

仮想化環境におけるクラスタ構成の動的変更制御による省電力化方式の提案

宮田 康志^{1,a)} 太田 智也¹ 天野 雅志¹ 西山 博泰¹

受付日 2011年8月13日, 採録日 2012年2月3日

概要: 仮想サーバで構築したクラスタ構成の IT システムを, 負荷に応じて構成変更することで消費電力を削減できる. 従来の構成変更方式としてスケールイン/アウトとコンパクションがある. スケールイン/アウトでは, 負荷変動に応じて IT システムを構成する仮想サーバ数を増減させ, 仮想サーバの稼働しない物理サーバの電源を停止し消費電力を削減する. コンパクションでは, 仮想サーバを物理サーバ間移動することで仮想サーバが稼働しない物理サーバを作り出す. ただし, コンパクション時に仮想サーバの移動数が多いと, 移動負荷の増大により省電力効果が減少する. この課題を解決するために 2 方式を組み合わせ, コンパクション時の仮想サーバ移動負荷を低減させるスケールイン/アウト対象選択方式を提案する. 本提案方式により物理サーバ間の仮想サーバ数を偏らせ, 1) 仮想サーバ数が多くリソース使用効率が高い物理サーバ, 2) 仮想サーバ数が少なくコンパクション時に仮想サーバ移動数が少ない物理サーバ, を作り出す. 評価では, 構成変更がない場合に比べ 48.2%の電力を削減できた. また, 偏らせない場合に比べ, コンパクションの回数を 90%削減し, 6.2%の消費電力削減率を向上させることができた.

キーワード: 省電力, 仮想化, クラスタ, スケールイン/アウト, コンパクション

Power Saving Method for Virtualized Computer Environments via Dynamic Cluster Configuration

YASUSHI MIYATA^{1,a)} TOMOYA OHTA¹ MASASHI AMANO¹ HIROYASU NISHIYAMA¹

Received: August 13, 2011, Accepted: February 3, 2012

Abstract: Power consumption of clustered virtual servers can be reduced by dynamically controlling their configurations according to workload changes. Dynamic resource allocation using scale-in/scale-out and server consolidation among physical servers can be used for realizing this. In this paper, we propose the power saving method by combining the scale-in/scale-out method and the server consolidation method in order to reduce the overhead for server movements. Experimental evaluations show that the proposed method can reduce power consumption 48.2% compared to an environment without power saving. In addition, the proposed method can reduce the power consumption 6.2% and the number of the server consolidation 90% compared to a method without consideration of virtual server movements.

Keywords: power saving, virtualization, cluster, scale-in/scale-out, compaction

1. はじめに

近年, 地球環境の保護や電力コストの削減要求により, 企業の IT システムが消費する電力量の削減が求められて

いる. このような消費電力削減を目的とし, IT システムを構成するサーバ群やその設備まで含んだデータセンタ全体の消費電力削減技術が着目されている. この消費電力削減技術を, 次の (a)~(d) に大別し, その技術例を示す.

(a) データセンタ設備の改良による消費電力削減
空調機器や UPS (無停電電源装置) の改良による電力ロスの低減, サーバや空調機器の配置により空気の流

¹ 株式会社日立製作所横浜研究所
Yokohama Research Laboratory, Hitachi, Ltd., Yokohama,
Kanagawa 244-0817, Japan

a) yasushi.miyata.bz@hitachi.com

れを最適化し、冷却効率を向上させる [1].

- (b) サーバ装置を構成する CPU などの改良による消費電力削減

CPU の消費電力は動作電圧の 2 乗とクロック周波数に比例して増加する. この特徴に着目した消費電力削減方式の一例として, 動作電圧とクロック周波数を動的に変更する DVFS (Dynamic Voltage and Frequency Scaling) 機構の活用がある [2].

- (c) IT システム構築にサーバ仮想化を活用することによる消費電力削減

企業の IT システムを構成するサーバ群を仮想化し, 少数の物理サーバで業務を稼働させる集約を行う [3]. この集約により, 企業の IT システムを稼働するために必要な物理サーバの稼働台数を削減し, 消費電力を削減する.

- (d) サーバ装置の運用による消費電力削減

サーバ装置は待機状態でも電力を消費しており, 最大負荷時と比べて 40%以上の電力量を消費する [4]. 複数のサーバ装置からなる IT システムの構成を変更し, 待機状態となった不要なサーバ装置の電源を停止させることで消費電力を削減する [5].

サーバ装置の運用による消費電力削減 (d) では, 負荷に応じて IT システムを構成するサーバ数を減少または増加させるスケールイン/アウトと, サーバ仮想化を前提とし仮想サーバの配置を動的に変更し集約するコンパクションの 2 方式がある.

(1) スケールイン/アウト

スケールイン/アウトは複数サーバで IT システムを構成するクラスタに適用する技術である. クラスタは一般的に負荷分散器を持ち, 負荷となるクライアントのリクエストを複数サーバのうちの 1 つに転送し複数サーバ間で負荷を分散させる. このようなクラスタにおいて, スケールインは負荷が減少した場合にサーバ台数を減少させ, スケールアウトは負荷が増加した場合にサーバ台数を増加させる. スケールイン/アウトにより, 変動する負荷に対して過不足のない処理能力を提供できる.

(2) コンパクション

仮想化環境では, ライブマイグレーション [6] により物理サーバ上で稼働する仮想サーバをその状態を保持したまま別の物理サーバに移動できる. この移動により, 仮想サーバ群を少数のサーバに集約するコンパクションを実行し, 不要となった移動元の物理サーバを停止させることで消費電力量を削減できる. また, 状態を持たないステートレスな IT システムを対象とした場合には, 次のように, スケールインとスケールアウトを組み合わせることで疑似的に移動することでコンパクションを実行することもできる.

- (a) 移動先の物理サーバで仮想サーバを起動し, IT システ

ムをスケールアウトする.

- (b) クライアントから IT システムへのリクエストを移動元の物理サーバで稼働する仮想サーバから (a) で起動した仮想サーバに割り振る.
- (c) 移動対象となった仮想サーバを停止させ, IT システムをスケールインする.

複数のクラスタが仮想化環境に存在する場合, 仮想化環境を構成する物理サーバには各クラスタを構成する仮想サーバが混在する. この場合, 特定のクラスタでスケールインし仮想サーバ数が減少しても, 他クラスタを構成する仮想サーバが同じ物理サーバ上に存在すると物理サーバの電源を停止できず, 大幅な消費電力の削減が見込めない. 一方, 稼働中の仮想サーバの配置を変更するコンパクションを組み合わせると, 物理サーバの電源を停止可能な構成に変更することができるため, 消費電力削減量の増加が見込まれる. ただし, コンパクションでは仮想サーバを動的に移動するために負荷がかかるという問題がある. ライブマイグレーションでは, 処理実行中に仮想サーバの状態であるメモリを移動元から移動先にすべてコピーする. そのため, コピー処理にともなう CPU 負荷や, メモリ転送によるネットワーク負荷が増え, 消費電力が増大する. また, 一時的な処理停止も発生するため処理の遅延を招く. スケールアウトとスケールインを利用する場合にも, スケールアウト時の仮想サーバの起動処理やスケールイン時の仮想サーバの停止処理により CPU 負荷が増え消費電力量が増大する. このような課題を解決するためにコンパクション時の仮想サーバ移動回数を低減させる方式が複数提案されている [7], [8], [9]. しかし, これら従来方式では, 本論文で対象とする IT システムのように負荷の増減に応じてスケールイン/アウトし, クラスタを構成する仮想サーバ数やその配置が変化することは考慮されていない.

