

ハイブリッドクラウドにおける 遠隔ストレージアクセスを用いたジョブの最適配置ミドルウェア

笠江優美子[†] 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: [†]yumiko@ogl.is.ocha.ac.jp, ^{††}oguchi@computer.org

あらまし 近年、コンピュータシステムにおける情報量が爆発的に増加している。その処理プラットフォームとして、ハイブリッドクラウドが注目されている。ハイブリッドクラウドには、プライベートクラウドとパブリッククラウドを併用することで、自社の持つリソースを無駄なく使いながら、スケーラブルな処理が実現できるというメリットがある。しかし、実社会において、特にパブリッククラウド内でのデータ処理に対する信頼性の問題により、ハイブリッドクラウドの導入があまり進められていない。また、ハイブリッドクラウドでの処理には、処理性能と従量制料金という、トレードオフの関係にあるコストが生じる。これらコストに対して、適切なコストバランスを実現するシステムも求められている。そこで本研究では、ビッグデータ処理基盤としてのハイブリッドクラウド環境において、ローカルに保持するストレージプールに対し、パブリッククラウドから遠隔ストレージアクセスを行い、データ自体を外部サイトに保存しない前提において、生じるコストという評価軸のもと、最適なジョブ配置を行う手法を提案し、ミドルウェアとして実装した。また、本提案手法を適用することによって、想定環境でのアプリケーション処理において、最適なジョブ配置が行えること示した。

キーワード ハイブリッドクラウド, 負荷分散ミドルウェア, コストバランス, iSCSI

Proposing the Method for Load Distribution on the Hybrid Cloud Environment using Remote Storage Access

Yumiko KASAE[†] and Masato OGUCHI[†]

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

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

1. はじめに

近年、情報通信の発達に伴って、あらゆる分野において、コンピュータシステムにおける情報量が爆発的に増加している。それら大量のデータは、1箇所に集められ、さらに分析を加えることで、新たな価値のある情報を生み出すという、ビッグデータ処理の効率の良い実現が求められている。その処理のプラットフォームとして、ハイブリッドクラウドが注目されている。ハイブリッドクラウド環境では、プライベートクラウドとパブリッククラウドを併用することで、それぞれの持つ拡張性や安全性などの長所と短所を補い合う事ができる。特にデータ処理においては、時々刻々と増加するデータに対して、必要に応じてスケーラブルな処理ができるハイブリッドクラウド環境は、処理に適していると言える。

しかし、実社会において、ハイブリッドクラウドの積極的な

導入はあまり行われていない。これは、特にパブリッククラウドでの処理の不透明性により、自社の持つデータを外部サイトで保存することへの懸念があるためである。また、ハイブリッドクラウド環境での性能と処理に伴うコストには、トレードオフの関係が生じてしまう点もその要因である。つまり、多くのデータ処理を高速に行うために、パブリッククラウドを多く使って処理を行えば、全体の実行時間は短くなるが、従量制料金がかかり過ぎる。逆に、プライベートクラウドのみで行うと、従量制料金は少なくなるが、処理に時間がかかる。このように、ハイブリッドクラウド環境での効率的な処理に関して、どちらのクラウドをどれくらい使えば良いかを、性能と必要なコストに基づき適切に判断し、最適なジョブ配置を行えるシステムが必要である。

そこで、本研究では、ハイブリッドクラウド環境での処理について、企業の持つセキュリティポリシーに従って、データを外

部サイトに保存せず、ローカルにあるストレージに保持し、パブリッククラウドから遠隔アクセスを行う場合において、性能と金銭的成本という2つの評価軸に基づき、最適なジョブ配置を目指す手法を提案し、ミドルウェアとして実装する。

本ミドルウェアでは、主に計算処理を行うCPUインテンシブアプリケーションだけでなく、ディスクへI/O処理の割合が多いデータインテンシブアプリケーションの効率的な処理へ向けたジョブ配置も対象とする。しかし、一般的に、CPU使用率などで負荷を正確に判断できるCPUインテンシブアプリケーションと異なり、データインテンシブアプリケーションにおける処理の場合、リソースを飽和するまで使い切るために必要な、効率的なリソースの利用状況が判断しづらい。これに対し、本手法では、CPUとI/Oのバランスを可変とする自作ベンチマークを用い、実行環境となるハイブリッドクラウドの性質を調べた上で、リソースへのI/O状況を元にジョブの実行状況を判断する手法を提案する。

本論文では、本研究で提案する手法と、開発したミドルウェアについて詳細に述べると共に、ローカルストレージに対しパブリッククラウドから遠隔アクセスを行った場合において、実際にデータインテンシブアプリケーションの処理を行う際に、本ミドルウェアを適用した。そして、その実行コストについて、性能と金銭的成本という評価軸で評価し、リソースを効率的に使う最適なジョブ配置が提供できることを示す。金銭的成本には、パブリッククラウドの従量制料金とプライベートクラウドの消費電力料金が含まれる。

