

仮想マシン PC クラスタにおける並列相関関係抽出実行時の動作解析

豊島 詩織[†] 原 明日香[†] 小口 正人[†]

[†] お茶の水女子大学 〒112-8610 東京都文京区大塚 2-1-1

E-mail: [†]{shiori,asuka}@ogl.is.ocha.ac.jp, ^{††}oguchi@computer.org

あらまし 近年の情報化社会において情報量の爆発的な増大や IT コストの増加が問題になっている。それに対して計算機資源の効率的な運用方法を考えることは必須事項である。本研究では各ノードに汎用のパーソナルコンピュータとネットワークを用いた PC クラスタに、リソース使用率の向上が期待されているサーバ仮想化技術を適用する。クラスタリングソフトウェアである Rocks を用いて構築した仮想マシン PC クラスタでディスク I/O 性能、ネットワーク帯域の測定を行い基本性能の測定を評価した。またデータマイニングの中でも膨大なデータから有益な規則や関係を抽出する、処理の重い相関関係抽出のアプリケーションと、膨大なデータ処理を扱う技術計算のアプリケーションをそれぞれを並列化したものを動作させたときの振舞いを調べ、仮想マシン PC クラスタの動作を解析する。

キーワード 仮想化、Xen、PC クラスタ、並列分散処理、データマイニング、科学技術計算

Analysis of system behavior of Virtual PC cluster when parallel association rule mining is executed

Shiori TOYOSHIMA[†], Asuka HARA[†], and Masato OGUCHI^{††}

[†] Ochanomizu Univ 2-1-1 Otsuka, Bunkyo-ku Tokyo 112-8610 JAPAN

E-mail: [†]{shiori,asuka}@ogl.is.ocha.ac.jp, ^{††}oguchi@computer.org

Abstract An explosive increase of volume of information and an increase in the IT cost become problems in the information society recently. It is necessary to consider about the method of effective use of computer resources for it. In this study, using a general-purpose personal computers and networks in the PC cluster, server virtualization technology is applied which is expected to improve resource utilization. We have measured disk input/output performance and the network bandwidth on the virtual machine PC cluster which we built with clustering software called Rocks. Moreover, we have executed a data mining of the parallel association rule mining that extracts useful rule and relations from enormous data, and the application of the scientific computation that treats huge data processing, and evaluated detailed behavior of virtual cluster.

Key words Virtualization, Xen, PC cluster, Parallel and Distributed Processing, Data mining, science computing

1. はじめに

情報発信の増加やネットワーク上へのデータ蓄積が進み、利用可能な情報量が爆発的に増大している。またそれに伴ってストレージ機器などの IT コストの増加も大きな問題となっている。それを解決するために、IT プラットフォームの分野では「仮想化」という考え方が企業などに一般的に採用されるようになってきた。仮想化を用いた実体であるハードウェアからシステムを分離することにより、物理的な仕様に依存せず、柔軟にインフラを構築することができる。

最も利用されている仮想化技術の 1 つがサーバ仮想化である。図 1 に示すように 1 台の物理サーバ上で複数の仮想サーバを稼働させることでサーバの集約を行なうことができる。それによ

り個々のシステムの稼働率が低くても、CPU やメモリの有効利用が可能になり、省スペース化やランニングコストの節約も期待される。本研究では仮想環境を作り出すソフトウェアの中でも多くのベンダがサポートを表明している Xen を使用する。

大量のデータを効率よく処理する技術としてクラスタリングが知られている。複数のコンピュータを論理的にひとまとめにし、全体として 1 台のコンピュータとして動作させるシステム構築手法で、あるノードがダウンしたとしても別のノードがその処理を引き継ぐことでサーバの処理が継続され、耐障害性を高めることができる。さらに全体的なパフォーマンス向上も図ることができる。本研究では汎用のコンピュータと汎用のネットワークを用いることで安価にクラスタを構築することができる PC クラスタを使用し、そのクラスタに仮想化技術を取

