仮想マシン PC クラスタにおける並列アプリケーション実行時の動作解析

豊島 詩織　(指導教員：小口 正人)

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。それに伴い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。その解決策として「仮想化」という考え方が一般的に採用されるようになってきた。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。

本研究では汎用機器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。この仮想マシン PC クラスタ上で並列アプリケーションを実行させたときの振舞を解析する。

2　仮想化

仮想化技術を用いることにより１つの物理システム上に２つ以上の仮想的なシステムが存在するかのような環境を作り出すことができる。その仮想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、サーバ集約やシステム利用率向上を図ることができる。本研究では仮想化のためのソフトウェアとしてオープンソースの Xen を使用した。Xen は複数の OS を動かすための基盤となるプラットフォームのみを提供するため仮想化による処理の性能低下が比較的小さい。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、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 PC クラスタ

システム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の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、利用用途に応じて規模の拡大が容易である。

PC クラスタの管理・構築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 を使用した。[2] Rocks のインストール時に仮想環境を作りだす Xen[3] や、モニタリングツールである Ganglia[4] などをオプションとしてインストールした。Rocks ではさまざまなサービスが動作する 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　実験概要

仮想化に対応した Rocks5.0 を用い計算ノード数が４の仮想マシン PC クラスタの構築を行なった (図 1)。Rocks の各計算ノード内にドメイン 0 とドメイン U がそれぞれ１つずつの状態で以下の実験を行なった。PC は CPU が Intel(R)Xeon(TM)3.60GHz で Gigabit Ethernet で接続した。メインメモリが 4GB、OS は Linux 2.6.18-53.1.14.el5xen(CentOS 5.0) である。計算ノードの仮想マシンのメモリ振り分けは 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 となっている。

まず基本性能測定としてローカルおよびリモートディスクアクセスの性能、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、クラスタの Compute ノードがドメイン 0 のみ４台のクラスタ (以下システム A)、ドメイン U のみ４台のクラスタ (以下システム B) の実行性能を比較した。

１つ目は有益な規則や関係を抽出する相関関係抽出の代表的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) を使用した。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繰り返し計算のため大容量のメモリや繰り返しのデータスキャンが必要とされる可能性がある。

２つ目はバイオインフォマティクスで 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アライメントを行なうためのアルゴリズムである 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 を使用した。バイオインフォマティクスの分野ではしばしば膨大なデータを扱うことが多く、BLAST は正確さよりも速度を重視したアルゴリズムになっている。

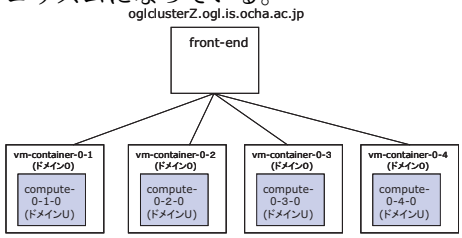

図 1: 実験環境

5　基本性能測定

5.1　Bonnie++

ハードディスクベンチマークの Bonnie++を用い、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アクセスに加え、NFS を用いたリモートディスクアクセスについても測定を行なった。

Write、Read とも同じ傾向が見られた (図 2)。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 となった。これは各形態における処理のオーバ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。

5.2　ネットワーク帯域測定

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 3 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 間を測定した。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

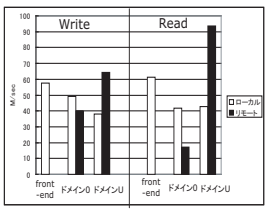

図 2: HPA 実行時間

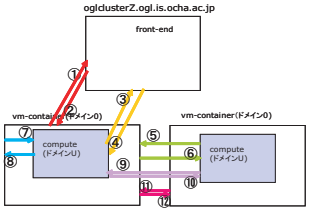

図 3: iperf 測定箇所

極めて高いスループットが得られた。これは性能のよいドメイン 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。逆の通信 (7) が悪かったのはドメイン 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 (図 4)。

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 は性能が悪いと考えられるため送信するパケット数にも限界があるからだと考えられる。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 5)。

