Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

仮想デスクトップ配置アルゴリズムに関する検討

カオ　レタンマン†　萱島 信†

仮想マシンを複数の仮想化サーバ上に最適に配置する問題は，NP 困難のビンパッキング問題に相当する．本研究は，仮想マシンをクライアント PC として利用する場合を対象として，同配置問題への有効な近似解を提案する．具体的には，各仮想デスクトップ（VD）の資源使用量は，一定周期でほぼ同一パターンの変化を繰り返す特性を利用して，各パターンの相関を考慮し，相関が低い VD 同士を同じサーバへ配置するアルゴリズムを検討した．提案アルゴリズムにより，同一サーバ上により多くの VD を搭載しても互いに資源を不足させず，従来アルゴリズムと比べてサーバ台数を削減しながら，資源不足による VD の再配置回数を抑えられる．

A Study on Virtual Desktop Allocation Algorithms

Cao Le Thanh Man †　and Makoto KAYASHIMA †

Finding optimal placement of virtual machines on numerous physical servers is one kind of bin-packing problem, which is known as a NP-hard. We propose a heuristic algorithm for this problem, in the particular case where the virtual machines play the roles of personal desktops. As the changes in the amount of hardware resources required by a desktop often repeat a certain pattern, the proposed algorithm considers the correlation between the patterns to find the group of desktops that most suitable for sharing the same physical server. Simulation results show that, comparing to existing ones, the proposed algorithm can reduce the number of physical servers required for hosting a certain number of virtual desktops.

1. はじめに

近年，情報システムの管理，運用作業を簡単化するプラットフォーム仮想化技術 Xen[1]，KVM[2] ，VMware[3]が注目されている．同技術により，計算機物理資源を複数の論理資源に見せかけることができ，一台のサーバ装置上に複数の仮想マシンを稼動させることができるようになる．また，仮想マシンはサーバ装置のハードウェア構成に依存しないため，仮想マシンを他のサーバ装置に移動して動作させることが容易である．プラットフォーム仮想化技術は Web サーバやデータベースサーバなど，サーバの役割を果たす計算機に利用するケースが多かった．

プラットフォーム仮想化技術の新しい用途として，デスクトップ用 OS を仮想マシン化し，利用者が遠隔操作で利用するという，デスクトップ仮想化技術も普及し始めている．従来のオフィス端末システムは，汎用の PC（Personal Computer）を用いて構成されたものがほとんどであり, PC 内でデータを処理すると共にローカルディスクにデータを保存することが一般的であった. このため, 多数の PC を所有する企業では，ハードウェアの障害対応，更新の手間やローカルハードディスクに保存したデータの管理の難しさが課題となっている．デスクトップ仮想化技術を利用すると，オフィスには仮想デスクトップ（Virtual Desktop, VD）の画面を表示する機能のみを有する端末（シンクライアント端末）を設置するだけでよく，オフィスにおける管理を簡単化できる．

プラットフォーム仮想化により，仮想マシンとハードウェアは疎結合化されるため，仮想マシンを任意のサーバ上で動作させることが可能になる．また，あるサーバ上で動作中の仮想マシンを別のサーバへ移動することも可能である．そのため，複数の仮想サーバからなる仮想化基盤において，仮想マシンをサーバに配置することは柔軟にでき，適切に配置することにより，サーバ資源の利用効果を大幅に向上させることが期待できる. 例えば, 電源の節約効果は, 最大で 50%節約できることも期待できる[4]. そのため，仮想化基盤向けの最適仮想マシン配置問題が多数の研究の対象となっていた．

仮想マシンを複数の仮想化サーバに最適に配置する問題は，NP 困難のビンパッキング問題に相当する．さらに，仮想デスクトップの場合，デスクトップの利用者の操作により，デスクトップが消費する資源量（CPU 処理時間，メモリ量，HDD アクセス帯域，ネットワーク帯域など）が短い間隔で大きく変化するため，同最適配置問題は一層難しくなる．

本研究では，各仮想デスクトップが常に快適なパフォーマンスを実現できるための資源を確保しながら，コストの削減やエネルギー節約のためのサーバ装置台数削減を

† 日立製作所 システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.

