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

2007－ARC－172 (20)
2007－HPC－109 (20)
2007／3／1

仮想クラスタを用いた複数サイト上でのMPI実行環境

立薗真樹[†] 丸山直也[†] 松岡聡[†,††]

近年、大規模なMPIアプリケーションの実行を広域分散上で行うことが求められている。本稿では、仮想クラスタ技術を用いることで広域分散環境上でのMPI実行環境を提案し、プロトタイプ環境を構築してその評価した結果、アプリケーション特性により広域環境での実行が現実的なものであることを確認した。また仮想クラスタ環境では、計算資源の仮想化技術により仮想ノードの動的な再配置が可能となる。この特徴を利用することで、実行中のアプリケーションの特性を取得し、最適な実行環境へと実行ノードの再配置を行うシステムを提案した。プロトタイプ実装による評価により、システムのスループットの向上を確認し、同時にサイト間通信量がアプリケーションの複数サイト実行への適合度を測る目安となることを確認した。

Multi-Site MPI Execution with Virtual Cluster

Masaki Tatezono ,[†] Naoya Maruyama [†]
and Satoshi Matsuoka [†,††]

Recently, a large-scale MPI application requests a large amount of computation resource. We propose a MPI execution environment on the the geographically distributed computation resource using virtual clusters, and we confirm that our proposal is feasible according to the application characteristics since experiment on prototype of virtual cluster. Moreover, virtual cluster makes dynamic relocation of virtual node possible. By using this feature, we propose a system which automatically relocates virtual nodes includes MPI process to suitable resources, besed on MPI application characteristics and resource usage. We confirm our approach in a experiment on the prototype, and amount of Cross-site communication gives an indication of possibility of cross-site MPI execution.

1. はじめに

現在、並列プログラミングライブラリであるMPIは科学技術計算において高いシェアを誇っており、MPIを用いるプログラムは実行に長時間要するものも少なくない。今日、このように大量の計算資源を要求するアプリケーションは、計算資源が単一サイトのみならず、複数サイト間にまたがるグリッド環境での実行が一般的になりつつあるが、MPIのように高密度な直接通信によるメッセージパッシングを行うアプリケーションの実行は、サイト間の運営ポリシーやネットワーク性能などにより実現が難しい。

我々は、このような広域分散環境でのMPI実行環境として、仮想クラスタ技術を用いることを提案する。仮想クラスタ技術を用いることで、NATやファイアウォール等の非対称ネットワークの問題や、アプリケーション実行資源のソフトウェア的不均質性の解消、さらに仮想クラスタの動的再配置により、実行中のMPIアプリケーションのマイグレーションを実現する。

本稿では、仮想クラスタ技術を用いた広域分散環境でのMPI実行を提案し、それに適した最適な仮想クラスタ構成の検討を行い、評価によってその妥当性を確認した。また、実行されるアプリケーションの特性と資源利用状況から、仮想クラスタの最適な配置を決定し、動的な再配置を行うことで、刻一刻と利用状況が変化するシステムのスループットの向上を実現できるシステムを提案した。そのプロトタイプシステムを実装、評価することで、動的な再配置によるスループットの向上を確認した。また、仮想クラスタ環境でのMPIアプリケーションの実行結果についての考察を行い、複数サイト実行での有効性はアプリケーションのサイト間通信量に直接的に影響している事を確認した。

2. 仮想クラスタ環境でのMPI実行

2.1 仮想クラスタとは

仮想クラスタ技術は、広域に分散する各計算資源を

† 東京工業大学
Tokyo Institute of Technolory
†† 国立情報学研究所
National Institute of Informatics

仮想化し、それらを統合的に利用する環境を構築することで、ユーザーからは単一のクラスタ環境に見える計算環境を提供する。各仮想クラスタの仮想ノードは仮想化された独自のネットワークと仮想計算機上のゲスト OS によって、仮想化、サンドボックス化された環境であり、ユーザーや組織ごとに固有の設定にて運用することが可能である。

