

# ハイブリッドクラウド環境における 各種コストバランスに基づくデータ処理最適配置ミドルウェア

笠江 優美子<sup>†1</sup> 小口 正人<sup>†1</sup>

高度 IT 社会の進展に伴い、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おり、大量のデータを効率よく処理するシステムが求められる一方で、処理時に生じるコストのバランスに対して柔軟な対応のできる手段も望まれている。本研究では、そのプラットフォームとしてハイブリッドクラウドに注目し、データインテンシブアプリケーションによるジョブを効率よく処理すると共に、処理時に生じるパブリッククラウドの従量制料金やクラウドの消費電力料金に対しても、ユーザの要望するコストバランスになるように、ジョブを最適配置するミドルウェアの構築を目指す。本稿においては、ハイブリッドクラウド上で処理されるデータインテンシブアプリケーションとして、異なる特徴を持つ 3 種類のアプリケーションを実行し、本ミドルウェアで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ることを示した。

## A Proposal of Optimal Job Allocation Middleware for Data-processing based on Multiple Cost Balancing on Hybrid Cloud Environment

YUMIKO KASAE<sup>†1</sup> and MASATO OGUCHI<sup>†1</sup>

In recent years, by explosive increase of amount of informations on computer systems, we need the system that can process a large amount of data efficiently. And also, we have hope the system that can take into account flexibly for cost balance. In this research, we focus on hybrid cloud environment, and we proposed method for processing jobs of data-intensive applications efficiently and responding flexibly to cost balance. And, we have developed it as middleware. In this paper, we have shown usefulness of this middleware by executing it on three applications with different characteristics.

### 1. はじめに

近年、高度 IT 社会の進展に伴い、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、大量のデータを効率よく処理するシステムが求められている。そのシステムを実現する有用な手段として、クラウドが期待されている。クラウドには、プロバイダに従量制料金を支払うことで、スケーラブルなリソースが提供されるパブリッククラウドと、ユーザが自身の持つリソースを使ってクラウド環境を構築することでセキュリティ等に配慮できるプライベートクラウドがあるが、パブリッククラウドにはセキュリティやコスト、プライベートクラウドにはスケーラビリティなどの問題点がある。そこで注目されているのが、この 2 つを併用するハイブリッドクラウドである。ハイブリッドクラウドでは、自前のシステムをプライベートクラウドとすることで、セキュリティに配慮しつつ効率的に運用し、リソースが足りなくなったらスケーラブルにパブリッククラウドからリソースを借りることが出来る。

また、近年のエコ志向や節電、節約の必要に伴って、ユーザがクラウド環境における処理に求めるものが変化しつつある。つまり、単にジョブの処理速度だけでなく、パブリッククラウドの従量制料金や、クラウドの消費する電力などとのコストバランスへの柔軟さも求められている。

上記の背景を踏まえ、本研究では、ハイブリッドクラウド環境において、大量のデータを効率よく処理しながら、生じるコストに対し、ユーザの要望するコストバランスとなるようにジョブを最適配置するミドルウェアの構築を目指す。

本ミドルウェアの特徴は 2 つある。1 つは、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ている点である。処理すべき情報量が増えると、単なる計算処理だけでなく、主としてディスクに負荷のかかる処理が重要になる。本ミドルウェアでは、データインテンシブアプリケーション実行時の負荷の状況をディスクアクセス量を用いて判断し、ジョブを最適配置していく。もう 1 つの特徴は、前述したとおり、ユーザの要望するコストバランスとなるジョブ配置を提供することである。本ミドルウェアでは、処理時間という時間的コストと、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する電力料金、およびプライベートクラウドの管理費用を足した金銭的成本を考え、この 2 つの指標に対し、ユーザの要望に応じてパレート最適にジョブを配置する。

---

<sup>†1</sup> お茶の水女子大学  
Ochanomizu University

本稿では、提案するミドルウェアを、異なる特徴を持つ3種類のデータインテンシブアプリケーションを用いて実行し、パレート最適なジョブ配置が提供出来ることを示す。

## 2. 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<sup>1)</sup>や文献<sup>2)</sup>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なジョブは考慮していない。本研究のようにデータインテンシブなジョブが対象の場合、CPUがI/O待ちとなっていることが多く、CPU利用率から適切な判断が難しいため、本ミドルウェアでは、負荷の指標としてディスクアクセス量を用いる。

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<sup>3)</sup>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、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。

さらに、クラウドの省電力化の検討も活発に行われている。文献<sup>4)</sup>と文献<sup>5)</sup>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り組みである。また、文献<sup>6)</sup>は、クラウドのデータセンタに対する省電力化への取り組みであり、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。

