

## 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

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

近年, ネットワーク技術や仮想化技術の発展などによりクラウドコンピューティングが様々な形で社会に普及している。しかし, それに伴いクラウドを構築した側における IT 機器の消費電力量増加が問題となっている。そこで本研究では, 特にセキュリティの観点からも今後の発展が予想されるプライベートクラウドに注目し, クラウド構築ソフトウェア Eucalyptus を用いて実際にプライベートクラウドを構築し実験を行った。構築したクラウドにおいて, インスタンスを動作させる条件を様々に変え, それぞれの場合の性能や消費電力量を実際に測定し, 評価することで, 性能と消費電力量を考慮したプライベートクラウドの実現を目指す。

### Evaluation of Energy Consumption using Private Cloud System with Various Condition

YUMIKO KASAE,<sup>†1</sup> SHIORI TOYOSHIMA<sup>†1</sup>  
and MASATO OGUCHI<sup>†1</sup>

In recent years, by the development of network technology and virtualization technologies, cloud computing is popular in society. However, this spread has caused the problem of power consumption increase of the equipment used to build a cloud. In this study, we focused on private cloud which is expected particularly from a security perspective, actually built the private cloud using the software Eucalyptus. In this private cloud, we change the conditions in many ways, and actually measure and evaluated the power consumption and performance of it. And we aim to achieve private cloud considering performance and power consumption.

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

### 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。このことから, クラウドコンピューティングは様々な形となって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。

このことから, 本研究ではクラウドの中でも特にプライベートクラウドに注目し, プライベートクラウドに対して性能と消費電力を考慮した負荷分散を行うミドルウェアの作成を目指す。そのための第一段階として, クラウド構築ソフトウェア Eucalyptus<sup>1)</sup> を用いてプライベートクラウドを実際に構築し, 仮想化ソフトウェアとして KVM と Xen を使うなど, インスタンスを動作させる条件を様々に変えた上で, それぞれの場合の性能と消費電力を実際に測定し, 評価を行った。

### 2. 研究背景

#### 2.1 プライベートクラウド

クラウドは, プブリッククラウドとプライベートクラウドに大別できる。パブリッククラウドとは, プロバイダから提供されるクラウドであり, Amazon EC2<sup>2)</sup>, Google App Engine<sup>3)</sup>, Microsoft Windows Azure<sup>4)</sup> が代表例としてあげられる。インターネットの向こうから提供される様々な機能を, 手軽に利用できることがメリットであるが, その際に一時的にでもデータをプロバイダ側に預けなければならず, セキュリティ上の不安がある。また, 一般にパブリッククラウドは, 地理的に遠方に存在する可能性が多いため, アクセス性能などが問題になることがある。

その問題点を解決するものとして, プライベートクラウドが期待されている。プライベート

トクラウドは、ユーザ自身がクラウドを構築することによって、セキュリティに配慮するなどのユーザのニーズに合ったクラウドを提供することが可能である。特に大規模な企業などが使うクラウドとして注目されている。

これら 2 種類のクラウドの普及により、クラウドを構築する側における消費電力量の増大が問題となっているが、その解決へのモチベーションは両者では異なる。パブリッククラウドでは、すべてのユーザにサービスとして様々な機能を提供しなければならないため、安定供給を重視しなければならないが、省電力化へ取り組みづらいため。それに対し、プライベートクラウドは、その取り組みがユーザ自身に還元されるため、今後取り組みへの活発化が予想される。そのことから、本研究では、プライベートクラウドに注目し、実際にプライベートクラウドを構築したうえで、その消費電力量の削減を考えていく。

## 2.2 Eucalyptus

本研究では、クラウド構築基盤ソフトウェア Eucalyptus を用いてプライベートクラウドを構築した。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。

そのシステム構成を図 1 に示す。

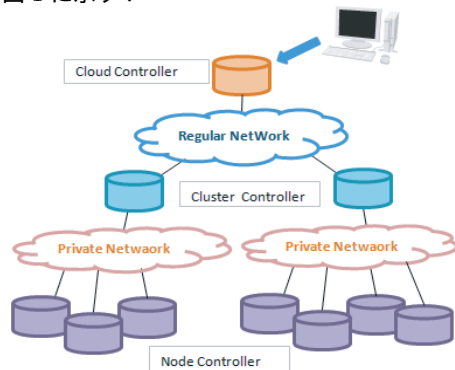


図 1 Eucalyptus のシステム構成

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。

## 2.3 仮想化ソフトウェア

Eucalyptus では仮想化ソフトウェアとして、Xen と KVM の 2 種類をサポートしている。本実験においては、その両者を用いて実験環境を構築し、それぞれの実験環境においてインスタンスの配置を変化させ、実験を行った。