2.2 仮想ノードの動的再配置

仮想クラスタ環境では、任意の物理計算機上で動作中の仮想ノードを、他の物理計算機上へと動的に再配置する機能を実現可能である。この機能は仮想計算機のゲスト OS のマイグレーション機能を用いる。仮想ノードであるゲスト OS は、物理ネットワークに依らない仮想ネットワーク上で動作している。従って、このゲスト OS は現在配置されているのサイトだけではなく、他サイトへのマイグレーションをも可能とし、これによって仮想クラスタの仮想ノードを任意の資源上に再配置可能となる。一般的に仮想クラスタ環境では、仮想ノードの物理資源上の配置はユーザーに透過に決定される。

2.3 仮想クラスタ上で MPI を実行するメリット

広域に分散した複数サイト環境での MPI 実行基盤として、仮想クラスタ技術を用いることで、サイトごとのソフトウェアの不均質性、NAT やファイアウォールによるプロセスの相互アクセス不可といった問題を解消することができる。従来難しいとされていた、異なるサイトへの MPI プロセスのマイグレーションも、それを実行する仮想ノード自体の再配置により可能となる。またユーザーの視点からは、仮想クラスタ環境を用いることで、広域分散環境においても単一クラスタでの実行と同様に MPI の実行を行うことができる。

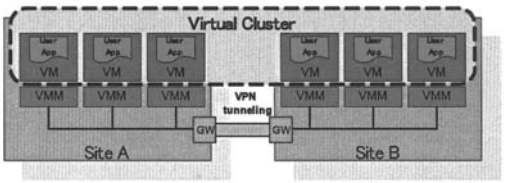

図 1 仮想クラスタを用いた広域 MPI 実行

3. 提　　案

3.1 問題設定

仮想クラスタを用いた複数サイトでの MPI 実行環境を実現するための問題点を述べる。

MPI に適した仮想クラスタ構成が未知

複数サイトで MPI を効率よく実行するために最適な仮想クラスタの構成は未知であり、用いる仮想化技術やその適用方法等の検討が必要である。

サイト間リンクによる性能への影響

複数サイトにまたがる仮想クラスタでは、通信経路にサイト間リンクが存在するが、一般的に高遅延・低バンド幅であり、それを用いることは MPI アプリケーションでは性能低下の一因となる。

仮想クラスタ配置ポリシー

仮想クラスタ環境では動的な再配置が可能であるが、再配置を行うための根拠となるポリシーを明確に定める必要がある。

再配置による通信の性能低下

仮想ノードの再配置を行った場合、それまで MPI が使用していた集団通信トポロジとは異なる資源配置となり、通信経路が実際のノード配置と合致しなくなり、迂回や重複により性能が低下する。

3.2 MPI 実行に最適な仮想クラスタ構成

3.2.1 計算資源の仮想化

計算資源の仮想化を行う仮想計算機への要件を述べる

高速な演算・I/O 性能

MPI アプリケーションを前提とした本研究では、演算性能、ネットワーク性能、ディスク性能のいずれもがアプリケーションの性能を左右する要因だと考えられる。

ゲスト OS の動的なマイグレーションが可能

本研究の想定する仮想クラスタ環境では、仮想ノードの動的再配置が可能であることが前提である。したがって、仮想計算機上で動作するゲスト OS の動的なマイグレーションが可能でなければならない。これを実現するためには、ゲスト OS の状態を保存し、任意のノードで再開する機能が必要となる。多くの仮想計算機ではメモリイメージ等のゲスト OS の状態の保存・再開がサポートされているほか、Xen や VMware ESX Server では、動作中のゲスト OS を止めることなくマイグレーションが可能である。

3.2.2 ネットワークの仮想化

仮想ネットワークに求められる要件を以下に示す。

高性能な通信

仮想計算機と同様、MPI の性能を左右する要因としてネットワークは非常に大きな部分を占める。

NAT、ファイアウォール問題の解消

任意のサイトの計算資源に配置された仮想ノード同士が相互に直接通信が可能である必要がある。

各仮想ノードへの仮想ネットワークの提供

仮想ノードに物理ネットワーク設定に依存しない仮想的なネットワークを提供する必要があり、それを用いることで他サイトへの再配置後も、ネットワーク設定の変更が不要になり、継続的な動作が可能となる。

