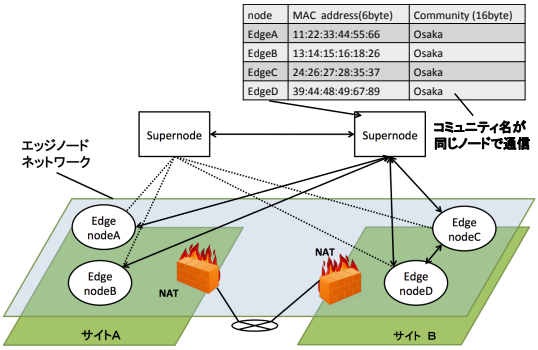

図 4 N2N のアーキテクチャ

タに接続されたスイッチの各ポートが任意の VLAN タグ付きパケットを通過できるように，全てのポートをトランクポートとして設定する．次に，2) Rocks が提供する「rocks add cluster」というコマンドを実行することによって，各物理計算機から仮想計算機を割り当てる計算機の選定及びその情報がフロントエンドの保持するデータベースへ登録される．これと同時に，VLAN ID も新たに発行され，VLAN ID が割り当てられたネットワークデバイスが生成され，ブリッジデバイスを介して仮想計算機に接続される準備がされる．ここまでで仮想計算機を起動する準備が整う．次に，3) 仮想クラスタのフロントエンドノードのインストールを開始する．フロントエンドのインストールが終われば，4) 各仮想計算機ノードのインストールを順次開始できる．このようにして，Rocks は仮想クラスタを構築する．

VLAN を用いた仮想クラスタ間のネットワーク独立性の確保は，これらの計算機リソースを単一サイト内での運用する上では道理にかなった方法であると考えられる．しかし，VLAN ID は限りのあるリソースで，かつ組織ごとに割り当てポリシが異なる．そのため組織を超えて共有することは非常に困難を伴う．VLAN ID を一致させるネゴシエーションを動的にオンデマンドで実施することはほぼ不可能であり，このままのアーキテクチャでマルチサイト仮想クラスタへの拡張はできないと考えられる．

3.2 N2N のアーキテクチャ

N2N は P2P によるオーバレイネットワーク技術をベースとしたレイヤ 2 レベルの仮想プライベートネットワークを提供する．N2N の構成は図 4 に示すように supernode と edge の組みからなる．edge が各末端の計算機上に配置されるプログラムであり，Linux の tap ドライバをベースとしたレイヤ 2 レベルの仮想ネットワークデバイスを提供する．この仮想ネットワークデバイスを通じて送受信されるデータは N2N の supernode と edge が構成する P2P ネットワーク内で交換され，目的の edge まで運ばれる．super node は各 edge の MAC アドレスとネットワーク上のルーティングを管理し，グローバル IP の割り当てのない edge がある場合は，その通信の中継も実施する．グローバル IP アドレスの割り当てのある edge 同士は supernode の管理テーブルに従って，互いに直接的に通信を行う．それ故に N2N が P2P ベースであると言われている．

また，N2N は，各 edge に割り当てられた MAC アドレス及びあらかじめ決定した固有のコミュニティ名を用いて，オーバレイネットワーク上に存在する特定の edge 間で独立したレイヤ 2 仮想ネットワークを構築することが可能である．edge の起動の際に，super node 及び固有のコミュニティ名を指定して起動することで，指定した supernode が，各 edge に割り当てられた MAC アドレスをベースに共通のコミュニティ名が割り当てられた edge 間でパケットの転送を行う．

本研究の基本的なアイデアはこの N2N の edge によって生成されるレイヤ 2 レベルの仮想ネットワークデバイスを Rocks の VLAN のネットワークデバイスの代わりに仮想クラスタのネットワーク構築に利用することにある．

3.3 MVC コントローラの実装

以下，本研究で提案した MVC コントローラを構成する資源管理機構，ネットワーク仮想化，VM 制御機構に関して説明し，Rocks と N2N を統合することで実現したマルチサイト仮想クラスタ構築システムに関して記述する．