2. ハイブリッドクラウド導入への懸念について

クラウドコンピューティングは、ネットワーク上に存在するサーバから、必要なソフトウェアやハードウェア資源を利用するサービス形態を示し、その提供するサービスの種類から、SaaS(Software as a Service)や、PaaS(Platform as a Service)、IaaS(Infrastructure as a Service)などに分類できる。

本研究では、特にIaaSを対象として検討を行うが、IaaSについてのクラウドの利用形態の種類として、パブリッククラウドとプライベートクラウドがある。パブリッククラウドは、インターネットを通じて利用でき、クラウドを提供するプロバイダに従量制の料金を支払うことで、スケーラブルなクラウドサービスを受けることができる、一般的なクラウド形態である。プライベートクラウドは、社内などの閉じたネットワーク上で利用するクラウドのことで、セキュリティに配慮した構成で、クラウド環境を利用することができる。また、これらに加え、近年、ハイブリッドクラウドという利用形態も注目されている。ハイブリッドクラウドでは、プライベートクラウドとパブリッククラウドの2つのクラウドを併用することで、自社の持つリソースを無駄なく使いながら、スケーラブルなクラウド環境を利用することができる。これらの点から、時々刻々と増加するデータを処理するビッグデータ処理の、効率的処理基盤として、ハイブリッドクラウドは適していると考えられる。

しかし、実社会において、ハイブリッドクラウドの積極的な導入が進められていない。総務省によって行われた平成23年

通信利用動向調査[1]では、日本におけるクラウドの導入状況は、導入予定を入れても、半数以下であるとされている。これは、以下の様な懸念があるためだと考えられる。

2.1 パブリッククラウドにおけるデータ配置

ハイブリッドクラウド導入が進まない理由として、データ配置の難しさが挙げられる。

特に、パブリッククラウドでは、企業の持つセキュリティポリシーなどにより、データを外部サイトに保存できないという制約が存在することが多い。平成23年通信利用動向調査の結果では、企業がクラウドサービスを導入しない理由の第2位にセキュリティへの不安を挙げている。また、データを外部サイトに持ち出すことが許されたとしても、データ自体が巨大である場合、データ転送にコストがかかってしまう。さらに、パブリッククラウド内のストレージサービスを利用した場合、ローカルのストレージのデータとの同期を取る必要がある。パブリッククラウドにデータを保持する場合、その容量に応じて課金されることも大きな理由である。

本研究では、これらのことを考慮し、企業の持つセキュリティポリシーに従って、データはローカルにあるストレージプールに保持しておき、外部サイトにデータを保存しないという前提のもと、ハイブリッドクラウドでデータ処理を行う場合には、パブリッククラウドから遠隔アクセスを行う。これは、パブリッククラウドのインスタンス(仮想マシン)から、処理能力のみを借りる場合を想定したものである。

2.2 コストのトレードオフ関係

また、ハイブリッドクラウド環境で処理を行う際には、性能と必要なコストには、トレードオフの関係が生じてしまう。つまり、多くのデータ処理を行うために、プライベートクラウドに加えて、パブリッククラウドを多く使って処理を行えば、処理は早く終わるが、多くのインスタンスを使って処理を行うため、従量制料金がかかり過ぎる。逆に、パブリッククラウドをほとんど使わずに、プライベートクラウドのみで処理を行なうと、従量制料金は少なくなるが、処理に時間がかかり過ぎる。このように、ハイブリッドクラウド環境での効率的な処理に関して考える場合、どちらのクラウドをどれくらい使えば良いのか、性能と必要なコストに基づき判断し、最適なジョブ配置を行えるシステムが必要である。

そこで本研究では、そのようなシステムを実現する手法を提案し、ミドルウェアとして実装した。本ミドルウェアで考慮するコストは、ジョブの処理時間という時間的コスト、および、パブリッククラウドの従量制料金とプライベートクラウドの消費電力料金を足した金銭的成本の2つである。本手法で考える金銭的成本の中に、プライベートクラウドの消費電力料金が含まれるのは、最近の世界的なエコ志向に加え、大量のデータを処理する上での電力の増加問題、更には電力供給力の減少問題など、消費電力削減の重要性が増しており、電力においても、無駄のない利用が求められるためである。プライベートクラウドの金銭的成本には、システム導入時の固定的な費用も含まれる筈であるが、その金額は一概には定義しにくいため、ここでは減価償却済みであるケースを想定し、固定費用はゼロ

として評価を行った。また、金銭的成本の中に、パブリッククラウドの消費電力料金が含まれていないが、パブリッククラウドを提供するプロバイダが、利用者に対して個々のリソースが消費した電力料金を公開するのは、一般に困難であると考えられ、パブリッククラウドにおいては、従量制料金にその消費電力料金が含まれると考えることができる。

