

協調フィルタリングを用いた 探索パラメータ推定による仮想マシン配置高速計画法

Ha Tuan Minh, 小林良至, 鮫島正樹 (大阪大学)

A High-speed Scheduling Method of Virtual Machine Placement
with Search Parameter Estimation by Collaborative Filtering
Ha Tuan Minh, Ryoji Kobayashi, Masaki Samejima (Osaka University)

Abstract

We address a scheduling problem of dynamically provisioning virtual machines (VMs) in order to minimize power consumption on servers. Our research group has already developed a scheduling method of k VMs on n servers every λ minutes based on predicted power consumption during the following T minutes. We propose a parameter estimation method of k, n, λ based on the following CPU resource usage to make schedules more efficiently. The proposed method determines appropriate parameter values for past time-series data of CPU resource usage. The appropriate parameter values for scheduling are estimated by collaborative filtering with multiple similar time-series data and the appropriate parameter values of it. Performing a simulation experiment with an assumption of 200 servers and 400 VMs, we have confirmed that the proposed method can reduce the power consumption by 2.5% and computational time by 88% compared to our conventional method.

キーワード：仮想マシン動的配置計画問題、分枝限定法、協調フィルタリング、スケジューリング問題
(Virtual machine dynamic placement problem, Branch and bound, Collaborative filtering, Scheduling problem)

1. はじめに

クラウドコンピューティングを支えるデータセンタでは、サービス要求の増加に対応するためサーバを多数稼働させており、その消費電力は増加傾向にある [1]。そこで、仮想化技術を用いてサーバのリソースを論理的に分割した仮想マシン (Virtual Machine:VM) [2,3] を稼働させ、少数のサーバに集約することで省電力化を図っている [4]。一方で、VM を過度に集約することによって、VM の総リソース使用量がサーバのリソース容量を超えると、VM 上のアプリケーションの実行に遅延が生じる。そこで、VM を停止させることなく他のサーバに移動させるライブマイグレーション技術 [5-7] を用いて、VM の集約またはリソース不足の回避が行われる。

従来、VM をサーバに配置する問題はビンパッキング問題 [8,9] として定式化され、ヒューリスティック [10] やマッチング [11] による解法が適用されている。しかし、実運用時にはリソース使用量が変化するため、ビンパッキング問題を繰り返し解いて、消費電力量を最小化するように、VM の配置を変更する必要がある。しかし、ライブマイグレーションを実行すると、VM を稼働させているサーバがリソースを使用して電力を消費する。本研究では、ライブマイグレーションの実行回数を抑えて、一定期間の消費電力量を最小化する VM 配置計画を作成する問題 (VM 動的配置計画問題) を対象とする。

本研究グループでは、期間 T の CPU 使用率を予測し、全てのサーバをいくつかのサーバ群に分割した後、サーバ

群ごとに最大 k 台の VM を n 台のサーバに再配置する計画を、 $\lambda (< T)$ 分毎に分枝限定法 [12] を用いて作成する手法を開発した [13,14]。既開発の手法では、パラメータ λ, k, n は固定値を使用しており、常に λ 分以内に計画作成が完了するような k, n を設定する。しかし、省電力化に有効なパラメータは、CPU 使用率の変動によって異なる。変動が大きい場合、 λ を小さく設定しつつ、 λ 分以内の計画作成が可能な範囲で k, n をなるべく大きく保ち、多くの VM を繰り返し移動することで消費電力を低減できる。また、変動が小さい場合、 λ, k, n を大きく設定し、なるべく再配置を行わず、時間をかけて計画を探索することで消費電力を低減できる。パラメータ固定の既開発の手法では、様々な変動に対応して、消費電力量を十分に低減できないため、CPU 使用率の変動に応じてパラメータを動的に変化させる必要がある。

本研究では、過去の CPU 使用率の時系列データを用いて、パラメータごとの消費電力量を事前に評価し、省電力化に貢献したパラメータを、予測期間の計画作成に利用する手法を提案する。