## 3. データ処理パレート最適配置ミドルウェア

### 3.1 ミドルウェア概要

本ミドルウェアでは、主にディスクに負荷のかかるデータインテンシブなジョブに対して、ハイブリッドクラウド環境でそのデータを連続的に処理する際に、生じるコストのバランスを考慮しながらジョブを最適配置していく。その時に、ミドルウェアでは、ハイブリッドクラウド環境で起動するリソースの`/proc/diskstats` ファイルをモニタリングすることで、実行されているジョブ量を推定し、現在の負荷状況を判断する。

また、本ミドルウェアでは、処理時間という時間的コストと、パブリッククラウドの従量制料金、クラウドの消費電力料金、プライベートクラウドの管理費用の合計である金銭的コストと言う2種類のコストに対し、ユーザによってパラメータが設定されることで、そのコストバランスを指定できる。ミドルウェアでは、ユーザによって指定されたコストバランス

となるように、まずはプライベートクラウド内のインスタンス(仮想マシン)へジョブを配置していき、必要に応じ、ユーザの許すコスト内で、パブリッククラウド内のインスタンスにもジョブを配置していく。ハイブリッドクラウド内のインスタンスにおいても、優先順位をつけ、高い順にジョブを配置していくことで、効率良くデータ処理を行なっていくことを目指す。図1に、ミドルウェアの構造と動作を示す。

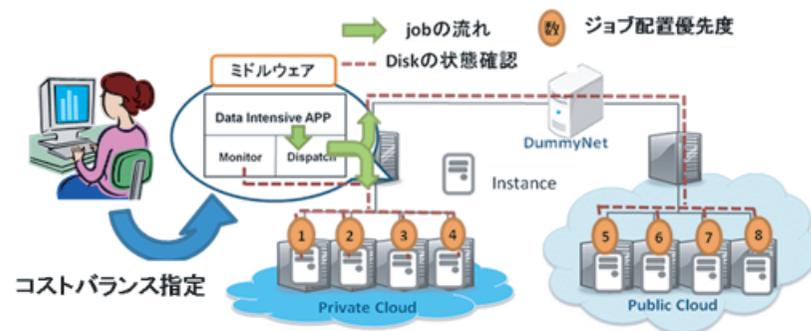


図1 ミドルウェアの構造と動作

本ミドルウェアはプライベートクラウドのFrontend上で動作しているシェルスクリプトと、C言語のプログラムで、図1に示す様に、ディスク状況を測定するMonitor部と、ジョブを振り分けるDispatch部から成る。Monitor部では、ハイブリッドクラウド内のリソースに対して、`/proc/diskstats`の内容を定期的に収集している。一方Dispatch部では、Monitor部で収集した情報を元に負荷状況の判断を行い、ジョブを振り分けていく。

次の節から、本ミドルウェアの詳細について述べる。

### 3.2 ディスク状況による飽和判断手法

本ミドルウェアは、主にディスクに負荷のかかるデータインテンシブなジョブを対象とするため、リソースのディスクアクセス量を用いて負荷状況を判断する。そのディスクアクセス量は、リソースの`/proc/diskstats`ファイルの内容を用いる。`/proc/diskstats`ファイルは、ディスクデバイスの統計情報を持つファイルである。`/proc/diskstats`ファイルの内容を元に、ディスクの負荷状況を判断するには、まず、ミドルウェアで実行するジョブの一部を用いて、ディスクの飽和状態におけるファイル内の値を確認する必要がある。

そこで、本手法では、本ミドルウェア実行前に、ディスク状況による飽和判断基準を知る

ための測定を行う。この測定では、ジョブの同時投入数を増加させた場合の/proc/diskstats ファイルの内容を測定する。同時投入することで、ジョブは並列に処理されていくため、ディスクの状態に余裕のある時は、それらのジョブを逐次的に処理させた場合より処理時間が速いはずである。しかし、同時投入数を増やしていくことで、ディスクがある時点で飽和状態となり、そのジョブを逐次的に処理させるより遅くなってしまふ点がある。これをディスクの飽和状態として、その時の/proc/diskstats ファイルの値を”飽和の基準閾値”と名付ける。本飽和判断手法では、/proc/diskstats ファイルの中でも、Disk Read とキューの長さに注目する。これは、本手法を提案する上で評価と検討を重ねた結果、/proc/diskstats に書かれるすべての統計情報の中で、この 2 つの指標のみでも正確に判断が行えたためである。この飽和判断手法については、具体的な例と共に後述する。

### 3.3 ミドルウェアの考慮するコスト

本ミドルウェアが考慮するコストとしては、ジョブの実行時間という時間的コストと、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量料金および管理費用を合わせた金銭的コストを考える。ここでは、パブリッククラウドの消費電力料金は、従量制料金に消費電力料金が含まれると考える。

この 2 種類のコストに対して、近年の節電や節約の志向により、ユーザが考える好ましいコストバランスは、状況によって異なる。金銭的コストが多少嵩んでも良いから処理時間を早くしたい場合や、処理時間を遅くしても良いから金銭的コストは抑えたい場合、もしくは 2 つのコストをバランス良く取りたい場合などがある。本ミドルウェアでは、”負荷分散の閾値”というパラメータをユーザが設定することで、希望するコストバランスに応じて、パレート最適な負荷分散を実現する。