3. 想定するハイブリッドクラウド環境の構築

本研究で想定するクラウド環境を、IaaS のクラウド環境構築ソフトウェア Eucalyptus [2] を使って構築した。また、Eucalyptus を用いて、2つのクラウド環境を構築し、その間を人工的な遅延を発生させる Dummynet で接続することで、ハイブリッドクラウド環境を実現した。さらに、ローカル環境側には、3台の RAID-0 構成のストレージから成るストレージプールを設けた。構築したハイブリッドクラウド環境を、図1に示す。また、それぞれを構成するサーバは表1から5に示す通りである。

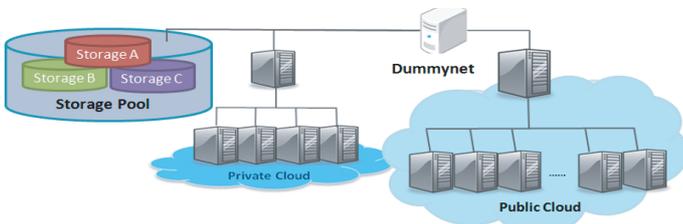


図1 構築したハイブリッドクラウド環境

表1 Private Cloud Frontend

OS	Linux 2.6.38 / Debian GNU / Linux 6.0
CPU	Intel(R) Xeon(R) CPU @ 3.60GHz 1 core
Memory	4GByte

表2 Public Cloud Frontend

OS	Linux 2.6.38 / Debian GNU / Linux 6.0
CPU	Intel(R) Xeon(R) CPU @ 2.40GHz 1 core
Memory	1GByte

表3 Private Cloud NodeServer

OS	Linux 2.6.32-xen-amd64 and xen-4.0-amd64 / Debian GNU / Linux 6.0
CPU	Intel(R) Xeon(R) CPU @ 2.66GHz 1 core
Memory	8GByte

表4 Public Cloud NodeServer

OS	Linux 2.6.32-xen-amd64 and xen-4.0-amd64 / Debian GNU / Linux 6.0
CPU	Intel(R) Xeon(R) CPU @ 3.60GHz 1 core
Memory	4GByte

表5 Storage Pool

OS	Linux 2.6.32-xen-amd64 and xen-4.0-amd64 / Debian GNU / Linux 6.0
CPU	Intel(R) Xeon(R) CPU @ 1.60GHz 1 core
Memory	2GByte
Disk	SAS Seagate Cheetah 15K.5 SAS 73G * 2
RAID Controller	SAS/iR

この環境において、それぞれのクラウドからストレージプールに対し、iSCSI アクセスを行う。iSCSI とは、データ転送を行う SCSI プロトコルを TCP/IP ネットワーク上で使用する規格のことである。専用のケーブルを用いて構築される FC-SAN に比べ、安価に SAN 環境を構築できる。

本実験環境において、インスタンスのローカルのディスクへアクセスする場合と、インスタンスからストレージに対し iSCSI アクセスする場合の Sequential Read/Write の性能測定をベンチマークツール Bonnie++ を用いて行った。

図2にその測定結果を示す。横軸は人工遅延発生装置 dummynet に設定した、往復遅延時間である。実環境では、東京-大阪間の遅延が約 16msec である。

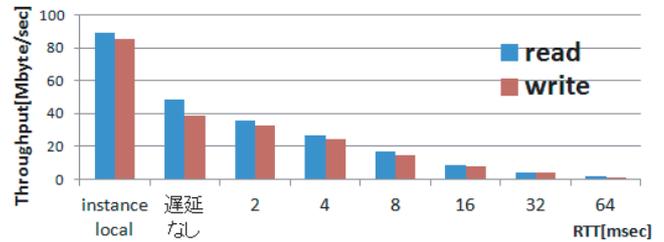


図2 iSCSI 性能測定結果

図2が示す通り、ローカル環境で iSCSI アクセスを行った場合に比べると、遅延が増加するにつれて、Read 性能、Write 性能の低下がみられる。

4. 提案する負荷判断手法

本論文で提案するミドルウェアでは、CPU インテンシブなジョブだけでなく、データインテンシブなジョブも処理の対象としている。どちらのジョブにおいても、ハイブリッドクラウド環境での無駄のないリソース利用のためには、その負荷を適切に判断し、それぞれのリソースを「効率良く使える限界まで使い切る」ということが大変重要である。そこで、本ミドルウェアでは、CPU 処理とディスク処理それぞれにおいて、リソースを使い切った状態を決め、その情報を元に、リソースの負荷の状況を判断していく。それぞれの処理での負荷の判断手法は以下の通りである。