2. VM 動的配置計画問題

2.1 VM 動的配置計画問題の定式化

サーバのリソースの中で CPU が最も多くの電力を消費することが知られている [11]。そこで、本研究で扱う VM 動的配置計画問題を、CPU 使用率を中心として消費電力量を最小化するように、以下のように定式化する。

Minimize

$$\sum_{t=1}^T \left\{ \sum_{i=1}^N (C_i(t) \times rate_i + B_i \times E_i) + m(t) \cdot E_M \right\}$$

subject to

$$C_i(t) = \sum_{j \in J_i(t)} V_j(t)$$

$$C_i(t) \leq 100$$

$$B_i = \begin{cases} 1 & (C_i(t) > 0) \\ 0 & (C_i(t) = 0) \end{cases}$$

ここで $C_i(t)$ はサーバ i の時刻 t における CPU 使用率 [%]、 $rate_i$ はサーバ i が CPU を 1% 使用するために必要な電力 [W/%]、 E_i はサーバ稼働のためのベース電力 [W]、 B_i はサーバの稼働状態を表す変数 (1 で稼働、0 で停止)、 E_M はマイグレーションを 1 回行うために必要な電力 [W]、 $m(t)$ は時刻 t におけるマイグレーション回数を表す。 $C_i(t)$ は、サーバ i における $j \in J_i(t)$ 番目の VM の CPU 使用率 V_j の総和とし、100% 以下に抑える。本問題は、制約条件を満足しつつ、消費電力を最小化する $J_i(t)$ を決定するスケジューリング問題の一種である。

2.2 既存の VM 配置計画作成法における問題点
 図 1 に本研究グループで開発した VM 再配置計画作成法 [13, 14] の概要を示す。期間 T における各 VM の CPU 使用率を自己回帰モデル [15, 16] を用いて予測する。予測にもとづいて、 λ 分後に VM の再配置を行う期間 T の配置計画を作成する。まず、対象とするデータセンタの規模に応じて、全サーバを一定数のサーバ群に分割する。サーバ群ごとの CPU 使用率の総和が均一になるよう、CPU 使用率の大きいサーバから順に、CPU 使用率の総和が最小のサーバ群に割り当てて分割する。次に、サーバ群ごとに分枝限定法を用いて、期間 T の消費電力量が最小となる VM 配置計画を探索する。サーバに配置されている VM から最大 k 台を選択し、電源効率の良い上位 n 台のサーバに順次再配置して探索する。このとき、1 台配置することに見込みの消費電力量を算出し、 k 台の再配置が完了した時点の最小消費電力量より大きければ、VM 配置計画の探索を打ち切

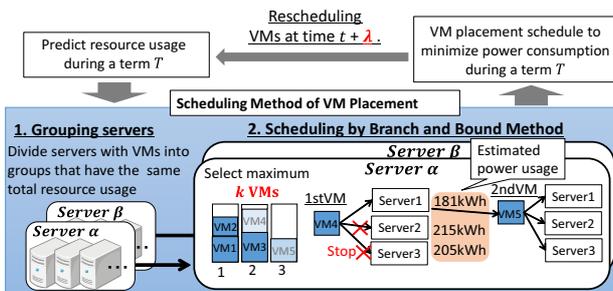


図 1 VM 配置計画作成法の概要

Fig. 1. Scheduling problem for dynamically provisioning VMs

る。探索で得られた VM 配置計画を λ 分後に実行し、その配置を初期配置として、再び VM 配置計画の探索を行う。

CPU 使用率の変動によって、省電力化に有効なパラメータ λ, k, n は異なる。変動が大きい場合、 λ を小さく設定しつつ、 λ 分以内の計画作成が可能な範囲で k, n をなるべく大きく保ち、多くの VM を繰り返し移動することで消費電力を低減できる。また、変動が小さい場合、 λ, k, n を大きく設定し、なるべく再配置を行わず、時間をかけて計画を探索することで消費電力を低減できる。しかし、全ての計画作成が再配置までに完了するように調整された固定のパラメータを用いる既開発の手法では、様々な変動に対応して、消費電力量を十分に低減できない。そこで、CPU 使用率の変動に応じて、パラメータを逐次変更し、計画を作成することを本研究の目的とする。