### 3.4 本ミドルウェアを用いた負荷分散制御方法

ユーザがコストバランスを指定する際、本ミドルウェアでは、前述したように、”負荷分散の閾値”というパラメータを設定する。この負荷分散の閾値とは、ユーザがどれくらいの負荷状況となるまで、一つのインスタンス内でジョブを実行させるかに相当する。そのため、事前の飽和測定で得られた飽和の基準閾値である 2 つの値を元に、負荷分散の閾値をこれより下の値に設定すれば、ジョブはディスク飽和前に負荷分散され、多くのインスタンスを用いることとなる。それぞれのインスタンスの負荷は軽いため、高速な処理が期待できる。反対に、これより上の値に設定すれば、ディスクは飽和状態となっても、その値に達するまで処理はそのインスタンス内で実行されることで、パブリッククラウドの従量制料金が抑えることができる。ユーザは自身の希望する負荷分散状況に応じた閾値に設定することで、負

荷分散を制御できる。また、指定された負荷分散の閾値を元に、まずはプライベートクラウドとパブリッククラウドのインスタンスへ、優先順位の高い順に、ジョブを配置していく。

## 4. ミドルウェア実行実験概要

### 4.1 実験概要

本ミドルウェアの実験環境としては、図 1 に示すように、パブリッククラウドとプライベートクラウドとなる 2 つのクラウドをクラウド構築ソフトウェア Eucalyptus<sup>7)</sup> を用いて構築し、その間を人工遅延発生装置 Dummynet で繋ぐことで、ハイブリッドクラウド環境を再現した。それぞれのクラウドでは、Frontend サーバを 1 台、Node サーバを 4 台とし、すべての Node サーバに表 1 の性能のインスタンスを 1 つずつ生成した。つまり、プライベートとパブリックで合計 8 つのインスタンスに対し、ミドルウェアを実行する。実験に用いたサーバは、シングルコア CPU であるため、各サーバに 1 インスタンスずつの配置としている。パブリッククラウドでは、インスタンス数は制限が無く利用できることが一般的であると考えられるが、本実験で投入したジョブの総量は、プライベートクラウドに加えて、パブリッククラウド 4 つ以内で十分に負荷分散可能なものであった。

実験では、前章で述べた負荷分散の閾値を変化させる。つまり、事前の飽和測定から得られた飽和の基準閾値を元に、負荷分散の閾値を増減させた場合の、処理時間、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を測定する。ここで、電力測定においては、システムアウェア製のワットアワーメーター SHW3A<sup>11)</sup> を使い、プライベートクラウドを構成する Node サーバ全体の消費電力を測定する。

本実験では、様々な負荷分散状況の上記 3 つのコストを測定、評価し、本ミドルウェアがパレート最適な負荷分散を実現できているか評価を行う。

表 1 Instance

OS	Linux 2.6.27.21-0.1-xen / x86_64 GNU / CentOS 5.3
CPU	Intel(R) Xeon(R) CPU @ 3.60GHz 1 core
Memory	1024MByte
Disk	10GByte or 20GByte

### 4.2 実験におけるデータ配置

クラウド環境において、特にデータインテンシブなジョブを負荷分散させる際には、その

データ配置を考えることは大変重要である。本ミドルウェア実行環境として想定する、ハイブリッドクラウド環境でのデータ配置として、クラウドに付随するブロックストレージを使う場合や、ローカルにストレージを設けてパブリッククラウドからは遠隔アクセスを行う場合などが考えられるが、本実験では、遠隔バックアップなどにより、インスタンス内に既にデータが配置されている場合を想定し、実験を行っている。つまり、本ミドルウェアでは、すべてのインスタンスのローカルに必要なデータが揃っており、その処理をどのインスタンスで行うのかを本ミドルウェアに判断させる。文献<sup>3)</sup>では、パブリッククラウドからローカルのストレージに iSCSI を用い遠隔アクセスした場合の取組みも行われている。今後上記データ配置についても取り組んでいきたい。

### 4.3 実行するジョブ

本論文では、本ミドルウェアの有用性を示すため、異なる3つの特性を持つデータインテンシブなジョブのワークロードでそれぞれ実験を行った。

まず1つは、自作したベンチマークツールを用いたワークロードである。このベンチマークでは、ディスクヘラダムに read アクセスのみを行う。2つ目は、汎用データベースベンチマークとして配布されている pgbench を用いたワークロードである<sup>8)</sup>。pgbench は PostgreSQL に同梱されているシンプルなベンチマークツールであり、オンライントランザクション処理を模した TPC-B を基に作成され、1秒間に実行できるトランザクション数で性能を判断できる。自作ベンチマークツールと異なり、データベースに対し read 処理や write 処理を様々に行う。この2種類のベンチマークツールを用いた実験では、2秒間で200回のジョブ実行命令に対し、ミドルウェアに配置判断させる。