### 2.3.1 Xen

Xen<sup>5)</sup> 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 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。ただし、Xen を利用するには、Xen 用に特別にコンパイルされたカーネルを利用しなければならない。

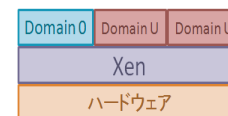


図 2 Xen のアーキテクチャ

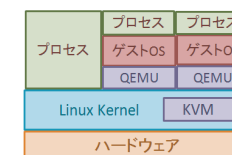


図 3 KVM のアーキテクチャ

### 2.3.2 KVM

KVM<sup>6)</sup> (Kernel-based Virtual Machine) は、Linux カーネルにハイパーバイザの機能を追加し、CPU の仮想化技術支援を必要としながら、完全仮想化により仮想マシン環境を提供する。図 3 に示すように、KVM ではハードウェアのエミュレーションやゲスト OS の管理用のフロントエンドとして QEMU を使っている。導入が容易で、ゲスト OS としても既存 OS がそのまま動作するが、準仮想化に比べればオーバーヘッド等によりパフォーマンスが劣る。Ubuntu 9.10 では、UEC (Ubuntu Enterprise Cloud) という、Eucalyptus をベースに作られた仮想化環境構築ソフトウェアセットが含まれており、仮想化ソフトウェアとして KVM を採用している。

### 3. 実験システム

#### 3.1 実験環境

構築したプライベートクラウドを図4に示す。本実験ではEucalyptus 2.0を用い、CLCとCCが動作するFrontendサーバ1台と、NCが動作するNodeサーバ4台の計5台でクラウドを構築した。Frontendサーバとしては、HP Workstations xw8200、CPUがIntel Xeon 3.60GHz、Memoryが4GB、OSがLinux 2.6.38/Debian GNU/Linux 6.0を用いた。Nodeサーバとしては、Power Edge T100、CPUがIntel Quad-Core Xeon 2.60GHz、Memoryが8GB、OSは、Xenの場合がLinux 2.6.32-5-xen-amd64 and xen-4.0-amd64/Debian GNU/Linux 6.0、KVMの場合はLinux 2.6.38/Debian GNU/Linux 6.0を用いた。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Nodeサーバは外部のネットワークから独立している。

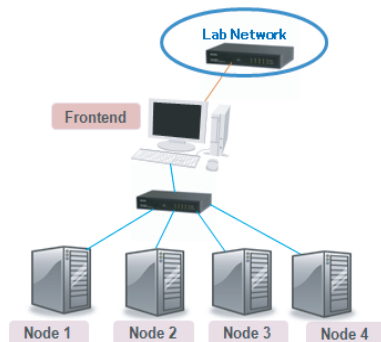


図4 構築したプライベートクラウド

この実験環境で電力を測定するための測定機として、システムソフトウェア製の高精度小型電力計ワットアワーメータ SHW3A<sup>7)</sup>を用いた。これは、コンセントに接続させ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。

#### 3.2 実験概要

本研究の最終目的は、プライベートクラウドに対する性能と消費電力を考慮した負荷分散ミドルウェアの作成である。そのための第一段階として、インスタンス配置の違いにより、プライベートクラウド全体でどのような性能と消費電力量の違いが出るかを評価する。本論文では、この実験環境を使って、表1にあるように5パターンにインスタンスの配置を変

化させ、実験を行った。

表1 インスタンス配置の変化

測定対象 実験環境	Nodeサーバ1	Nodeサーバ2	Nodeサーバ3	Nodeサーバ4
I	4			
II (3-1)	3	1		
II (2-2)	2	2		
III	2	1	1	
IV	1	1	1	1

表中の実験環境の数字は、インスタンス数を示しており、例えば実験環境Iなら、Nodeサーバ1にインスタンスを4つ配置し、Nodeサーバ1のみを電力の測定対象とする。電力の測定対象は、インスタンスが動作するNodeサーバとする。本論文の実験では、どの実験環境においても、インスタンス数の合計が4となるようにしている。これは、本実験環境において1つのサーバ内に生成できるインスタンス数の最大値が4であるためである。この数字は、EucalyptusにおいてNodeサーバ機のコア数に制限されている。これら5つの実験環境において、さらに仮想化ソフトウェアとしてXenとKVMをそれぞれ使い、合計10パターンの実験環境を用いて実験を行った。