ⓒ 2011 Information Processing Society of Japan

Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

図る仮想化デスクトップ配置アルゴリズムを検討する．一般オフィスユーザのデスクトップの資源使用量がある周期で一定のパターンを繰り返す特徴を利用し，資源利用パターンの相関が少ないデスクトップ同士をできるだけ同じサーバへ配置するアルゴリズムを提案する．提案アルゴリズムにより，サーバの仮想デスクトップ集積度を上げながらデスクトップ同士の資源競争の発生可能性を抑えることができる．

本論文の構成は次となっている．第２章では，関連研究について述べる．第３章では，仮想デスクトップの特性について説明する．次に，第４章および５章では提案する仮想デスクトップ配置アルゴリズムおよびその評価結果について述べる．第６章では結論及び今後の課題について述べる．

2. 関連研究

仮想化サーバへ仮想マシン配置に関する研究が盛んに行われている．[5]によると，同配置問題を次の二つの内容を含んでいる．一つ目は，新規仮想マシンが生成・導入される時の仮想化サーバ装置への割り当てる問題であり，もうひとつは，稼働中の仮想マシンを別の仮想化サーバ装置間へ再割当て問題である．前者を中心に検討した論文は，各仮想マシンの資源要求とサーバ装置の資源容量を考慮し，配置を行う[7-9]．後者のアルゴリズムは，仮想マシンが稼働している最中に無停止でサーバ間移動できる機能（Live migration [10]）を活用してサーバ上の仮想マシンを再配置する．基本的には，サーバの資源が過不足した場合，そのサーバ上の仮想マシンを転出または転入する．再配置アルゴリズムは，資源変動を発見するアルゴリズムと，それに基づいて対応方法によって異なる．後者の問題では，仮想マシンの再配置回数の削減などの工夫が必要である[11]．なお，配置の基準としては，仮想マシンの CPU の計算能力[9]，電源消費量[5]または複数の指標[8]が用いられる．

前者の仮想化基盤全体の配置問題は，ビンパッキング問題という組み合わせ最適化問題とみなされ，さまざまな近似解がある．例えば，[11]は Best Fit または First Fit の一種のアルゴリズムを提案している．また，電源の利用を削減するために，Best Fit を改善したアルゴリズムが提案された[5]．予測結果をベースに，仮想マシンの予測資源使用量を使った First Fit アルゴリズム[9]もあった．さらに，[8]は First Fit, Next Fit, Best Fit を使った配置結果の比較を行った．

既存の全体配置記アルゴリズムでは，仮想マシンの資源利用量の予測値として，平均値など，ひとつの値を利用することがほとんどである．この表し方は，仮想マシンの資源量の変動が時間的に激しくない場合に有効である．しかし，仮想デスクトップの場合は，時間的変動が多いため，ひとつの値で表すことが効率的ではない．

上記の関連研究と違って，本研究は，仮想マシンをクライアント PC として利用する場合を対象として，仮想マシン配置問題を検討する．仮想デスクトップの場合，一定周期でほぼ同一パターンの変化を繰り返す特性があるため，その資源使用量をパターンで表すことで，サーバの資源をより有効に活用するに配置方法を提案している．

3. 仮想デスクトップの特性

仮想デスクトップが動作するために必要なハードウェア資源は, HDD アクセス帯域, CPU 処理能力，メモリ容量，ネットワーク帯域である．その中で，ネットワーク帯域に関しては，一般オフィスユーザの仮想デスクトップの場合は仮想サーバの NIC（通常は高速の 1Gbps または 10Gbps の NIC を利用する）がボトルネックになることが少ない. 一方, メモリ量が不足の場合は仮想デスクトップの性能が顕著に低下するため，本研究では仮想デスクトップ基盤が各仮想デスクトップの OS が正常に動作するためのメモリ量（通常 1GB から 2GB 程度の目盛りを必要とする）を搭載していることを前提として配置問題を検討する．十分なメモリ容量がある場合，一般オフィスユーザのデスクトップのパフォーマンスに影響する資源は，HDD アクセス帯域および CPU 負荷となる. HDD 性能の問題に関しては, HDD を SSD へ換装することで解決できる．つまり，SSD を導入したサーバにおいては，CPU 負荷が仮想デスクトップの性能低下の要因となる可能性がもっとも高くなる．そのため，本研究では，CPU 処理時間の最適化を対象に仮想デスクトップの配置問題を検討する．

まず，デスクトップの CPU 負荷について検討を行った．12 名のオフィスユーザが画面転送ソフトウェアで遠隔利用しているデスクトップのプロセッサがアイドル以外のスレッドを実行する時間のパーセンテージを示すパフォーマンスログを収集した．ログの収集期間は一週間である．収集期間中，ログ収集対象デスクトップのユーザは通常通り作業を行った．

