Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

データインテンシブアプリケーション実行時におけるクラウドリソースへの負荷分散ミドルウェア

豊島詩織[†1]　山口実靖[†2]　小口正人[†1]

高度 IT 社会の進展に伴いデータの管理や IT コストの問題が深刻になっている．その問題に対して各々の組織でサーバを構築するのではなく，クラウドコンピューティングを利用することでシステムの構築コストを抑えることができる．コストパフォーマンスのよいデータ処理システムが望まれている中，手元のクラスタ使用状況を観察し，リソースが不足している場合は外部のクラウドリソースへ動的に負荷分散を行う，クラスタのスケーラブルな運用のためのミドルウェア構築を目指す．クラウドコンピューティングは不足している CPU パワーを補うことには有効であるが，本研究で議論を行うデータインテンシブアプリケーションは通常の計算処理とは異なる特徴がある．ミドルウェアにおいてはディスク I/O の状態により，投入されたジョブをローカルクラスタまたはクラウドコンピューティングに振り分ける．データベースベンチマークを使用しミドルウェアの精度を評価したところ，理想的な配置に近い振り分けができることが確認でされた．

Shiori TOYOSHIMA,[†1] Saneyasu YAMAGUCHI[†2]
and Masato OGUCHI[†1]

In recent years, data management and IT cost have become serious problem as the advanced IT society progresses. Cloud computing is introduced in order to save the construction cost of local cluster at each organization. This mechanism is suitable for supplying insufficient CPU power to a computing system. However, data-intensive applications have a totally different feature from CPU-intensive applications. Thus, execution of data-intensive applications making use of cloud computing resources is discussed in this paper. Since a data processing system with better cost/performance ratio is required, we have constructed middleware for load distribution among cloud computing resources and a local cluster. Scalable resource management is achieved by monitoring resource usage of a local cluster and insufficient resources are acquired dynamically from a cloud service. Our middleware allocates injected jobs optimally among the local cluster and cloud computing, based on the result of the I/O disk load conditions. According to the evaluation using a data-intensive benchmark, our middleware has achieved an excellent job allocation, which is close to an ideal case.

1. はじめに

高度 IT 社会の進展によりコンピュータシステムにおいて利用可能なデータの量が増大している近年，よりスケーラブルなリソース管理の実現が望まれている．そこで期待が高まっているのがクラウドコンピューティングである．クラウドコンピューティングにおいてユーザはリソースを利用するだけであるため，システムの運用コストが大幅に削減できる．さらに必要なときに必要な機能を，必要な分だけ利用することが可能となる．

本研究においてデータインテンシブアプリケーションのためのデータ処理システムを構築するにあたり，全てのシステムをクラウドで構築することも考えられるが，元々自前のデータ処理システムを所有していることが考えられる他，運用面やコストの面でのリスクが懸念される．そのため上記で述べたクラウドコンピューティングのメリットを活かし，使用しているクラスタのシステム状況をモニタリングし，負荷が高い場合には外部のクラウドリソースへ負荷分散するミドルウェアの構築を目指す．クラウドリソースを利用しているため，特に急激に大量のキャパシティが必要となる場合に有効になると考えられる．

ローカル環境における負荷が高い場合にネットワーク越しの遠隔リソースへ負荷を分散すること自体は，グリッドコンピューティングなどの枠組みで実現できるため，新しい考え方ではない．しかし遠隔リソースとしてクラウドを利用した場合，以下のような点で従来とは異なる特徴がある．まずユーザのニーズに応じてリソースを大幅に増減できることが期待される．またセキュリティポリシにより社外にデータを置けないユーザが，データは社内に保存したまま，計算能力だけクラウドから借りるようなケースが想定される．加えてクラウドでは従量制のコストがかかることから，単に性能向上のみを目的として負荷分散を行うことはできず，コストパフォーマンスが良くなることを目指す必要がある．

さらに本研究ではデータインテンシブアプリケーションを対象負荷としている．クラウドコンピューティングにおけるロードバランスを議論している論文として[1)] や[2)] がある．しかしこれらの論文では CPU リソースの負荷分散を対象としており，データインテンシブアプリケーションは対象としていない．科学技術計算など計算処理が中心となるアプリケー