3.3 仮想クラスタでの効率的な MPI 実行

仮想クラスタ環境では、仮想ノードの再配置機能を利用した実行中の動的な再配置により、実行性能の向

上を目指す。再配置を行う際には実行中のアプリケーションの特性と計算資源の利用状況をもとに、アプリケーション実行中の仮想ノードの再配置を行う。これによりMPIアプリケーションを最適な実行資源上に配置することが可能となり、性能の向上を期待できる。同時に、再配置に伴いMPIアプリケーションが利用する集団通信トポロジと、仮想ノードの新配置とに相違が生じるため、MPIライブラリは新たな配置情報を受け取り、トポロジの再構築を行うことで高い通信性能を維持する。

4. プロトタイプ実装

4.1 プロトタイプ仮想クラスタ環境

プロトタイプ環境では、仮想計算機としてXen[1]、仮想ネットワークとしてOpenVPN[2]を利用した。Xenは英ケンブリッジ大学で開発されたVirtual Machine Monitorで、仮想化により一つの計算機上で複数のゲストOSを同時並行的に動作が可能となる。Xenでの仮想化は準仮想化手法をとっており、ゲストOSとして動作させるカーネルの改変が必要となるが、高い性能を発揮することができる。Xenの大きな特徴の一つとして、ゲストOSのマイグレーションがある。これは動作中のゲストOSを他のXenホストへ動的にマイグレーションするもので、メモリイメージ転送の工夫[3]により、非常に短いダウンタイムでこれを実現している。

OpenVPNはSSL-VPN実装の一つで、TUN/TAPデバイスを用いることで、仮想NICを提供する。2ホスト間の仮想NICの間では、暗号化パケットとなりセキュアな通信を実現する。OpenVPNを用いた全ノードでのネットワーク仮想化は、そのサーバークライアント型の通信方式により、サイト内の通信もすべてVPNゲートウェイを通過するため、オーバーヘッドが非常に大きい。[4] そこで、プロトタイプ環境では、サイト間通信にのみ仮想ネットワークを適用し、各ノードでの仮想化を含めた性能は議論にて推測する。

4.2 資源管理モニタ

MPI実装からのアプリケーション実行ログと、各ノードから送られてくる資源利用状況を基に仮想ノードの再配置を行う。プロトタイプ実装では簡単に、半数ずつを2サイトに配置した場合のサイト間通信量の値により、アプリケーションをネットワーク・インテンシヴとコンピュート・インテンシヴとに分類、ネットワーク・インテンシヴは単一サイトに集約、コンピュート・インテンシヴは複数サイトでの実行も可とする。

4.3 MPI実装

MPI実装のプロトタイプは、mpich-1.2.7p1をベースに、ログの送信と集団通信の最適化を実装した。ログ収集は、MPI関数のラッパー関数を用意し、関数呼び出し時の宛先、メッセージサイズを関数名とともに一定数蓄積した後に、資源管理モニタへと送信する。集団通信の最適化は、集団通信時に、呼び出し時のMPI_Comの中から、現在の仮想ノード配置情報をもとに、同一サイトに配置される仮想ノード同士で新たなMPI_Comを作成、これを各サイトごとに行い、各サイト間の通信は代表ノードによる通信を行うことで、サイト間リンクの通信量を削減する。仮想ノードの再配置時には、資源管理モニタから新配置情報を受信し、上記MPI_Comの作成に利用することで、再配置後も最適な集団通信トポロジを利用できる。

表1 PrestoIII クラスタの構成ノード

CPU	AMD Opteron 242*2
Memory	2048MB
Network	Broadcom NetXtreme BCM5702XT
Kernel	Linux 2.6.12.6