収集したデータの中で, 各デスクトップの CPU 利用率の２時間ごとの平均値を時系列として, その自己相関性を調査した．図 1 では次数を 0 から 84 まで変化させた時の自己相関係数の結果を示す．同図により，24 時間ごとに各デスクトップのログが高い自己相関を示している．この結果から, 24 時間（1 日）ごとに，各デスクトップの CPU 利用率の変化が同一のパターンを繰り返すことが明らかになった. なお, 図 1 の場合，この周期の長さが 1 日である．別の職場になると，仕事の特性により，この CPU 利用率の変化周期は一日ではなく，一週間や一カ月など，パターンの長さが異なることも考えられる．

各デスクトップの CPU 利用パターンの例を図 2 に示す. 各パターンの最大値および時間により変化速度がそれぞれ異なることが分かる．これはデスクトップのユーザの仕事の習慣や利用するするソフトウェア，出張状況などにより異なるからである．

ⓒ 2011 Information Processing Society of Japan

Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

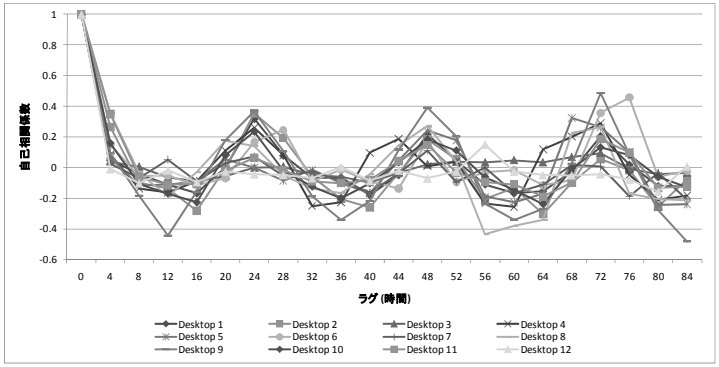

図 1　CPU 利用率のコレログラム

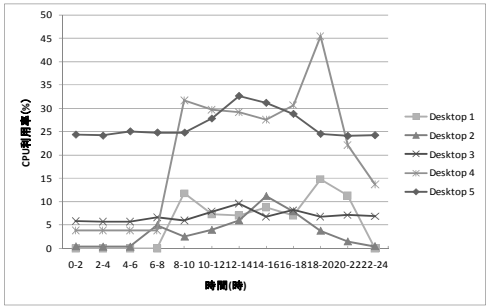

図 2　CPU 利用パターンの例

4.　仮想デスクトップの配置アルゴリズム

本章では, 各仮想化サーバ装置の CPU の資源を最適に活用するための仮想デスクトップ配置アルゴリズムについて提案する.

4.1　仮想マシン配置の概要

仮想マシン配置問題は，一般的に，初期導入時に全体の仮想サーバへ仮想デスクトップを配置する，いわゆる“全体配置”，と，仮想デスクトップ運用中の SLA（Service Level Agreement）条件を満たさない個所に対する“再配置”がある．本研究では，前者の全体配置に関する問題を対象にする．