り入れ、仮想 PC クラスタを構築した。これにより大量の情報を効率よく処理することが期待される。

本研究ではまずこの仮想 PC クラスタ上で基本性能を測定するためディスク I/O 性能とネットワーク帯域の測定を行なった。また近年クラスタ上では単なる計算処理よりもデータ処理アプリケーションを動作させることが重要となってきた。そのため構築したクラスタ上で2種類の並列データ処理アプリケーションを動作させ、アプリケーション実行時にクラスタのノード間通信や I/O の実行がどのように振舞うかを解析する。

2. 仮想化技術

2.1 サーバ仮想化

近年の情報システムの問題の一つとしてサーバの乱立が挙げられる。別の仕事をする複数のアプリケーションを一つの OS 上に共存させることは難しかったり、共存させられても OS の動作が不安定になることが懸念されるため、通常のサーバ構成は提供するアプリケーションごとに最低1台のサーバを割り当てている。そのため新たなサービスが増える度に次々とシステムの拡張やサーバの増設が繰り返され、さまざまなサーバが乱立し、環境ごとの管理作業やランニングコストが発生していた。こうした無駄を解消するにはサーバを集約することが効果的である。そこで1台のサーバ上に複数の仮想サーバを設け、そこに既存サーバを移行させることを考える。

仮想化技術を用いて1つのコンピュータ上で仮想的に複数のコンピュータが稼働しているようにシステムを構築することができる。この擬似的なコンピュータの一つひとつを仮想マシン (Virtual Machine) と呼び、この技術を用いサーバ仮想化を実現することが主流になりつつある。仮想マシンを利用することにより最新のハードウェアではサポート切れの OS も、最新ハードウェア上で仮想サーバとして動かすことが可能になる。さらにシステム使用率のピークが異なる複数のシステムを仮想マシンとして同一サーバ上に移行させることによりシステム使用率の向上が期待される。

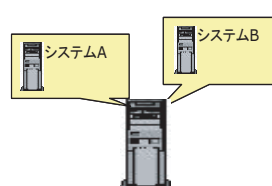


図1 サーバ集約

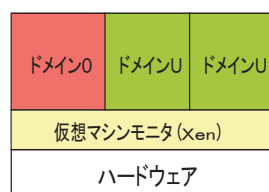


図2 Xenの構造

2.2 Xen

仮想化技術を用いて仮想マシンを実現するソフトウェアとして、VMware や VirtualPC などがある。これらはホスト OS 上にゲスト OS と呼ばれる OS のインストールを可能にするソフトウェアで、ゲスト OS はアプリケーションとして動作する。既存 OS 上のアプリケーションとして動かすため処理性能が実機よりも低下するといった欠点がある。

一方 Xen[5] は図2に示すように複数の OS を動かす為の基盤となるプラットフォームのみ提供する。そのため仮想マシン

からリソースへアクセスする際に基本的にはホスト OS を介さずに実行が可能で、仮想化による処理の性能低下が比較的小さい。Xen はオープンソースのソフトウェアで、イギリスのケンブリッジ大学の研究プロジェクトとして始まった。オープンソースとしては非常に高性能で、現在ビジネスユースにおいても用いられるようになっている。Xen 上で動作する仮想マシンは「ドメイン」と呼ぶ単位で管理されドメインの中で各 OS が動作する。実ハードウェアへのアクセスやその他のドメインを管理する特権を持つドメインをドメイン 0 と呼び、ドメイン 0 以外をドメイン U と呼ぶ。

3. 仮想マシン PC クラスタ

各ノードが独立して動作する CPU、メモリ、二次記憶を保有し、ノードが必要に応じてネットワークを介し互いに通信することで全体として並列分散処理を実現する分散メモリ型並列計算機において各ノードに汎用のパーソナルコンピュータとネットワークを用いたものを PC クラスタという。汎用製品をそのまま利用できるため価格対性能比が優れており、また利用用途に応じて規模の拡大が容易である。