本論文では, 複数のクラスタにスケールイン/アウトとコンパクションを組み合わせる場合に, 両方を制御することで消費電力量を削減する省電力制御方式を提案する. 本方式では, 各クラスタにおいてスケールイン対象の仮想サーバや, スケールアウト対象の物理サーバをコンパクション時の仮想サーバ移動回数が少なくなるように選択する. この選択により, 各物理サーバで稼働する仮想サーバ数を適切に制御し, コンパクションにおいて特定の物理サーバから移動する必要のある仮想サーバ数を低減させる. その結果, コンパクション時の移動負荷が低減し, 消費電力量も削減できる.

以下, 2 章でスケールイン/アウトとコンパクションを用いた消費電力削減方法とその特徴を述べる. 3 章では, 2 章で述べた方式の組合せにより仮想サーバの移動回数を低減させる方式を提案する. 4 章で提案方式による省電力性能を評価し, 5 章で関連研究を概観した後に 6 章でまとめる.

2. クラスタ構成の動的変更制御による省電力化

本章では、仮想化環境においてスケールイン/アウトとコンパクションを利用して消費電力を削減するための従来技術について述べる。

2.1 スケールイン/アウト

スケールイン/アウトでは、負荷の上昇と下降に応じてクラスタを構成する仮想サーバ数を増減させる。仮想サーバ数の削減はスケールイン、増加はスケールアウトである。この増減の結果、仮想サーバ数が減少し、物理サーバ上の稼働仮想サーバがなくなれば物理サーバを停止し消費電力を削減できる(図1)。以下にスケールインおよびスケールアウトの具体的な処理を示す。

スケールイン

- (1) クラスタを構成する仮想サーバ1台を任意に選択。
- (2) 負荷分散器のリクエスト割り振り先から選択した仮想サーバを削除。
- (3) 選択した仮想サーバで実行中のリクエスト処理がすべて完了した後に選択した仮想サーバを停止。

スケールアウト

- (1) クラスタを構成する仮想サーバをリソース消費量に余裕のある任意の物理サーバ上で新たに起動。
- (2) 負荷分散器のリクエスト割り振り先に新規起動した仮想サーバを追加。

スケールイン/アウト契機の判断や、スケールアウト先として選択する物理サーバの判断を行うための負荷指標として、サーバ装置がリクエストを処理するのに必要なリソースの消費量が用いられる。具体的には、CPU、メモリ、ディスク I/O、ネットワーク I/O があげられる。

CPU リソースはリクエストを処理するロジック実行に消費されるため、リクエスト量やその内容に応じて消費量が増減する。メモリリソースはリクエスト処理過程での一時データの保持とその他の継続状態を保持するために一定量が消費され続ける。たとえば、Java^{*1}はJava VMがつねに一定のメモリリソース量を確保する。このような場合、

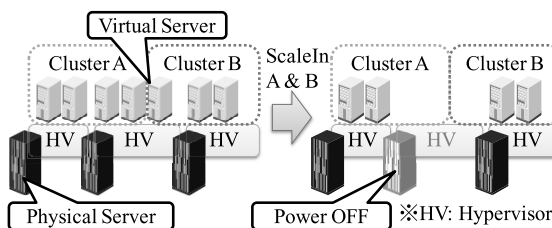


図1 スケールイン/アウトによるリソース使用量調整

Fig. 1 Reconfiguring the resource usage with scale-in/out.

*1 Java は、米国およびその他の国における Oracle, Inc. の商標または登録商標です。

メモリリソース消費量は IT システムの負荷となるリクエスト量やその内容と相関が低い。ディスク I/O のリソースは、リクエスト処理内容をログとして記録する場合や、リクエスト処理過程で DB やファイルシステムのデータを読み書きする場合に消費される。ネットワーク I/O のリソースは、クライアントからサーバへのリクエストとそのレスポンスを送受信するために消費される。また、クラスタ構成システムでは、複数サーバ間でネットワークを介してデータを共有しており、DB やファイルシステムのデータを読み書きする場合にもネットワーク I/O のリソースを消費する。このようにディスク I/O やネットワーク I/O のリソース消費はともにリクエスト処理と連動しており、リクエスト量やその内容に応じて消費量が増減する。

以上のように、CPU、ディスク I/O、ネットワーク I/O のリソース消費量は IT システムの負荷であるリクエスト処理量に連動して増減するため、スケールイン/アウト契機を判断する負荷指標とされることが多い。一方、メモリリソースは IT システムの負荷と相関が低く、スケールイン/アウト契機を判断する負荷指標とされることが少ない。ただし、リクエスト処理に必要なリソースであるため、スケールアウト先として選択する物理サーバを判断する場合には、他リソースとともに負荷指標とされる。

2.2 コンパクション

仮想サーバを物理サーバ間で移動することで動的に仮想サーバを集約し、不要な物理サーバを停止できる。このコンパクションと 2.1 節のスケールイン/アウトを組み合わせることでより高い省電力効果を得ることができる。

コンパクション実行の契機は図2のように物理サーバごとにリソース使用率が分散するときである。たとえば、使用率 25% の物理サーバを移動元、使用率 50% の物理サーバを移動先として仮想サーバを移動させる。これにより、移動先でリソース使用率が上昇するが、移動元の仮想サーバがなくなるため物理サーバを停止させ、消費電力量を削減できる。このコンパクションを実行した後も再びコンパクションする契機が発生する。これは負荷に応じて、クラスタをスケールイン/アウトすることで物理サーバ上の仮想サーバ数が増減するためである。この仮想サーバ数の増減により、リソース使用率が再び分散する。そのため、省電力制御のためには定期的あるいは負荷変動に応じてコン

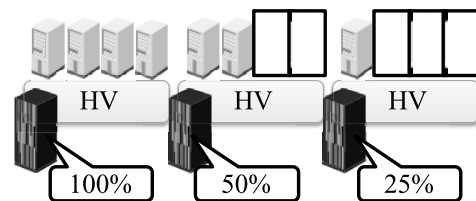


図2 使用リソースのフラグメンテーション化

Fig. 2 Fragmentation of resources.

パクションを繰り返す必要がある。

3. コンパクションの負荷を低減する動的変更制御システム

前章で述べたスケールイン/アウトとコンパクションを組み合わせることでより高い省電力効果を得ることを期待できる。一方、仮想化環境でクラスタ構成のITシステムを複数稼働させる場合、各ITシステムの負荷変動にともなうスケールイン/アウトの繰返しにより、各物理サーバで稼働する仮想サーバ数が変わり、使用リソースのフラグメンテーション（断片化）が起こる。連続使用領域が減少するメモリのフラグメンテーションとは異なり、物理サーバのリソース使用効率が低下し、使用効率が高い場合に比べて消費電力量が増加するという問題が発生する。この問題を回避するため、フラグメンテーションのたびに稼働物理サーバ数を低減可能か確認し、可能ならばコンパクションする。これにより、ITシステム全体の消費電力量を削減できると考えられる。ただし、リソース利用率の断片化状態によっては、コンパクションに必要な仮想サーバの移動回数が増加する可能性がある。この場合、コンパクション完了までに発生する移動負荷の総量が増大し、省電力効果が低減することになる。また、仮想サーバ移動負荷は移動先と移動元の物理サーバに影響を及ぼすため、それら物理サーバで稼働する移動対象以外の仮想サーバが利用可能なリソースも減少し、過負荷状態に陥りやすくなる。本章では、これらの課題を解決するために、スケールイン/アウト対象を適切に選択し、コンパクション時の仮想サーバの移動回数を減らすシステムを提案する。