しかし、上記2種類の1つ1つのジョブは、人工的に作られたジョブであり、実環境では想定しづらい。そこで、3つ目のワークロードとして、意思決定支援システム環境におけるベンチマークツール TPC-H の簡略版である、OSDL-DBT-3 の検索クエリを用いた。意思決定支援システムでは、キャッシュが効果的に使えないほど大規模なデータベースにおいて、インデックスが効果的に使えないくらいに複雑なクエリの実行が要求される。そのため、DBT-3 では、インデックスが使えない全行検索となるような複雑な SELECT 文で構成された、22の番号が付けられた複雑な検索クエリをラン順に実行する。

図2に、DBT-3の検索クエリを、クエリラン順に実行した時の処理時間を示す。このように、DBT-3の検索クエリでは、クエリによって処理時間がかなり異なることがわかる。本実験では、DBT-3実行時に構築される大規模なデータベースと、DBT-3の22個の複雑なクエリの内、処理時間が50秒以下の11個を利用し、ミドルウェアを実行する。また、そ

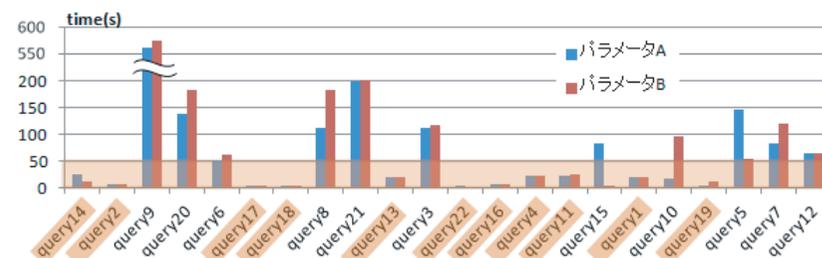


図2 DBT-3で実行される検索クエリの処理時間

の11個のクエリに対して、SQL内の検索パラメータを2通り(パラメータA, パラメータB)用意することで、計22個のSQLを用意した。実験では、2秒間でクエリを投入していき、22個の順次クエリ実行を1タームとした時の、5ターム分のクエリ実行を、ミドルウェアに配置判断させる。

### 4.4 評価方法

本ミドルウェアの評価は、前述した通り、時間的コストと金銭的コストの2つの指標で行う。トータルコストは、以下の式で規定する。

#### Total Cost

時間的コスト:  $T_{exec}$

金銭的コスト:  $T_R * N_R * C_R + P_L * C_L + M_L$

$T_{exec}$ : ジョブ全体の実行時間

$T_R$ : パブリッククラウドでのジョブの実行時間

$N_R$ : パブリッククラウドの使ったインスタンス数

$C_R$ : パブリッククラウドの従量制料金

$P_L$ : プライベートクラウドの消費電力量

$C_L$ : 消費電力料金単価

$M_L$ : プライベートクラウドの管理費用

本実験での金銭的コストを考える場合、従量制料金単価と消費電力料金単価の価格設定による議論を行う必要がある。すなわち、この2つの金額は常に一定ではないことを考慮する。最近ではクラウドを提供するプロバイダが数多く存在しているが、その従量制料金は必ずしも一定ではなく、今後このようなプロバイダが更に増えれば、価格競争により下がる可能性もあり、何らかの事情によって現在より高くなる可能性もある。これは、消費電力料金においても言えることである。そのため、本評価では、パブリッククラウドにおいては AmazonEC2 の価格設定を参考に \$0.5/hour とし、消費電力料金としては東京電力の価格

設定を参考にして\$0.24/kwh とした上で、金銭的成本として足し合わせる際に、比率を従量料金単価:消費電力単価=3:1, 1:1, 1:3 と変化させ、本ミドルウェアがパレート最適な負荷分散を実行可能であるか評価する。

## 5. 実験結果と評価

### 5.1 自作ベンチマークツールを用いた実験結果と pgbench を用いた実験結果

#### 5.1.1 自作ベンチマーク:飽和測定

ミドルウェア実行前に、まずミドルウェアで投入するジョブを用いたディスクの飽和測定を行い、飽和の基準閾値を決める。この測定では、自作ベンチマークの同時投入数を変化させた場合の処理時間と、/proc/diskstats ファイル内の Disk Read とキューの長さを測定し、この2つの指標の飽和の基準閾値を決定する。そのため、まず飽和測定の処理時間を見ることで飽和状態の同時投入数を確認する。図3に、自作ベンチマークの同時投入数を変化させた時の処理時間を示す。

