

ハイブリッドクラウド環境における 省電力指向の負荷分散ミドルウェア開発に向けた一検討

笠江 優美子^{†1} 小 口 正 人^{†1}

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、そのデータを効率よく処理するためのシステムが求められている。それを実現する有用な手段としてクラウドコンピューティングがあげられ、すでに世界中で普及している。また、そのクラウドコンピューティングの枠組として、ハイブリッドクラウドが注目されている。さらに、近年は世界的なエコ志向であり、クラウドにおいてもその省電力化が求められている。そこで、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省電力を指向しながら負荷分散するミドルウェアの構築を目指す。本論文では、上記ミドルウェア実現をめざし、その前段階として、まずは大量のデータを効率よく処理するためのミドルウェアを構築し、その際の消費電力量の評価を行った。

Study for Development of Power-Saving-Oriented Middleware for Data Processing Load Distribution in Hybrid Cloud

YUMIKO KASAE^{†1} and MASATO OGUCHI^{†1}

In recent years, the amount of information in computer systems is increasing explosively. Therefore, the system for processing efficiently the data is required. Cloud computing is a useful means to realize the system, and has already spread all over the world. Moreover, hybrid Cloud attracts attention as a framework of the cloud computing. Furthermore, power-saving Cloud computing is required by eco-oriented society. So, this research aims at the middleware construction which carries out load distribution of a lot of data to a power-saving-oriented in hybrid Cloud environment. In this paper, We built the middleware for processing a lot of data efficiently, and evaluated the power consumption at that time.

1. はじめに

近年、インターネット回線の高速化や、仮想化技術、グリッドコンピューティング技術の進展によって、クラウドコンピューティングという新しいコンピュータの利用形態が生まれた。クラウドコンピューティングとは、ネットワークの向こう側からコンピュータシステムを使う上での様々な機能をサービスとして提供するというものである。ユーザとしては、最低限の接続環境のみでも利用ができ、デバイスを選ばないなど多くのメリットがある。さらに、現在、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、そのデータを効率よく処理するシステムとしてクラウドコンピューティングが期待されている。これらのことから、クラウドコンピューティングは社会に急速に普及している。

しかし、この普及に対し、世界的なエコ志向による IT 機器のグリーン化の考えが浸透することによって、クラウドを提供する側がかかえるサーバなどの IT 機器の消費電力の増加が問題となっている。その解決策として、電源配給装置や空調設備などのファシリティの改善や、IT 機器の消費電力量低減があるが、これらの新しい技術や機器を導入することはコストの観点からも必ずしも容易ではない。そこでソフトウェアおよびシステム構成の観点からの消費電力量の削減が望まれている。

そこで本研究では、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ハイブリッドクラウド環境において、大量のデータを省電力を指向しながら負荷分散するためのミドルウェアを開発している。

関連研究¹⁾では、ローカルのクラスタとパブリッククラウドを使い、大量のデータを効率良く処理するためのミドルウェアが開発された。このミドルウェアでは、データインテンシブなアプリケーションを対象としており、まずローカルクラスタでジョブを実行させ、定期的にリソースの使用状況を測定し、飽和状態であると判断したら、パブリッククラウドに負荷分散を行う。このミドルウェアを実環境に近い環境で実行させた結果、クラウドの従量制のコストと実行時間の両立を実現できた。本研究では、ローカルの環境においてはプライベートクラウドを用い、また上記ミドルウェアに負荷分散の指標として消費電力量を取り入れることで、ハイブリッドクラウド環境で省電力を指向しながら負荷分散を行うミドルウェア作成を目指す。本研究で消費電力量を指標として実現することで、近年の世界的なエコ志

^{†1} お茶の水女子大学
Ochanomizu University

向においても活用されるミドルウェアが作成可能であると考えられる。

2. 研究背景

2.1 Eucalyptus

本研究では、クラウド構築基盤ソフトウェア Eucalyptus²⁾ を用いて 2 つのクラウドを構築し、その間を人工的に遅延を発生させる dummynet で繋ぐことで高遅延環境を再現させ、ハイブリッドクラウド環境を構築した。Eucalyptus とは、カリフォルニア大学サンタバーバラ校の研究プロジェクトとして開発されたものであり、クラスタ化した複数のサーバのリソース管理機能を備えたオープンソースソフトウェアである。Amazon EC2/S3 と互換性があることも特徴である。