4.1 CPU 処理に基づいた負荷の判断

CPU 処理を伴う CPU インテンシブなジョブについての負荷分散の研究は、今まで多くなされており、負荷の判断として CPU 使用率が使われている。本ミドルウェアにおいても、CPU 使用率を元に判断し、リソースを使い切った状態を CPU 使用率が 100% とし、ジョブ処理時に CPU 使用率が 100% に達すると、負荷分散を行う。この CPU インテンシブなジョブの最適配置については、他の研究でされている手法と本質的には変わらないため、本提案の特徴とはしない。

4.2 ディスク処理に基づいた負荷の判断

4.2.1 負荷の判断における問題点

ディスク処理を伴うデータインテンシブなジョブの場合、CPU 処理に比べ、その負荷が処理的な飽和に達しているかどうかの確定的な判断が難しい。データインテンシブなジョブの場合は、処理が I/O 待ちとなってしまう場合が多く、CPU 使用率を用いても適切な判断が行えない。また、ディスクの処理性能は、そのシステムに依存するという問題もある。そこで、本提案手法では、ミドルウェア実行前に学習フェーズを実行する。

4.3 学習フェーズ

4.3.1 概要

学習フェーズでは、Random Read 処理のみを行う自作ベンチマーク Disk Bench を用い、ミドルウェア実行環境のディスクに対し負荷をかける。そして、その負荷の処理時間からディスクの処理的なディスクの飽和状態を判断し、その飽和状態と判断されたディスクの情報を分析することで、処理的なディスク飽和を定義する。特に、本手法では、ディスクの情報として、飽和状態をディスクに溜まるキューの長さを取得することで、「ディスクリソースを使い切った状態の条件」を定義する。次節で、学習フェーズの具体的な手法について説明する。

4.3.2 実行手法

学習フェーズは、ミドルウェア実行環境のディスク処理性能測定を行っていく。まず、ミドルウェア実行環境のディスクリソースに対し、自作ベンチマーク Disk Bench を使ったジョブを複数同時処理させていく。その時、その処理にかかった処理時間と、その時のディスクに溜まるキューの長さを、ディスク処理の統計情報を記述する `/proc/diskstats` ファイルから測定する。そして、処理時間の測定結果と、同時処理した分のジョブを逐次的に処理する場合の処理時間を比較する。

一般的に、ディスクリソースでの処理に余裕があれば、ジョブを同時処理した方が、その分のジョブを逐次処理した時に比べ実行時間が短いはずである。しかし、同時処理するジョブ数を増加させていくと、同時処理した場合の方が遅くなる点がある。そこをディスクリソースを使い切った状態と判断し、その時のディスクに溜まったキューの長さを、「ディスクリソースを使い切った状態の条件」とする。ディスクアクセスは複数のジョブからのリクエストがランダムに到着し、ブロックサイズが一定ならジョブの処理時間はほぼ一定、窓口はディスクごとに1つという待ち行列モデルに従うと見なせる。従って、キューの長さは、ジョブからの I/O リクエストの混雑度、すなわち入出力の飽和の程度を正確に反映していると考えられる。

また、Disk Bench 内には、2つの可変なパラメータが存在する。一度にどれだけ読み込むかという読み込み量と、この読み込みを何回繰り返すか、というパラメータである。学習フェーズでは、これらパラメータを網羅的に変化させることで、あらゆるディスクアクセスパターンを想定し、ディスク処理の飽和の条件を明確に定義していく。

5. 開発した最適配置ミドルウェア

学習フェーズにおいて、処理実行環境における「ディスクリソースを使い切った状態の条件」を定義した後、開発したジョブの最適配置ミドルウェアを介し、ジョブ処理を行う。本章では、開発したミドルウェアについての詳細を述べる。

5.1 ミドルウェアの構造

図3に、ミドルウェアの動作を示す。ミドルウェアは、Monitor 部と Dispatch 部から成る。Monitor 部では、ハイブリッドクラウド内に存在するそれぞれのリソースに対して、ジョブの優先配置度順に、定期的にリソースの状態確認を行っている。リソースの状態確認とは、前章で述べた通り、CPU 処理において

は CPU 使用率、ディスク処理においては、ディスクに溜まるキューの長さを測定している。また、リソースの負荷状況は、CPU 処理、ディスク処理の両方を同時に確認しており、どちらかが使い切った状態となると、そのインスタンスはリソースを使い切られた状態であると判断する。Dispatch 部では、ジョブを受け取り、Monitor 部からの情報を元に、ジョブを配置していく。ただし、ジョブがディスク処理を行うもの場合、実際のデータは、ストレージプール内にあるため、そこへデータアクセスすることとなる。