†1 お茶の水女子大学
Ochanomizu University
†2 工学院大学
Kogakuin University

Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

ションの場合は，各ノードの CPU 負荷により判断して適切な負荷分散を行うことができるが，データインテンシブアプリケーションの場合，CPU は I/O 待ちとなっていることが多く，CPU 負荷では適切な判断が行えない．そこで本研究では負荷の指標として，ディスクアクセス量を用いた．

このようにリソースとして伸縮性が高く従量制のコストがかかるクラウドを用いている点，そしてデータインテンシブアプリケーションを対象負荷として用いている点が，本研究の特徴である．

2. ローカルクラスタのシステム構成

2.1 仮想マシン PC クラスタ

ローカルシステムはクラスタのワーカノードに仮想マシンを配置した仮想マシン PC クラスタとした．IT リソースを効率的に活用するためにしばしば仮想化が用いられる．仮想化ソフトには Xen[3] を使用した．

Xen は図 1 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を提供することで仮想マシンのオーバヘッドを少なくし，物理マシンに近い性能が発揮されるよう工夫されている．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．仮想マシンモニタが仮想化のための土台となり，その上で Domain と呼ばれる仮想マシンが動作する．ホスト OS として動いているものが Domain0，ゲスト OS として動いているものが DomainU で，Domain0 は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．クラウドコンピューティングでは仮想化が重要な構成要素となっており，仮想マシンを利用することでクラウドの特徴であるフレキシブルなリソースの利用が可能となっている．

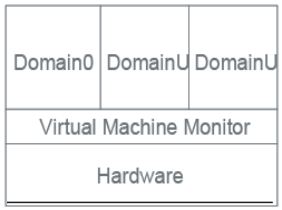

図 1 Xen の構造

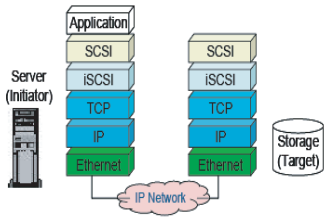

図 2 iscsi の階層構造

2.2 IP-SAN によるストレージ統合

ローカルクラスタのストレージネットワークには IP-SAN(IP-Storage Area Network) を使用した．近年情報システムにおいて処理されるデータの量が膨大になってきたことから，ネットワークストレージ技術が発展し，PC クラスタのストレージに SAN を用いることが多くなっている．SAN は，分散したストレージをネットワークで統合し，ストレージの集中管理とディスク資源の効率的な活用を可能にする．特に本研究にて使用した IP-SAN は Ethernet インタフェースと TCP/IP 対応ネットワークさえあれば導入でき，また専用網も含め広範囲に IP ネットワークのインフラが整備されているため長距離接続が可能で，クラウドコンピューティングなどの枠組を用い，計算機リソースのアウトソーシングに利用されることが期待される．

IP-SAN のプロトコルとしては iSCSI (Internet Small Computer System Interface)[4] を使用した．iSCSI の構造を図 2 に示す．iSCSI は SCSI コマンドを TCP/IP パケットの中にカプセル化することでブロックレベルのデータ転送を行う．Gigabit Ethernet/10Gigabit Ethernet が広く普及していくであろうことを考慮すると，IP-SAN をバックエンドに持つ PC クラスタ多くが使用されるようになると考えられる．

3. クラウドリソースの調達とデータ配置

3.1 評 価 環 境

負荷分散先として使用するクラウドには東京大学生産技術研究所が構築したプライベートクラウド (以下 Cloko) を使用した．これはオープンソースのクラウド構築ソフトウェアである Eucalyptus を用いて構築されたものである．Eucalyptus の特徴の一つとして米 Amazon.com 社が提供しているクラウドである Amazon EC2 (Amazon Elastic Compute Cloud)[5] の API と互換性を持っていることが挙げられ，Eucalyptus を使用することで，Amazon EC2 上のサービスをそのまま Eucalyptus で構築したプライベートクラウドに移すことが可能である．

既存研究ではこの Amazon EC2 を負荷分散先のクラウドとして使用した負荷分散ミドルウェアを構築した[6]．Amazon EC2 と互換性があること，そしてプライベートクラウドであるため拡張性を期待し本論文では Cloko をクラウドリソースとして利用する．

3.2 データの配置

本研究で対象としているデータインテンシブアプリケーションでは頻繁にデータベースアクセスが発生するため，サーバに対するデータの位置が実行時間に大きな影響を及ぼす．リ

Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

モートサイトのサーバに計算処理の一部をマイグレートする場合，処理を行うデータの配置については，以下の 3 つのケースが考えられる．

(a) 処理のマイグレート時に処理に必要なデータについてもオンデマンドでリモートサイトにコピーする場合

(b) ローカルクラスタでの処理中に遠隔バックアップが行われており，あらかじめリモートにデータが存在する場合

(c) セキュリティポリシや，データが巨大すぎるなどの理由でリモートにデータを置くことができない場合

(a) については一般にリモートへのデータ転送はスループットが低いため，この方式はデータ量が多い場合には，性能低下を招く可能性がある．ただしコピーが終わればリモートサイトからデータへ高速アクセスが可能となるため，データ量が少ない場合やアプリケーション全体の処理時間が長い場合には有効であると考えられる．

(b) の場合にはデータアクセスに関する制約が無くなるため，積極的にリモートサイトのリソースを利用した方が性能面では有利になると考えられる．

(c) の場合には，計算処理のみリモートサイトのリソースを利用しながら，データはローカルに置きリモートサイトからアプリケーション実行時にアクセスする事が考えられる．例えば Google 社が提供するクラウドサービスである Google Apps[7] においては Secure Data Connector という仕組みが提供されており，これを利用するとクラウドとローカルサイトの間にセキュアトンネルが構築され，クラウドからローカルサイトのデータに対し，安全にアクセスを行う事ができるようになる．このケースの場合には，リモートサイトから計算処理サーバだけ借りれば良く，容易に負荷分散のマイグレーションが実現できる．

ただしリモートサイトとローカルクラスタの間の通信性能が全体の実行性能に大きな影響を与えるため，ネットワークの帯域幅が小さい場合やデータアクセス頻度が高いアプリケーションの場合には，性能の大幅な低下が予想される．また，リモートサイトからローカルのストレージへのアクセスには制限がある場合もあり，これらの問題をクリアしなければならない．

このようにリモートサイトのリソースを利用して負荷分散を行う事を考える際，データインテンシブアプリケーションの場合には，データをどのように扱いどこに置いて実行するか考える事が重要である．さらに上記の (a) と (b) のケースのように，データをリモートサイトに配置して計算処理を実行する場合には，リモートサイトにおけるストレージについても，何台用いデータをどのように配置すべきかについて検討する必要がある．

本研究ではひとまず (b) 利用したいデータが既にリモートサイトに存在する環境での実験を行う．また実験の後半では (c) データが常にローカルサイトに存在する環境を意識し，ローカルサイトとリモートサイト間のネットワークスループット，ディスクアクセス性能，また実際にアプリケーションを実行した際の性能を調べた．

4. データ処理アプリケーション最適配置ミドルウェア

4.1 最適配置の概要

本研究ではデータインテンシブアプリケーション実行時に，ローカルクラスタの I/O 負荷をモニタリングすることでジョブ量を測定し，そこから現在の負荷が大きいと判断した場合には外部のクラウドリソースへ動的に負荷分散を実現するミドルウェアの構築を目指す．

クラウドコンピューティングの一つの大きな特徴として，必要なときに必要な分だけリソースが使用可能であることがあげられる．ローカルクラスタでの処理量を減らし，その分クラウド側で多くの処理を並列して実行すればアプリケーション全体の実行時間が短くなることが期待される．しかしクラウドは従量制のコストがかかるということも特徴であるため，実行時間のみを考慮した場合コストが膨大になる可能性がある．そのため現実にクラウドリソースを使用した負荷分散システムを使用する場合は，実行時間といったパフォーマンスとともに，コストパフォーマンスも考慮してリソースを配分することが必要である．

本研究ではアプリケーションの実行時間に制限が設けられた場合にローカルクラスタとクラウドコンピューティングリソース間でどのようにジョブを分けるのがいいのか，実行が制限時間内に終わりかつ最もコストパフォーマンスが良くなるよう検討する．すなわち与えられた時間内に実行可能な最もコストが低いリソースの使用法を求めることが本研究における最適配置である．図 3

システム環境については，マシン台数はローカルクラスタでは使用できる台数に制限があるという現実的な環境で，その不足分をクラウドリソースで補う．クラウド側は使用ノードの数の制限がない．またジョブは順次投入されていく環境を想定している．