2.3 パラメータ推定の課題 パラメータ推定の方針と課題を図 2 に示す。計画作成の目的関数と制約条件を構成する CPU 使用率が、計画作成期間 T で類似していれば、同一のパラメータを用いて類似した計画を得ることができる。そこで、類似した過去の CPU 使用率の時系列データ (以降、単に時系列データと呼ぶ) において、省電力化に貢献した再配置時間 λ 、再配置 VM 数 k 、移動先サーバ数 n を特定し、予測期間の計画作成に利用する。まず、省電力化に貢献したパラメータを、過去の CPU 使用率の時系列データに対して事前に求めるため、パラメータを変更しながら計画作成を行い、パラメータごとの期間 T の消費電力量を評価しておく。過去の時系列データのうち、サーバごとの予測 CPU 使用率と類似した時系列データにおいて、省電力化に貢献したパラメータを予測期間の計画作成に利用する。

単純なマッチング手法として、ユークリッド距離で表される非類似度が最小となる時系列データのみに着目し、最も消費電力を低減したパラメータを設定する手法が考えられる。すべての仮想マシンを統合的に判断して、パラメータを決定するため、過去の時系列データ $s \in S$ と期間 T で比較した場合の非類似度 $D_{T,s}$ を以下のように求める。

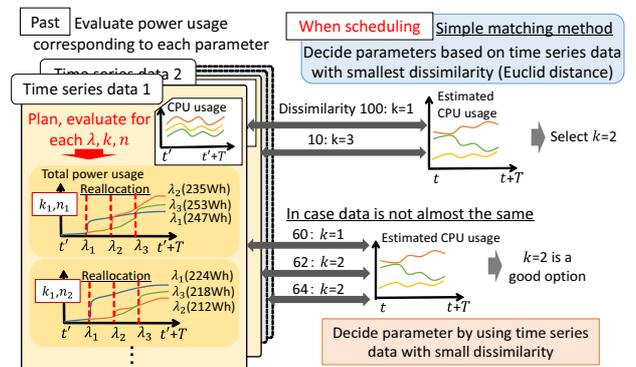


図 2 パラメータ推定の方針と課題

Fig. 2. Issues on estimating parameter values

- (1) 全 VM の過去の時系列データ s と、全 VM の現在の時系列データの組み合わせを網羅的に作成し、全組み合わせのユークリッド距離を求める。
- (2) ユークリッド距離の小さい過去の VM と現在の VM の組み合わせから順に確定させる。
- (3) 確定した組み合わせのユークリッド距離の総和を求め、非類似度 $D_{T,s}$ とすることで、全ての組合せの中でなるべく小さい $D_{T,s}$ を効率的に求める。

すべての仮想マシンで、ほぼ一致した時系列データが存在する場合は、単純なマッチング手法でパラメータを設定できる。しかし、非常に類似した時系列データが存在することはまれであるため、有効なパラメータを設定できない可能性がある。

3. VM 配置計画作成のためのパラメータ推定手法

3.1 提案手法の概要

非類似度の小さい時系列データを複数用いて、より多くの時系列データによって支持されるパラメータを設定する方が、省電力の計画を作成できる可能性が高い。そこで、非類似度の小さい複数の時系列データを協調的に用いて、パラメータを設定する手法を提案する。過去の CPU 使用率の時系列データのうち、類似した複数の時系列データにおいて、省電力化に貢献したパラメータを重視する協調フィルタリング [17] を用いて、計画作成時のパラメータを決定する。図 3 に提案手法の概要を示す。

再配置 VM 数 k と移動先サーバ数 n は、 λ 分後に再配置するためのパラメータであるため、 λ 分後以降の CPU 使用率の変動に大きく依存することが分かっている。そこで、再配置時間 λ を決定した後に、 λ 分後以降の消費電力量を最小化するように k, n を決定する。 λ の決定は、期間 T の時系列データ s 間の非類似度 $D_{T,s}$ が小さい時系列データを用いて行う。また、 k, n の決定は、期間 $T - \lambda$ の時系列データ s 間の非類似度 $D_{T-\lambda,s}$ が小さい時系列データを用いて行う。決定した λ, k, n を用いて、VM 配置計画を作成し、 λ 分後に実行する。