システム構成を図 1 に示す。

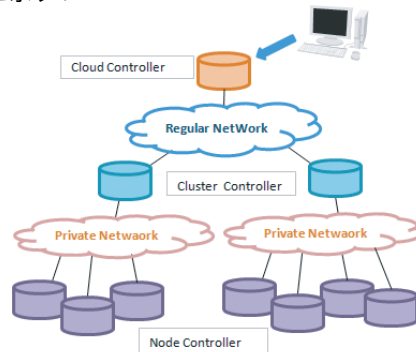


図 1 Eucalyptus のシステム構成

Eucalyptus は、Cloud controller(CLC)、Cluster controller(CC)、Node controller(NC) の 3 層構造である。CLC では EC2 互換のインターフェイスを備え、ユーザから指示された内容に基づいた各種の制御を行い、CC ではインスタンス間、インスタンスとクラウドの外部の間のネットワークの制御を行う。NC は実際にインスタンスを動作させる。インスタンスとは、クラウドから提供される仮想マシンである。また、CLC と CC を同一サーバ 1 台、NC サーバ 1 台の、最低 2 台のサーバでもクラウドを構築可能である。

2.2 Xen

Eucalyptus では仮想化ソフトウェアとして、Xen⁴⁾ と KVM⁵⁾ の 2 種類をサポートしている。先行研究³⁾ では、Eucalyptus で構築したクラウドにおいて、データインテンシブな

ジョブを実行させる場合、KVM より Xen の方が仮想化ソフトウェアの特性からも優れていることがわかった。そのため、仮想化ソフトウェアとしては Xen を採用し、実験システムを構築する。Xen は、図 2 に示すように、複数の OS を動かす為の基盤のみを提供することで仮想マシンのオーバーヘッドを抑え、物理マシンに近い性能が発揮されるよう工夫し、オープンソースながら非常に高性能で、現在 Amazon EC2 でも Xen を使ってインスタンスが提供されている。仮想マシンモニタが仮想化のための土台となり、その上でドメイン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 ハードウェア資源の管理や、その他のドメインを管理する特権を持つ。

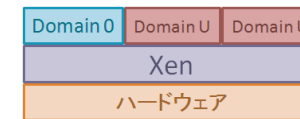


図 2 Xen のアーキテクチャ

3. ミドルウェア開発方針

本研究の目指すミドルウェアである、ハイブリッドクラウド環境にてデータインテンシブなジョブを省電力指向に負荷分散を行うミドルウェア開発の方針を以下に述べる。まず第 1 段階として、ハイブリッドクラウド環境でデータインテンシブなジョブを効率よく処理するためのミドルウェアを作成した。このミドルウェアでは、DiskI/O による判断により、インスタンスのリソースを使い切っていると判断したら、別のインスタンスへジョブを負荷分散する。そして、第 2 段階では、第 1 段階のミドルウェアに消費電力量を指標として取り入れ、本研究の目指すミドルウェアを構築する。

本論文では、第 2 段階の前段階として、第 1 段階のミドルウェアである、大量のデータを効率よく処理するためのミドルウェアを実行した際に、リソースの飽和判断の基準を変化させた場合の、ジョブ全体の実行時間、パブリッククラウドクラウドの従量制コスト、プライベートクラウド側が消費する電力の評価を示す。

4. 実験システム

図 3 に本研究で構築した実験システムを示す。ハイブリッドクラウド環境として、プライベートクラウドとして、CC と CLC が動作する Frontend サーバ 1 台とインスタンスが生成される Node サーバ 4 台、パブリッククラウドも同様に、Frontend サーバ 1 台と Node

サーバ 4 台とし、その間を人工的に遅延を発生させる dummynet で繋ぐ。それぞれの性能は表 1 から 4 に示す通りである。2 台の Frontend サーバはネットワークの口を 2 つ持ち、1 つが Node サーバへ、もう 1 つが外部のネットワークへ繋がっており、これによってそれぞれの Node サーバは外部のネットワークから独立している。