3.1 スケールイン/アウト対象選択によるコンパクション負荷低減

本節では、コンパクション回数を低減させるためリソース利用率のフラグメンテーション抑止を実現するスケールイン/アウト方式を提案する。

リソース利用率のフラグメンテーションを抑止するためには、できる限り多くの仮想サーバを同一物理サーバで稼働させる必要がある。つまり、リソース容量の異なる仮想サーバを、リソース容量の上限を持つ物理サーバ群に対し、その容量の無駄なく配置する必要がある。これは一般にビンパッキング問題 [10] と呼ばれ、NP 困難な問題だが、その近似解法としてファーストフィット法が知られている。このファーストフィット法を、仮想サーバをより少数の物理サーバ群に配置する問題に適用できる。

稼働する物理サーバ数を最小限にするためには、リソース容量の大きい物理サーバで多くの仮想サーバを稼働させた方がよい。そのため、リソース容量の大きい順番に物理サーバに番号を付与し、仮想サーバをリソース容量の大きい順番に最も若い番号を持つ物理サーバに配置していく。

このとき、物理サーバの容量が不足すれば次に番号の若い物理サーバに仮想サーバを配置する。

本論文のスケールアウトでは、すでに物理サーバに配置された状態から物理サーバを1台選択し仮想サーバを増加させる。ファーストフィット法に従った場合、最も若い番号を持つ物理サーバはリソース容量が多く、仮想サーバの配置を試行する回数も多いため、最も仮想サーバ数が多いと考えられる。したがって、最も仮想サーバ数が多い物理サーバの仮想サーバ数を増加させるようにスケールアウトする。また、その物理サーバが容量不足ならば次に仮想サーバ数が多い物理サーバの仮想サーバ数を増加させるようにスケールアウトする。このようにスケールアウト対象を選択することで、仮想サーバ数の多い物理サーバにはより多くの仮想サーバを稼働させられる。スケールインにより、物理サーバの仮想サーバ数を低減する場合も仮想サーバ数の多い物理サーバでより多くの仮想サーバを稼働させることを考慮する。そのため、仮想サーバ数の少ない物理サーバを選択し、その仮想サーバ数を減らすようにスケールインする。

以上のスケールイン/アウト対象の選択を繰り返すことで、複数のクラスタを構成する仮想サーバ群をできる限り少ない物理サーバ数で稼働させ、フラグメンテーションを抑止できると考えられる。このようなフラグメンテーションの抑止では、仮想サーバ群を特定の物理サーバに偏らせた状態となる。この偏りを維持することでフラグメンテーションを抑止する制御方式を本論文では偏り維持方式と呼ぶ。偏り維持方式では、クラスタのスケールイン/アウト対象となる物理サーバを次のように選択する。

- A) スケールイン：クラスタを構成する仮想サーバが稼働する物理サーバのうち、稼働仮想サーバ数が最も少ない物理サーバ
- B) スケールアウト：クラスタを構成する仮想サーバを追加するリソース余裕がある物理サーバのうち、稼働仮想サーバ数が最も多い物理サーバ

3.2 動的変更制御による過負荷状態の防止

フラグメンテーション状態では、各物理サーバに未使用リソースが存在する。そのため、各物理サーバで稼働する仮想サーバ群の一部の使用リソース量が想定以上に上昇しても、物理サーバが保持する未使用リソースを割り当てることでリソース量不足に陥らない。一方、フラグメンテーションを抑止し偏り状態とすると、仮想化環境を構成する物理サーバ群の一部に処理が集中するため、その物理サーバの未使用リソース量が減少する。つまり、この偏り状態では、フラグメンテーション状態に比べ物理サーバにリソース量の余裕が少なく、リソース量不足に陥り処理が滞る過負荷状態となる可能性が高くなる。この過負荷状態の発生を抑止するには、各物理サーバで稼働する仮想サーバ

のリソース消費量を適切に管理する必要がある。

管理対象のリソースとしては、2.1 節に示したように CPU、ディスク I/O、ネットワーク I/O があげられる。以降では、CPU を一例とし、各リソースにおいて過負荷状態を防止するためのスケールイン/アウト閾値設定方式について述べる。なお、他のリソースに関しても、同様に扱うことが可能である。

CPU 使用率が一定値を超え過負荷状態となると、IT システムを構成するクラスタの処理性能が劣化する。この性能劣化を防ぐため、スケールアウトを実施し、クラスタを構成する仮想サーバ数を増加させる。これにより負荷を分散し各仮想サーバの CPU 使用率を下げる。ただし、負荷上昇中にスケールアウトを実施するため、スケールアウト中の負荷上昇も考慮する必要がある。つまり、スケールアウト完了前に過負荷状態に陥らないことを保証する必要がある。また、各仮想サーバの負荷上昇率は、クラスタを構成する仮想サーバ数が多いと負荷が分散するため少なく、仮想サーバ数が少ないと負荷が集中するため大きい。以上から、スケールアウト閾値 S_{out} は、過負荷状態の CPU 使用率を S_{limit} 、スケールアウトにかかる時間を T 、負荷上昇率の最大値を R 、スケールアウト前のクラスタを構成する仮想サーバ数を N として式 (1) で決定できる。式 (1) では、スケールアウト閾値 S_{out} が負の値になる場合がある。この場合の N で示される仮想サーバ数では、想定する最大負荷上昇が発生すれば過負荷状態になることを示す。このような過負荷状態を防ぐため、スケールアウト閾値 S_{out} が正の値となる N の最小値をクラスタ構成の IT システムに含まれる仮想サーバの最小数と設定する。

スケールインでは、負荷分散器のリクエスト割り振り先からスケールイン対象とする仮想サーバを取り除くことで、クラスタを構成する仮想サーバ数を削減する。そのため、スケールイン後のリクエストはスケールイン前より少ない仮想サーバ数で処理される。つまり、従来はスケールイン対象の仮想サーバを含めて分散した負荷がより少数の仮想サーバに集中する。この集中により、仮想サーバ 1 台あたりの負荷が上昇し過負荷状態に陥る可能性がある。このスケールイン時に、クラスタにかかる負荷 S_{after} は、スケールイン前のクラスタを構成する仮想サーバ台数 N (≥ 2)、スケールインで停止する仮想サーバ台数 M ($< N$)、スケールイン前の負荷 S_{before} を用いて式 (2) で表すことができる。スケールインによる過負荷状態を防ぐにはスケールイン後に、各仮想サーバにかかる負荷 S_{after} が式 (1) で導出したスケールアウト閾値を超えて過負荷状態となることを防ぐ必要がある。したがって、スケールイン閾値 S_{in} は、負荷 S_{before} をスケールイン閾値 S_{in} 、負荷 S_{after} をスケールアウト閾値 S_{out} 以下とし、式 (3) により表される。

$$S_{out} \leq S_{limit} - T * R / N \tag{1}$$

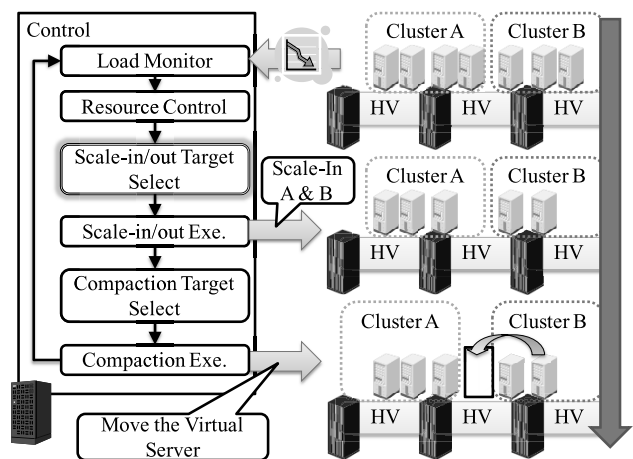


図 3 コンパクション負荷を考慮した省電力制御の動作
Fig. 3 Power saving control with consideration of virtual server movements.