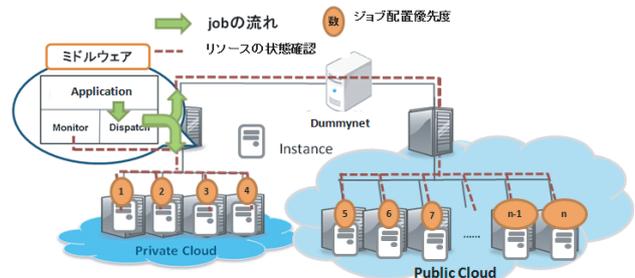


図3 ミドルウェアの動作

5.2 負荷分散基準

ミドルウェアの負荷分散の基準として、前章で述べた通り、そのリソースが処理的飽和状態になり次第、負荷分散を行う。具体的には、ジョブ処理をしているリソースから取得した情報が、CPU 処理においては CPU 使用率 100%、ディスク処理においては、ディスクに溜まるキューの長さが学習フェーズで得られた、「ディスクリソースを使い切った状態の条件」にそれぞれ達したら、負荷分散を行なっていく。つまり、CPU 処理における負荷分散の閾値は CPU 使用率 100%、ディスク処理における負荷分散の閾値は、学習フェーズで得られたディスクに溜まるキューの長さとなる。特に、ディスク処理における負荷分散の閾値は、学習フェーズで得られた「ディスクリソースを使い切った状態の条件」を元に、ユーザが設定可能である。

5.3 アルゴリズム

本ミドルウェアのアルゴリズムを以下に示す。初めてミドルウェアを利用するハイブリッドクラウド環境で実行する際には、前章で示した学習フェーズにおけるディスクの性能測定を行い、「ディスクリソースを使い切った状態の条件」を定義する必要がある。

- (1) ユーザは、学習フェーズで得た閾値の範囲を元に、ディスク処理における閾値を設定し、ミドルウェアを実行する。
- (2) ミドルウェアは、投入されたジョブを受け取る。
- (3) プライベートクラウド内のインスタンスで、優先配置度順に、リソースの負荷状態が閾値以上であるか調べ、閾値以下であればそこで実行し、(2)へ。プライベートクラウド内のリソース全ての負荷状態が閾値以上であれば、(4)へ。
- (4) パブリッククラウドでジョブの優先配置度順に、リソースの負荷状態が閾値以上であるか調べ、閾値以下であればそこで実行して、(2)へ。見つからなければ、(5)へ。

(5) パブリッククラウドにおいて、借りるインスタンスを増やし、そこで実行して (2) へ。

6. ミドルウェア適用実験と評価

本研究では、3章で示した想定環境において、実際のアプリケーションを処理する際に、ハイブリッドクラウド内のリソースを効率良く使用すると同時に、生じるコストのバランスも考慮しながら、最適なジョブ配置を行えることを目標とする。そこで、適用実験において、実際のアプリケーションによるジョブ処理を本ミドルウェアを介して行い、上記目標に対して、本提案手法を適用することが有用であるか評価を行った。

6.1 実験概要

本適用実験では、東京都内のある企業が、自社にプライベートクラウド環境と、大量のデータを格納する大規模なDBを構築したストレージプールを保持し、また、都内近郊にデータセンタを持つパブリッククラウドプロバイダから、必要なリソースを利用できる環境が整っていることを想定する。この環境の実現のため、本実験環境においては、プライベートクラウドとパブリッククラウドの間に、往復遅延時間 4msec の人工的な遅延を *DummyNet* を使って加えた。

また、本適用実験では、ハイブリッドクラウド環境で、ストレージプールにある大規模なDBに対し、アドホックなクエリ処理を、断続的に行う場合を想定する。この想定を実現するために、本適用実験では、DBT-3の検索クエリを用いた。DBT-3とは、TPC-Hの簡易版で、大規模なDBと、それに対する22の複雑でアドホックな検索クエリが提供されている。本適用実験では、ストレージプール内に、DBT-3を用いて大規模なDBを作成後、それに対し検索クエリを断続的に処理していく際にミドルウェアを適用する。今回の適用実験においては、一定時間間隔でジョブが投入されるようなタイプの実行形式を用いている。この一定間隔で割合小粒度なジョブが投入されていく状況は、特定の環境における評価である。異なるタイプのアプリケーション実行形式の場合、ジョブの分割や投入といった機能を別途用意する必要があるが、それら機能さえ実装すれば、本提案をそのまま利用することが出来る。

6.2 実験手順

本適用実験では、まず、ミドルウェア適用環境において、学習フェーズを実行する。これによって、「ディスクリソースを使い切った状態の条件」を定義する。次に、前述したアプリケーションに対し、開発したミドルウェアを介して、処理を実行する。このとき、本提案手法である、学習フェーズで得られた「ディスクリソースを使い切った状態の条件」をミドルウェアに負荷分散の閾値として設定した場合と、学習フェーズで得られる値より、小さな値や大きな値に設定した場合の、様々な閾値設定での実験も行う。また、それぞれの閾値でアプリケーション処理時に、処理したジョブの処理時間と、パブリッククラウドの従量制料金、プライベートクラウドの消費電力料金を測定する。これら測定コストを評価することで、本提案手法の有用性を示していく。