横軸が同時投入数であり、同時投入数が増えるほど、インスタンスにかかる負荷が増えることを意味する。図3のように、同時投入数が増えると、飽和していない状態を表したbaselineに場合に比べ、処理時間が長くなっていくことが分かる。今回投入した自作ベンチマークの大きさでは、同時投入数3の場合でbaselineより処理時間がかかってしまっている。baselineとは、自作ベンチマークを横軸の個数分、逐次的に実行させた時の処理時間である。つまり、1つのジョブ終了後すぐに次のジョブを投入していく場合の処理時間を意味する。この飽和した同時投入数を”飽和同時投入数”とする。今回の飽和測定では3である。この値を元に、リソースの飽和状態を考えていく。

図5, 6に、飽和測定実験時の同時投入数ごとのDisk Read とキューの長さの推移結果を示す。この2つの図では、飽和同時投入数以降の、飽和と判断される推移結果は破線で示している。図5の、緑の横軸に平行な直線は2800byteを示す。図5では、同時投入数1の場合は低い値で推移しているが、同時投入数3以降になると2800byte以上を記録している。そこで、この値をDisk Readの飽和の基準閾値とする。また、より正確な判断を行うため、キューの長さも飽和の判断材料とする。そのキューの長さを示す図6では、同時投入数が増えるに連れ、確実に値が大きくなっていくことが読み取れる。この図から、飽和する同時投入数が3であった点を考えると、キューの長さの飽和の基準閾値を2000に決定できる。

ここで、本実験では、/proc/diststat内のその他のディスク情報による飽和判断も検討し

たが、上記2つの値に比べると正確な判断が行いづらく、逆に、この2つの値の情報のみで正確な判断が行えたため、本手法では、Disk Read とキューの長さに注目する。

以上の飽和測定結果により、自作ベンチマークによるDisk Read とキューの長さの飽和の基準閾値を、2800と2000に決定した。

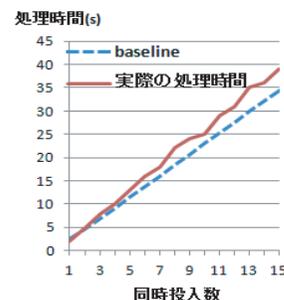


図3 自作ベンチマークの飽和測定:処理時間

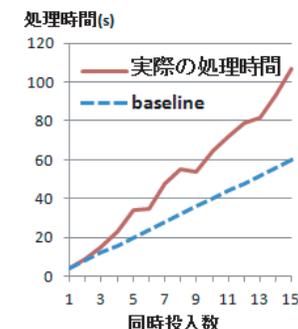


図4 pgbenchによるジョブの飽和測定:処理時間

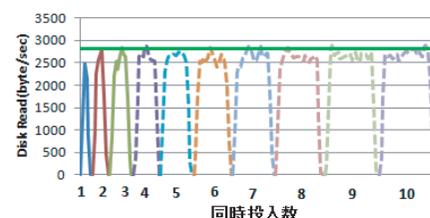


図5 自作ベンチマークの飽和測定:Disk Readの推移

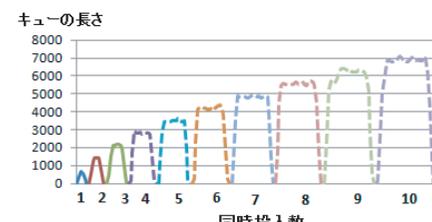


図6 自作ベンチマークの飽和測定:キューの長さの推移

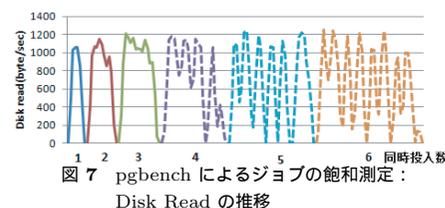


図7 pgbenchによるジョブの飽和測定:Disk Readの推移

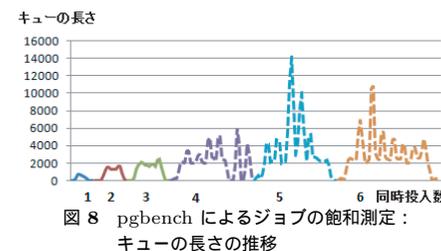


図8 pgbenchによるジョブの飽和測定:キューの長さの推移

#### 5.1.2 pgbench: 飽和測定

pgbenchを用いたジョブにおいても、事前の飽和測定により、飽和の基準閾値を決定する。また、このベンチマークは、Readアクセスのみ行う自作ベンチマークと異なり、様々

なディスクアクセスを行うため、指標の汎用性が求められる。そこで、同時投入数するジョブの大きさを変化させ、それぞれのジョブの大きさでの飽和測定を行った。pgbench ではクライアント数とトランザクション数が指定できるため、クライアント数を 1~2、トランザクション数を 100~300 に変化させて実験を行った。しかし、どれも同じ傾向が見られたため、本論文では、その代表例としてクライアント数:2、トランザクション数:300 の飽和測定の結果を示す。