$$S_{after} = S_{before} * N / (N - M) \tag{2}$$

$$S_{in} \leq S_{out} * (N - M) / N \tag{3}$$

3.3 スケールイン/アウトとコンパクションの組合せ

本節では、3.1 節で示した偏り維持と 3.2 節で示した閾値設計によるスケールイン/アウトとコンパクションとの組合せによる省電力制御システムについて述べる。

本論文で提案する制御を実現する全体制御部 (Control) の動作を図 3 に示す。全体制御部の中で負荷監視部 (Load Monitor) はクラスタを構成する仮想サーバの CPU 使用率を定期的に取り得ることで各クラスタの負荷を監視する。リソース調整部 (Resource Control) は、監視結果から各クラスタに対してスケールイン/アウトが必要か判断する。

スケールイン/アウトの判断では、省電力化のために無駄なリソース使用を防ぐ必要がある。一方、クラスタの過負荷状態を防ぐため負荷に対して十分なリソースを与える必要がある。そのため、3.2 節で示した閾値を用いて過負荷状態になる前にスケールアウトし、過負荷状態にならないことを保証してスケールインする。クラスタのスケールイン/アウトを決定した後、スケールイン/アウト対象選択部 (Scale-in/out Target Select) が偏り維持方式に従いスケールイン/アウト対象物理サーバを選択する。その後、スケールイン/アウト実行部 (Scale-in/out Exe.) がスケールイン/アウトを実行する。

スケールイン/アウト実施後は図 3 のコンパクション判定部 (Compaction Target Select) がコンパクション可能性を判定し、可能ならばコンパクション実行部 (Compaction Exe.) が仮想サーバを物理サーバ間で移動させる。コンパクション時の移動回数が少なくなり、かつ偏りが維持されるように、移動対象を次の手順で判定する。

- (1) 仮想サーバ数が最小の物理サーバと仮想サーバ数が最大の物理サーバを選択。

- (2) 仮想サーバ数が最小の物理サーバから仮想サーバ数が最大の物理サーバへ仮想サーバをすべて移動できるか判定.
- (3) すべて移動できないならば稼働する仮想サーバ数が次に多い物理サーバを追加し、複数台の物理サーバへ仮想サーバをすべて移動できるか再判定.

以上の処理で、稼働仮想サーバが存在しない物理サーバがあればその物理サーバの電源を停止させる. その後、再び各業務の負荷監視を続ける.

4. 評価

本章では 3 章で示した偏り維持方式のスケールイン/アウトとコンパクションを組み合わせた制御の省電力効果を、実環境とシミュレーションで評価する. 実環境では、制御ありと制御なしを比較し、提案方式の省電力効果を確認する. また、予測した負荷上昇率を超え、過負荷状態になった場合の挙動も実環境で評価する. シミュレーションでは、複数の負荷モデルを利用して、提案方式の省電力効果を評価する.

4.1 実環境での省電力性能評価

Hypervisor 型の仮想化ソフトウェアをインストールした物理サーバ 3 台により仮想化環境を構築した. 各物理サーバのスペックを表 1 に示す. 各サーバの消費電力は、停止状態 (Standby) で待機電力分の 7W/10W を消費する. 起動状態で負荷がない場合 (Idle) には 110W/180W, 最も負荷が高い場合 (Max) では 170W/270W の電力を消費する. Idle と Max の間では負荷に比例して消費電力が増加する.

表 1 に示す物理サーバ群にクラスター A, B を稼働させた. 両クラスターを構成する仮想サーバでは、リクエスト数に比例して負荷が上昇するようにリクエストに対しランダムな文字列を返却する処理を実行させた. このような単純な処理では、CPU 処理以外に、ログ出力とネットワークを介したリクエストの送受信が実施される. ただし、それぞれで扱うデータ量は少なく、CPU リソースが主に消費されることとなる. したがって、CPU 使用率がスケールイン/アウト契機を判断する負荷指標とした. また、クラスター A を構成する仮想サーバには 1 台あたり 1CPU コアと 1GB メモリ、クラスター B を構成する仮想サーバには 1 台あたり 2CPU コアと 2GB メモリを割り当てた.

各クラスターの負荷は増加から減少、減少から増加の両方が含まれるように 1 時間で図 4 のように変化する負荷と

した. なお、クラスター A は低負荷時に 2 台、中負荷時に 4 台、高負荷時に 6 台の仮想サーバリソースを必要とする. 一方、クラスター B は低負荷時に 1 台、高負荷時に 2 台の仮想サーバのリソースを必要とするよう設計した. なお、負荷分散器は負荷を各クラスターの仮想サーバ群に均等に割り振るものとする.

本評価では、CPU 使用率をスケールイン/アウトの契機を判定する負荷指標とし、過負荷状態を 80%, スケールアウト中に増加する CPU 使用率の最大量を 20% とした. これら設定と式 (1), (2), (3) から、CPU 使用率が 60% を超えるとスケールアウト、仮想サーバ台数が 2 台の場合に CPU 使用率が 30% を下回るとスケールインすることとした.

コンパクションを含む偏り維持制御 (Bias Control with Compaction) と制御なし (No Control) を実環境に適用した場合の消費電力の変化を図 5 に示す. 1 時間の総消費電力量は、コンパクションを含む偏り維持制御で 0.29kWh, 制御なしで 0.41kWh であり 29% の消費電力を削減できた. 両制御を比較すると、負荷の低い 1,000 秒までの期間と 1,700 秒から 2,500 秒の期間で 180W 程度の消費電力量の差がある. これらの期間中、コンパクションを含む偏り維持制御では表 1 の物理サーバ S3 の電源を停止させている.

高負荷中は両制御とも消費電力量に大差ないが、コンパクションを含む偏り維持制御では、物理サーバを起動し消費電力が増加した直後と物理サーバを停止させて消費電力が減少する直前に消費電力が増大している. これは物理サーバの起動/停止処理が原因であり、起動時は 50W 程度の増加が 110 秒間、停止時には 50W 程度の増加が 120 秒間続いた. 本評価中には 3 回の物理サーバの電源操作が発生し、その消費電力増加の総和は 4.8Wh であった. これは、総消費電力量 0.29kWh の 1.7% 程度であり、物理サー

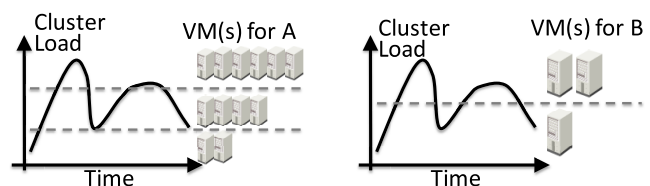


図 4 業務 A と B にかかる負荷
Fig. 4 Load for the cluster A and B.

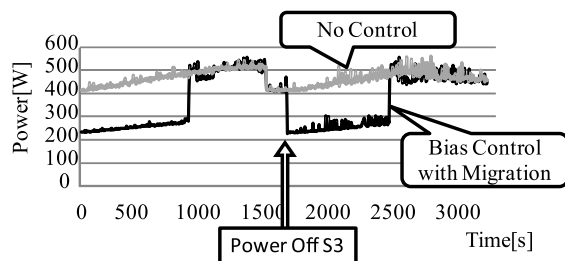


図 5 消費電力: コンパクションを含む偏り維持制御/制御なし
Fig. 5 Power consumption: Bias control with compaction/No control.

表 1 物理サーバスペック

Table 1 Physical server specification.