この環境で電力を測定するための測定機として、システムアートウェア製の高精度小型電力計ワットアワーメータ SHW3A⁶⁾ を用いた。これは、コンセントに接続したワットアワーメータに電気製品を繋ぐと、瞬時に消費電力を測定、表示するというものである。本研究では、電力の測定対象はプライベートクラウドのみとし、プライベートクラウドの Node サーバが消費する電力を測定する。これは、現実としてパブリッククラウド環境で消費電力量がわかることは想定しづらいためである。その代わりパブリッククラウドにおいては、従量制コストのコストがかかるものとする。これについては後述する。測定された消費電力量は、電力モニタ用 PC で収集される。

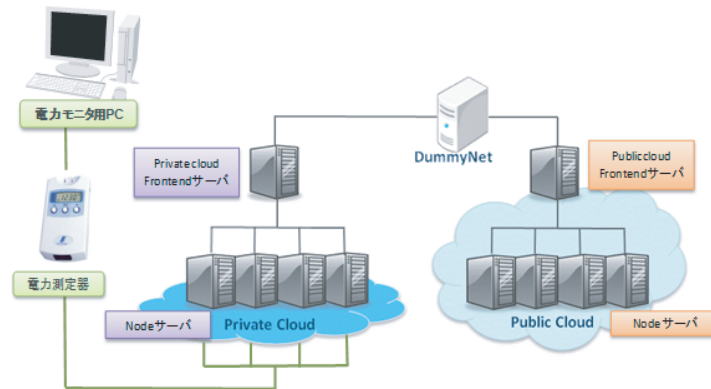


図 3 実験システム

表 1 Private Cloud Frontend

OS	Linux 2.6.38/Debian GNU/Linux 6.0
CPU	Intel(R) Xeon(R) CPU @ 3.60GHz
Main Memory	4GByte
Disk	141GByte

表 2 Private Cloud Node

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
Main Memory	4GByte
Disk	222GByte

表 3 Public Cloud Frontend

OS	Linux 2.6.38/Debian GNU/Linux 6.0
CPU	Intel(R) Xeon(R) CPU @ 2.40GHz
Main Memory	1GByte
Disk	72GByte

表 4 Public Cloud Node

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
Main Memory	4GByte
Disk	141GByte

5. 負荷分散ミドルウェア

5.1 ミドルウェア概要

本節に、本研究の第 1 段階のミドルウェアの概要を示す。この段階では、データインテンシブアプリケーション実行時に、プライベートクラウドとパブリッククラウドの Disk I/O 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、投入されたジョブの負荷分散を行うミドルウェアを構築した。すなわち、データインテンシブアプリケーションのジョブが連続的に投入されている状況において、プライベートクラウドのリソースを使い切ったら、パブリッククラウドへ負荷分散を行う。パブリッククラウドにおいても、ジョブが投入されたインスタンスのリソースを使い切ったら、新たにインスタンスを借り、ジョブを投入していく。このように、リソースを使い切ってから負荷分散を行なっていくことで、後述するミドルウェアの評価指標として考えるコストを抑えていき、効率よくジョブを処理する。

システム環境は、図 4 に示すように、どちらのクラウドにおいても、Node サーバに 1 イ

インスタンスずつ生成し、合計 8 台のインスタンスに対し、負荷分散を行なっていく。パブリッククラウドでは、インスタンス台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、本実験で投入するジョブの総量は、パブリッククラウド 4 台以内で十分に負荷分散可能なものである。

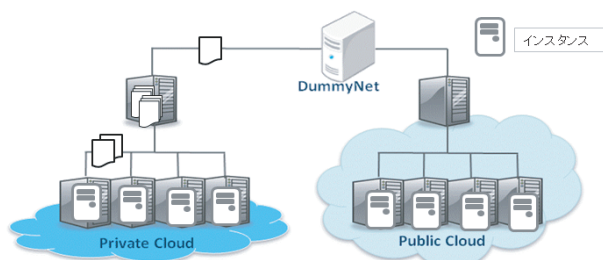


図 4 システム環境

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、ローカルクラスターおよびクラウドリソースの Disk I/O 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、dstat コマンドを利用して定期的に Disk I/O の情報を収集している。一方 Dispatch 部では、投入されたジョブを受け取り、Monitor 部で収集した Disk I/O の情報を基に、プライベートクラウドまたはパブリッククラウドへジョブを負荷分散する。