PC クラスタの構築および管理の一部自動化を行うクラスタリングソフトウェアとして Rocks がある [6]。Red Hat Linux をベースとした OS とセットでパッケージングされており、Rocks のダウンロードサイトから iso イメージを取得したものをを用いて PC クラスタの構築を行う。またクラスタ構築時に様々なソフトウェアをオプションとして機能追加が可能である。本研究ではオプションとして仮想環境を作り出す Xen や、自動的に各ノードからデータを収集し、それらを視覚的にグラフ化するモニタリングツールである Ganglia などをインストールした [7]。Rocks ではさまざまなサービスが動作するマスタノードである front-end ノードから計算を行う Compute ノードへジョブの投入を行う。また仮想マシンの起動等も Front-end ノードより操作する。

4. 研究内容

4.1 既存研究

文献 [3] ではシングルコアマシン2台とデュアルコアマシン2台にそれぞれ1つずつ仮想マシンを配置、シングルコア、デュアルコアマシン同士をそれぞれ FastEthernet、GigabitEthernet、10GigabitEthernet を用いて接続した。そのローカルディスクアクセス、iscsi アクセスの性能評価を行なった実験で、ドメイン 0 とドメイン U の違いや配置によりネットワーク性能に大きな差が出ることが分かった。

また文献 [4] では PC クラスタの記憶装置において、計算ノード-ストレージ間のバックエンドのネットワークに SAN (Storage Area Network) を用い (iscsi)、そのフロントエンドとバックエンドのネットワークを同じ IP ネットワークに統合した IP-SAN 統合型 PC クラスタを構築している。IP ネットワークを使用することで安価にクラスタが作成でき、またフロントエンドとバックエンドが同じ IP ネットワークを使用することから構築および管理コストの削減が期待されるが、ノード間通信と

ストレージアクセスで同じネットワークリソースを使用するため、互いに衝突し、性能が低下する可能性が推測される。このシステム上で相関関係抽出のアルゴリズムである Apriori アルゴリズムをハッシュ関数を使用して並列化した HPA (Hash Partitioned Apriori) と、FP-growth アルゴリズムを並列化した PFP (Parallelized FP-growth) を実行し、IP-SAN 統合型クラスタの詳しい振舞を明らかにしている。評価を行なった範囲では iSCSI のネットワークを統合してもネットワークバウンドにはならないということが分かっている。

4.2 研究概要

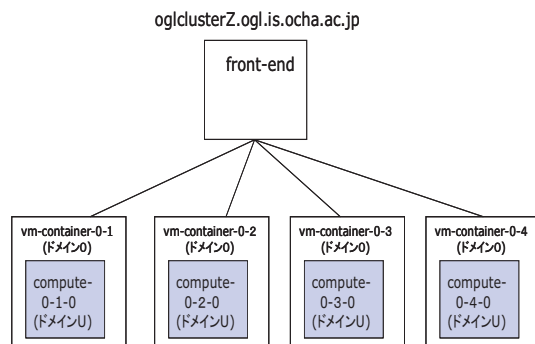


図3 仮想マシン PC クラスタ

本研究では、クラスタリングソフトウェアである Rocks を用いて、計算ノード数が4の仮想マシン PC クラスタを構築した。実験環境を図3に示す。Rocks の仮想マシンクラスタでは Compute ノードを vm-container と呼び、これが Xen におけるドメイン 0 となる。また vm-container の中にはそれぞれ compute と呼ばれる仮想マシンを一つずつ作成した。これが Xen におけるドメイン U となる。バージョンは仮想化に対応した Rocks5.0 を使用した。front-end も含めた5台の PC は CPU が Intel(R) Xeon(TM) 3.60GHz でそれらを Giga-bit Ethernet で接続した。メインメモリが 4GB、OS が Linux 2.6.18-53.1.14.el5xen(CentOS 5.0) である。計算ノードのメモリの振り分けは Rocks が自動で行なったものを用い、ドメイン 0 が 3GB、ドメイン U が 1GB である。

そのクラスタ上で、まず基本性能測定としてハードディスクベンチマークの Bonnie++[8] によりローカルおよびリモートディスクアクセスの性能、Iperf[9] を用いてネットワーク帯域の測定を行なった。

次に以下2種類の並列アプリケーションを動作させた。この場合クラスタの Compute ノードのドメイン 0 だけにジョブを与えた場合とドメイン U だけにジョブを与えた場合との動作を比較した。