3.2 再配置時間 λ の決定

過去の時系列データ s との期間 T の非類似度 $D_{T,s}$ が小さい時系列データにおいて、消費電力量が小さい再配置時間 λ を採用する。 λ の決定方法を図 4 に示す。 λ が同一の全ての k, n の組み合わせの中で、最小消費電力量が小さい順に λ のランク $R_{\lambda,s}$ を決定する。 $D_{T,s}$ と $R_{\lambda,s}$ の重み付け和の逆数を、 λ に対する推薦度 P_{λ} として以下の式で算出し、 P_{λ} が最大の λ を採用する。

$$P_{\lambda} = \frac{1}{\sum_s R_{\lambda,s} \times D_{T,s}}$$

図 4 の例では非類似度 70 の時系列データ 1 と非類似度 60 の時系列データ 2 があり、パラメータ λ_1 による消費電力量のランクは、時系列データ 1 で 2 位、時系列データ 2 で 1 位である。非類似度とランクに基づいて λ_1 の推薦度は 0.005 と計算される。 λ_1 の推薦度が、他の推薦度よりも大きいので、 λ_1 が使用される。

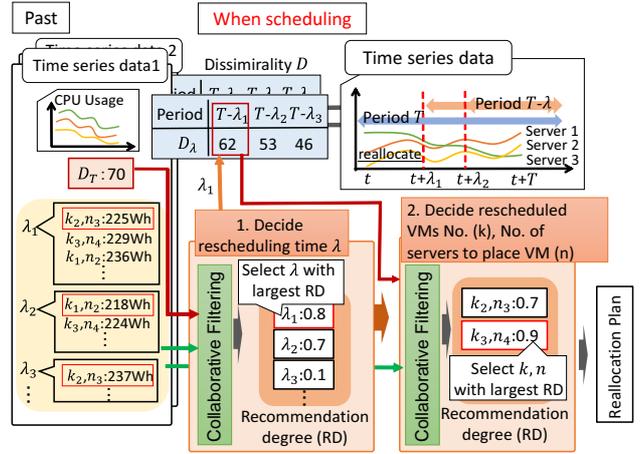


図 3 提案手法の概要

Fig. 3. Outline of the proposed method

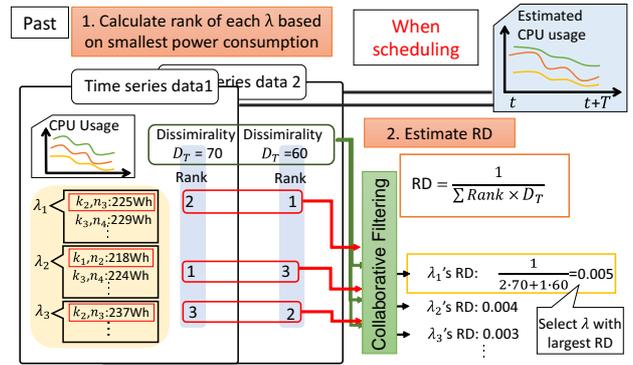


図 4 再配置時間 λ の決定

Fig. 4. Estimation of rescheduling time λ

3.3 再配置 VM 数 k 、移動先サーバ数 n の決定

過去の時系列データ s との期間 $T - \lambda$ の非類似度 $D_{T-\lambda,s}$ が小さい時系列データにおいて、消費電力量の小さい再配置 VM 数 k 、移動先サーバ数 n を採用する。 k, n の決定方法を図 5 に示す。3.2 節で決定した λ にもとづいて、 k, n ごとの消費電力量が小さい順に k, n のランク $R_{k,n,s}$ を決定する。 λ の決定と同様に、決定済みの λ を用いた場合の k, n ごとの推薦度 $P_{k,n}$ を以下の式で算出し、 $P_{k,n}$ が最大の k, n を採用する。

$$P_{k,n} = \frac{1}{\sum_s R_{k,n,s} \times D_{T-\lambda,s}}$$

図 5 の例では非類似度 60 の時系列データと非類似度 50 の時系列データが存在している。図 4 の例と比べると比較間隔 T から $T - \lambda$ へと短くなったため、ユークリッド距離で定義される非類似度も小さくなっている。パラメータ k_2 と n_3 による消費電力量のランクは、時系列データ 1 で 1 位、時系列データ 2 で 2 位である。このため、 k_2 と n_3 の推薦度は 0.006 と計算され、他の推薦度よりも大きいため、 k_2 と n_3 が使用される。