5.2 飽和判断

本ミドルウェアでは、リソースを使いきるための判断として、DiskI/O を使っている。これは、ジョブとしてデータインテンシブなジョブを使用しているためである。データインテンシブなジョブの場合、処理が I/O 待ちとなっていることが多く、CPU 負荷による判断が難しいため、DiskI/O から判断を行う。図 5 に、データインテンシブなジョブを次々に投入していった場合の DiskI/O と実行時間のグラフを示す。

この図から、ジョブが次々と投入されていくと、DiskI/O の値は飽和状態となり、この状態になると実行時間は飽和していない場合に比べ、長くなっていくことが読み取れる。このことから、本ミドルウェアでは飽和する前のリソースを使用範囲とする。そのために、ミドルウェアでは、定期的に DiskI/O の値を測定し、飽和状態の DiskI/O の値 (以下、S 値とする) を任意回連続して測定したら、別のインスタンスへ負荷分散を行い、S 値を任

意回連続で下回ったら、そのインスタンスは飽和していないと判断している。DiskI/O の値は不安定であることから、上記方法は飽和状態を見積もる適当な方法の 1 つであると考えられる。

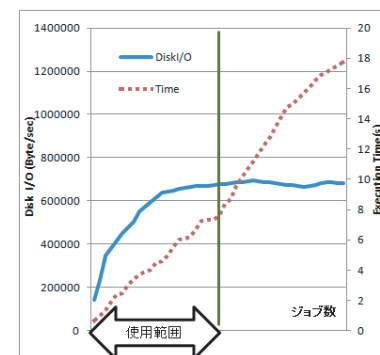


図 5 データインテンシブジョブ実行の DiskI/O と実行時間

5.3 ミドルウェアの評価指標

クラウドに負荷分散することを考えた場合、パブリッククラウド側で多くのインスタンスを用い、多くの処理を並列して実行すれば、全体の実行時間が短くなることが期待される。しかしパブリッククラウドは従量制のコストが発生するため、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。従ってパブリッククラウドを使用した負荷分散システムを実現する場合は、実行時間などのパフォーマンスとともに、従量制コストも考慮してリソースを配分することが重要である。そこで、第 1 段階のミドルウェアでは、ジョブ全体の実行時間とパブリッククラウドの従量制コストを考える。また、ここでのパブリッククラウドの従量制コストとしては、パブリッククラウドでジョブを実行した実行時間と借りたインスタンスの数をかけたものとする。

5.4 ミドルウェアのアルゴリズム

ジョブは連続的に投入されていることを想定している。

- (1) 連続投入されたジョブを受け取る。
- (2) プライベートクラウドにおいて実行優先順位が高いインスタンスから順次飽和しているか調べ、飽和していなければプライベートクラウド内のインスタンスで実行して (1) へ。飽和していれば (3) へ。

- (3) パブリッククラウドにおいて実行優先順位が高いインスタンスから順次飽和しているか調べ、飽和していないものが見つかり次第実行して (1) へ。見つからなければ (4) へ。
- (4) 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、実行して (1) へ。

まずはプライベートクラウド、次はパブリッククラウドの優先順位が高いものから Disk I/O の状態を調べ、空いているところから実行を行うことで、アプリケーションの実行時間とクラウドの従量制コストの両方がバランス良くなるようにしている。

5.5 データ配置

クラウドを利用した負荷分散の際には、データ配置を考えることも重要である。その配置として、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、本研究では、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。今後上記データ配置についても取り組んでいきたい。