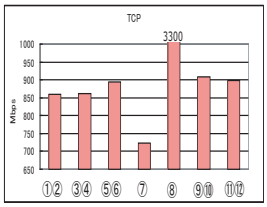

図 4: TCP 通信でのネットワーク帯域

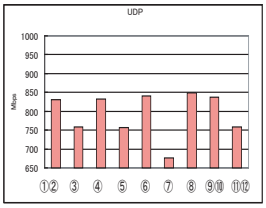

図 5: UDP 通信でのネットワーク帯域

6 並列アプリケーション実行結果と考察

6.1 HPA

HPA アルゴリズムについてアイテム数を 1000、最小支持度を 0.7 %、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実行した際のシステム A と B の動作を比較した。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 B の実行時間よりわずかながら長くなった (図 6)。モニタリングの結果からシステム A、B ともメモリ使用率、ネットワーク帯域にはまだ余裕があることが分かった。またシステム B の場合はドメイン 0 のネットワーク帯域も消費しており、このときの両ドメインの合計スループットがシステム A の最大スループットと等しくなったことから一部の処理はドメイン U を介さずドメイン 0 が代わりに処理していると考えらる。

通常、仮想化によるオーバ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、それに伴い Compute ノードがドメイン U のみの場合は実行時間も遅くなると予想されるが、実行時間はシステム B の方がわずかながら短いという結果になった。

これは、HPA においてはノード間通信が割合多く、ドメイン U を使った場合には上述のようにその処理の一部をドメイン 0 が肩代わりしているからではないかと考えられる。

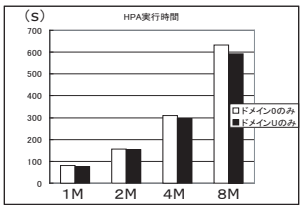

図 6: HPA の実行時間

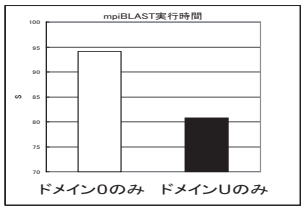

図 7: mpiBLAST の実行時間

6.2 mpiBLAST

mpiBLAST を実行したときの動作を比較した。データは約 4GB のものを使用し、データはリモートの NFS 上にある状態で行なった。実行時間はシステム B の方が早いという結果になった (図 7)。mpiBLAST は共有ディスク領域から各ノードのローカルディスクにデータベースがコピーされ個々のノードが検索を行なっていく。Bonnie++の実験でもリモートアクセスの場合はドメイン U の実行時間がドメイン 0 に比べ早かったことから NFS アクセスの通信性能についてはドメイン 0 よりもドメイン U のほうがよいのではないかと考えられる。

7 まとめと今後の課題

仮想マシン PC クラスタにおいてディスク I/O 性能とネットワーク帯域の測定より仮想化による形態のオーバヘッドが性能差に現れる結果が得られた。次に 2 つのアプリケーションを動作させクラスタの計算ノードがドメイン 0 のみのクラスタとドメイン U のみのクラスタの動作を比較した。HPA の場合の実行時間は仮想化によるオーバヘッドがあると考えられるにも関わらず、Compute ノードがドメイン U のみで実行した場合のほうが少し早いという結果になった。またドメイン U での通信処理は一部をドメイン 0 が代わりに行なっているのではないかということが分かった。mpiBLAST をリモートディスクアクセスで動かした場合はシステムの計算ノードがドメイン U だけの場合のほうが実行時間が早いという結果になった。このことは Bonnie++の結果を合わせて考えると、NFS アクセスの通信性能はドメイン 0 よりドメイン U のほうがいいのではないかと考えられる。

今後は mpiBLAST においてローカルアクセスで実行した場合の振舞を解析し、今回の実験結果と比較する。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を動作解析をする。

参考文献

[1] 原明日香、神坂紀久子、山口実靖、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”, DICOMO2008,2008 年 7 月

[2] Rocks Cluster:http://www.rocksclusters.org/

[3] Xen:http://www.xen.org/

[4] Ganglia:http://www.ganglia.info/

[5] 豊島詩織、原明日香、小口正人:”仮想マシン PC クラスタにおける並列アプリケーション実行時の動作解析”, DEIM2009,2009 年 3 月発表予定