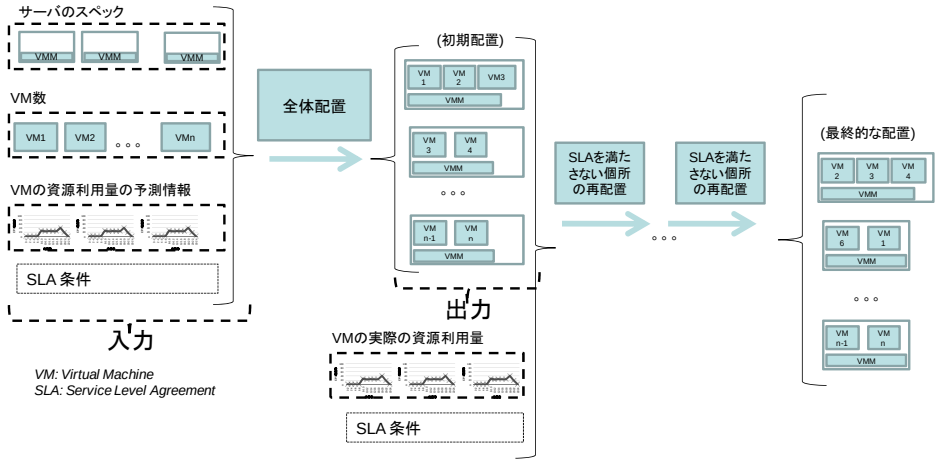

図 3　仮想マシン配置アルゴリズムの概要

図 3 では仮想マシン配置アルゴリズムの入力・出力を示す．仮想マシンの全体配置アルゴリズムは，サーバ装置の資源容量，仮想化基盤へ配置すべき仮想マシン数およびそれぞれの仮想マシンの資源使用量の予測値，SLA 条件を入力とする．全体配置の出力は, 各サーバ上の仮想マシンの初期的配置図である. この初期的配置図に従って，仮想マシンを各サーバに配置し，動作させる．動作中の仮想マシンおよびサーバの資源は，SLA 条件を満たしているかどうかを周期的に確認する．SLA 条件を満たさないサーバに対し，サーバ上の仮想マシンを別のサーバへ移動し，SLA 条件違反をなくすための再配置を行う．再配置を繰り返し実行した結果，システム停止時の配置が全体配置の出力と異なる場合もある．システム停止時の最終的な配置におけるサーバ台数が，実際に必要なサーバ数となる．

コスト削減のため，最終的な配置におけるサーバ台数が少ないことが好ましい．また，再配置に伴い，システム内のトラヒック量が増加するため，再配置仮想マシン数は少ない方が好ましい．しかし，仮想マシンを複数の仮想化サーバに最適に配置する問題は，NP 困難のビンパッキング問題に相当するため，最適解を求めることが難しい．

4.2　提案仮想デスクトップ配置アルゴリズム

本研究は，仮想マシンをクライアント PC として利用する場合を対象として，全体配置問題への有効な近似解を提案する．第 2 章で述べたように，各仮想デスクトップ（Virtual Desktop, VD）の資源使用量が一定周期でほぼ同一パターンの変化を繰り返

ⓒ 2011 Information Processing Society of Japan

Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

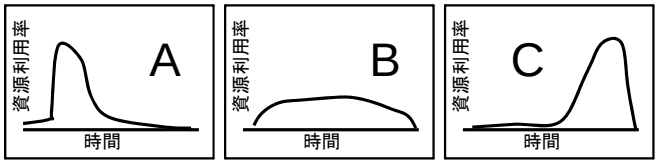

図 4　デスクトップの資源利用パターンの例

す特性を利用して，各パターンの相関を考慮し，相関が低い VD 同士を同じサーバへ配置するアルゴリズムを提案する．例えば，図 4 に示すような 3 つの VD の CPU 使用量の変化パターンがある場合，A と C を組み合わせた方が資源の競合が発生しにくく，A と B は資源の競合が発生する．つまり，相関が少ない利用パターンを持つ VD 同士を同じサーバに配置した方が，同一サーバ上により多くの VD を搭載しても互いに資源の競合を起こさせずに動作可能となる．

本提案の配置アルゴリズムを PBA（Pattern-based Virtual Desktop Allocation）と名付ける．PBA は各 VD の CPU 使用量のパターンを配慮して VD を配置する．その基本動作は以下の通りである．PBA は一台のサーバごとに，SLA 条件を満たす範囲内でできるだけ多くの仮想デスクトップを配置し，すべての仮想デスクトップが配置されるまでサーバ台数を増やしていく．一台のサーバに VD を配置する作業は以下のようになる．まず，未配置 VD の中からもっとも CPU 利用量の平均が大きい VD をサーバに入れる．次に，サーバに配置した VD と相関がもっとも小さい VD を順に選んでサーバに追加する．当該サーバが SLA 条件を満たす限り，VD を追加する．このような配置アルゴリズムをとることにより，同じサーバ上に配置される VD の資源利用パターンの相関が低く，資源競合の可能性を低くすることができる．以下に，提案アルゴリズムの入力・出力の定義および詳細な動作を示す．

PBA の入力は次のとなる．仮想化基盤全体で配置すべき VD の集合を V とする．

$$V = \{VD_1, VD_2, \ldots VD_N\}$$

ここで， N は収容すべき仮想デスクトップの個数となっている．仮想化基盤の仮想化サーバ装置の集合を W とする．仮想化サーバの台数を M とし，M は，N 台の VD を格納するために十分な数であると仮定する．

$$W = \{SV_1, SV_2, \ldots SV_M\}$$

W 内の各サーバSV_jに対して，その資源容量として CPU のコア数をU_jとする．各サーバの CPU コアが異なる場合は，あるサーバの CPU のコアを基準として，残りの CPU をその処理能力に基づいて基準コアに換算し，換算結果によりU_jを決定する．また，V 内の各デスクトップVD_iに対して，資源消費量の変化パターンを$F_i(t)$とする．$F_i(t)$ はVD_iが W 内のサーバの CPU の一つのコアを利用した場合の使用率（%）である．ここで，t□[0..T]，T はパターンの長さとする．また，$F_i(t)$の平均をM_iとする．SLA 条件を，ある VD の集合を特定のサーバへ搭載することが可能かどうかを決めるための基準である．ここで，SLA 条件を表すための関数 SLA()を定義する．

$$SLA(SV_i, S) = \begin{cases} \text{True} & (SV_i\text{に S を配置しても SLA 条件違反とならない}) \\ \text{False} & (SV_i\text{に S を配置すると SLA 条件違反となる}) \end{cases}$$