一つ目の並列アプリケーションは、膨大なデータから有益な規則や関係を漏れなく、効率的に抽出するデータマイニングの一種である相関関係抽出を使用した。相関関係抽出で扱うデータは巨大であることが多く、データベースを分散、計算処理を並列化し、多数台のコンピュータをネットワークで接続した PC クラスタなどの環境でマイニング処理を実行する並列相関関係抽出の研究が行なわれている。相関関係抽出の代表的なアルゴリズムとして Apriori アルゴリズムが上げられる。Apriori

アルゴリズムは、信頼度 (confidence) と支持度 (support) に基づいてルールを評価し、「パンとバターを購入した取引の 90 % がミルクも購入している」などといった、大量のデータの中の隠れた規則や関係を抽出する。信頼度とは、X が発生したときに、Y が起こる割合を示す。この数値が高いほど、X が発生したときに Y が起こるというルールが強いことを意味する。一方、支持度とは、X と Y を同時に満たすトランザクションが全トランザクションに占める割合をいう。つまり、ルールそのものの出現率である。信頼度と支持度に閾値を設け (ユーザーが指定)、それを超える信頼度と支持度を持つルールを相関があると見なし、候補アイテムセット (ルールとして抽出される候補) から頻出アイテムセットを抽出するという動作を繰り返し行なう。この Apriori アルゴリズムをハッシュ関数を使用して並列化した HPA (Hash Partitioned Apriori) を動作させた。HPA は候補アイテムセットを各ノードに分割し、その後全ノード対全ノード通信により頻出アイテムセットを繰り返し検索していく。ノード間通信や、繰り返し計算のためネットワーク通信量は多く、ディスク I/O も繰り返し行なわれるといった特徴を持つ。

二つ目の並列アプリケーションは、バイオインフォマティクスにおいて用いる科学技術計算の一種である BLAST を並列化した mpiBLAST を使用した。BLAST は DNA の塩基配列あるいはタンパク質のアミノ酸配列のシーケンスで、ペアワイズのシーケンスアライメントを行うもので、シーケンスデータベース中のシーケンス断片と類似するクエリシーケンス中のシーケンス断片を見つけ出す。例えば、ハツカネズミの未知の遺伝子を発見したときに、ヒトがそのシーケンスと類似した遺伝子をもつかどうかを調べることができる。BLAST はアルゴリズムの正確さより速度を重視しており、膨大なデータが蓄積されているゲノムのシーケンスデータベースに対して検索を行うような場合において実用的なものとされている。この BLAST を並列処理記述の標準ライブラリである MPI を用いて並列化した mpiBLAST を動作させた。mpiBLAST はその性質上膨大なデータを扱うことが多く、共有ディスク領域から一度各ノードにデータベースをコピーし、その後各ノードにて計算処理を行っていく。ノード間通信が少ないためネットワーク通信量は HPA に比べ少なく、ディスク I/O は主にデータを一度読み出す際のみ使用するといった特徴を持つ。またメモリはある程度の大きさが必要であると考えられる。

上記2つのアプリケーションを、クラスタの Compute ノードのドメイン 0 だけにジョブを与えた場合 (以下システム A) とドメイン U にだけジョブを与えた場合 (システム B) で動作させたときの通信状況をモニタリングツールである Ganglia を用いて観察し、システムの振舞を解析する。

5. 基本性能測定

5.1 Bonnie++

ハードディスクベンチマークツールの Bonnie++ を用いて、Front-end、ドメイン 0、ドメイン U それぞれのディスク I/O 性能の測定を行なった。ドメイン 0 とドメイン U について

はローカルディスクアクセスの性能に加え、リモートディスクアクセスについても測定を行なった。その際 NFS(Network File System) を用いて、ネットワーク経由でフロントエンドのディレクトリをマウントした。測定結果を図 4 に示す。Write、Read とも同様の傾向が見られた。まずローカルでは性能が高い順に front-end、ドメイン 0、ドメイン U となっている。これは各形態における処理のオーバーヘッドが性能差に現れたものと考えられる。またリモートではドメイン U の性能が非常によいという結果になった。