まず、飽和測定の処理時間におけるディスクの飽和する同時投入数を決定する。図 4 に、飽和測定における処理時間を示す。この図が示すように、自作ベンチマーク同様の判断で、本実験環境において、ディスクの飽和同時投入数は 3 であることがわかる。

次に、ディスクの飽和状態における Disk Read とキューの長さを求める。図 7, 8 は、その推移の結果であり、飽和同時投入数以降の推移結果は破線で示している。この図より、飽和している状態であると、Disk Read の値は 1200byte を、キューの長さは 2000 を越していることが確認できる。これは、他のジョブの大きさの場合も同様の傾向にあった。それを示すため、表 2 に、それぞれのジョブの大きさの処理時間から得た飽和同時投入数、Disk Read が 1200byte を初めて上回った同時投入数、キューの長さが初めて 2000 を上回った同時投入数を示す。

表 2 pgbench:飽和の基準閾値の決定

クライアント数	トランザクション数	飽和する同時投入数	Disk Readの値が初めて1200を上回る投入数	キューの長さが初めて2000を上回る投入数
1	50	7	7	5
1	100	4	3	4
2	100	3	2	3
1	200	3	3	4
2	200	2	2	2
1	300	3	3	3

この表のように、Disk Read の値が初めて 1200byte を上回るジョブ数とキューの長さが初めて 2000 を上回るジョブ数が、飽和同時投入数と一致、もしくは近い値となっていることがわかる。このことから、この 2 つの値を飽和の基準閾値として決定した。

### 5.1.3 ミドルウェアの実行と評価

それぞれの飽和測定により決定した飽和の基準閾値を元に、ミドルウェアの実行実験を行った。この実験では、負荷分散の閾値として、飽和の基準閾値を元に、その前後の値を様々に指定して実行させた時の、ジョブの処理時間、パブリッククラウドの従量制料金、プ

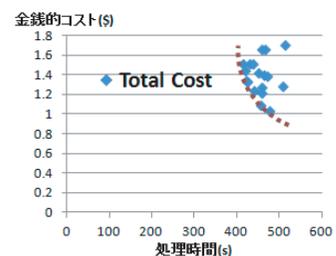


図 9 自作ベンチマーク Total Cost (従量制単価：電力単価=3 : 1)

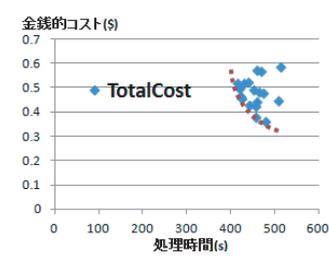


図 10 自作ベンチマーク Total Cost (従量制単価：電力単価=1 : 1)

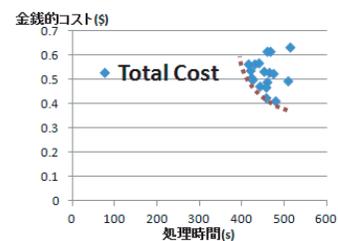


図 11 自作ベンチマーク Total Cost (readbench: 従量制単価：電力単価=1 : 3)

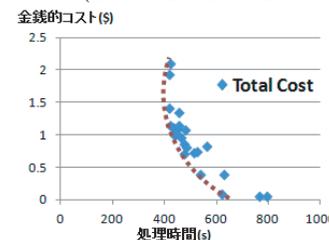


図 12 pgbench Total Cost (従量制単価：電力単価=3 : 1)

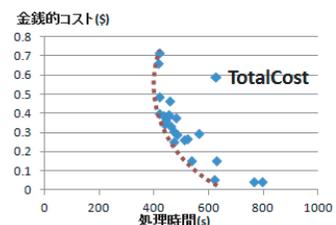


図 13 pgbench Total Cost (従量制単価：電力単価=1 : 1)

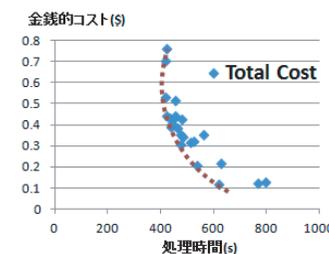


図 14 pgbench Total Cost (従量制単価：電力単価=1 : 3)

ライバートクラウドの消費電力を測定し、コスト評価を行う。図 9~11 に、自作ベンチマークを用いたミドルウェア実行によるコスト評価結果、図 12~14 に、pgbench を用いたミドルウェア実行によるコスト評価結果を示す。どちらのワークロードでも、金銭的コストの中の単価比率を 3:1, 1:1, 1:3 に変化させている。

これらの図からわかるように、本ミドルウェアは、どの価格設定においても、パレート最

適なジョブ配置を提供できることがわかる。つまり、示したパレート曲線上の点を指定するような負荷分散の閾値を指定すれば、それぞれの条件下でのパレート最適なジョブ配置を提供する。また、コストのスケールが変わりながらも、それぞれのワークロードで、ほぼ同じような分布となっているが、これは金銭的成本のうち、従量制単価の影響が電力単価に比べてかなり強いからだと考えられる。