ここで，S は VD の集合である．

PBA の出力は V 内の VD のすべてがそれぞれどのサーバに配置されるかを表す次の配列である．

$$x_{ij} = \begin{cases} 1 & (SV_i\text{に}VD_i\text{を配置する}) \\ 0 & (SV_i\text{に}VD_i\text{を配置しない}) \end{cases}$$

ここで，$i = 1..N, j = 1..M$である．

PBA は，次の関数を利用する．

Max_average_index(S): VD の集合 S の中で，平均が最大の VD を返す．

Min_correlation_index(F(t),S) : VD の集合 S の中で，その資源利用パターンと F との相関係数が最小となる VD を返す．

PBA アルゴリズムの詳細動作は次の 7 ステップである．

1. すべての VD が未配置の状態にする．

$$x_{ij} = 0$$

2. 最初のサーバを選択する．

$$j = 1$$

3. サーバSV_jに最初の VD を配置する．

サーバSV_jに最初に配置される VD は，未配置 VD の中で最も CPU 利用率の平均が大きい VD とする．

$$VD_q = \text{Max\_average\_index}\ (\{VD_w | \textstyle\prod_{v=1}^{M}(1 - x_{wv}) = 1\})$$

とすると，サーバSV_jにVD_qを配置する．

$$x_{qj} = 1$$

ⓒ 2011 Information Processing Society of Japan

Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

ここで, 各サーバには SLA 条件違反せずに最低でも一つの VD を格納できると仮定する．一台の VD でも格納できないサーバがある場合，このサーバを仮想化基盤として利用するメリットが少ないからである．

4. サーバSV_jに 2 台目以降の VD を配置する．

まず，未配置 VD の中から，サーバSV_jにさらに配置しもよい VD の集合 R を作る．

$$R = \{\{VD_w|\prod_{v=1}^{M}(1-x_{wv})=1\} \cap \{VD_w|SLA(SV_j, L \cap \{VD_w\}) = True\}\ \}$$

ここで，L はサーバSV_jにこれまで配置された VD の集合である．

$$L = \{VD_w|\ x_{wj}=1\}$$

R が空集合ならば, サーバSV_jにさらに追加できる VD がない場合, ステップ 6 へ進む．

5. 次に，R の中から，サーバSV_jにこれまで配置された VD のパターンの総和と相関が一番小さいVD_qを特定する．

$$VD_q = \text{Min\_correlation\_index}(\sum_{w|x_{wj}=1} F_i(t), R)$$

VD_qをサーバSV_jに配置する．

$$x_{qj} = 1$$

6. 未配置 VD があれば配置を続ける．

$Q = \{VD_w,\ \prod_{v=1}^{M}(1-x_{wv})\}$が空集合なら終了する．そうでなければステップ 7 へ移動する．

7. 次のサーバに VD を配置する．

$$j = j + 1$$

ステップ 3 へ移動する．

上記の各ステップを実行すると，配置に用いるサーバ台数 j が一つずつ増加していく．ステップ 5 において，Q が空集合なら，配置すべき VD をすべて配置したので，配置作業を終了する．配置に必要となるサーバ台数は終了時の j の値となる．また，