3.3.1 資源管理機構

我々の開発した MVC コントローラでは，まず，資源管理機構において仮想クラスタ構築に用いる計算リソースを選択する．各サイトに配備されている Rocks

クラスタは，そのクラスタシステムにおいて利用されている計算リソースの管理に必要なデータベースを保持している．資源管理機構においては，そのデータベースの情報を基に仮想計算機をホスト可能な計算リソースを発見する．選択したリソースの情報は，Rocks クラスタの DB 内で MVC コントローラ用に拡張して作ったテーブル上に保存され，VM 制御機構における仮想マシンの起動・破棄の際に利用される．

また，ネットワーク仮想化機構においては，選択したリソースにおける仮想計算機の MAC アドレスが必要になる．そのため，資源管理機構では，あらかじめ，利用する仮想計算機の MAC アドレスの取得・管理も行う．

3.3.2 ネットワーク仮想化機構

ネットワーク仮想化機構は，従来の Rocks が構築する単一サイト内の仮想クラスタ用の VLAN による仮想ネットワーク構築機能を置き換える形で実装されている．ネットワーク仮想化機構は，資源管理機構によって選択された複数サイトに分散する計算リソース間で，先ずは N2N のオーバレイネットワークを構築する．この N2N による仮想ネットワーク構築は VM 制御機構における仮想計算機の起動に先立って実施される．

N2N は起動されると，レイヤ 2 レベル仮想化ネットワークを提供するため，tap ベースの仮想ネットワークデバイスを作成する．この tap ベースのネットワークデバイスを Rocks の VLAN の代わりに仮想計算機に割り当てる．ただし，イーサネットのパケットをこの tap デバイスを介して仮想計算機と送受信するためには，この tap デバイスと仮想計算機上のネットワークカードの MAC アドレスが一致している必要がある．ネットワーク仮想化機構は，資源管理機構より渡された各仮想計算機の MAC アドレスを指定して，N2N を実行することにより，起動する仮想計算機の MAC アドレスと N2N によって作成される tap デバイスの MAC アドレスを一致させる．これにより，N2N で作成されたネットワークデバイスを仮想計算機に割り当てる準備が整う．

3.3.3 VM 制御機構

VM 制御機構は，資源管理機構が選択した計算リソースにおいて，仮想計算機を起動する．この機構は従来の Rocks における仮想計算機の起動機能とほぼ同じ実装である．唯一違う点は，仮想クラスタにネットワークを提供する際に，VLAN の代わりに，上記のネットワーク仮想化機構によって作成された N2N の仮想ネットワークデバイスをブリッジデバイスを経由して仮想計算機のネットワークデバイスに繋ぎ込むことである．

これにより，起動される仮想計算機に対して，透過的に N2N が構築したオーバレイネットワークが割り当てられることになる．したがって，仮想計算機の中で実行される OS にとっては，接続されているネットワークデバイスは通常の物理的なネットワークカードのように認識することになるが，実際にはこのネットワークデバイスを介して行われた通信は全て N2N が構築したオーバレイネットワークを通じて他の仮想計算機に転送されることになる．

以上の手順により，複数サイトにまたがる仮想クラスタ環境が構築される．

4. 評　　価

本節では，最初に我々の構築したマルチサイト仮想クラスタ構築システムを用いて，実際に複数サイトにまたがった仮想クラスタが構築できることを確認した．次に，構築した仮想クラスタが実際の WAN 環境でどの程度のアプリケーション実行性能の影響を受けるか確認すると共にその結果を示す．

4.1 マルチサイト仮想クラスタの実用性に関して

我々の構築したマルチサイト仮想クラスタ構築システムの実用性を確認するために，図 5 のような環境下でマルチサイト仮想クラスタの構築を行った．まず，複数サイトに位置する複数の Rocks クラスタを想定し，1 台のフロントエンドノードと 5 台の VM コンテナで構築される Rocks クラスタ A と，1 台のフロントエンドと 4 台の VM コンテナで構築される Rocks クラスタ B,C をセットアップした．個々のクラスタは 1Gbps のローカルネットワークスイッチで単一セグメントを成すクラスタを構成しており，それらは 1Gbps の単一のルータで集約されている．

これら 3 個の Rocks クラスタを用いて，Rock クラスタ A のフロントエンド上に起動した仮想フロントエンドを起点とした仮想クラスタを構築した．構築した仮想クラスタは，1 台の仮想フロントエンドと各 Rocks クラスタを構成する VM コンテナにそれぞれ 1 台ずつ起動された 13 台の仮想計算機から成る．各仮想計算機は，N2N の仮想ネットワークを介して接続されており，supernode は，Rock クラスタ A の物理的なフロントエンド上で起動している．