	Server1(S1)	Server2(S2)	Server3(S3)
CPU, Memory	4×2.7GHz, 4GB	4×2.7GHz, 4GB	2×2.8GHz, 2GB
Standby/Idle/Max Power	7W/110W/170W	7W/110W/170W	10W/180W/270W

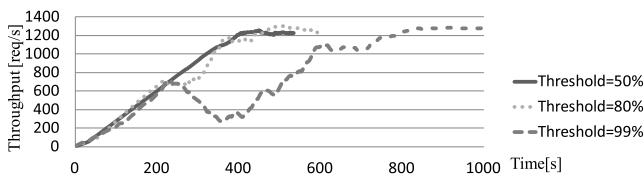


図 6 スケールアウト閾値設定とスループット変化

Fig. 6 Change of throughput according to threshold setting for scale-out.

バの電源操作が頻発しない限り、消費電力の削減量への影響は少ないと考えられる。

本評価では、リクエスト処理において主に消費されるリソースを CPU とした。この CPU 以外のディスク I/O やネットワーク I/O を主に消費する処理でも、負荷指標を適切に選択すれば同様の結果を得られると考えられる。また、リクエスト内容によって主に消費するリソースが動的に変わる場合もある。この場合、全リソースにおいて過負荷状態を防ぐため、全リソースを負荷指標とし、1つでもスケールアウトが必要と判定すればスケールアウトし、すべてでスケールイン可能と判定すればスケールインを実施する制御方式が考えられる。この制御では、スケールイン条件が厳しく、消費電力削減量は少なくなるが本評価と同様の変動傾向を持つ結果を得られると考えられる。

4.2 動的構成変更時のスループット性能評価

負荷の上昇率が予測範囲内であれば、設定したスケールアウト閾値を基準とし構成を動的に変更することで安定した応答性能を保つことができる。一方、負荷の上昇率はクライアントからのリクエスト頻度によって決まるため、閾値の設定を誤る場合がある。本節では、クラスタ構成システムにおいて、複数のスケールアウト閾値設定における応答性能をスループットの変動を測定することで評価する。

評価対象は、表 1 に示す Server1 に 1 台あたり 1CPU コアと 1GB メモリを割り当てた仮想サーバから構成されるクラスタである。このクラスタの最小構成を 2 台とし、スケールアウト閾値 (Threshold) を 50%, 80%, 99% と設定した。また、クライアントはレスポンス受信後に次のリクエストを送付するものとし、負荷の最大値は 1,200 リクエスト/秒、上昇率は 3 リクエスト/秒とした。評価結果が図 6 である。

閾値が 50% の場合は経過時間に応じてスループットが上昇し、400 秒で 1,200 リクエスト/秒に到達する。80% の場合はスループットが上昇しない期間がある。これは、スケールアウトが間に合わず、スループットが頭打ちになったためと考えられる。99% の場合は、スループットが頭打ちになってからスケールアウトするため、スループットが上昇しない期間が 80% の場合に比べ長い。さらに、スループットの低下も発生している。これは、過負荷状態において、リクエストを処理する複数のスレッド間でロック競

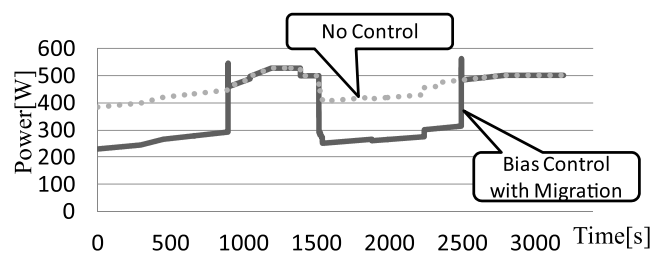


図 7 消費電力シミュレーション：コンパクションを含む偏り維持制御/制御なし

Fig. 7 Simulation of power consumption: Bias control with compaction/No control.

合やコンテキストスイッチが多発したためと考えられる。一方、すべての閾値においてスケールアウトは実施され、最終的にスループットは 1,200 リクエスト/秒に到達する。このように閾値設定を誤り、スケールアウトが遅れるとスループットの頭打ちや低下が発生する。これにより、クライアント側はリクエスト処理を待たされ、応答性能が低下したことになる。したがって、応答性能の保証を重要視するような IT システムでは、負荷上昇に対して早めにスケールアウトする閾値設定が必要だと考えられる。

4.3 シミュレーションによる省電力性能評価

本節では、提案手法の効果をシミュレーション評価した結果を述べる。まず、4.1 節の評価と同じサーバ環境と負荷モデルをシミュレーションした結果を図 7 に示す。

図 7 はコンパクションを含む偏り維持制御と省電力制御なしをシミュレーションした場合の消費電力の変化である。図 5 と同様に 1 時間の消費電力量の評価であり、コンパクションを含む偏り維持制御で 0.30 kWh、省電力制御なしで 0.41 kWh であり 27% の消費電力を削減できた。実環境とは 3% 程度の誤差であり、シミュレーションでも有用な結果が得られると考えられる。

4.1 節の評価で用いた負荷モデルのほかに省電力制御に関する複数の研究 [11], [12], [13], [14] では、World Cup 1998 の Web サイトへのアクセスログ (WC'98Web ログ) [15] を負荷モデルとして利用している。WC'98Web ログは、1 日のうち 12:00~18:00 に負荷が高く、それ以外の時間帯では負荷が低い傾向があるアクセスログである。このようにある特定時間帯に負荷が高くなる傾向は日本のブロードバンドサービス契約者のインターネットトラフィック量が 19:00~23:00 の間でピークを迎える傾向 [16] と同様に一般的に見られる。ここでは WC'98Web ログを複数組み合わせることで、ある特定時間帯にピークを迎える複数のクラスタ構成システムを対象として、本論文で提案する省電力制御の適用効果を評価した。具体的には、WC'98Web ログの 1998 年 5 月 1 日から 5 月 11 日の負荷情報を複数のクラスタへの負荷として使用した。これら 11 日分の負荷から重複を許して 3 種類組み合わせ、それぞれの負荷を 3 種

表 2 シミュレーションでの仮想サーバスペック

Table 2 Virtual server spec for simulation.

	VM for Cluster1	VM for Cluster2	VM for Cluster3
CPU Core, Memory	1, 1.0GB	1, 1.5GB	2, 1.5GB
Max Request	600/minute	600/minute	600/minute

表 3 シミュレーションでの物理サーバスペックと電力コスト

Table 3 Physical server spec and the power cost for simulation.

Server Spec	Power Consumption	Switch Cost	Compaction Cost
CPU Core: 4	Standby: 7W	On: 50W×110sec	70W×60sec
Memory: 4GB	Idle: 110W	Off: 50W×120sec	
The Number of Server: 7	Max: 220W		

類の IT システムを構成するクラスタ 1, 2, 3 に処理させた。3 種類のクラスタはそれぞれ表 2 で示す CPU コア数 (CPU Core), メモリ割当て (Memory) と限界処理性能 (Max Request) を持つ仮想サーバで構成されるものとした。また, リクエスト処理数と CPU 使用率は比例するとした。たとえば, 毎分 300 リクエストを処理する場合は最大処理性能である毎分 600 リクエストの半分である 50% の CPU 使用率となる。

スケールイン/アウトの契機となる閾値は 4.1 節の実環境と同じであり, 各仮想サーバの CPU 使用率が 60% を超えた場合にスケールアウト, CPU 使用率が 30% を下回った場合にスケールインすることとした。たとえば, クラスタ 1 へのリクエスト数が毎分 300 の場合は仮想サーバ 1 台で IT システムを稼働させる。リクエスト数が毎分 450 になれば, 仮想サーバ 1 台では CPU 使用率が 75% (= 450/600) になるので, 仮想サーバ 2 台にスケールアウトし各仮想サーバの CPU 使用率を 38% (450/1,200) に低減させる。