5. 評　価

5.1 評価環境

評価には東工大松岡研究室のPrestoIIIクラスタを用いた。表2にその物理構成を示す。ネットワーク遅延測定にはNetPIPE3.6.2[5]を、MPIアプリケーション性能測定にはNas Parallel Benchmarks 3.2[6](以下、NPB)のクラスCを用いた。性能比較対象である「Native」は、Xenとの比較では通常のLinuxカーネル、OpenVPNとの比較では通常のクラスタネットワークを指す。

5.1.1 2サイト・エミュレート環境

広域分散環境での性能を示すために、PresotoIIIクラスタ上に2サイト・エミュレート環境を構築した(図2)。疑似サイトはそれぞれ32ノードずつの計算ノードが一つのスイッチと接続され、2サイトの外部へのリンクがGtrcNET-1[7]により接続されている。GtrcNET-1はハードウェア・ネットワークエミュレータであり、本稿では2サイトの通信に任意の遅延を挿入するために用いる。

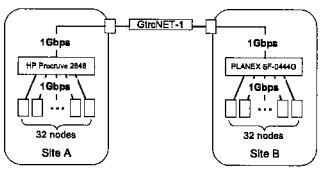

図2 2サイト・エミュレート環境の構成

5.2 Xenの性能

図3にXen環境でのホストOS(Dom0)とゲストOS(DomU)のネットワーク遅延を示す。図においてy切片であるメッセージサイズ0での遅延値が、Nativeで0.04msに対してXenゲストOSで0.08msとなっている。

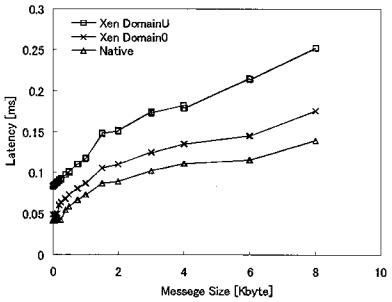

図 3　Xen における遅延

5.3　OpenVPN の性能

図 4 に OpenVPN での遅延を示す。y 切片の値はそれぞれ OpenVPN が 0.15、Native が 0.05ms となっている。次にネットワークバンド幅を図 5 に示す。こ

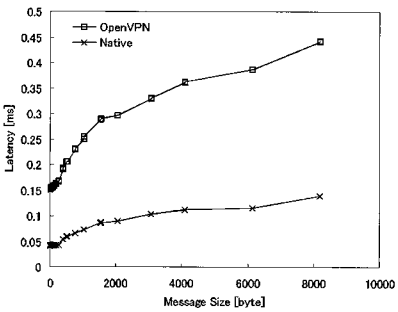

図 4　OpenVPN におけるメッセージサイズと遅延

れは 2 サイト・エミュレート環境において、サイト間を OpenVPN により接続し、両サイトの計算ノード同士のバンド幅を遅延を変化させながら測定した。遅延が 0ms に近いところでは、OpenVPN が 290Mbps 程度なのに対して、Native は 940Mbps である。一方、サイト間遅延が大きい場合、両者の遅延差は 0.1ms 程度なのでその影響差がなくなり、同様の値となる。

5.4　仮想クラスタ環境での MPI 性能

図 6,7,8,9 に 2 サイト・エミュレート環境上に構築したプロトタイプ仮想クラスタでの Nas Prallel Benchmarks 3.2 CLASS=C[6] の LU,CG,BT,EP ベンチマークを 16,32,64 ノードで実行した結果を示す。2 サイトに対して 2*8,2*16,2*32 ノードの配置で遅延を 0ms から 10ms まで変化させた場合の実行結果である。y 軸は 1 サイト 8,16,32 ノードで実行した結果に対する相対性能値であり、この値が 1 を超えている系列では、1 サイト上 x ノードでの実行と比較して 2 サイト上 $2x$ ノードでの実行時に性能が向上している