今回の実験では、インスタンスの起動、ベンチマークによる負荷、インスタンスの終了の消費電力と実行時間を測定した。これらの測定結果をもとに、プライベートクラウド全体の消費電力量と性能を評価していく。

### 4. 基本性能測定結果

本章では、10パターンの実験環境において、インスタンスの起動、負荷、終了にかかった消費電力量と処理時間を示す。XenとKVMによる違いや、実験環境による違いを述べていく。

#### 4.1 インスタンスの起動

図5, 6, 7に、各場合のインスタンスの起動処理時間、利用するサーバ台数分の単位時間あたりの平均消費電力、消費電力量を示す。図では、KVMとXenで同じインスタンス配置のものは隣り合っている。

図5からわかることは、KVMはXenに比べ、どの実験環境においても圧倒的に起動処理時間が短いということである。また、KVMでは、インスタンス配置により起動処理時間が変化しないのに対し、Xenでは、インスタンスが分散して配置されるほど、起動時にサー

バ機にかかる負荷も分散し、処理時間が短くなってきていることがわかる。また、実験環境 II(3-1) と II(2-2) では、同じサーバ台数にもかかわらず、インスタンス配置が違うことにより処理時間に差が生じている。これは、実験環境 II(3-1) では、サーバに不均等にインスタンスを起動させたことにより、その負荷が偏り、処理に時間がかかってしまっていると考えられる。図 6 では、KVM と Xen では起動処理時の平均消費電力はさほどかわららず、インスタンスが分散して配置されると、値が大きくなってきていることがわかる。図 7 でも、図 5 と同様に、KVM の方が、Xen より消費電力量が少ない。図 6 に示されているように、単位時間当たりの消費電力は KVM と Xen でほとんど同じであることから、これは処理時間の影響が強く表れているためと考えられる。また、KVM では、インスタンスが分散して配置されるほど、消費電力量が大きくなっていくが、Xen では異なっており、特に複数台のサーバに不均等にインスタンスを配置している実験環境 II(3-1) や III は、消費電力量が大きくなってしまいう結果となった。

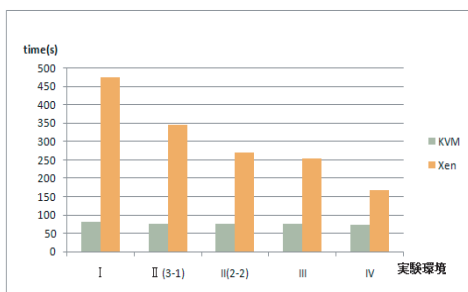


図 5 起動処理時間

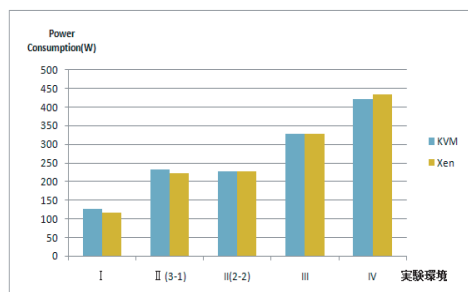


図 6 起動処理平均消費電力

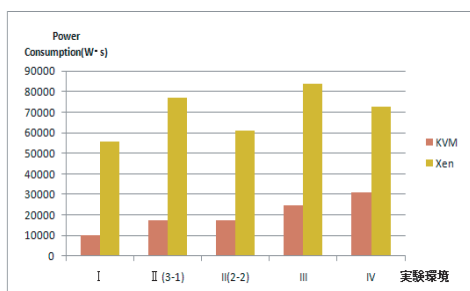


図 7 起動処理消費電力量

#### 4.2 ベンチマークによる負荷

本実験では、ベンチマークとして PostgreSQL のベンチマークテストを行うためのプログラムである pgbench<sup>8)</sup> を用いた。これはオフラインバッチ処理的なストレステストを指向する「TPC-B」ベンチマークを参考に、石井達夫氏によって開発された。複数のクライアントからの接続をシミュレートしたり、処理するトランザクション数を増やして、より大きな負荷をかけたりといったことが可能なデータインテンシブアプリケーションである。