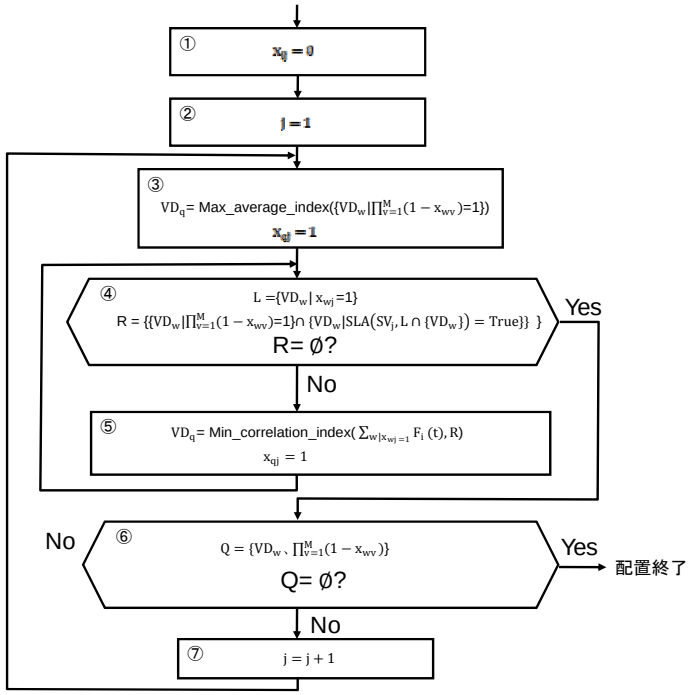

図 5　提案アルゴリズムのフローチャート

配置結果は x_{ij}の値となる．図 5 に上記の各ステップのフローチャートを示す．

従来の First Fit アルゴリズムと比べて，PBA はステップ 5 において，相関係数による VD 選択作業（計算量：0(N)）が新しく加わる．初期配置はシステム稼働する前に一回だけ行う作業であり，リアルタイム性に関する要求が高くないため，この程度の計算量が増えることによる影響が少ないと考える．

PBA は N 台の VD を M 台のサーバへの最適に配置する問題の最適解を求められないが，サーバごとに配置する VD はできるだけ互いに資源利用パターンが異なるものとする．そのため, SLA 条件を満たす範囲で多くの VD をサーバに搭載できる．また，利用するサーバの台数を最小限に抑える効果もある．特に，各 VD が資源利用パターンのピックになる時間帯が異なる VD が複数存在する環境では大きな効果があると期待できる．

ⓒ 2011 Information Processing Society of Japan

Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

表 1　比較対象の仮想デスクトップ配置アルゴリズム

#	全体配置	再配置
1	FF-ave	SLA 違反時別サーバへ仮想デスクトップを移動する
2	FF-max	
3	PBA（提案アルゴリズム）	

5. 評価

本章では，4 章で提案したアルゴリズムの性能を評価するためのシミュレーション結果を示す．本シミュレーションは，独自に作成したプログラムによるものである．

5.1 シミュレーションの設定

表 1 に示す 3 種類の全体配置方法について性能評価を行う．FF-ave は，仮想デスクトップの平均 CPU 利用率を使った First-Fit[12] 配置アルゴリズムである．FF-max は，仮想デスクトップの CPU 利用率の最大値を使った First-Fit 配置アルゴリズムである．FF-max に近いアルゴリズムは[9]で提案されている．一方，PBA は，3 章で説明した提案アルゴリズムである．

本評価の対象は全体配置であるため，再配置アルゴリズムに関しては，もっとも簡単なアルゴリズムを利用する．SLA 違反となるサーバにおいては，仮想デスクトップを別のサーバへ移動するが，移動先のサーバ上には FF-ave アルゴリズムによる配置が行われる．

以下のシミュレーションでは，SLA 条件は，各仮想化サーバ装置では，“2 時間の平均 CPU 利用率が 85%を超えない”とする．また，サーバのコア数は

$$U_j = 2, j = 1..M$$

とする．仮想デスクトップの CPU 利用パターンはパターンの特徴を簡単に表すために式（1）に示す Gaussian 分布に従う仮定する．式（1）のa_iは当該デスクトップの CPU 利用率の最大値であり，b_iは当該デスクトップがもっとも CPU を利用する時間を表し，c_iは当該デスクトップが集中して CPU を利用する時間帯の長さを表すパラメータである．図 6 は，VD のパターンの例である．実際の VD の変化は，このパターンで示す値から D%の範囲内に変化すると仮定する．以下のシミュレーションでは，明言しない限り D%=10%とする．