4.2 ミドルウェアの動作概要

図 4 にミドルウェアのアーキテクチャを示す．ユーザはまずアプリケーション全体の実行制限時間を決定し，それを Limit time とする．ミドルウェアは Limit time を越えない範囲で，かつクラウドのコストが最も最小になるようローカルクラスタとクラウドに仕事を振り分ける．

ミドルウェアはローカルクラスタのディスク I/O を測定する Monitor 部とジョブを振り

Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

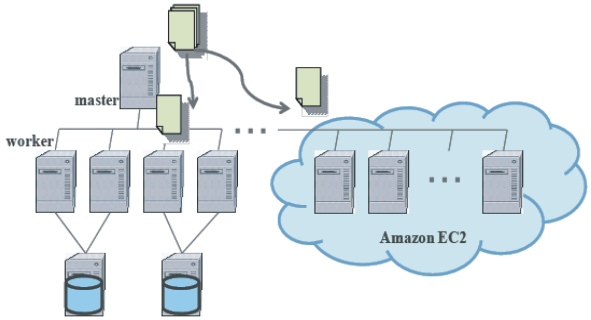

図 3 ローカルクラスタからクラウドリソースの調達

分ける Dispatch 部に分けられる．以下にそれぞれの仕事を示す．

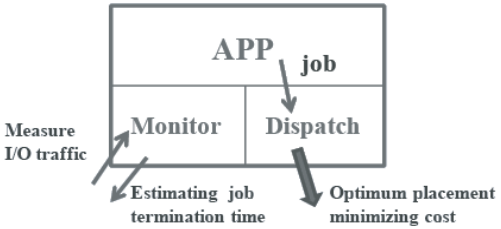

図 4 ミドルウェア

【Monitor 部】

- 定期的にローカルクラスタの I/O 負荷を dstat コマンドにより測定.
- I/O 負荷からジョブ量をチェック.
- ジョブの実行終了時間を見積もる.

【Dispatch 部】

Dispatch 部では Monitor 部が見積もったジョブの終了時刻を元にクラウドへの仕事の振り分けを判断する.

- 順次投入されるジョブを，まずはローカルクラスタで実行していく.
- Monitor 部が予測したジョブの終了時刻が，ユーザ指定の Limit time を超える場合はクラウドに処理を依頼.
- クラウド内では Limit time 内で最もコストが低くなるようマシンの台数やジョブの配置を決定.

クラウドでのジョブの振分けについては多くのマシンを借りたほうが実行時間が短くなるが，その分コストが課金される．そのため最もコストパフォーマンスが良くなるマシンの台数を求める．このときコストは（実行時間×台数）で計算している．

5. 最適配置ミドルウェアの評価

5.1 評価実験の概要

ここからはジョブの最適配置ミドルウェアの精度を評価した実験について示す．評価のためのデータインテンシブアプリケーションとしては PostgreSQL のベンチマークである pgbench[8] を用いた．データベースの大きさは 7.5GByte，サーバで実行するユーザ数 1 に対するトランザクション数を 1000 とした.

pgbench のデータを 5 秒ごとに 4 ユーザずつ合計 10 回を順次投入し (図 5)，1 回毎にローカルクラスタとクラウドのどちらに投げるかを決定する．またシステムに大きな負荷をかけ，より分かりやすくミドルウェアの評価を行うため，1 回の実行につき 10 回 pgbench を繰り返している．ジョブ投入間隔は全体の実行時間よりも十分に短い.

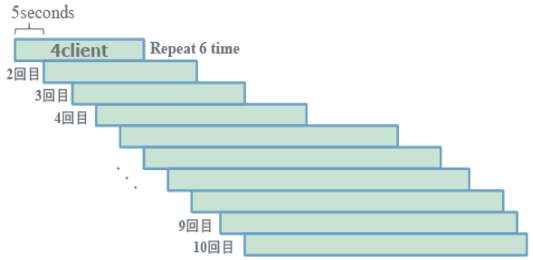

図 5 ジョブ投入イメージ

評価では上記のように pgbench を 10 回投入する際に，ミドルウェアを使用した場合と，ミドルウェアを使用しない理想的なリソース配置をした場合の振舞を比較し，ミドルウェアの精度を測定する.

Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

このときミドルウェアを使用した場合は，ローカルディスクの I/O 負荷から実行にどのくらいの時間がかかるかを見積り，Limit time を超える場合にはクラウドリソースへ負荷分散を行う．また理想的なリソース配置とは，単純にローカルクラスタとクラウドにおける pgbench の実行時間より判断した配置である．

5.2 ミドルウェアを使用した場合

ローカルクラスタの負荷が大きくなった場合にクラウドリソースに負荷分散を行う仕組みについて，ディスクアクセス量により自動で判断するミドルウェアを使用した場合と，ローカルのクラスタとクラウドにおける pgbench の実行時間とコストから求めた，最も理想的に振り分けた場合との振舞を比較し評価を行う．ミドルウェアにおいてディスク I/O の値から実行の終了時刻を見積もる際には，下記の図 6 と図 7 を使用する．

図 6 はローカルクラスタにおいて pgbench のクライアント数を変化させたときのそれぞれのディスク負荷を測定したものである．この図の Disk-Read よりクライアント数 1〜7 において負荷が右肩上がりになっていることが確認された．

また図 7 はローカルクラスタに pgbench を一度に 1client ずつ 5 秒毎に投げる繰り返しの数を変化させ，それぞれの実行時間を測定したものである．

ミドルウェアはモニタリングで得られた I/O 負荷の値と図 6 の表の Disk-Read の値を見比べることで，現在行われているジョブ量を推測し，かつ図 7 から終了時刻を見積もる．

User	Disk Read (kByte/sec)	Disk Write (kByte/sec)
1	3273	29000
2	3248	33666
3	3932	32000
4	4568	35000
5	5048	38000
6	5391	38333
7	5440	33000
8	5327	28666
9	5484	42500
10	7093	42000

図 6 ディスクアクセス負荷

ただ図??にてクライアント数が 7 以降の場合，処理が飽和状態となり，ジョブが増えてもモニタリングには現れていない．そのためクライアント数が 7 相当以上の場合にはローカルクラスタでの処理が飽和状態であると判断し，クラウドに処理を依頼することとする．

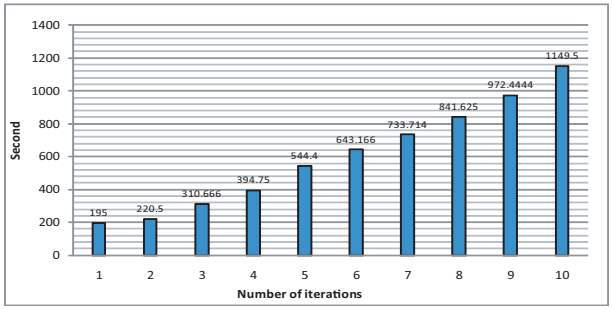

図 7 pgbench 実行時間 (Local)

5.3 最も理想的な振分けをした場合

理想的なリソース配置とは，単純にローカルクラスタとクラウドにおける pgbench の実行時間より判断する．図 7 よりローカルクラスタに pgbench を 1client ずつ 5 秒毎に投げた場合の実行時間が分かる．また図 8 はクラウドにおいて pgbench を 1client ずつ 5 秒毎に投げた場合の実行時間である．これらのグラフからローカルクラスタとクラウドへの理想的なジョブの振り分けを判断する．

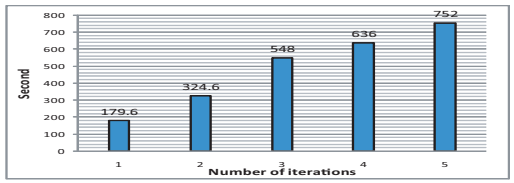

図 8 pgbench 実行時間 (Cloko)

【理想的なジョブの振分け例】

例えばユーザが Limit time を 700 秒と指定した場合，実験において 4 クライアントずつ 5 秒間隔で 10 回ジョブを投入するうち，図 7 より 6 回目まではローカルに投げることがで

Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

きるが，7 回目もローカルに投げると実行時間が 700 秒を超える．そのためローカルでは 6 回，残りの 4 回はクラウドで実行するのがよい．

クラウドでは 4 回× 4 クライアント＝ 16 クライアント分の処理をする必要がある．マシンの台数については 16 クライアントを全て別々のマシンに投入する場合の 16 台借りる場合から，全て同一のマシンで実行する場合の 1 台借りる場合が考えられる．ただ実行時間を考慮した場合，3 台以下になると 1 台につき 5 クライアント実行しなければならないマシンが発生し，実行時間が 700 秒以上になってしまう (図 8)．