6.3 コスト評価概要

本適用実験のコスト評価では、ミドルウェアに提案手法である「ディスクリソースを使い切った状態の条件」という適切な閾値を設定した場合と、それ以外の値を設定した場合で、生じるコストのバランスにどのように違いがあるかについて評価する。ここで考えるコストとしては、ジョブの処理時間という時間的コストと、金銭的コストの2種類である。金銭的コストは、以下の式で算出したものである。

金銭的コスト： $T_R * N_R * C_R + P_L * C_L$

T_R :パブリッククラウド上のインスタンスを使った時間 [時間]

N_R :パブリッククラウドのインスタンスを使った数 [数]

C_R :パブリッククラウドの従量制料金単価 [\$/時間]

P_L :処理時にプライベートクラウドが消費した電力 [kWh]

C_L :消費電力料金単価 [\$/kWh]

それぞれの単価として、従量制料金単価は Amazon EC2 の値段を参考に処理時間あたり \$0.5、消費電力料金単価は、東京電力の値段を参考に 1kWh あたり \$0.24 とした。

6.4 適用実験：学習フェーズの実行

まずミドルウェア実行前に学習フェーズの実行を行う。本実験環境では、インスタンスからストレージプール内のストレージへ iSCSI アクセスを行った場合のディスク処理性能を測定することとなる。学習フェーズ実行手法は、4.3 節で示した通り、自作ベンチマーク *Disk Bench* を使用して、ディスクに負荷をかけ、そのときの処理時間と、ディスクに溜まるキューの長さを測定し、「ディスクリソースを使い切った状態の条件」を定義していく。また、学習フェーズでは、*Disk bench* 内のパラメータを変化させ、様々なアクセスパターンで測定を繰り返すが、本稿では、その代表例として、10Kbyte の読み出しを 512 回繰り返した場合の、逐次処理と同時処理の処理時間の比較結果を、図 4、5 に示す。

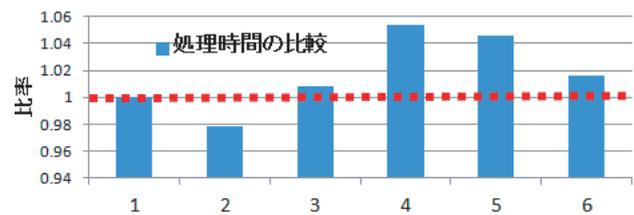


図4 処理時間の比較 (一例)

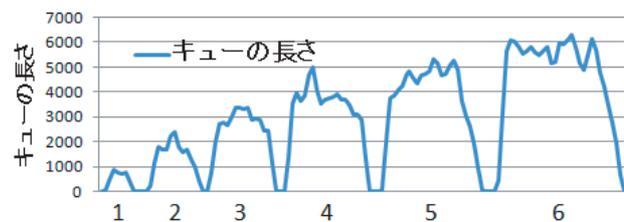


図5 キューの長さ (一例)

図4は、ジョブの同時処理と逐次処理時の処理時間の比較結果である。縦軸を処理時間の比率で表しており、横軸のジョブ数分のジョブを同時処理した時の処理時間を、逐次処理した場

合の処理時間で割ったものである。縦軸が1となる所に、赤の破線をひいており、それより下だと同時処理の方が速く、上だと逐次処理の方が速い、つまりディスクリソースを使い切った状態であるとうことを示す。つまり、本例では、同時処理数3で、ディスクの処理的な飽和状態となっていることがわかる。また、図5は、同時処理ジョブ数ごとのキューの長さの推移を示している。横軸が、同時処理数、縦軸がキューの長さである。このグラフから、同時処理数と、ディスクリソースに溜まるキューの長さは比例関係にあることがわかる。

これらの考察を様々なアクセスパターンで行い、処理時間によるディスク処理的な飽和の判断から、そのときのディスクに溜まるキューの長さを分析した結果、「ディスクリソースを使い切った状態の条件」としてのディスクに溜まるキューの長さを、2200~3200と定義できることがわかった。これらの値は、小粒度のアクセスのみを行い、物理ディスクの細かいレベルまで解析すれば、一意に定まる可能性があるが、実際のジョブはそれほど小さくないため、本手法では、この値の範囲を、「ディスクリソースを使い切った状態の条件」と定義する。

6.5 適用実験：評価結果

前節の学習フェーズにより決定した、「ディスクリソースを使い切った状態の条件」を元に、ミドルウェアに対して、上記の範囲内の値を含む、様々な値を負荷分散の閾値として設定した上で、ジョブ処理を実行し、その時に生じたコストを測定した。