実際に構築された仮想クラスタ上では，仮想計算機全体で物理的なクラスタと同様に利用できるクラスタ環境が構築されていることを確認した．また，構築された仮想クラスタ上でセットアップされたジョブスケ

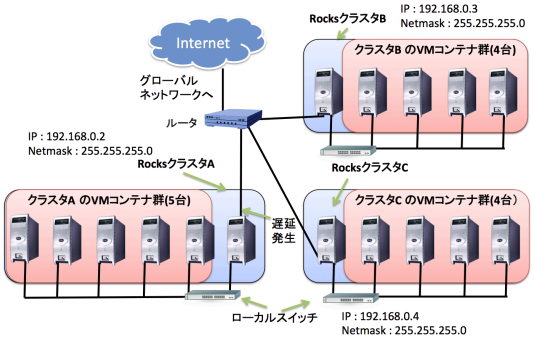

図 5 評価環境

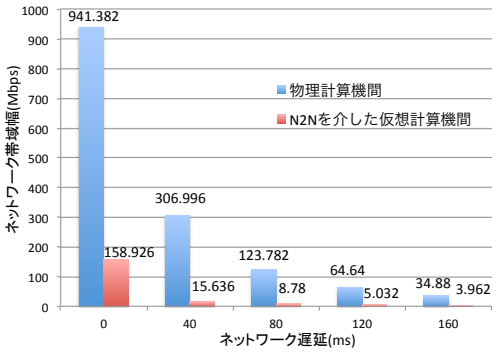

図 6 サイト間遅延によるネットワーク帯域の変化

ジューラなどを用いて，MPI などのノード間通信が発生するアプリケーションの実行が可能なことも確認できた．

4.2 構築されたマルチサイト仮想クラスタの性能評価

構築した複数サイトのリソースから成る仮想クラスタを用いた性能評価を行った．N2N を介して接続されたマルチサイト仮想クラスタが，実際の WAN 環境におけるサイト間のネットワーク遅延によって，アプリケーションにどの程度影響を与えるかをアプリケーションの実行時間から性能評価した．遅延時間に関しては，複数サイトにまたがる仮想クラスタを想定し，大阪と筑波間の往復遅延 (Round-trip Time; RTT) が 40 ミリ秒 (ms)，大阪とアメリカのカリフォルニア州間の RTT が 120ms であることを考慮して，設定した．

まず，最初にサイト間のネットワーク遅延によって，構築したマルチサイト仮想クラスタのネットワーク帯域幅がどの程度小さくなるか，Netperf [9] を用いて計測した．計測は，Rocks クラスタ A のフロントエンドと仮想クラスタ C の VM コンテナの物理計算機間での帯域幅及び，Rocks クラスタ A 上に起動した仮想フロントエンドと Rocks クラスタ C の VM コンテナ上に起動した仮想計算機間での帯域幅を，ネットワーク遅延を加えながら測定した．ネットワーク遅延は，Rock クラスタ A 上のグローバルネットワークに接続されたネットワークデバイスに対して加えた．このネットワークデバイスにネットワーク遅延を発生させることで，N2N により仮想ネットワーク上でも同様のネットワーク遅延が生じるようになっている．

測定した結果を図 6 に示す．図 6 が示す通り，物理計算機間及び N2N を介して仮想クラスタを構築する仮想計算機間のどちらにおいても，ネットワーク遅延によって，大きく帯域幅が狭められている．

次に，実際に分散計算アプリケーションを構築した仮想クラスタ上でサイト間のネットワーク遅延時間を増加させながら実行し，計算リソース間のネットワーク遅延時間がどの程度アプリケーションに影響を与えるか，評価した．分散計算アプリケーションとしては，ジョブの実行中に各仮想計算機間ではほとんど通信が発生しない，パラメータスタディ型のアプリケーション DOCK を用いた．DOCK は，薬物候補となる化合物を探索する分散計算アプリケーションであり，MPI を用いて一連の化合物ごとに複数の計算機へ分散して処理できるように実装されている．評価では，DOCK を用いてあるターゲットとなるタンパク質に対して，400 個の化合物の Docking シュミレーションを行った時に要した処理時間を計測した．