$$F_i(t) = a_i \times e^{\frac{-(t-b_i)^2}{2c_i^2}} \quad (1)$$

ここで，$a_i \in [20,80], b_i \in [4,11], c_i \in (0,6]$とする．

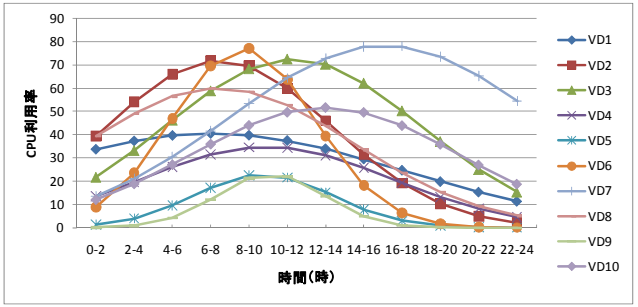

図 6　仮想デスクトップ CPU 利用率変化パターンの例

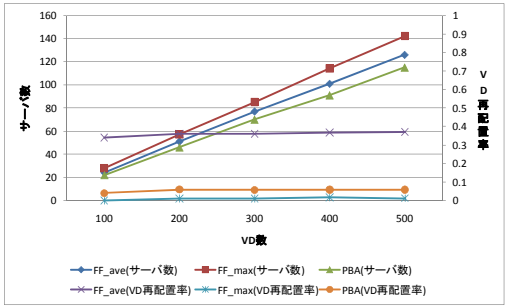

図 7　VD 数変化時の配置アルゴリズムの比較結果

5.2 VD 数変化時の配置結果

まず，一定の VD 数を配置する問題に対して，各配置アルゴリズムで必要となるサーバ装置の台数および再配置の割合について検討する．

図 7 は VD 数が 100,200,300,400,500 の場合の FF_ave，FF_max，PBA のそれぞれのサーバ数および VD 再配置率を示す．ここで，サーバ台数は再配置が行われた，最終的な配置においての仮想化サーバ数である．また，VD 再配置率は，再配置された VD の VD 全体に対する割合を示す．同図により，サーバ数に関しては，PBA は FF_ave と FF_max と比較して，VD 数と関係なく，サーバ台数は常にそれぞれ 9%，20%程度削減できることが分かる．また，VD 再配置率に関しては FF_max がもっとも低い（1%）値を示している．これは，FF_max が最大値を使って，大目に VD の利用資源量を見積もるからである．一方，FF_ave は FF_max よりサーバ台数が少ないが，VD 再配置率が平均で 36%となっている．それに対して，提案アルゴリズムの PBA はもっとも少ないサーバ台数を実現しながら，VD 再配置率も 5%程度に抑えることができること

ⓒ 2011 Information Processing Society of Japan

Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

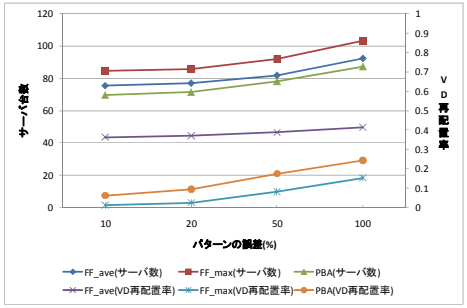

図 8　パターン相対誤差 D%変化時の配置アルゴリズムの比較結果

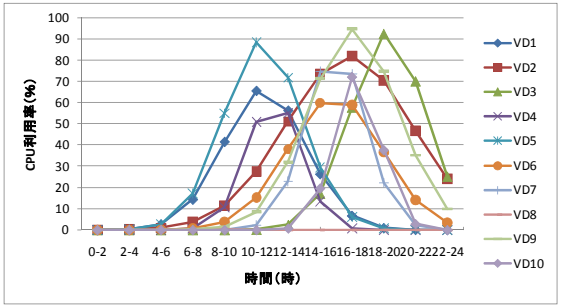

図 9　分散が狭いパターンの例

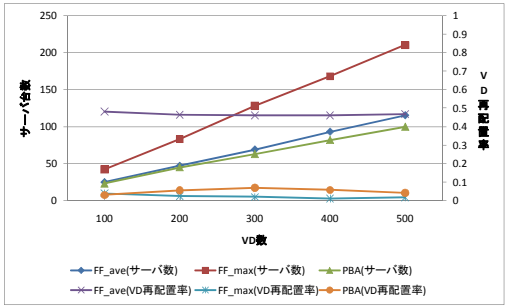

図 10　配置アルゴリズム比較（分散が狭いパターンの場合）

が分かる．

5.3　パターン情報の相対誤差変化時の配置結果

提案アルゴリズムは，各 VD の CPU 利用率のパターン情報に依存する．ここで，このパターン情報の精度を下げた場合の提案アルゴリズムの効果を確認する．

図 8 は，各 VD パターンの相対誤差 D%を 10%，20%，50%，および 100%に設定した場合，最終配置においてサーバ台数および VD 再配置率がどのように変化するかを示すグラフである．なお，VD 数は 300 とする．各データは１０回のシミュレーションの平均である．同図により，情報の誤差が増えると，3 つの配置アルゴリズムとも VD 再配置率が増すことが分かった．しかし，パターン情報の相対誤差が 100%の場合でも PBA は 14%程度に抑えることができ，FF_min, FF_max と比べてそれぞれ 6%，16%のサーバ数を削減する．つまり，パターン情報の相対誤差が 100%の場合でも，提案アルゴリズムを用いることにより，サーバ台数を削減する効果が得られることが分かる．

5.4　仮想デスクトップの資源利用パターン特徴変化時の配置結果

最後に, 仮想デスクトップの CPU 利用パターンが提案アルゴリズムへどのように影響するかを検討する. そこで, 式 （1） のパラメータ c_iをより小さい領域に絞る. $F_i(t)$ の各パラメータを次のように設定する.

$$a_i \in [10,70], b_i \in [4,11], c_i \in (0,2]$$

つまり，分散が狭い（0..2）パターンのデスクトップだけを利用することになる．このパターンの一例を図 9 に示す.