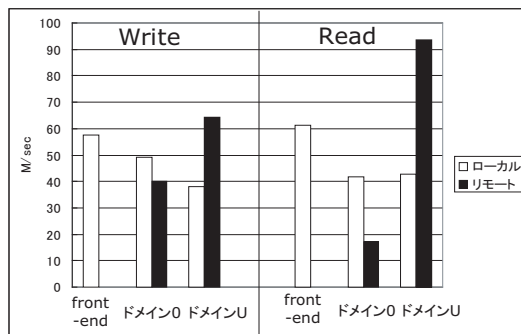


図 4 ディスク I/O 性能

5.2 Iperf

Iperf を用い front-end、ドメイン 0、ドメイン U 間のネットワークスループットの測定を行なった。図 5 に測定箇所を示した。5 と 6、7 と 8 の測定はそれぞれ異なるマシン間と同一マシン上でドメイン 0 とドメイン U 間のスループットを測定している。TCP 通信における測定結果を図 6 に、UDP 通信における測定結果を図 7 に示す。

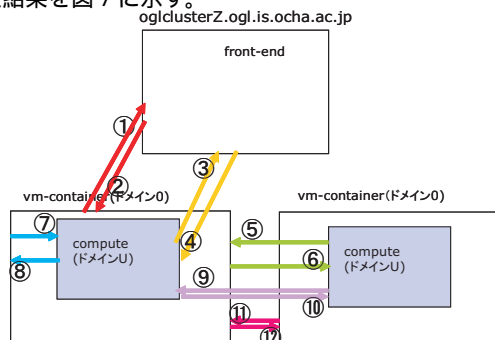


図 5 Iperf 測定図

まず TCP では同一マシン上でドメイン U からドメイン 0 の通信 (8) は 3.30Gbps と高いスループットが得られた。これは仮想化によるオーバーヘッドが少ないと考えられるドメイン 0 が高い通信速度でデータを受信できているからと考えられる。逆方向の通信 (7) のスループットが低いのはオーバーヘッドが大きいドメイン U の受信バッファが小さいためではないかと考えられる。

UDP においては、異なるマシン間ではドメイン U が送信側のときの性能が悪い結果になった。(3、5) これは一般にドメイン U が送信するパケット数に限界があるからではないかと考えられる。同一マシン上の通信ではドメイン 0 が送信側の場合の性能が悪い。(7) これは TCP と同様にオーバーヘッドが大きいドメイン U がデータを受けきれないためではないかと考

えられる。

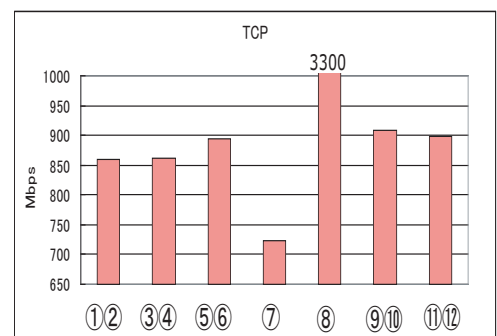


図 6 ネットワーク帯域 (TCP)

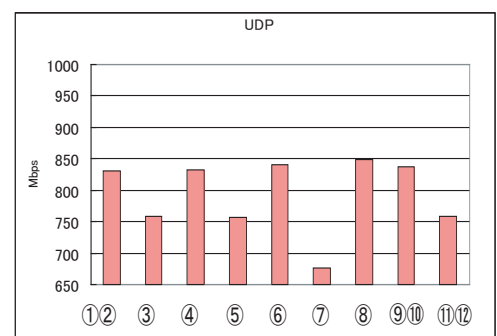


図 7 ネットワーク帯域 (UDP)

6. 並列アプリケーション実行

6.1 並列データマイニング

HPA アルゴリズムについてアイテム数を 1000 とし、トランザクション数が 1M、2M、4M、8M のトランザクションデータを用い、最小支持度を 0.7 % に設定し実験を行なった。ドメイン 0 のみを用いたシステム A、ドメイン U のみを用いたシステム B においてそれぞれ HPA アルゴリズムを実行したときの実行時間を測定、そのときの CPU 使用率、ネットワークトラフィック、メモリ使用率をモニタリングし 2 種類のクラスタの動きを比較した。図 8 に HPA アルゴリズムの実行時間を示す。