図6に、本適用実験での、様々な閾値設定における、生じたコストの評価結果を示す。

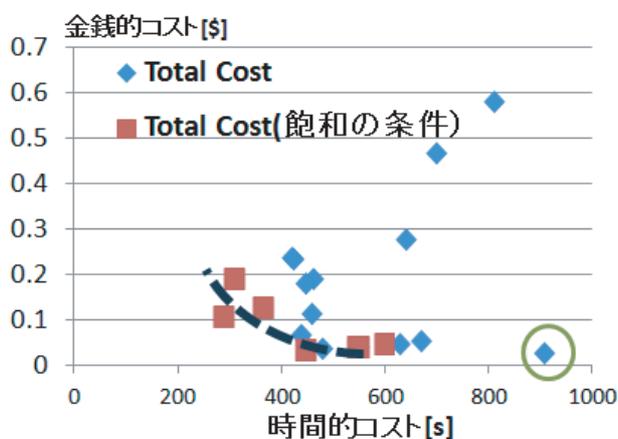


図6 想定環境における DBT-3 の検索クエリ処理を適用時のコスト評価

図6においても、縦軸は金銭的コスト、横軸は時間的コストを示している。また、図では、閾値として、学習フェーズで得た飽和の条件の範囲内の値を設定したものを赤のプロットで示す。

図6が示す通り、本提案手法である、学習フェーズにより決定した「ディスクリソースを使い切った状態の条件」となるキューの長さ、つまり 2200~3200 の値に設定することで、リソースを効率的に使いながら、コスト評価においてパレート最適の点となっていることがわかる。逆に、性能測定で決定した値の範囲内になくても、パレート最適曲線上の点となっている

こともあるが、これは、処理性能を重視するために、リソースを非常に多く使っているか、金銭的コストをかけないようにリソースに多大な負担を与えている負荷分散のコスト評価が、曲線上の点となった例である。パレート最適曲線上にのっていない点については、その閾値に比べ、より良いコストバランスを提供できる他の閾値が存在する点である。

また、緑の丸で囲まれた点は、プライベートクラウドのリソースのみでジョブ処理を行った場合のコスト評価結果である。このように、遅延が生じる環境においても、プライベートクラウドのリソースのみを使って処理する場合に比べて、本提案手法を適用した方が、効率の良い処理が提供されていることが確認できる。

6.6 適用実験：結論

本適用実験における評価によって、本提案手法である、学習フェーズにおいて決定した「ディスクリソースを使い切った状態の条件」をミドルウェアに設定することにより、ジョブのパレート最適配置が提供出来ることがわかった。つまり、「ディスクリソースを使い切った状態の条件」をミドルウェアに設定することは、パレート最適配置を提供するための十分条件であり、本提案手法により、リソースを無駄なく利用しながら、最適なジョブ配置を提供できることが示せた。

また、今回の金銭的コスト計算においては、パブリッククラウドの従量制料金の比重が大きいため、プライベートクラウドの消費電力料金の違いによる影響はかなり小さかった。しかし、この金銭的コストは、技術的・社会的な理由などにより、大幅に変わる可能性がある。消費電力料金の比重が大きくなったならば、本提案方式においても、より電力料金を念頭に置いた負荷分散が実行されるようになると考えられるが、これはコスト計算の式を変更するだけで対応が可能である。

7. 関連研究

クラウドにおけるロードバランスに関する論文として、例えば文献[5]などがある。しかしこれらは、CPU インテンシブアプリケーションを対象としており、データインテンシブアプリケーションは考慮していない。

また、本研究と同じように、ジョブの対象をデータインテンシブアプリケーションとし、その負荷の指標としてディスクアクセス量を用いて開発された負荷分散ミドルウェアに文献[6]がある。このミドルウェアでは、ディスクアクセス量を元に、ローカルのクラスターとパブリッククラウド間で動的な負荷分散を行うことで、ジョブの最適配置を行った。本研究ではこのミドルウェアを更に発展させ、コスト評価軸を設け、リソースを使い切りつつも、ユーザの要望に応じたコストバランスとなるジョブ配置を提供する。

クラウドの省電力化の検討も活発に行われている。文献[8]などは、CPU インテンシブアプリケーションをクラウド上で実行するときの省電力化への取組みである。また、文献[9]は、クラウドのデータセンターに対する省電力化への取組みであり、プライベートクラウドを含むクラウド全体の省電力化を目指す点で、本研究と異なる。

8. まとめと今後の課題

企業の持つセキュリティポリシーに従ったデータ配置のもと、ハイブリッドクラウド環境での効率的なビッグデータ処理に関して、どちらのクラウドのどれくらい使えば良いのかを、性能と必要なコストに基づき判断して、最適なジョブ配置を行える手法を提案し、ミドルウェアとして実装した。