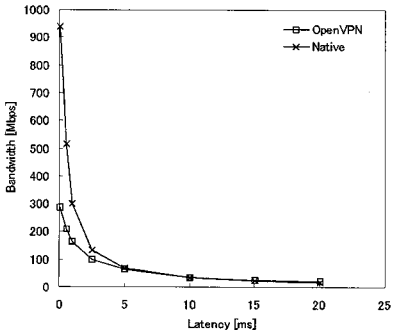

図 5　TCP における遅延とバンド幅

表 2　提案システムの有用性

	提案システム	通常環境
EP-1	105.28	98.44
LU	420.45	537.38
CG	118.57	101.97
EP-2	102.67	98.59

ことを示し、MPI アプリケーション実行時に他サイトの資源を追加利用する方が性能が向上する。結果として、LU の一部と EP の全ての場合で、2 サイト実行による性能向上が認められた。

5.5　提案システムによる効率的な資源利用

プロトタイプシステムの有効性を示すために、それぞれ 24 ノード、8 ノードの 2 サイト・エミュレート環境において、最初に Nas Parallel Benchmarks の EP(EP-1) が動作、続けて LU が投入、EP 終了後に CG を投入、CG 終了後に EP(EP-2) を投入というシナリオで評価した。仮想クラスタの動的な再配置の様子を図 10 に示す。また提案システム環境と再配置を行わない環境での各投入ベンチマークの実行時間を表 ??に示す。 結果より、提案システムでは LU 以外の

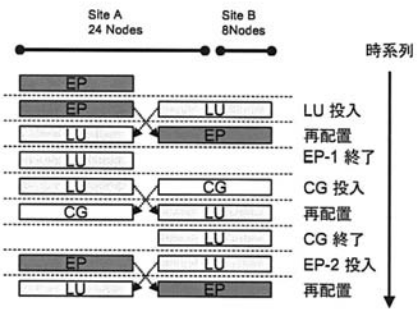

図 10　動的再配置による効率的な資源利用

実行時間が増加しているものの、LU では 20%以上の実行時間短縮になっている。

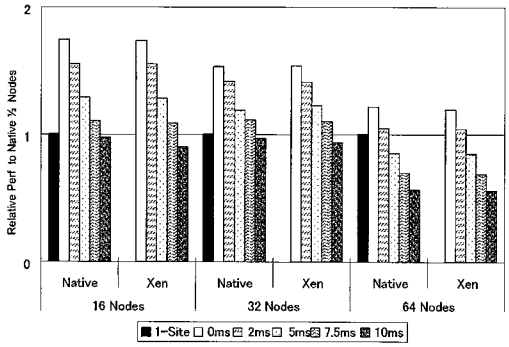

図 6 単一サイト性能に対する 2 サイト実行による性能向上 (LU)

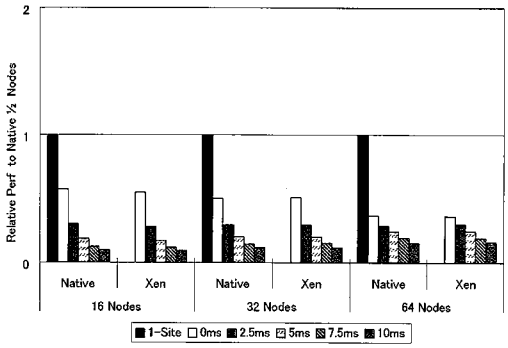

図 7 単一サイト性能に対する 2 サイト実行による性能向上 (CG)

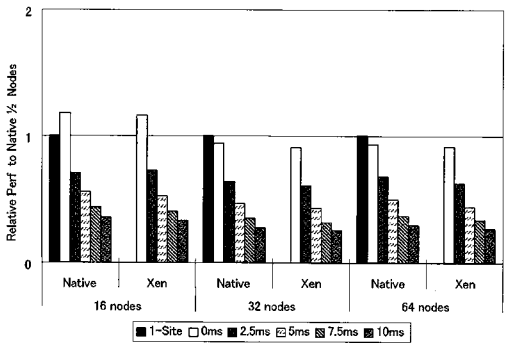

図 8 単一サイト性能に対する 2 サイト実行による性能向上 (BT)