図 8, 9, 10 に、ベンチマーク処理時間、利用するサーバ台数分の単位時間あたりの平均消費電力、消費電力量を示す。

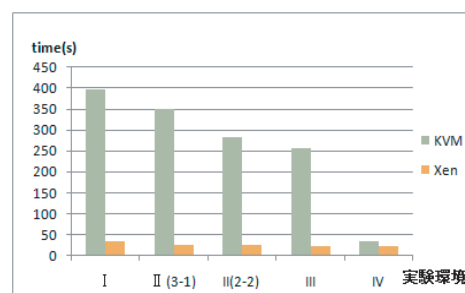


図 8 ベンチマーク処理時間

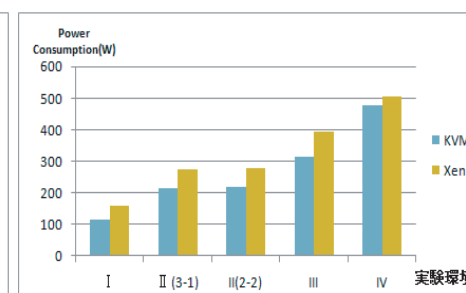


図 9 ベンチマーク処理平均消費電力

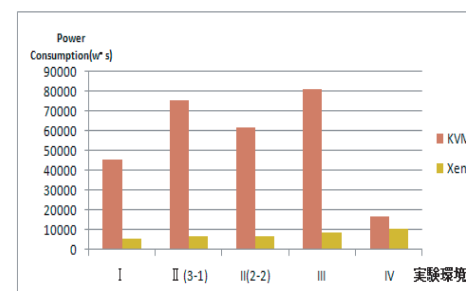


図 10 ベンチマーク処理消費電力量

今回の実験では、各インスタンスに 6 クライアントずつ、1 クライアントあたり 5000 のトランザクションの負荷をかけた。

図 8 からわかるように、起動時とは逆に、Xen の方が KVM より処理時間が大幅に短いことがわかる。また、実験環境 I から III までは、KVM ではかなりの時間がかかっているが、実験環境 IV では Xen と近い値となっている。これは、KVM の実験環境 III までは、

負荷の大きさからオーバーヘッドが多発し、処理時間がかかってしまっていたのに対し、実験環境 IV では、各サーバあたりに 1 インスタンスしか配置されていないため、その負荷が小さくなり、オーバーヘッドが抑えられ、処理時間が大幅に減少したと考えられる。Xen の環境においては、インスタンス配置により、分散して配置するほど処理時間は短くなってはいくが、違いはさほど見られなかった。図 9 では、KVM より Xen の方が平均消費電力が大きいが、またどちらにおいても分散して配置されるほど値が大きくなっていくことがわかる。図 10 では、どの実験環境でも、KVM の方が Xen より消費電力量が大きいことがわかるが、これは処理時間の長さの影響が強く表れたと考えられる。また、Xen のみを考えると、消費電力量には各実験環境の平均消費電力の違いが影響し、実験環境 I のようにインスタンスを集中して配置したほうが、消費電力量が小さい。