6. ミドルウェアの実行と評価

5 章のミドルウェアを実行させる際、以下のような実験を行った。

6.1 実験概要

本研究の目指すミドルウェア構築の前段階として、5 章で述べたミドルウェアを飽和判断レベルを変化させ実行させた場合、ジョブ全体の実行時間と、パブリッククラウドの従量制コスト、プライベートクラウドの消費電力にどのような相関関係があるかについて、それぞれの値を測定すべく、実験を行った。この飽和判断レベルを変化させるとは、「インスタンスの DiskIO の値が S 値を超えた連続回数（以下、E 値とする）」と「インスタンスの DiskI/O 値が S 値を下回った連続回数（以下、L 値とする）」を変化させることである。今回は、E 値=L 値とした。例えば、E 値=L 値=2 だとすると、S 値を 2 回連続で上回ったらそのインスタンスは飽和状態であるとし、S 値を 2 回連続で下回ったらそのインスタンスは飽和していないと判断する、というものである。この E 値、および L 値のセット (E 値, L 値) を、(2, 2) (3, 3) (4, 4) (5, 5) と変化させ、その時のジョブ全体の実行時間、パブリッククラウドの従量制コスト、プライベートクラウドの消費電力を測定し、評価した。また、一度の実験でのジョブ数は 50 とし、投入するジョブは、データインテンシブアプリケーションの一つである PostgreSQL のベンチマークの pgbench を用いる。インスタンスの性能はすべて同じであり、表 5 に示す。

表 5 Instance

OS	Linux 2.6.27.21-0.1-xen /x86_ 84 GNU / CentOS 5.3
CPU	Intel(R) Xeon(R) CPU @ 3.60GHz 1 core
Main Memory	1024MByte
Disk	10GByte

6.2 実験結果と評価

ミドルウェアで負荷分散を行う際の、E 値および L 値を上述したように変化させた時の、ジョブ全体の実行時間、パブリッククラウドの従量制コスト、プライベートクラウドの消費電力を図 6 から 8 に示す。

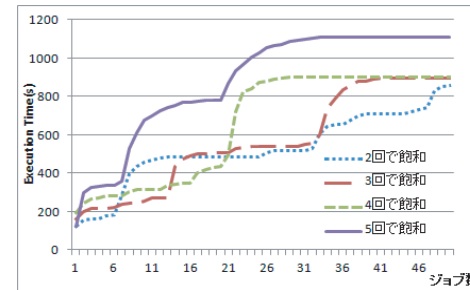


図 6 ジョブ全体の実行時間

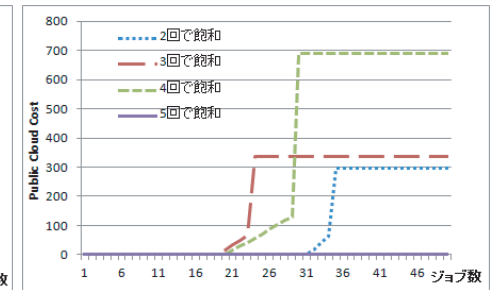


図 7 パブリッククラウドの従量制コスト

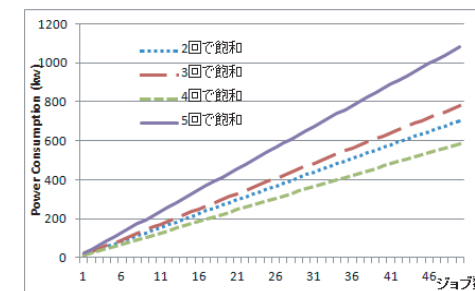


図 8 プライベートクラウドの消費電力

どのグラフにおいても、横軸は投入したジョブ数である。

図5から図8において(E 値, L 値) = (5, 5)の結果を見ると, 飽和判断レベルが高いことから, プライベートクラウド内のみで処理しており, パブリッククラウドの従量制コストが全くなかかっていないことがわかる。しかし, その結果, ジョブ全体の実行時間とプライベートクラウドの消費電力が大きくなってしまっている(E 値, L 値) = (4, 4)の結果を見ると, プライベートクラウド内においてもジョブはある程度実行しているが, パブリッククラウドにもジョブを負荷分散していることが読み取れる。このことから, 実行時間は(E 値, L 値) = (5, 5)の時に比べ, 抑えられている。しかし, 一度パブリッククラウドに負荷分散してしまうと, E 値やL 値が高めなため, プライベートクラウドの飽和状態の解消の判断も遅く, プライベートクラウドへのジョブの投入に戻らず, 余分にパブリッククラウドにジョブを投入してしまっている。そのため, プライベートクラウド内の消費電力量は小さいが, パブリッククラウドの従量制コストは高い(E 値, L 値) = (3, 3) (2, 2)においては, 極力ジョブをプライベートクラウド内で実行し, そのリソースを使い切りながら, 足りなくなったらパブリッククラウドへ負荷分散を行っており, ジョブ全体の実行時間, パブリッククラウドの従量制コスト, プライベートクラウドの消費電力のバランスが上手く取れていると言える。