仮想サーバを稼働させる物理サーバの想定スペックと電力コストは表 3 に示すとおりである。ここでは, 4CPU, 4GB メモリの物理サーバが 7 台存在する環境を想定した。物理サーバの待機電力は 7W, アイドル状態の電力は 110W, 最大消費電力は 220W と設定した。また, 4.1 節の実験や他の省電力に関する研究 [12], [13] では, 物理サーバにおける CPU 使用率とアイドル状態からの消費電力増加率は比例関係にあるとしている。本シミュレーションでも同様の電力増加モデルを採用した。たとえば, 50% の CPU 使用率の物理サーバの消費電力は, 165W (110W + (220W - 110W) × 50%) とする。

仮想サーバを物理サーバ間で移動するコンパクションコストは, 使用するライブマイグレーション方式により異なる。本評価では 4.1 節の実環境で取得した情報から 70W の消費電力増加が 60 秒間続くと想定した。

表 2, 表 3 で示す環境において, 下記 5 つの制御方式による省電力性能を評価した。

- (1) コンパクションを含む偏り維持制御 (Bias Control with Compaction)
- (2) コンパクションを含む分散維持制御 (Non-Bias Control with Compaction)

- (3) コンパクションを含まない偏り維持制御 (Bias Control)
- (4) コンパクションを含まない分散維持制御 (Non-Bias Control)

- (5) 制御なし (No Control)

(2), (4) の分散維持とは, 偏り維持の効果を評価するために比較対象となる制御方式である。分散維持では, 仮想サーバの多い物理サーバからスケールイン, 仮想サーバの少ない稼働中の物理サーバを対象にスケールアウトを実施する。(3), (4) はコンパクションを含まず, 偏り維持によりコンパクションせずに得られる省電力効果を評価する。

図 8 に, WC'98Web ログの 2 日目, 6 日目, 10 日目の毎分のリクエスト数変動を示す。これらの負荷をそれぞれクラスタ 1, 2, 3 にかけた場合の (1)~(5) の制御方式における消費電力の変化を図 9 に示す。図 9 の左はコンパクションを含む偏り維持制御の消費電力量 ($P_{2,6,10}(1)$) と制御なしの消費電力量 ($P_{2,6,10}(5)$) であり, $P_{2,6,10}(1)$ は $P_{2,6,10}(5)$ に比べて 52.7% の消費電力を削減できている。図 9 の右はコンパクションを含む偏り維持制御の消費電力量とコンパクションを含む分散維持制御の消費電力量 ($P_{2,6,10}(2)$) であり, $P_{2,6,10}(1)$ は $P_{2,6,10}(2)$ に比べて 3.8% の消費電力を削減できている。また, コンパクション実施回数は $P_{2,6,10}(1)$ が 10 回で $P_{2,6,10}(2)$ が 129 回であり, コンパクション回数を 90% 以上削減できた。図 9 の真ん中はコンパクションを含まない偏り維持制御の消費電力量 ($P_{2,6,10}(3)$) とコンパクションを含まない分散維持制御の消費電力量 ($P_{2,6,10}(4)$) であり, $P_{2,6,10}(3)$ は $P_{2,6,10}(4)$ に比べて 7.1% の消費電力を削減できている。以上から, 図 8 に示す負荷に対しても偏り維持制御により省電力効果が得られることが分かる。

図 10 に WC'98Web ログの組合せに対する制御方式別の消費電力量を示す。図 10 では, クラスタ 1, 2, 3 それぞれに “App1” 日目, “App2” 日目, “App3” 日目の負荷をかけたことを (App1, App2, App3) で表現する。一番左の列が, クラスタ 1, 2, 3 すべてに 1 日目の負荷をかけた場合の消費電力量である。以降, クラスタ 3 に対する負荷を 2 日目, 3 日目と変更し, 11 日目の次は 1 日目に戻しクラスタ 2 の負荷を次の日 (2 日目) にした。なお, クラスタ 1 に関しても, クラスタ 2 の負荷が 11 日目から 1 日目に戻る時点で次の日とした。図 10 から消費電力量や削減率は変わるが, コンパクションを含む偏り維持制御が最も省電力効果が高いことが読み取れる。

表 4 に 11 日間の WC'98Web ログの全組合せ 1,331 通りでシミュレーションした場合の各制御方式における消費電力量, 物理サーバの電源操作回数, コンパクション回数の平均値を示す。(1) コンパクションを含む偏り維持制御は (5) 制御なしに比べて, 平均して 48.2% の消費電力量を削減できている。(1) コンパクションを含む偏り維持制御は (2) コンパクションを含む分散維持制御に比べて, 平均

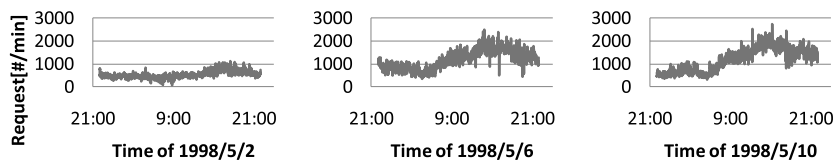


図 8 World Cup '98 Web サイトの 1 日間の負荷変動

Fig. 8 Load change of World Cup '98 Web site for 1 day.

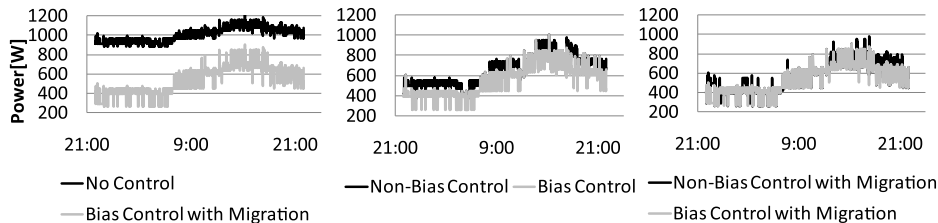


図 9 図 8 の負荷変動を持つクラスタを稼働させるシステムの消費電力変化

Fig. 9 Power consumption change of the clusters with Fig. 8's load.

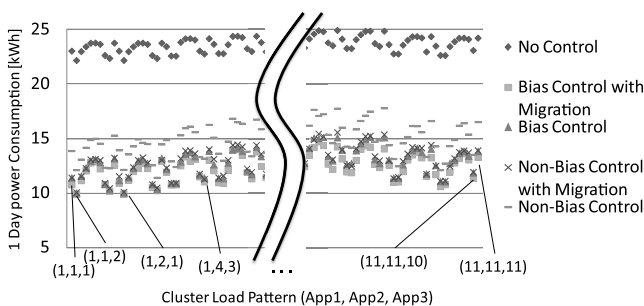


図 10 各負荷変動での 1 日の消費電力量

Fig. 10 Daily power consumption for application load patterns.

表 4 World Cup '98 Web サイトの負荷 (5/1~5/11) 組合せに対する省電力性能

Table 4 Power saving efficiency for World Cup '98 Web site's load combination.