### 4.3 終了処理

図 11, 12, 13 に、終了処理時間、利用するサーバ台数分の単位時間あたりの平均消費電力、消費電力量を示す。

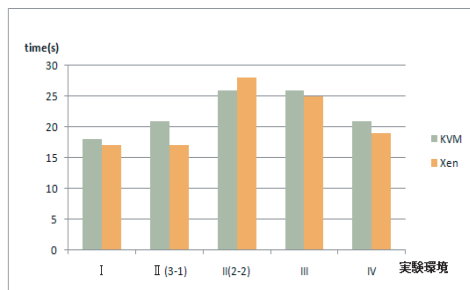


図 11 終了処理時間

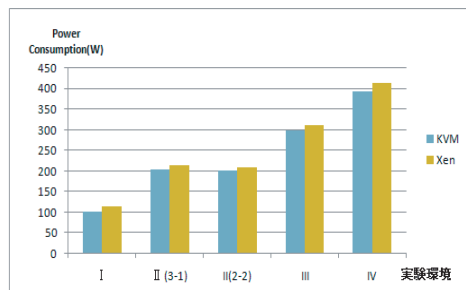


図 12 終了処理平均消費電力

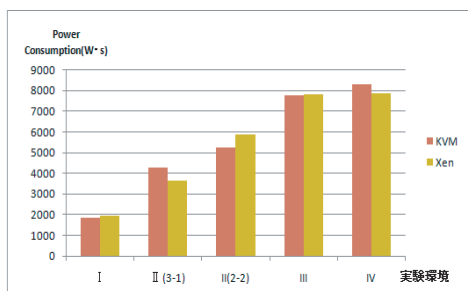


図 13 終了処理消費電力量

これら 3 つの図において、KVM と Xen を比べると、傾向の違いがほとんど見られない結果となった。これは、終了処理が必ずしも最短時間での実行を目指して設計されている訳ではないことによっていると考えられる。インスタンス配置においても、分散して配置すれば、消費電力量が大きくなる結果となった。

## 5. 評価

これら測定結果をもとに、本実験でのプライベートクラウドの消費電力量、および性能を考えていく。本論文では、以下の 5 パターンについて評価する。

### 評価 1

インスタンス配置によるデータインテンシブなジョブのみの性能と消費電力量の評価

### 評価 2

インスタンス配置による、インスタンスの起動、データインテンシブなジョブによる負荷、インスタンスの終了の性能と消費電力量の評価

### 評価 3

処理時間の長さの違いを考慮した、インスタンス配置による、インスタンスの起動、データインテンシブなジョブによる負荷、インスタンスの終了を含めた、プライベートクラウド全体の消費電力量の評価

### 評価 4

処理時間とサーバ台数の違いを考慮した、インスタンス配置による、インスタンスの起動、データインテンシブなジョブによる負荷、インスタンスの終了を含めた、プライベートクラウド全体の消費電力量の評価

### 評価 5

処理時間とサーバ台数の違いを考慮した、インスタンス配置による、データインテンシブなジョブによる負荷のみの、プライベートクラウド全体の消費電力量の評価

評価 1 については、インスタンスのジョブ実行のみの性能と消費電力量の評価である。

評価 2 は、評価 1 にインスタンスの起動、終了処理を加え、これらの処理時間と消費電力量を評価する。

評価 3 は、評価 2 の実験環境の違いによる処理時間の違いを考慮した。そのために、一番処理時間の長いものに電力の計算時間を合わせ、その間はサーバのアイドル状態であると考え、アイドル状態のサーバの消費電力量を加えた。これは、ユーザ側としては、サーバがイ

インスタンスの動作する分だけ起動しており、その上でそれぞれに配置にあわせ、インスタンスを起動させ、仕事をさせたあと、インスタンスのみ終了させている状態であると言える。クラウド構築側としては、必要最低限のサーバ台数において、様々なインスタンス配置で仕事をさせたときの消費電力量を、同じ時間分で切り取った場合の評価である。評価3では、処理時間の違いなしに、処理にかかる消費電力量がインスタンス配置によりどのように変化していくか評価する。

評価4では、評価3にさらにサーバ台数の違いを考慮していく。評価3では、処理時間の違いは考慮しても、サーバ台数の違いは考慮していなかった。そこで、評価4ではどの実験環境においても、サーバ台数をすべて同じとし、インスタンスが動作していないサーバはアイドル状態であるとして、その消費電力量も加えた。これは、サーバがすべて起動しており、その上でそれぞれの配置に合わせ、インスタンスを起動させ、仕事をさせたあと、インスタンスのみ終了させている状態であると言える。このことにより、処理時間、およびサーバ台数の違いなしに、処理にかかる消費電力量がインスタンス配置によりどのように変化するのか評価する。

評価5では、評価4同様、サーバ台数と処理時間の違いを考慮した上で、データインテンシブなジョブによる負荷のみのプライベートクラウドの消費電力量の評価を行う。これは、サーバとインスタンスがすべて事前に起動しており、ただジョブが投げられるのを待っている状態から、データインテンシブなジョブが投げられた場合の消費電力量の評価であると言える。このことにより、より実環境に近い状況において、インスタンス配置の違いによるクラウド全体の消費電力量の評価を行っていく。

この5つの評価により、評価1, 2においては、インスタンス配置による仕事のための処理時間と消費電力量の評価、評価3, 4, 5では、クラウド構築側としてのクラウド全体の消費電力量の評価を行う。