それぞれのワークロードで、より詳しい分析結果を図 15, 16 に示す。

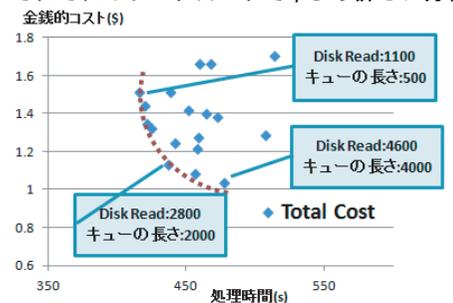


図 15 Total Cost の拡大図  
(自作ベンチマーク: 従量制単価:電力単価= 3:1)

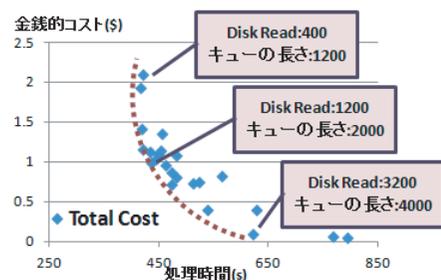


図 16 Total Cost の拡大図  
(pgbench: 従量制単価: 電力単価= 3:1)

図 15, 16 は、単価比率 3:1 のときのコスト評価の拡大図である。それぞれの図において、基準閾値は線上に載っており、この点がパレート最適の 1 つであることが分かる。また、飽和の基準閾値より下の値を指定すると、処理速度重視である負荷分散を提供し、逆に、上の値を指定すると、金銭的成本重視の負荷分散を提供することがわかる。

以上により、この 2 つのワークロードにおいて、本ミドルウェアによって、パレート最適なジョブ配置が提供可能であることが示された。

## 5.2 DBT-3 の検索クエリを用いた実験結果

上記 2 つのワークロードと同様の方法で、データインテンシブアプリケーションとして、DBT-3 で使われる検索クエリを用いたミドルウェア実行実験を行った。

### 5.2.1 DBT-3 のジョブ: 飽和測定

まず、飽和測定を上記 2 つのワークロードと同様の方法で行った。図 17 に、飽和測定における処理時間を示す。図 17 に示す通り、他のベンチマークと同様に、飽和測定における処理時間が baseline を越す点があり、その点をディスクが飽和した点として判断する。DBT-3 の検索クエリを用いたジョブでは、飽和同時投入数は 2 であることがわかる。この値を元

に、その時の Disk Read とキューの長さから、飽和の基準閾値を決定する。

図 18, 19 に、本飽和測定における Disk Read とキューの長さの同時投入数における遷移を示す。飽和状態である値の遷移は、どちらのグラフも破線とした。紫の直線は、図 18 では 1900byte、図 19 では 2400 を示している。図 18 において、飽和状態であると、その推移が、1900byte を下限としているということがわかる。同様に、図 19 においても、飽和状態であると、その推移が、2400 を下限としていることがわかる。これら 2 つのグラフから、飽和の基準閾値として、Disk Read は 1900、キューの長さは 2400 とした。

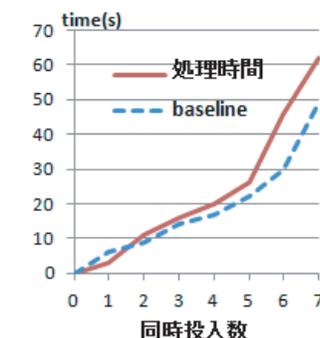


図 17 DBT-3 の検索クエリを用いたジョブの飽和測定:処理時間

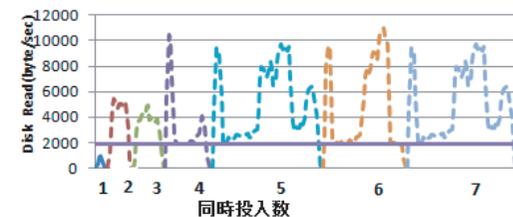


図 18 DBT-3 の検索クエリを用いたジョブの飽和測定: Disk Read の推移

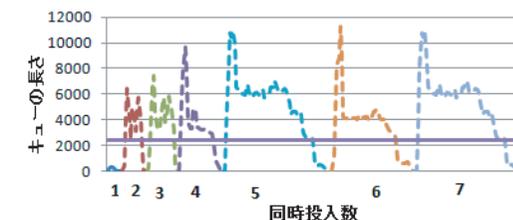


図 19 DBT-3 の検索クエリを用いたジョブの飽和測定: キューの長さの推移

### 5.2.2 ミドルウェアの実行と評価

決定した飽和の基準閾値を元に、負荷分散の閾値を様々に変化させてミドルウェアの実行を行った。ミドルウェア実行時においても、前記ワークロードと同様に、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を測定しており、得られた結果から、コスト評価する。また、この評価においても、金銭的成本の単価比率を 3 パターンに変化させている。図 5.2.2~22 に、その評価結果を示す。