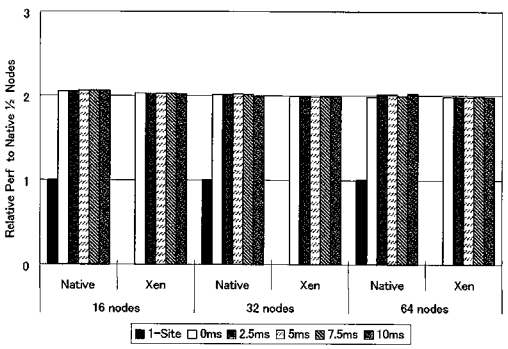

図 9 単一サイト性能に対する 2 サイト実行による性能向上 (EP)

6. 考　　察

6.1 仮想計算機性能

計算資源の仮想化技術として、プロトタイプ環境では Xen を用いた。Xen のゲスト OS でのネットワーク遅延は、Native 環境の 2 倍程度であったが、NPB でのベンチマークの結果、アプリケーション性能に及ぼす影響は少ない。一例として CG の場合、クラス C、64 ノードでの実行時に、1 ノードあたり約 20000 回の MPI_Send を呼び出す。Xen と Native のネットワーク遅延の差が 0.04ms であることから、バンド幅が変わらないとすれば Native と Xen での CG の実行時間差は $20000 * 0.04 = 800ms$ となる。実際の評価結果は 64 ノード時に Xen と Native はほぼ同等である。LU、クラス C、64 ノードの場合、1 ノードあたり約 140000 回の MPI_Send が呼ばれる。実行時間増加は $140000 * 0.04 = 5600ms$ となる。これに対し、実際の実行結果は 3-5 秒程度の実行時間増加が起こっており、3-4%程度の性能低下といえる。

6.2 仮想ネットワーク性能

サイト間リンクでのオーバーヘッドに関しては、OpenVPN の生み出すネットワーク遅延により、サイト間遅延が小さいときには、バンド幅の差が大きかったが、サイト間遅延が大きくなるにつれ、OpenVPN の遅延の影響が小さくなり、サイト間遅延 5ms 程度で、Native とのバンド幅の差はほとんどなくなる。これは、広域での使用に対して耐性があることを示している。

プロトタイプ環境では OpenVPN をサイト間接続としてしか使用していない。しかし実際の仮想クラスタ環境では、全仮想ノードのネットワーク仮想化が要件となるため、これら二つの仮想化を実現した仮想ネットワークの性能の推測を行う。想定する仮想ネットワークは、OpenVPN と同様のサイト間通信と、各ノードにて OpenVPN と同様の TUN/TAP デバイスによる仮想 NIC の提供を行う。そこで、プロトタイプ環境での各仮想ノードの通信性能に TUN/TAP によるオーバーヘッドが加わる。TUN/TAP による単純な遅延増加は片道で 0.04ms である。(OpenVPN にて測定) この値は、Xen 環境と Native 環境との遅延の差と同値であり、アプリケーションにおいては評価における Xen と Native との性能差と同様のオーバーヘッドが生じると考えられる。前節で述べたとおり、複数サイト環境でのアプリケーション性能低下は微少量であり、仮想 NIC の提供によって更なるオーバー

ヘッドが加わるとしても、その性能は許容範囲と思われる。

6.3 2サイト実行の判断基準

2サイト・エミュレート環境での仮想クラスタプロトタイプ環境における評価にて、2サイト実行の有用性が確認されたものについて、単位時間あたりのサイト間リンクの通信量を表3に示す。逆に2サイト実行が不利であるものについて、同様のものを表4に示す。

この二つから、サイト間通信量によって2サイト実行の性能が左右されていることがわかる。各結果はサイト間通信の値の大きさにより、性能向上もしくは性能低下を起こしており、この値が直接的に性能に影響を与えていると考えられる。これらの結果から、今回のプロトタイプ環境では、サイト間通信が10MB/sec前後を超えるかが複数サイト実行の可否を決める閾値と考えられる。