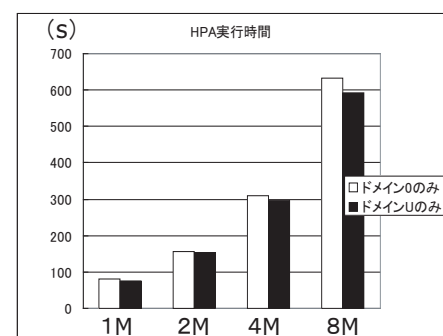


図 8 HPA アルゴリズム実行時間

この結果から、トランザクションデータが大きくなるにつれてシステム A で実行した場合の実行時間がシステム B での実行時間よりわずかながら長くなっている。

図 9 にシステム A、図 10 にシステム B で実行した時の CPU 使用率を示す。トランザクション数を多くした場合にはどちらの場合も CPU をほぼ 100 % 使用していることが分かる。これは、HPA のアルゴリズムは頻出アイテムセットから候補アイテムセットを生成していくことを繰り返すため、この過程においてアイテムの比較演算処理等が行われて計算量が増えるためであると考えられる。

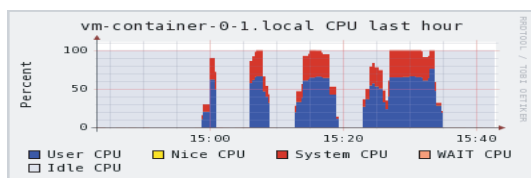


図 9 HPA をシステム A で実行した際の CPU 使用率

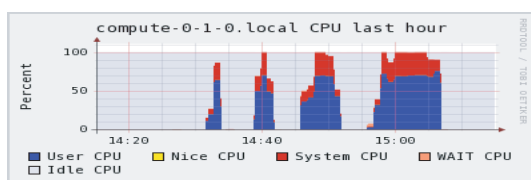


図 10 HPA をシステム B で実行した際の CPU 使用率

図 11 にシステム A、図 12 にシステム B で実行した時のメモリ使用率を示す。

どちらの場合にも割り当てられたメモリは使い切っておらず、メモリ不足が原因となる性能低下は起こっていないと予想される。

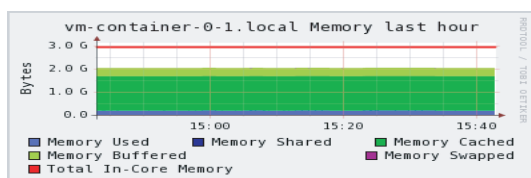


図 11 HPA をシステム A で実行した際のメモリ使用率

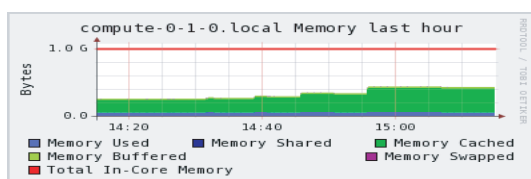


図 12 HPA をシステム B で実行した際のメモリ使用率

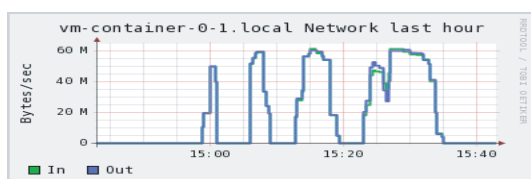


図 13 HPA をシステム A で実行した際のネットワークトラフィック

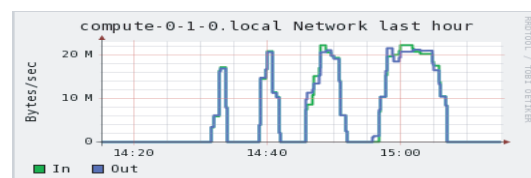


図 14 HPA をシステム B で実行した際のネットワークトラフィック

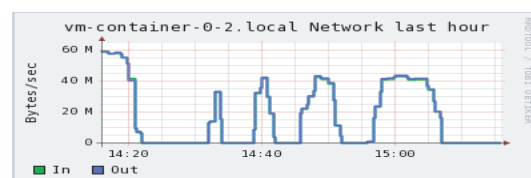


図 15 HPA をシステム B で実行した際のドメイン 0 のネットワークトラフィック

図 13 にシステム A、図 14 にシステム B で実行した時のネットワークトラフィックを示す。

この結果から、どちらのクラスタもネットワークの帯域にはまだ余裕があることが分かる。またシステム B で実行した際にはドメイン 0 のネットワークも消費することが分かった (図 15)。この場合ドメイン 0 の最大ネットワーク帯域は約 40MByte/sec、ドメイン U は約 20MByte/sec でこれを合計すると、システム A で実行した際の最大ネットワーク帯域である約 60MByte/sec になる。このことからシステム B で実行する場合に、一部の処理はドメイン U を介さず、ドメイン 0 がバッファリング等を行い代わりに処理するのではないかと考えられる。逆にシステム A で実行した際にはドメイン U のネットワークを消費することはなかった。