### 5.1 評価 1

評価1は、単純にデータインテンシブなジョブのみによる評価であり、前述した4.2章と同じである。4.2章の図10から、データインテンシブなベンチマークによる負荷の場合には、Xenの方が優れており、その中でもインスタンス配置を考えると、性能を重視するには、実験環境IVのように多くのサーバを使い負荷を分散させた方が良いが、消費電力的には実験環境Iのように負荷をまとめさせたほうが良い傾向があり、これらはトレードオフの関係にあるといえる。

## 5.2 評価 2

図14, 15に評価2の処理時間と消費電力量を示す。

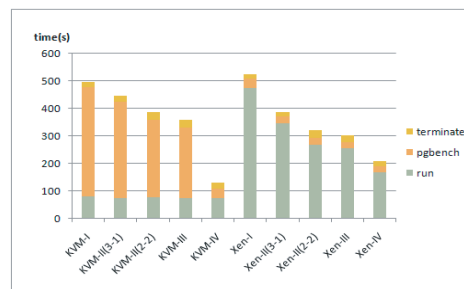


図 14 評価 2:処理時間

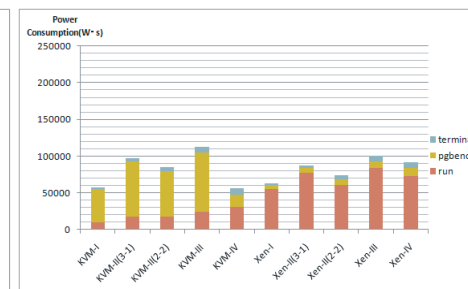


図 15 評価 2:消費電力量

図14では、それぞれの実験環境に、インスタンスの起動処理、ベンチマークによる負荷、インスタンスの終了処理の処理時間示されている。この図から、KVMの実験環境IVが一番速く処理を終え、その次に速いのがXenの実験環境IVであることがわかる。図15では、KVMの実験環境IとIV、Xenの実験環境Iがほぼ同じくらい消費電力量が小さい。

これらの結果から、評価2においては、性能と消費電力を考慮すると、KVMの実験環境IV、つまりインスタンスをサーバ機1台につき1インスタンスずつ分散して配置していった場合が一番良いことがわかった。しかし、KVMの環境において、この分散した状態が崩れ、サーバ内に2つ以上のインスタンスが生成されてしまうと、たちまち性能が悪くなってしまいう傾向にあることもわかった。また、Xenのみで考えると、評価1と同様で、性能を重視するにはインスタンスをすべて分散して配置する実験環境IVがふさわしく、消費電力量を考慮すると、一台のサーバ機に集中して配置した実験環境Iがふさわしいことがわかった。

### 5.3 評価 3

図16, 17に評価3の処理時間と消費電力量を示す。

図16では、評価2で考えた処理時間に加え、すべての実験環境の中で一番この処理時間が長かったXenの実験環境Iと各実験環境の処理時間との差を、インスタンスが動作していたサーバのアイドル状態の時間としている。これにより、電力の計算時間をすべて揃え、処理時間の違いが取り除かれた場合の消費電力量の評価が可能である。図17では、評価2の消費電力量の値に、図16にあるアイドル状態の時間分の消費電力量を加えた。この図から、KVMの実験環境Iがアイドル状態を含めた消費電力量が一番小さく、それとわずかな差でXenの実験環境Iがつづいていることがわかる。また、評価2において、消費電

力量が小さかった KVM の実験環境 IV が、この評価ではかなり大きくなってしまっていることも分かる。全体的に見ても、利用するサーバ台数が増えるにしたがって、消費電力量の値が順に大きくなっていく。これらのことから、評価 3 の場合においては、やはり実験環境 I のように、インスタンスをまとめて配置したほうが、クラウド構築側としては消費電力量が抑えられることがわかった。