本ミドルウェアでは、CPU処理とディスク処理から、それぞれの負荷の判断を行い、リソースを使い切った状態になった時に、負荷分散を行うことで、効率的な負荷分散を提供する。本手法では、CPU処理については、CPU使用率、ディスク処理においては、I/Oの混雑度を示す指標であるディスクに溜まるキューの長さから負荷を判断する。また、ディスク処理においては、事前に性能測定を行い、リソースを使い切った状態となる条件の範囲を決定した上で、負荷分散を行う閾値を、ユーザがその範囲内の値に設定することができる。これによって、生じるコストのバランスを、ユーザの要望に合わせるように負荷分散を実行することが出来る。

提案した手法を、実アプリケーションを用いてビッグデータ処理を模した環境へ適用した結果、データを外部サイトに保存せず、ローカルのストレージへiSCSIアクセスを行う状況においても、パレート最適なコストバランスとなるジョブ配置が提供できていることが確認され、本手法の有用性が示された。

今後の課題としては、様々なアプリケーションを本ミドルウェアを介して行なって行きたい。具体的には、データマイニングの代表的な手法である、FP-growthを使ったアプリケーション処理等に対し、本提案手法が有用かを評価したい。また、現在の実装では、学習フェーズで取得した閾値を、ミドルウェアに適用してから実行する形となっているが、この学習フェーズもミドルウェアの一部に取り込み、自動化を実現したい。

謝 辞

本研究は一部、文部科学省科学研究費基盤研究「電力消費を制御するスケーラブルな情報の蓄積と検索」によるものである。また、本論文作成にあたり、独立行政法人産業技術総合研究所竹房あつ子氏、中田秀基氏、高野了成氏、工藤知宏氏、東京大学生産技術研究所中野美由紀特任准教授、西川記史氏、横山大作助教、工学院大学山口実靖准教授には、大変有用なアドバイスをいただきました。深く感謝いたします。

文 献

- [1] 平成24年度情報通信白書:
<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/pdf/24honpen.pdf>
- [2] Eucalyptus:<http://www.eucalyptus.com/>
- [3] SHW3A:
<http://www.system-artware.co.jp/shw3a.html>
- [4] Gueyoung Jung, Kaustubh R. Joshi, Matti A. Hiltunen, Richard D. Schlichting and Calton Pu, "Generating Adaptation policies for Multi-Tier Applications in Consolidated Server Environments", In Proc. 5th IEEE International Conference on Autonomic Computing (ICAC2008), p23-32, June, 2008.
- [5] Evangelia Kalyvianaki, Themistoklis Charalambous and Steven Hand, "Self-Adaptive and Self-Configured CPU Resource Provision-

- ing for Virtualized Servers Using Kalman Filters", In Proc. 6th International Conference on Autonomic Computing and Communications (ICAC2009)", June, 2009
- [6] 豊島詩織, 山口実靖, 小口正人: "データインテンシブアプリケーション実行時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", 日本データベース学会論文誌, Vol.10, No.1, pp.31-36, 2011年6月
- [7] Kazuki Mochizuki and Shin-ichi Kuribayashi, "Evaluation of Optimal Resource Allocation Method for Cloud Computing Environments with Limited Electric Power Capacity," In Proc. 2011 14th International Conference on Network-Based Information Systems, pp.1-5, 2011
- [8] Che-Yuan Tu, Wen-Chieh Kuo, Wei-Hua Teng, Yao-Tsung Wang, Steven Shiau, "A Power-Aware Cloud Architecture with Smart Metering", In Proc. Parallel Processing Workshops (ICPPW), 2010 39th International Conference, pp.497-503, Sep, 2010
- [9] Cathryn Peoples, Gerard Parr and Sally McClean, "Energy-aware data centre management", In Proc. Communications (NCC), 2011 National Conference, pp.1-5, Jan, 2011
- [10] Gueyoung Jung, Matti A. Hiltunen, Kaustubh R. Joshi, Richard D. Schlichting, Calton Pu "Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures ", In Proc. Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference, pp.62-73, June, 2010
- [11] 笠江優美子, 小口正人: 「ハイブリッドクラウド環境における性能と実行コストに基づいたデータ処理最適配置ミドルウェアの提案と実装」, 電子情報通信学会データ工学研究会, DE2012-39, 2012年12月
- [12] Luna Mingyi Zhang, Keqin Li, Yan-Qing Zhang, "Green Task Scheduling Algorithms with Speeds Optimization on Heterogeneous Cloud Servers ", In Proc. Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on and Int'l Conference on Cyber, Physical and Social Computing (CPSCom), pp.76-80, Dec, 2010
- [13] Yumiko Kasae and Masato Oguchi "Proposal for an Optimal Job Allocation Method for Data-intensive Applications Based on Multiple Costs Balancing in a Hybrid Cloud Environment" In Proc. the 7th ACM International Conference on Ubiquitous Information Management and Communication (ICUIMC2013), 9-3, Kota Kinabalu, Malaysia, January 2013.