図 8 に示すように，DOCK のようにノード間で通信があまり発生しない分散計算アプリケーションの場合は，たとえノード間の通信が 40ms 増加したとしても，アプリケーションの実行性能には高々2，3 パーセントほどしか影響を与えない．したがって，我々の提案したマルチサイト仮想クラスタ構築システムが構築する仮想クラスタが実際の WAN 環境において構築されても，アプリケーションの実行性能には大きく影響することなく，実行できることが示された．

一方で，各計算機間で通信が発生するような計算の評価を HPL(High-Performance Linpack) [14] を用いて実施した．図 8 は挿入遅延が 0 ms の場合で，利用する計算ノード数を変化させながら，その時の実行性能を計測した結果である．この結果から，ネットワークを利用しない 1 ノードで計測を行った場合の性能は約 6.5Gflops であるが，利用するノード数を増やして行っても性能はスケールせず，すぐに頭打ちになっていることが分かる．

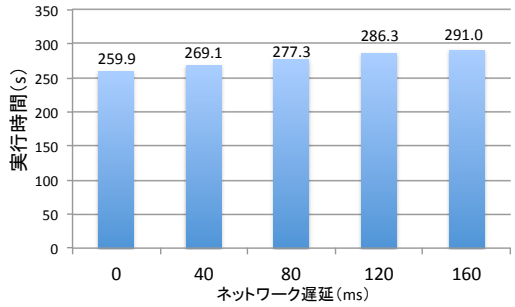

図 7 Dock の計算性能

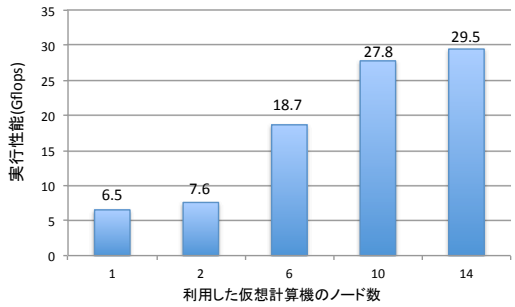

図 8 HPL の実行性能

5. 関 連 研 究

現在，複数サイトの計算リソースを集約し，単一の計算環境を構築する研究は幾つか進められつつある．[4, 15]．WoW [5] では，P2P ベースのミドルウェア IPOP [6] を用いて複数サイトの仮想計算機を集約することで，広域での分散計算環境を実現している．IPOP は，ユーザの利用するアプリケーションに対して，仮想ネットワークデバイスである tap を提供する．この tap を介して，Condor [11] などのジョブ管理ソフトウェアが用意するバッチキューにジョブを投入することで，IPOP の構築する仮想ネットワークを介して各計算リソース上でジョブを実行する．この論文では，IPOP による仮想ネットワーク構築方法のみが提案されているのみであり，ジョブ管理ソフトなどを含めたアプリケーションの実行環境の構築までを行うツールを提供しておらず，ユーザビリティは低い．また，tap デバイスがユーザの目に触れる形で提供されており，透過性に関しても低いため，ユーザは仮想ネットワークを意識して利用する必要がある．

一方で，我々と同様に，ユーザビリティを考慮してクラスタ構築ツールである Rocks をベースにしたマルチサイト仮想クラスタ構築システムも提案されている [7]．このシステムでは，各サイトに設置されたゲートウェイノード間に VPN を配備し，サイト内では VLAN による仮想ネットワークを構築し，サイトをまたぐ独立した単一のクラスタ構築を実現している．構築された仮想ネットワークを介した PXE ブートサーバ機能によるインストールも実現し，自動的な仮想クラスタ構築が可能にしている．しかし，このシステムでは，個々の仮想クラスタ配備のために異なる組織間で VPN を構築したり，VLAN の設定を調整する必要がある．したがって，オンデマンドかつ柔軟な仮想クラスタを提供することはできない．

我々のシステムは，ユーザの必要な計算リソースに応じて，各サイトに位置する計算機間で動的にオーバレイネットワークを配備することで，オンデマンドな仮想クラスタの提供を実現する．また，既存の仮想クラスタ構築システムを拡張する形でシステムを構築することで，研究者の仮想クラスタ利用にかかる負担を最小限にとどめている．