一般に仮想化によるオーバーヘッドのためドメイン 0 よりドメイン U の方が性能が悪いと考えられる。それに従い Compute ノードがドメイン U のみの場合はドメイン 0 の場合に比べ HPA の実行時間も遅くなると予想されるが、実行時間はほぼ同じで、ドメイン U のみの方がわずかながら短いという結果になった。またネットワークトラフィックのモニタリングにより、ドメイン U が通信するデータをドメイン 0 が一部行っているのではないかとということが分かった。

以上よりドメイン U を使用する場合はバッファなどのリソースがより多く使われ、逆にドメイン 0 のみの場合は、リソースが割り当て分のみしか使われていないため、このような結果になったのではないかと考えられる。

6.2 科学技術計算

mpiBLAST について、データは約 4GB のものを使用し、ローカルディスク上に置いた場合とリモートディスク上に置いた場合それぞれで動作させた。リモートの際は NFS を使用した。システム A、システム B において mpiBLAST を実行時間を比較した。図 16 よりローカルでもリモートでも実行時間はほぼ変わらず、システム B のほうが少しだけ早いという結果になった。

図 17 にシステム A、図 18 にシステム B にて、ローカルで

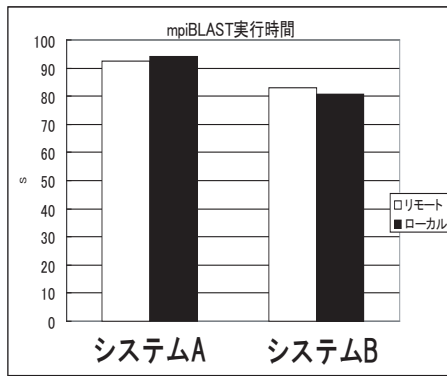


図 16 mpiBLAST 実行時間

連続 10 回実行した時のネットワークトラフィックを示す。この結果から、どちらのクラスタもネットワークの帯域にはまだ余裕があることが分かる。また HPA の場合と同様にシステム B で実行した際にはドメイン 0 のネットワークも消費することが分かった (図 19)。しかしこの場合ドメイン 0 の最大ネットワーク帯域は約 5KByte/sec、ドメイン U は約 3KByte/sec でこれを合計すると、システム A で実行した際の最大ネットワーク帯域よりも小さくなる。これはシステム B で実行する場合に、一部の処理はドメイン U を介さず、ドメイン 0 が代わりに処理するということは HPA と同様だが、ドメイン U のメモリが 1 GB と小さいため、メモリを使い切っている (図 20)。そのためメモリアクセスがボトルネックとなりその他の処理は待たされる形となってプログラムの処理動作が遅れてしまったことが原因と考えられる。