また実行時間×台数によりコストも考慮にいれると，4 台借りて 1 台につき 4 クライアントずつ実行するのが最もコストパフォーマンスがよい．

以上から制限時間が 700 秒と指定された場合には，10 回ジョブを投入するうち 6 回目まではローカルクラスタに投げ，残りの 4 回は 1 台に 4 クライアントずつ投入するのがよい．

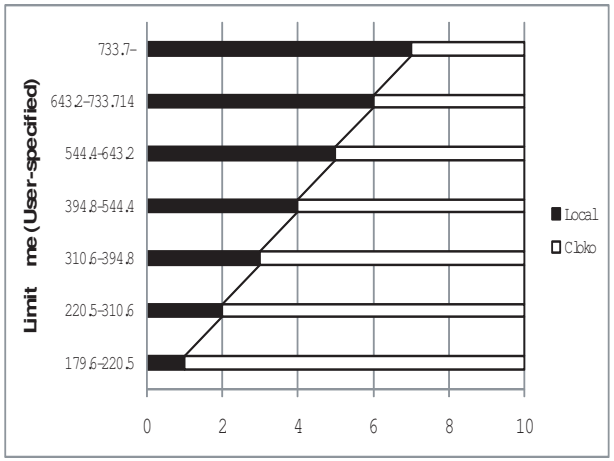

図 9 理想的なジョブの振分け

図 9 に pgbench を 10 回投入する際に，何回ずつローカルクラスタとクラウドで実行するのが理想的かを示す．

アプリケーション実行がローカルクラスタとクラウドの両方で行われるよう，図 7 と図 8 よりユーザが指定する制限時間は 179.6 秒から指定できるものとする．

縦軸はユーザが指定する実行の制限時間である．ユーザ指定の Limit time が比較的長い場合は，時間がかかってもたくさんの処理をローカルクラスタで行うほうがクラウドでのコストが割安になるためクラウドでの実行の割合が高くなっている．逆に Limit time が短い場合は速く実行することが望まれるため，クラウド側でたくさんのマシンを借りて並列に実行することが望まれる．そのためクラウドでの実行の割合が高くなっている．

5.4 実験環境

Eucalyptus を使用したプライベートクラウドのマシンスペックを表 1 に，ローカルクラスタのスペックを表 2 に，実験環境を図 10 に示す．ローカルクラスタの各計算ノードには DomainU(virtual machine) を一つずつ配置した．

使用するリソースは、ローカルクラスタは限定（サーバが 4 台)、リモートのクラウドは台数無制限とする．

表 1 Experimental setup：Cloud

OS	Linux 2.6.24-19-xen (Debian)
CPU	Intel (R) Xeon(TM) 2.40GHz
Main Memory	1GB

ストレージについては，ローカル環境では iSCSI ストレージを使用し，クラウドではローカルストレージを使用する．またデータについては上記の (b)「遠隔バックアップなどによりあらかじめリモートにデータが存在する」場合を考える．

表 2 Experimental setup：PCs

OS	Linux 2.6.18-128.el5(CentOS5.3)
CPU	Initiator：Intel (R) Xeon(TM) 3.6GHz Target：Intel (R) Xeon(TM) 2.66GHz
Main Memory	Initiator(DomainU)：1GB Target：8GB
iSCSI	Initiator：iscsi-initiator-utils Target：iSCSI-Enterprise-Target
Monitoring Tool	dstat

5.5 実験結果と考察

図 11 にローカルクラスタとクラウドの投げ分けをした場合，ユーザが指定したそれぞれの制限時間においてミドルウェアが 10 回中何回ローカルクラスタとクラウドへ投げたかを示す．7 回以上ローカルで実行するとモニタリングにおいてディスクアクセスが飽和となり，

Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

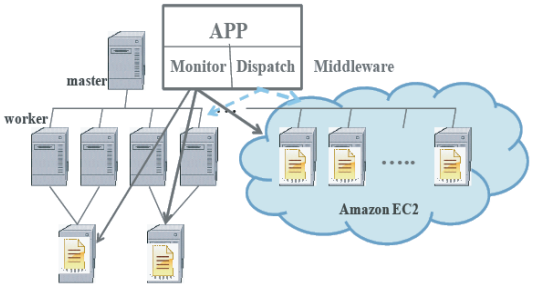

図 10 実験環境図

負荷が測定できないため 8 回以上ローカルで実行することはない．

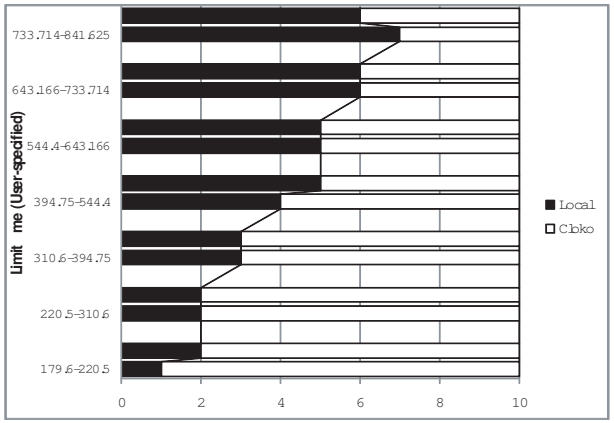

図 11 ミドルウェアを使用した際のジョブの振分け

この結果から，多少の誤差はあるものの，理想とかなり近い振り分けができていると言える．つまりローカルクラスタのモニタリング結果により，理想に近いジョブの振り分けができていることが分かる．

6. リモートサイトからのデータアクセス

上記の実験では本稿の「データの配置」の章での (b)「データインテンシブアプリケーションを実行するために必要なデータが常にアプリケーションを実行するサーバ内にある」ことを前提とし実験を行ってきた．今後はより現実的な環境を想定して，(a) や (c) の場合も考慮する必要がある．そこで本稿では (c)「データが常にローカルサイトに存在する」環境を想定し，ローカルサイトとリモートサイト間のネットワークスループット，ディスクアクセス性能，また実際にアプリケーションを実行した際の性能を調べた．

研究室内のサーバとクラウドを接続し，クラウドからローカルへのディスクアクセスには iSCSI を使用した．

図 12 にネットワークスループット，図 13 にディスクアクセスの結果を示す．ネットワークスループットの測定には，大きいサイズのファイルを転送する手法を用いた．またディスクアクセスの性能についてはベンチマークの dbench を使用した．

ディスクアクセスのグラフには比較としてクラウドのローカルディスクアクセス性能も示す．

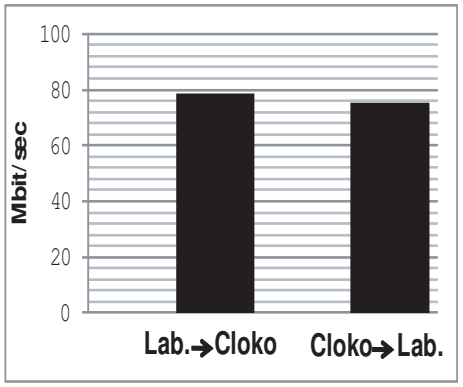

図 12 ネットワークスループット

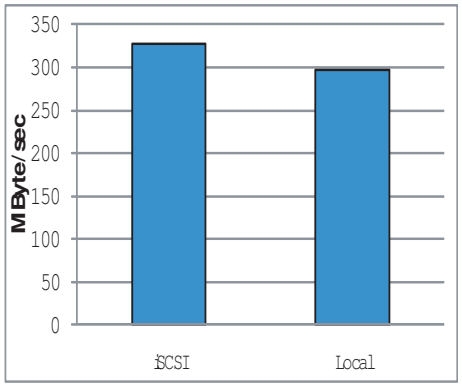

図 13 ディスクアクセス性能

以上のグラフよりネットワークについてもディスクについても十分実用的な性能がでていることが分かる．

図 14 には pgbench において，クライアント数が 1 のときと 10 のときの実行時間を示す．比較として研究室内のローカルのマシンで pgbench を実行した際のデータも示す．

その結果アプリケーション実行についてはローカルクラスタで実行するよりも性能が劣ることが分かった．今後は上記の結果を参考に，ミドルウェアにおいてデータが常にローカル