	Power Consumption	Power on/off	Compaction
Bias Control with Compaction	12.2kWh	330	13
Non-Bias Control with Compaction	13.0kWh	230	139
Bias Control	12.6kWh	300	0
Non-Bias Control	14.7kWh	14	0
No Control	23.5kWh	0	0

して 6.2%の消費電力量を削減できている。また、コンパクション回数を 90.6%削減している。(3) コンパクションを含まない偏り維持制御は(4) コンパクションを含まない分散維持制御に比べて、14.3%の消費電力量を削減できている。以上から複数の負荷変動に対しても偏り維持制御により省電力効果が得られることが分かる。

消費電力削減率は物理サーバの電源効率にも依存する。ここまでのシミュレーションでは物理サーバの最大/最小消費電力を 220 W/110 W とすることで 48.2%の削減率を得た。しかし、270 W/110 W を想定すると平均削減率は 44%に減少する。一方、170 W/110 W を想定すると、平均削減率は 53.4%に上昇する。このように最大消費電力に比べて最小消費電力 (Idle 状態) の割合が少ないと、本論文で

提案する制御による省電力化の効果が低くなる。一方、最大消費電力に比べて最小消費電力の割合が少ないと、CPU での処理で消費する電力が相対的に大きくなるため電力効率は良いといえる。このように電力効率が良いと、本論文で提案する省電力制御の効果が相対的に低くなると考えられる。

物理サーバの電力効率を加味して、本論文で提案する偏り維持制御方式を評価する。ここでは、データセンタの電力効率を示す指標である PUE [17] にならない、物理サーバの電力効率を示す PUE4S (PUE for Server) を定義し、偏り維持制御を評価する。PUE はデータセンタ全体の消費電力をサーバの消費電力で割った値であり、1 が最も電力効率が良い。物理サーバの PUE4S は、物理サーバの最大消費電力を物理サーバの処理実行で増加する消費電力により割った値と定義する。たとえば、最大/最小消費電力が 270 W/110 W の物理サーバが最大消費電力を維持した場合、最小に比べて 160 W 増加し続けるので電力効率は、270 W を 160 W で割った 1.69 と計算される。この値は物理サーバの計算能力を最大限利用したときの理想値である。この最大消費電力を最小消費電力で割った値を、ここでは最大電力効率 (Max PUE for Server) と定義する。

以降では、物理サーバの電力効率にかかわらず提案する省電力制御が電力効率を上昇させるか評価する。そのために、最大消費電力を一定とし、最小消費電力を変化させることで異なる最大電力効率を持つ物理サーバ環境を想定し、各制御方式別の消費電力量の平均値をシミュレーションにより取得した。その結果を図 11 に示す。図 11 では、(1) コンパクションを含む偏り維持制御 (Bias Control with Compaction), (2) コンパクションを含む分散維持制御 (Non-Bias Control with Compaction), (5) 制御なし (No Control) の消費電力量を比較している。図 11 に示すとおり、物理サーバの最大電力効率が大きくなっても、制

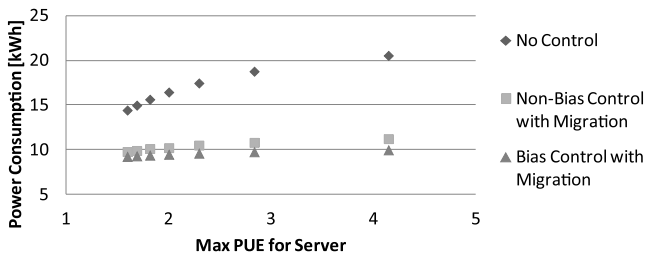


図 11 省電力制御実行時のサーバに対する消費電力効率
 Fig. 11 PUE for server with power-saving control.

御なしの場合に比べて制御がある場合の消費電力量が少なくなっている。また、電力効率にかかわらずコンパクションを含む偏り維持制御がコンパクションを含む分散維持制御より消費電力量が低く、提案手法の優位性を確認できる。

5. 関連研究

仮想化環境に構築した複数の IT システムを動的に構成変更することで消費電力を削減する技術に関して、様々な研究がなされている。

Imada らは、複数の物理サーバからなる IT システムにおいて、処理を集中させることで一部の物理サーバの負荷を高くし、不要な物理サーバの電源を停止させることで IT システム全体の消費電力量を削減する手法を提案している [5]。サーバ装置の消費電力量を削減する技術としては、DVFS [2] により CPU の消費電力を削減する手法が提案されているが、IT システム全体を停止させることでメモリ、I/O デバイスなど CPU 以外の電力の削減が期待できる。また、仮想サーバの移動をライブマイグレーションと、スケールイン/アウトの組合せとで行う場合に発生する移動負荷についても報告している。スケールイン/アウトの組合せによる仮想サーバ移動方式では、移動対象の仮想サーバが構成するクラスタを移動先にスケールアウトし、移動元となる仮想サーバを停止させてスケールインする。この方式は、クラスタを構成するどの仮想サーバでクライアントからのリクエストを処理しても同じ結果となる、いわゆるステートレスな IT システムで実現できる。このスケールイン/アウトの組合せでも、仮想サーバの起動や停止処理の負荷により、消費電力量が増加するが、ライブマイグレーションと比べると消費電力量を低減できると報告している。

Cardosa らは、仮想化環境を前提とし仮想サーバを最小の物理サーバ数で稼働するように配置する最適化問題の解法を提案した [3]。この方式では、必要リソース量の大きい仮想サーバから配置するファーストフィット法に加え、リソース使用効率を向上させる仮想サーバ配置入れ替えを実施する。この仮想サーバ配置入れ替えでは、特定の物理サーバに配置済みの仮想サーバ群のうち必要リソース量が最小の仮想サーバをより大きなリソース量を必要とする仮

想サーバと入れ替える。この入れ替えにより、物理サーバのリソース使用効率を向上させる。各物理サーバのリソース使用効率が増えれば、必要な物理サーバ数が低減し、消費電力を削減できる。シミュレーションによる評価では、ファーストフィット法のみを適用する場合に比べ、仮想サーバを稼働させるのに必要な物理サーバ数を削減できている。ただし、実際のコンパクションでは、仮想サーバを移動する時間や消費電力といった負荷が存在するが、Cardosa らの手法では考慮されていない。したがって、消費電力を効果的に削減するためには、仮想サーバの移動にかかる負荷低減を考慮した最適配置導出が課題となる。

Hermenier らは、コンパクションによる仮想サーバの最適配置化をビンパッキング問題 [10] に定式化して解いた後、移動効率を向上させる仮想サーバ移動方式を提案している [9]。ビンパッキング問題は、NP 困難な問題であり、物理サーバに配置する仮想サーバ群が増加すると最適配置を得ることが容易でなくなる。この問題に対して Hermenier らは、必要リソース量が同じ仮想サーバどうしを入れ替えても各物理サーバのリソース使用率が変わらないという特徴に着目している。この特徴を利用することで、組合せ量を削減し計算時間を短縮している。最適配置計算後の実際の移動においては、できる限り多くの仮想サーバ移動を並行して実施させることで移動時間を低減させる。以上の方式により、ファーストフィット法を使う場合に比べ、集約率の向上と仮想サーバ移動時間の削減を実現する。ただし、仮想サーバの移動回数は削減できていないため、移動にかかる負荷により消費電力が増加する。つまり、仮想サーバの移動回数を低減させる最適配置が課題である。

Kusic らは、仮想サーバのスケールイン/アウトとそれにとまなう物理サーバの停止/起動により省電力化する方式を提案している [12]。具体的には、現在の負荷から将来の負荷を予測し、それに応じて仮想サーバで構築したクラスタ構成の IT システムをスケールイン/アウトさせる。これにより、仮想サーバが存在しなくなった物理サーバを停止する。提案方式では、数分後の負荷をカルマンフィルタにより予測している。しかし、将来の負荷を予測する場合、時間が離れるにつれて予測精度が低下する。さらに、予測値より大きな負荷が発生すると、SLA (Service Level Agreement) 違反を引き起こす。Kusic らの方式では SLA 違反の発生回数を低減するために IT システムに割り当てるリソース量を増加させ負荷が予測値より大きくても過負荷状態にならずに処理できるための余裕を持たせる方式をとっている。しかしながら、リソース量を余分に与えると省電力効果が低下する。そこで、Kusic らは余分なリソース量をどの程度与えれば SLA 違反発生回数を低減させ、かつ省電力効果も得られるかを評価している。特定の負荷モデルに対する評価ではあるが、予測する負荷変化率の 2 倍のリソースまでは省電力効果を低下させずに SLA 違反発