6. ま と め

本論文では，オーバレイネットワークを用いることで，複数サイトのリソースを柔軟に割り当て，単一のクラスタを構築するマルチサイト仮想クラスタを提案した．提案したシステムは，計算機クラスタの管理者によって一般的に用いられているクラスタ運用，管理ツール Rocks をベースとして開発した．Rocks をベースに開発を行うことで，新たな運用コストが生じることがない．また，オーバレイネットワークを Rocks にシームレスに統合することで，研究者らは通常，Rocks 上で仮想クラスタの構築を行う方法と同様の方法で，複数サイトのリソースを用いたマルチサイト仮想クラスタを構築することができる．

構築したマルチサイト仮想クラスタ構築システムの評価を行うために，実際にバイオサイエンス分野で用いられている創薬シミュレーションを行う分散計算アプリケーションを実行し，その計算性能を測定した．その結果，WAN 環境でマルチサイト仮想クラスタを構築した際に，アプリケーションの計算性能にほとんど影響を及ぼすことなく，実行が可能であることが確認された．

謝辞 本研究の一部は科研費 (No. 22700052, 23700058) の助成を受けたものである．

参 考 文 献

1) Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield.

Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37(5):164–177, 2003.

2) L.Deri and R.Andrews. n2n: A layer two peer-to-peer vpn. *Resilient Networks and Services*, pages 53–64, 2008.
3) T J Ewing, S Makino, a G Skillman, and I D Kuntz. DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. *Journal of computer-aided molecular design*, 15(5):411–28, May 2001.
4) I Foster, T Freeman, and K Keahy. Virtual clusters for grid communities. In *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, number 16-19 May 2006, pages 513–520. IEEE, 2006.
5) Arijit Ganguly, Abhishek Agrawal, and PO Boykin. Wow: Self-organizing wide area overlay networks of virtual workstations. *Computing, 2006 15th*, 2006.
6) Arijit Ganguly, Abhishek Agrawal, P.O. Boykin, and Renato Figueiredo. IP over P2P: Enabling self-configuring virtual IP networks for grid computing. In *Proceedings of the 20th international conference on Parallel and distributed processing*, pages 46–49. IEEE Computer Society, 2006.
7) H. Hirofuchi, T. and Yokoi, T. and Ebara, T. and Tanimura, Y. and Ogawa, H. and Nakada. Multi-site virtual cluster: A user-oriented, distributed deployment and management mechanism for grid computing environments. In *Proceedings of the Fourth IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management*, pages 203–216, 2008.
8) Xuxian Jiang. Violin: Virtual internetworking on overlay infrastructure. *Parallel and Distributed Processing and Applications*, pages 937–946, 2005.
9) R.Jones et al. Netperf: a network performance benchmark. *Hewlett-Packard Company*, 1996.
10) A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori. kvm: the Linux virtual machine monitor. In *Proceedings of the Linux Symposium*, volume 1, pages 225–230, 2007.
11) M.J. Litzkow, M. Livny, and M.W. Mutka. Condor-a hunter of idle workstations. In *Distributed Computing Systems, 1988., 8th International Conference on*, pages 104–111. IEEE, 1988.
12) Hidemoto Nakada, Takeshi Yokoi, Tadashi Ebara, Yusuke Tanimura, H. Ogawa, and S. Sekiguchi. The design and implementation of a virtual cluster management system. In *Proceedings of the first IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management (EVGM2007)*, 2007.
13) Philip M. Papadopoulos, Mason J. Katz, and Greg Bruno. NPACI Rocks: tools and techniques for easily deploying manageable Linux clusters. *Concurrency and Computation: Practice and Experience*, 15(7-8):707–725, June 2003.
14) A. Petitet, R.C. Whaley, J. Dongarra, and A. Cleary. Hpl-a portable implementation of the high-performance linpack benchmark for distributed-memory computers.
15) Ala Rezmerita, Tangui Morlier, V. Néri, and Franck Cappello. Private virtual cluster: Infrastructure and protocol for instant grids. In *Euro-Par 2006 Parallel Processing*, pages 393–404. Springer, 2006.
16) M. Tsugawa and J.A.B. Fortes. A virtual network (vine) architecture for grid computing. In *Proceedings of the 20th international conference on Parallel and distributed processing*, pages 148–148. IEEE Computer Society, 2006.
17) Carl Waldspurger. Memory resource management in VMware ESX server. *ACM SIGOPS Operating Systems Review*, 36(SI):181–194, 2002.