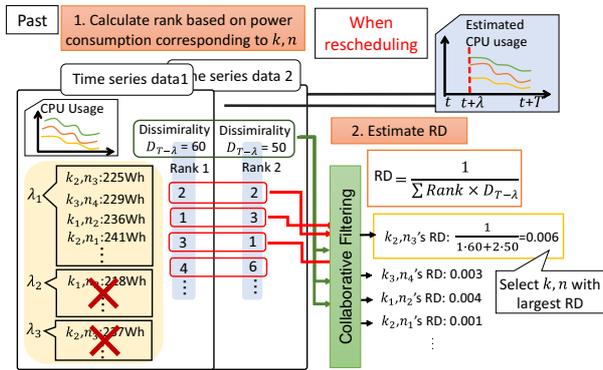


図5 再配置 VM 数 k 、移動先サーバ数 n の決定

Fig. 5. Estimation of the numbers of target VMs k and servers n in rescheduling

4. 評価実験

4.1 実験環境 CPU が Intel Core i7 3.4GHz、メモリが 8GB の計算機上にデータセンターの消費電力量を計算するシミュレータを実装した。VM の CPU 使用量は、公開された Web サービスのアクセス履歴 [18, 19] に対して線形であるとする。また、実際のサーバの消費電力量を計測し、 $rate_i$ を 0.5W/% からサーバごとに 0.1 刻みで増加させた値に、 E_i 、 E_M をそれぞれ 5W、30W に設定した。サーバ 200 台、VM400 台を対象として、16 個のサーバ群へ分割し、計画作成期間 $T = 40$ 分で逐次作成した計画を 24 時間運用した場合の消費電力量と平均計画作成時間を評価した。パラメータ群は $K = \{1, 2, 3, 4\}$ 、 $N = \{1, 2, \dots, 8\}$ 、 $\Lambda = \{5, 10, 15, 20\}$ とし、提案方式では過去 9 日分の CPU 使用率の時系列データを用いてパラメータを決定する。グリーディに再配置する手法 ($\lambda = 20$)、パラメータを最適値 ($k = 3, n = 6, \lambda = 20$) に固定した手法、非類似度最小の時系列データのみを用いた手法と比較した。

4.2 提案手法の評価 各手法における 24 時間の消費電力量、平均計画作成時間、変動係数 $CV (= \text{標準偏差} / \text{平均})$ と提案手法によって採用された λ の関係を、それぞれ図 6、図 7、図 8 に示す。

まず、24 時間の消費電力量について評価する。パラメータを固定した手法では、CPU 使用率の予測を利用することで、将来の変動を考慮した再配置が行えており、グリーディに VM を再配置する手法に対して 16% 削減できている。提案手法においては、パラメータを固定した手法からさらに 2.5% 削減できた。図 8 に示すように、変動係数が 0.4 を超える時間帯では 94% の割合で $\lambda = 5$ が採用されているのに対し、変動係数が 0.4 を下回る時間帯では、80% の割合で $\lambda = 5$ が採用されている。提案手法によって、変動が大きい場合は λ を小さく、変動が小さい場合は λ を比較的大きく設定できている。変動に応じたパラメータ設定により消費電力を低減できたと考えられる。非類似度最小の時系列データのみを用いた手法では、パラメータを固定した手法に対して消費電力量が増加しており、複数の類似する時系