Vol.2010-DPS-145 No.24
Vol.2010-GN-77 No.24
Vol.2010-EIP-50 No.24
2010/11/26

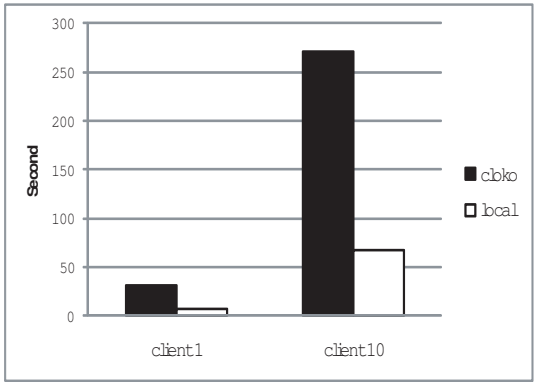

図 14 pgbench 実行性能

に存在する場合の評価を行っていく．

7. まとめと今後の課題

スケーラブルなデータ処理システムを実現するため，手元のクラスタ負荷をモニタリングし，負荷が大きい場合は外部のクラウドコンピューティングリソースへ負荷分散を実現するミドルウェアを構築した．その際に対象をデータインテンシブアプリケーションを対象とし，システムのボトルネックとなるディスク I/O を負荷の判断指標とした．

ローカルクラスタとクラウド間ののジョブの振分けを行ったところ，ミドルウェアを使用した場合，理想に近い振分けができていることが分かった．現在はジョブの振分け指標としてローカルクラスタのディスクアクセス量を用いているが，今後は負荷が少ないところでも正しく投げ分け判断が行えるようモニタリングの精度を高めていく．

本研究においてはデータが常に処理を行うサーバ上にあることが前提だったが，セキュリティポリシなどの制約でデータは常にローカルに存在する場合や，処理のアウトソーシングと同時にデータのコピーを行うなど，より現実的な条件を考慮した実験を行う．

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．

参 考 文 献

1) G. Jung, K. R. Joshi, M. A. Hiltunen, R. D. Schlichting, and C. Pu: ”Generating Adaptation policies for Multi-Tier Applications in Consolidated Server Environments,” In Proc. 5th IEEE International Conference on Autonomic Computing (ICAC2008), pp.23-32, June 2008.
2) E. Kalyvianaki, T. Charalambous, and S. Hand: ”Self-Adaptive and Self-Configured CPU Resource Provisioning for Virtualized Servers Using Kalman Filters,” In Proc. 6th International Conference on Autonomic Computing and Communications (ICAC2009), June 2009.
3) Xen : http://www.xen.org/
4) iSCSI RFC: http://www.ietf.org/rfc/rfc3722.txt
5) Amazon EC2: http://aws.amazon.com/jp/ec2/
6) 豊島 詩織, 山口 実靖, 小口 正人: ”データ処理アプリケーションのクラウドリソースとローカルクラスタ間における負荷分散ミドルウェアの検討” 並列/分散/協調処理に関するサマー・ワークショップ (SWoPP2010)，CPSY-6，金沢，2010 年 8 月．
7) Google Apps:http://www.google.co.jp/apps/intl/ja/business/index.html
8) pgbench: http://www.postgresql.jp/
9) Asuka Hara, Kikuko Kamisaka, Saneyasu Yamaguchi, and Masato Oguchi: ”Analyzing Characteristics of PC Cluster Consolidated with IP-SAN using Data-Intensive Applications,” In Proc. 20th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS2008), No.631-042, November 2008.
10) Amazon Web Service: http://aws.amazon.com/
11) Aravind Menon,Alan L.Cox,Willy Zwaenepoel: ”Optimizing Network Virtualization in Xen,” 2006 USENIX Annual Technical Conference, pp.15-28, May 2006.
12) Jose Renato Santos,Yoshio Turner,G.(John)Janakiraman,Ian Pratt:”Bridging the Gap between Software and Hardware Techniques for I/O Virtualization,” 2008 USENIX Annual Technical Conference, pp.29-42, June 2008.
13) Tanimura Yusuke, Ogawa Hirotaka, Nakada Hidemoto, Tanaka Yoshio, Sekiguchi Satoshi: ”Comparison of Methods for Providing An IP Storage to A Virtual Cluster System,” IPSJ SIG Notes 2007, pp.109-114, March 2007.