7. 関連研究

7.1 広域環境でのMPI実行

GridMPI[8]は広域での実行を目的としており、サイト間リンクの性能を考慮した集団通信やネットワークの流量制御により、広域分散環境でも高い性能でのじっこうを可能としている。しかし実行後の動的なプロセスの再配置には対応していない。

7.2 仮想クラスタでの資源管理

VIOLINプロジェクトでは、アプリケーションの内容に応じて、仮想クラスタの再配置を行う資源管理手法を提案している。[9] しかし、並列アプリケーションに対しては、仮想クラスタ環境ではなく単一の物理クラスタでの動作させるポリシーであり、MPIのような並列アプリの複数サイト実行を目標とはしていない。

8. おわりに

8.1 まとめ

本稿では、広域分散資源上でのMPIアプリケーションの実行基盤として、仮想クラスタ環境の利用を提案し、プロトタイプ実装を評価することで、MPIの実行に最適な仮想クラスタ環境の構成に対する考察を行い、Xenを用いることの有効性の確認および、最適な仮想ネットワークの構造と性能についての議論を行った。また、仮想クラスタ環境における動的な仮想ノード再配置機能を用いて、アプリケーション特性を考慮した効率の良い資源利用が可能なMPI実行環境を提案し、プロトタイプ実装によりその有用性を確認した。仮想クラスタ環境におけるMPIアプリケーション実行時の仮想クラスタの配置のため、アプリケーション実行結果の考察を行い、サイト間通信量により複数サイト実行可否の判断が可能なことを確認した。

表3 2サイト実行が有効なベンチマークとサイト間通信量

ベンチ	ノード数	許容遅延	サイト間通信量
LU	16	0-7.5ms	3.36MB/sec
LU	32	0-7.5ms	5.88MB/sec
LU	64	0-2ms	6.94MB/sec
EP	*	*ms	0MB/sec

表4 2サイト実行が不利なベンチマークとサイト間通信量

ベンチ	ノード数	サイト間通信量
CG	16, 32, 64	40.7, 19.9, 28.5MB/sec
BT	16, 32, 64	13.6, 15.4, 20.1MB/sec

8.2 今後の課題

本稿ではMPI実行に最適な仮想クラスタ環境の検討を通じて、仮想ネットワークの要件についての考察を行った。その結果得られた要件に合致する仮想ネットワークを開発もしくはサーベイすることによって、より現実的な環境でのシステム評価が可能となる。また仮想ノード再配置のポリシーを精緻に設定することで、より複雑なジョブ投入シナリオへの対応を行う。さらに仮想ノード再配置による資源の効率的利用の有効性を示すため、より大規模なジョブ投入シナリオに基づいた実験を行う。

謝辞 本研究の一部は科学研究費補助金特定領域研究(18049028)の補助による。

参考文献

1) Barham, P. et al.: Xen and the Art of Virtualization, *SOSP* (2003).
2) Hosner, C.: OpenVPN and SSL VPN revolution, Technical report, Sans Institute (2004).
3) Clark, C. et al.: Live Migration of Virtual Machines, *NSDI* (2005).
4) 立薗真樹ら: 仮想計算機を用いたグリッド上でのMPI実行環境, *SACSIS2006*, pp. 525–532 (2006).
5) Turner, D. et al.: Protocol-Dependent Message-Passing Performance on Linux Clusters, *Cluster 2002* (2002).
6) : Nas Parallel Benchmarks. http://www.nas.gov/Software/NPB.
7) Kodama, Y. et al.: GNET-1: Gigabit Ethernet Network Testbed, *2006 IEEE International Conference on Cluster Computing(Cluster 2004)* (2004).
8) 石川祐ら: GridMPI-通信遅延を考慮したMPI通信ライブラリの設計, SWoPP 松江 2003 (2003).
9) Ruth, P. et al.: Autonomic Live Adaptation of Virtual Computational Environments in a Multi-Domain Infrastructure, *ICAC'06* (2006).