図 20~22 に示すように、本ミドルウェアによって、DBT-3 の検索クエリを用いた、より複雑なクエリにおいても、パレート最適なジョブ配置を提供できることがわかる。また、図

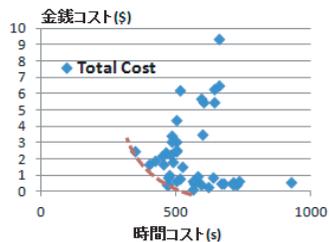


図 20 DBT-3 の検索クエリを用いたジョブ  
Total Cost(従量制単価：電力単価=3:1)

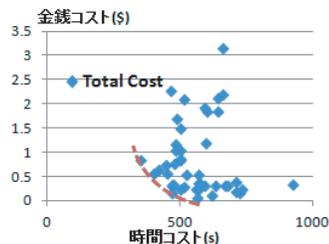


図 21 DBT-3 の検索クエリを用いたジョブ  
Total Cost(従量制単価：電力単価=1:1)

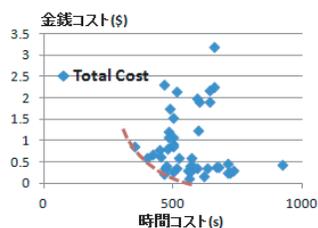


図 22 DBT-3 の検索クエリを用いたジョブ  
Total Cost(従量制単価：電力単価=1:3)

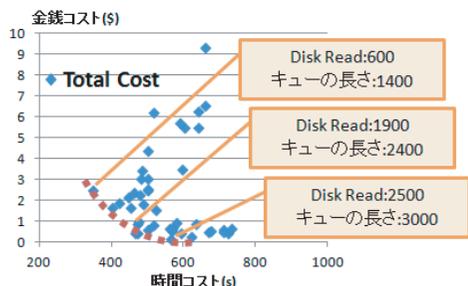


図 23 Total Cost の拡大図  
(DBT-3 によるジョブ: 従量制単価:電力単価= 3:1)

23 に示すように、より詳しい分析を行った結果においても、他のワークロードと同様に、飽和の基準閾値がパレート最適曲線上にあり、その値から前後の値を指定すれば、処理速度や金銭的コストのコストバランスに柔軟に対応できるジョブ配置を提供できることがわかる。

## 6. まとめと今後の課題

ハイブリッドクラウド環境において、異なる特性を持つ 3 種類のデータインテンシブなジョブに対して、それを連続的に処理する場合に、ディスクアクセス量を示すパラメータを用いて、パレート最適なジョブ配置を提供できるミドルウェアを提案し、評価した。本ミドルウェアによって、ユーザが考えるコストバランスをパレート最適にジョブ配置することが可能であることを示した。

しかし、現段階の評価では、様々なパラメータ設定で本ミドルウェアを実行させることにより、パレート最適なコストバランスとなるような負荷分散を提供できるということしか示せていない。今後としては、本ミドルウェアを実行することにより、パレート最適曲線上の

点となるコストバランスを提供するための取り組みを行なっていきたい。また、本実験では、アクセスを行うデータはすでに遠隔バックアップなどをされていると仮定したが、実環境においてそのような環境は想定しづらい。そのため、データウェアハウスのような、ローカルに持つストレージのデータに対し、遠隔アクセスを行う場合など様々な場合を想定していきたい。

## 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。

## 参 考 文 献

- 1) 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.
- 2) Evangelia Kalyvianaki, Themistoklis Charalambous and Steven 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) 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
- 4) 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
- 5) 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
- 6) Cathryn Peoples, Gerard Parr and Sally McClean, "Energy-aware data centre management", In Proc. Communications (NCC), 2011 National Conference, pp.1-5, Jan, 2011

- 7) Eucalyptus:<http://www.eucalyptus.com/>
  - 8) pgbench:<http://www.postgresql.org/docs/9.1/static/pgbench.html>
  - 9) Xen:<http://www.xen.org/>
  - 10) KVM:<http://www.linux-kvm.org/>
  - 11) SHW3A:  
<http://www.system-artware.co.jp/shw3a.html>
  - 12) 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
  - 13) 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
  - 14) 笠江 優美子, 豊島 詩織, 小口 正人: "Eucalyptus を用いたプライベートクラウドの様々な条件における消費電力量評価", マルチメディア, 分散, 協調とモバイル (DI-COMO2011) シンポジウム, 3H-1, pp.550-557, 天橋立宮津ロイヤルホテル, 2011年7月.
  - 15) 笠江 優美子, 小口 正人: "ハイブリッドクラウドの各種コストを考慮したデータ処理負荷分散ミドルウェアの提案", DEIM2012, C5-1, シーサイドホテル舞子ビラ神戸, 2012年3月.
-