生回数を低減できることを示している。

これら従来の研究では、コンパクション時の仮想サーバ移動回数の低減と消費電力量の削減との両方を目的として、ITシステムを構成するクラスタのスケールイン/アウト対象を選択することは検討されていない。本論文では、仮想サーバ配置の偏りを維持するスケールイン/アウトにより従来研究で課題となっていたコンパクション時の移動回数と移動にともなう負荷の低減を実現する省電力制御方式を提案した。

6. おわりに

本論文では、仮想化環境における複数クラスタの構成を動的に変更することで消費電力量を削減することを目的とした。従来はクラスタをスケールイン/アウトする方式と仮想サーバを物理サーバ間で移動してコンパクションする方式があった。本論文では2つの方式を組み合わせ、コンパクション時の仮想サーバ移動負荷を低減させるようにスケールイン/アウト対象を選択する省電力制御方式を提案した。このコンパクションを含む偏り維持制御をWC'98Webログから生成した負荷モデルを処理する3つのクラスタに適用すると、制御がない場合に比べて1日あたり48.2%の消費電力を削減できた。また、偏りを維持しない分散維持制御に比べて、コンパクション回数を1日あたり90%削減し、6.2%の消費電力削減率の向上を得ることができた。また、物理サーバの電力効率として、データセンタの電力効率の指標であるPUEにならぬ物理サーバ全体の消費電力をCPU処理などの実際の負荷で上昇した消費電力で割ったPUE4Sを定義した。この指標を用いて、様々な電力効率を持つ物理サーバ群を対象としても、コンパクションを含む偏り維持制御により消費電力量を削減できることと、偏りを維持しない分散維持制御に比べてより多くの消費電力量を削減できることを確認した。

今後の課題としては、WC'98Webログに見られる傾向とは異なる負荷傾向を持つクラスタ構成のITシステムにおける省電力性能評価があげられる。たとえば、日本のブロードバンドサービス契約者のアクセスが集中する時間帯と、日本の企業からのアクセスが集中する時間帯は異なると考えられる。また、世界中からアクセスされるITシステムが集まったデータセンタでは、各国の時差や生活習慣によりアクセスが集中する時間帯が異なるITシステムが存在する。このようなデータセンタを対象とした評価を行うために、本論文で用いたWC'98Webログの時間をずらして複数組み合わせた負荷モデルを作成しての評価が考えられる。

参考文献

[1] Sharma, R.K., Shih, R., Bash, C., Patel, C., Varghese, P., Mekanapurath, M., Velayudhan, S. and Manu, K.V.:

- On Building Next Generation Data Centers: Energy Flow in the Information Technology Stack, *Proc. 2008 ACM Compute Conference*, Bangalore, India (2008).
- [2] Mark, B.: データセンターのサーバ電力消費量を削減する5つの方法 (online), 入手先
(http://members.thegreengrid.org/japanese/gg-content/J.White_Paper_7-Five_Ways_Save_Power.pdf) (参照 2011-11-02).
- [3] Cardoso, M., Korupolu, M.R. and Singh, A.: Shares and Utilities Based Power Consolidation in Virtualized Server Environments, *Proc. IFIP/IEEE International Symposium on Integrated Network Management*, USA (2009).
- [4] Xiaobo, F., Wolf-Dietrich, W. and Luiz, A.B.: Power Provisioning for a Warehouse-sized Computer, *Proc. ACM International Symposium on Computer Architecture*, San Diego, CA (2007).
- [5] Imada, T., Sato, M. and Kimura, H.: Power and QoS Performance Characteristics of Virtualized Servers, *Proc. Energy Efficient Grids, Clouds and Clusters Workshop in the 10th IEEE GRID 2009*, pp.232–240 (2009).
- [6] Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I. and Wareld, A.: Live Migration of Virtual Machines, *Proc. 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation*, pp.273–286 (2005).
- [7] Garbacki, P. and Naik, V.K.: Efficient Resource Virtualization and Sharing Strategies for Heterogeneous Grid Environments, *Proc. 10th IFIP/IEEE International Symposium on Integrated Network Management*, pp.40–49 (2007).
- [8] Verma, A., Ahuja, P. and Neogi, A.: pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems, *Proc. ACM 9th International Middleware Conference* (2008).
- [9] Hermenier, F., Lorca, X., Menaud, J.-M., Muller, G. and Lawall, J.: Entropy: a Consolidation Manager for Clusters, *Proc. ACM/USENIX International Conference on Virtual Execution Environments*, pp.41–50 (2009).
- [10] Shaw, P.: A Constraint for Bin Packing, *Proc. Principles and Practice of Constraint Programming (CP 2004)*, Lecture Notes in Computer Science, Vol.3258, pp.648–662, Springer (2004).
- [11] Chase, J., Anderson, D., Thakar, P., Vahdat, A. and Doyle, R.: Managing Energy and Server Resources in Hosting Centers, *Proc. 18th Symposium on Operating Systems Principles*, pp.103–116 (2001).
- [12] Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N. and Jiang, G.: Power and Performance Management of Virtualized Computing Environments via Lookahead Control, *Proc. 5th ICAC International Conference on Autonomic Computing*, Washington DC, USA, pp.3–12 (2008).
- [13] Petrucci, V., Loques, O. and Mosse, D.: A Dynamic Optimization Model for Power and Performance Management of Virtualized Clusters, *Proc. 1st International Conference on Energy-Efficient Computing and Networking* (2010).
- [14] Santana, C. and Mosse, D.: Load Forecasting Applied to Soft Real-Time Web Clusters, *Proc. 2010 ACM Symposium on Applied Computing* (2010).
- [15] Arlitt, M. and Jin, T.: Workload Characterization of the 1998 World Cup Web Site, Hewlett-Packard Labs, Technical Report HPL-99-35R1 (Sep. 1999).

- [16] 総務省：我が国のインターネットにおけるトラフィック総量の把握 (online), 入手先 (http://www.soumu.go.jp/main_content/000109282.pdf) (参照 2011-11-02).
- [17] Belady, C.: The Green Grid Data Center Power Efficiency Metrics: PUE and DCiE (online), available from (http://www.thegreengrid.org/~media/WhitePapers/White_Paper_6_-_PUE_and_DCiE_Eff_Metrics_30_December_2008.pdf) (accessed 2011-06-17).



宮田 康志

2005年京都大学工学部電気電子工学科卒業。2007年同大学大学院工学研究科修士課程修了。現在、(株)日立製作所横浜研究所にてミドルウェア分野の研究に従事。



太田 智也

1998年静岡大学大学院電子科学研究科博士課程修了。博士(工学)。現在、(株)日立製作所横浜研究所主任研究員。Java実行環境等の言語処理系の研究に従事。



天野 雅志

1991年千葉大学大学院工学研究科修士課程修了。現在、(株)日立製作所横浜研究所にてミドルウェア分野の研究に従事。



西山 博泰 (正会員)

1993年筑波大学大学院工学研究科博士課程修了。工学博士。(株)日立製作所横浜研究所主管研究員。最適化コンパイラ, Java実行環境等言語処理系の研究に従事。ACM, IEEE各会員。