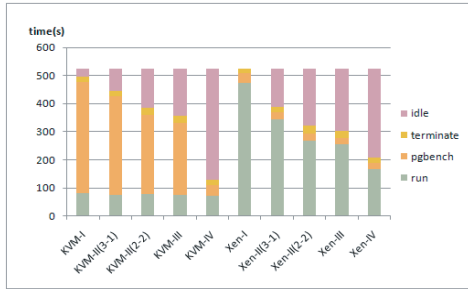


図 16 評価 3 と評価 4:処理時間

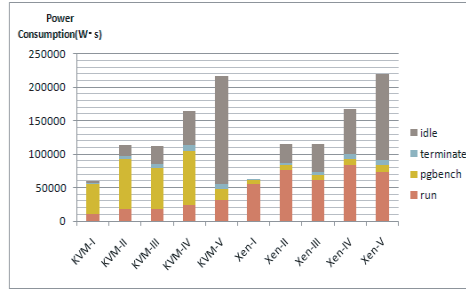


図 17 評価 3:消費電力量

#### 5.4 評価 4

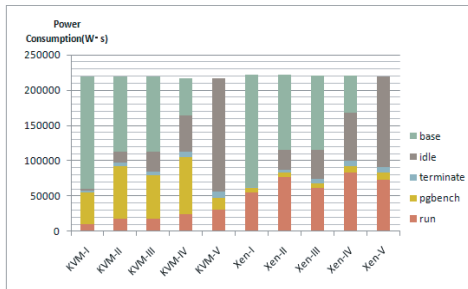


図 18 評価 4:消費電力量

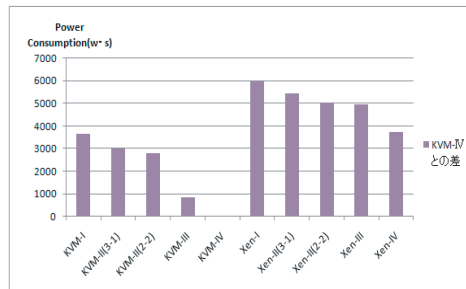


図 19 評価 4:消費電力量の差

図 18 に評価 4 の消費電力量を示す。評価対象の時間は評価 3 と同じである。

評価 3 により、各実験環境において、処理時間の違いは取り除かれたが、電力量を計算するサーバ台数の違いがある。そこで、評価 4 では、サーバ 4 台すべてを起動させており、インスタンスが動作していないサーバ機は常にアイドル状態にあるとし、その消費電力量を加える。図 18 では、その値を「base」とし、一番処理時間の長かった Xen の実験環境 I の時間分、アイドル状態とした場合に消費される電力量を、各実験環境のサーバ利用台数に合わせて加えた。これにより、どの実験環境でも、サーバ機は 4 台起動しており、仕事をし

ているサーバ機以外は常にアイドル状態である状況を想定できる。

図 18 の結果では、どの実験環境においても、全体の消費電力量が大きいことから消費電力的な差は見えづらい。そこで、図 18 を拡大し、中でも一番消費電力量が小さかった KVM-IV を 0 とし、その差を図 19 に示す。図 19 からわかるように、処理時間やサーバ台数の違いを排除しても、インスタンス配置による消費電力量の違いが生じている。この評価においては、Xen より KVM の方が消費電力量が小さいことがわかった。インスタンス配置を考えると、評価 3 と違って、分散して配置していく程、消費電力量は小さくなる。このことより、インスタンスをまとめて配置しても、その他のアイドル状態のサーバの消費電力が加わると、節電効果は得られず、この場合にはむしろインスタンスとジョブを分散配置した方が良いと言える。

#### 5.5 評価 5

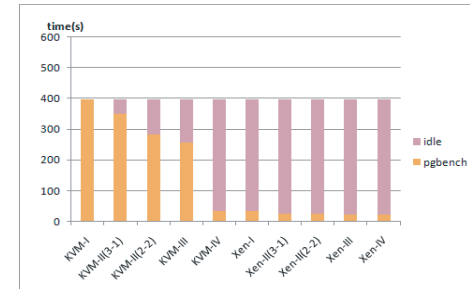


図 20 評価 5:処理時間

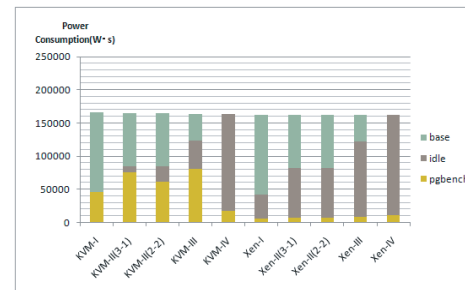


図 21 評価 5:消費電力量

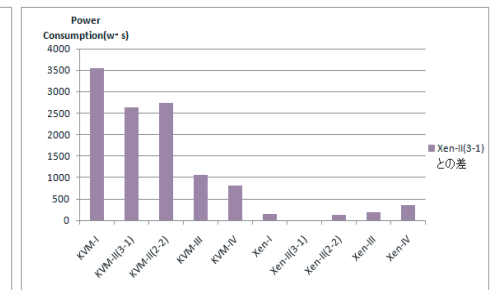


図 22 評価 5:消費電力量の差

図 20, 21 に評価 5 の処理時間と消費電力量を示す。評価 5 では、考えるジョブの対象を、データインテンシブなベンチマークのみとした。図 20 では、pgbench の処理時間に、一番