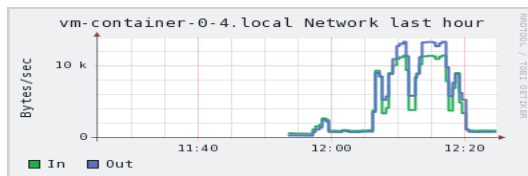


図 17 mpiBLAST をシステム A で実行した際のネットワークトラフィック

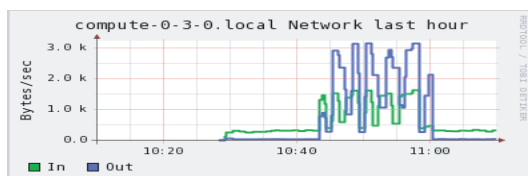


図 18 mpiBLAST をシステム B で実行した際のネットワークトラフィック

7. まとめと今後の課題

Rocks により仮想 PC クラスタを構築した。そのクラスタ上で基本性能測定としてディスク I/O 性能とネットワーク帯域の測定を行なった。

次に 2 種類の並列データ処理アプリケーションを動作させシ

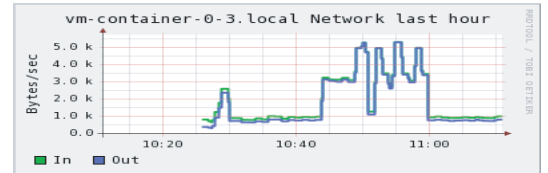


図 19 mpiBLAST をシステム B で実行した際のドメイン 0 のネットワークトラフィック

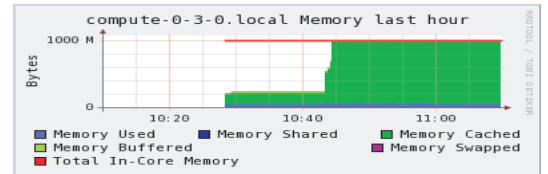


図 20 mpiBLAST をシステム B で実行した際のメモリ使用率

ステムの振舞いを調べた。HPA、mpiBLAST とともに仮想化によるオーバーヘッドのためドメイン U のクラスタのほうが実行時間が長くなるのではないかと予想されたが、実際には Compute ノードがドメイン U のみで実行された場合のほうが実行時間が短くなった。これはドメイン U にジョブを与えた場合 I/O や通信処理の一部をドメイン 0 が代わりに実行し、そのため全体性能が良くなったと考えられる。

今後は、ドメイン U のみのクラスタで動かす際に負荷を与え、リソースの使用量にどのような変化があるかなどを調べ、今回の実験結果の考察をさらに進める。負荷が複数のドメインに分散されるのであれば、今後マシンのマルチコア化が進むにつれ、ードメインにつきコアを割り当てることでシステムの実行性能が上がることを期待される。

また仮想 PC クラスタ上に iSCSI を導入し、ストレージアクセスを行うイニシエータとストレージを提供するターゲット間の通信の動作解析をする。

謝 辞

本研究は一部、文部科学省科学研究費特定領域研究課題番号 18049013 によるものである。

文 献

- [1] 谷村勇輔、小川宏高、中田秀基、田中良夫、関口智嗣: "仮想クラスタに対する IP ストレージの提供方法の比較", 「ハイパフォーマンスコンピューティングとアーキテクチャの評価」に関する北海道ワークショップ (HOKKE-2007)
- [2] 佐藤三久: "計算機クラスタコンピューティングの技術動向", 並列分散システム特論筑波大学 HPCS laboratory, 2005 年
- [3] 付思宇、小口正人: "マルチコア、仮想マシン環境における IP-SAN アクセスの性能評価", 情報処理学会第 70 回全国大会, 2008 年 3 月
- [4] 原明日香、神坂紀久子、小口正人: "並列データマイニング実行時の IP-SAN 統合型 PC クラスタのネットワーク特性解析", 分散、協調とモバイル (DICOMO2007) シンポジウム, 2007 年 7 月
- [5] Xen: <http://www.xen.org/>
- [6] Rocks Cluster: <http://www.rocksclusters.org/>
- [7] Ganglia Monitoring System: <http://www.ganglia.info/>
- [8] Bonnie++: <http://www.coker.com.au/bonnie++/>
- [9] Iperf: <http://dast.nlanr.net/projects/Iperf/>