CPU 利用パターンの分散が狭い場合, 4.2 節の評価結果が図 10 となる. 同図により，このようなデスクトップの資源使用量の変化では，FF_ave の VD 再配置率が 47%と非常に高くなる．また，FF_max のサーバ台数が 210 であり，FF_ave と PBA の 2 倍程度になる．一方，提案アルゴリズムは，この場合，VD の再配置割合を 4%程度に抑えながら，FF_max と比べて 51%，FF_min と比べて 13%のサーバ台数を削減できることが分かる．

5.5　結論

本提案アルゴリズムではデスクトップの資源利用パターンの相関を利用することにより，FF-max と比べて 20%のサーバ数を削減しながら，VD 再配置回数を同程度に抑えられる. また, FF-min と比べて VD 再配置率を 36%から 5%程度に引き下げる上に，１０%程度のサーバ数を削減する．さらに，提案アルゴリズムはデスクトップの資源利用パター情報を利用しているが同情報の精度に強く依存しない．例えば，デスクトップの資源利用パター情報の誤差多い場合（相対誤差が 100%）であっても FF_max と比べてサーバ台数を 16%削減できる．また，デスクトップの資源利用パターンの分散が狭い場合，FF-max と比べてサーバ削減率が 51%となることもあり，提案アルゴリズムの優越性が顕著になる．

ⓒ 2011 Information Processing Society of Japan

Vol.2011-DPS-146 No.47
Vol.2011-CSEC-52 No.47
2011/3/11

6. 終わりに

本研究では，各仮想デスクトップが常に快適なパフォーマンスを実現できるための資源を確保しながら，コストの削減やエネルギー節約のためのサーバ装置台数削減を図る仮想化デスクトップ配置アルゴリズムを提案した．今後は様々なオフィス環境において，仮想デスクトップの実際の資源利用パターンを調査して，それぞれの環境において提案アルゴリズムを適用する場合に，どれぐらいのサーバ台数削減効果があるかを検討する予定である．また，Eucalyptus[13]など，クラウド管理基盤に，一つの機能として提案アルゴリズムを実装することも予定している．

参考文献

1) Xen, available at http://www.xen.org.

2) Kernel-based Virtual Machine, available at http://www.linux-kvm.org.

3) VMware, available at http://www.vmware.com.

4) Andrzej Kochut, “Power and Performance Modeling of Virtualized Desktop Systems”, in Proceedings of MASCOT 2009.

5) Beloglazov, Anton Buyya and Rajkumar, “Energy Efficient Resource Management in Virtualized Cloud Data Centers”, in Proceedings of the 10th IEEE/ACM CCGrid, 2010.

6) Yazir, Y.O.; Matthews, C. et al., “Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis”, in Proceedings of the 3rd IEEE CLOUD, 2010.

7) A. Kochut and K. Beaty, “On Strategies for Dynamic Resource Management in Virtualized Server Environments”, in Proceedings of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007.

8) Paolo Campegiani and Francesco Lo Presti,” A General Model for Virtual Machines Resources Allocation in Multi-tier Distributed Systems”, in Proc. of the 5^th International conference on autonomic and autonomous systems, 2009.

9) Bobroff. N., Kochut. A. and Beaty. K., “Dynamic Placement of Virtual Machines for Managing SLA Violations”, in Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, IM 2007.

10) Anthony Nocentino and Paul M. Ruth, “Toward Dependency-Aware Live Virtual Machine Migration”, in Proceedings of the 3rd VTDC 2009.

11) F. Hermenier, et al., “Entropy: A Consolidation Manager for Clusters”, in Proceedings of ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, 2009.

12) E.G.C.Jr., M.R.Garey and D.S.Johnson. “Algorithms for Bin Packing: A Survey”, Approximation algorithms for NP-hard problems, pages 46-93, 1996.

13) Eucalyptus, Open Source Cloud Platform, available at http://open.eucalyptus.com.

ⓒ 2011 Information Processing Society of Japan