7. 指標としての消費電力量実現にむけて

本研究の目指すミドルウェア(第2段階のミドルウェア)では, 負荷分散を行う際の指標として消費電力量を用いる。それをいかにして実現するかについては, 以下の式によりミドルウェアの概念的コストを見積もることを考えている。

$$\text{Total cost} = a \times \text{Max}(T_L, T_R) + b \times (T_R \times N_R \times C_R + P_L \times C_L)$$

T_L : Execution time on Private site

T_R : Execution time on Public Cloud

N_R : Number of Instances used on Public Cloud

C_R : Charges of Public Cloud

P_L : Power Consumption on Private Cloud

C_L : Charges of Power Consumption on Private Cloud

第一項が実行時間という時間的コスト, 第二項がパブリッククラウドの従量制コストとプライベートクラウドの利用時間分の消費電力量料金を合わせた金銭的コストを表す。パブ

リッククラウドの消費電力量料金は, パブリッククラウドの従量制コストに含まれると考える。これは, 現実としてパブリッククラウドの消費電力量料金がわかるということは想定しづらいく, また分かったとしてもその電力を削減しようとするかどうかはユーザのモチベーションに関わるためである。また, 定数 a, b においては, ユーザが時間的コストと金銭的コストのバランスをどのように考えるかを決める値であり, その比率等にも今後取り組んでいきたい。

8. ま と め

ハイブリッドクラウド環境にて, 大量のデータを省電力指向に処理するためのミドルウェア構築を目指して, ハイブリッドクラウド環境を構築し, まずは大量のデータのデータを効率よく処理するためのミドルウェアを作成した。このミドルウェアでは, インスタンスのリソースを使い切る際の判断として DiskI/O を用い, 測定値が事前に測定した飽和基準に任意回数分達すると負荷分散を行う。本論文では, その回数を変化させたときの, ジョブの全体の実行時間, パブリッククラウドの従量制コスト, プライベートクラウドの消費電力を測定し, 評価を行った。その結果から, プライベートにジョブを多く投入すれば, 従量制コストは抑えられるが消費電力量は多くなり, 逆にプライベートを少なくしてパブリッククラウドを多用すれば, 従量制コストは高くなるが, 消費電力量は抑えられることがわかった。また, 3つの値をバランス良くするような飽和判断レベルも存在していることがわかった。

最後に, 本研究の最終目標であるミドルウェア実現の際に, 指標としての消費電力を実現するための概念的コストも示した。

9. 今後の課題

今回の実験においては, プライベートクラウド内において1つのサーバに1つのインスタンスを配置していた。先行研究³⁾では, クラウド内のインスタンス配置により消費電力の違いがあることがわかっている。そのことから, インスタンス配置も考慮し, 評価を行っていきたいと考えている。また, データインテンシブなジョブの負荷分散先としてパブリッククラウドを利用する際に, そのデータ配置を考えるのも重要なことである。今後としては, クラウドの持つブロックストレージを使った場合や, ローカルにあるストレージに遠隔アクセスを行った場合などを考えていきたい。

今回の実験では「インスタンスの DiskI/O の値が S 値を超えた連続回数」と「インスタンスの DiskI/O 値が S 値を下回った連続回数」を同じ数とし, ミドルウェアを実行した。

今後としては、この2つのパラメータを様々な変化させた場合の評価を行なっていきたい。
しかし、このパラメータ設定の際には、「プライベートクラウド内での実行を優先」や「できるだけジョブを早く終わらしたい」など、ユーザの希望から決まるため、最適は一意には決められない。これらのことから、今後は少し異なるパラメータセットでの評価も行いながら、最適（パレート最適）はどういう状態であるかなどを含め深く議論して行く。

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。

参 考 文 献

- 1) 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
 - 2) Eucalyptus:<http://www.eucalyptus.com/>
 - 3) 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.
 - 4) Xen:<http://www.xen.org/>
 - 5) KVM:<http://www.linux-kvm.org/>
 - 6) SHW3A:<http://www.system-artware.co.jp/shw3a.html>
-