その処理時間が長かった KVM-I に合わせたアイドル状態の時間を足している。図 21 では、pgbench の消費電力量に、pgbench 終了後のアイドル状態の消費電力量、およびサーバ台数を合わせるためのベースの消費電力量を足している。この評価は、より現在のクラウド利用法に近い環境における消費電力量の評価である。

図 21 では、やはり消費電力量の差が見づらいため、一番消費電力量の小さかった Xen-II(3-1) を 0 とした場合の差を図 22 に示す。図 22 より、評価 5 においても、評価 4 同様に、処理時間やサーバ台数の違いを排除しても、インスタンス配置による消費電力量の違いが生じている。この図から、Xen の環境は KVM より消費電力量が小さいことがわかる。これは、4.2 章でも述べたように、KVM ではオーバーヘッドが多発しているためだと考えられる。インスタンス配置としては、Xen においては、どの配置においても消費電力量的な違いは殆ど無いが、KVM においては分散して配置していく方が良いことがわかる。

## 6. ま と め

Eucalyptus を用いてプライベートクラウドを構築し、サーバ台数やインスタンス配置、仮想化ソフトウェアを変化させた場合の、インスタンスの起動処理、データインテンシブなベンチマークによる負荷、インスタンスの終了処理の処理時間と消費電力を測定した。また、データインテンシブなベンチマークのみの評価、インスタンスの起動や終了処理を含めた評価、処理時間の違いを含めた評価、サーバ台数の違いを含めた評価を行った。

ベンチマークのみや、インスタンスの起動、終了処理を含めた仕事を考えると、KVM では、性能や消費電力量を最適にするインスタンス配置があり、Xen の環境では、それらがトレードオフのインスタンス配置を必要としていることがわかった。

また、評価 3 のような、利用するサーバのみを起動させ、インスタンスを配置していく、省電力を考える上で理想的な場合では、やはり利用するサーバを最低限にし、その中に負荷が大きくなるともインスタンスをできるだけ集中して配置していくほうが、消費電力量的には良い。しかし、サーバをすべて起動している状況において、その中でインスタンス配置を考えていくことは消費電力量的に有効であることがわかった。今回の評価においては、インスタンスの起動、データインテンシブなベンチマークの負荷、インスタンスの終了を含めた場合だと、Xen より KVM の方が良く、インスタンス配置的には分散して配置していくほうが良い。データインテンシブなベンチマークによる負荷のみでは、仮想化ソフトウェアの特性からも Xen の方が良く、もし KVM を利用する場合ならインスタンスは分散して配置しておくべきであることがわかった。これらのことから、インスタンス配置による省電

力化を考える場合には、様々な状況に応じた場合分けが必要であることがわかった。

## 7. 今後の課題

今回は、どの実験環境でもインスタンス数 4 の場合における評価だったため、インスタンス数を様々に変化させて評価を行っていききたい。その上で、最適なインスタンス配置や、最適なインスタンス数などを見つけていきたいと考えている。また、その他のベンチマークによる評価や、ネットワークストレージを使った場合も評価していききたい。そして、それらの評価をもとに、本研究の最終目的である、プライベートクラウドに対して性能と消費電力を考慮した負荷分散ミドルウェアの開発を行っていききたい。

## 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。

## 参 考 文 献

- 1) Eucalyptus:<http://www.eucalyptus.com/>
- 2) Amazon EC2:<http://aws.amazon.com/jp/ec2/>
- 3) Google App Engine:<http://code.google.com/intl/ja/appengine/>
- 4) Microsoft Windows Azure:<http://www.microsoft.com/ja-jp/cloud/developer/default.aspx>
- 5) Xen:<http://www.xen.org/>
- 6) KVM:<http://www.linux-kvm.org/>
- 7) SHW3A:<http://www.system-artware.co.jp/shw3a.html>
- 8) pgbench:<http://www.postgresql.jp/document/9.0/html/pgbench.html>