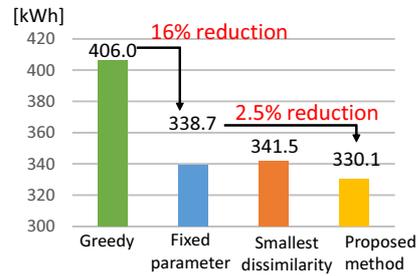


図 6 消費電力量の比較

Fig. 6. Comparison of power consumption

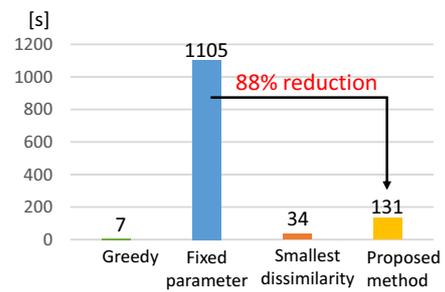


図 7 平均計画作成時間の比較

Fig. 7. Comparison of average computational time of scheduling

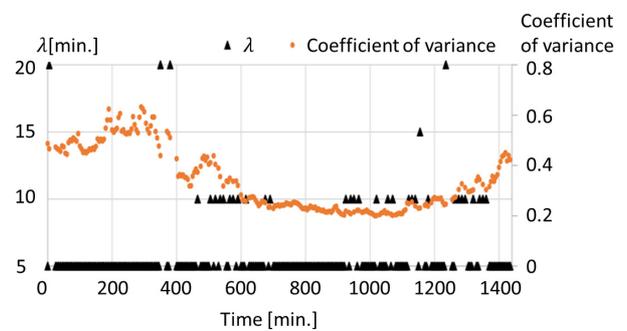


図 8 変動係数と λ の時系列変化

Fig. 8. Coefficient of variation and time-series variation of λ

列データに着目する協調フィルタリングを用いた手法の有効性を確認した。

次に、平均計画作成時間について評価する。提案手法では、パラメータを固定した手法に対して 88% 短縮されている。図 8 に示すグラフでは、211 回の計画作成のうち $\lambda = 5$ に設定された回数が 152 回と最も多く、少数の VM を短い間隔で移動したことによって、平均計画作成時間が大幅に短縮された。

5. 結論

本研究では、CPU 使用率を予測し、リソース容量を超えないように消費電力量を最小化する VM 動的配置計画問

題を対象とした。本研究グループで提案した、分枝限定法を用いた VM 配置計画作成法では、再配置時間 λ 、再配置 VM 数 k 、移動先サーバ数 n に固定値を使用している。そのため、CPU 使用率の変動に応じて、省電力化に有効なパラメータ λ, k, n が異なる VM 動的配置計画問題において、従来手法では消費電力量を十分に低減できないという問題があった。そこで本研究では、過去の CPU 使用率の時系列データのうち、類似した複数の時系列データにおいて、省電力化に貢献したパラメータを重視する協調フィルタリングを用いて、計画作成時のパラメータを決定する手法を提案した。

評価実験においては、サーバ 200 台、VM400 台を有するデータセンタにおいて、パラメータを固定した手法と比較して 2.5%、非類似度最小の時系列データのみを用いた手法と比較して 3.3%消費電力を削減できた。以上より、協調フィルタリングを用いた過去の CPU 使用率の時系列データとのマッチングの有効性を示した。また、パラメータを固定した手法と比較して、平均計画作成時間を 88%削減し、再配置時間 λ を短く設定した場合でも、再配置時間までに計画作成を完了できることを確認した。

謝 辞 本研究は JSPS 科研費 24360154 の助成を受けたものです。

参考文献

- (1) P. Thibodeau, "Data centers are the new polluters," *Computerworld* (2014)
- (2) R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F. C. Martins, A. V. Anderson, and L. Smith, "Intel virtualization technology," *Computer*, Vol. 38, No. 5, pp. 48–56 (2005)
- (3) M. Stillwell, V. Frederic, and C. Henri, "Dynamic fractional resource scheduling for hpc workloads," *IEEE International Symposium on Parallel & Distributed Processing (IPDPS'10)*, pp. 1–12 (2010)
- (4) N. Kim, J. Cho, and E. Seo, "Energy-Based Accounting and Scheduling of Virtual Machines in a Cloud System," In *Proc. of IEEE/ACM International Conference on Green Computing and Communications*, pp. 176–181 (2011)
- (5) H. Liu, H. Jin, C.Z. Xu, and X. Lia. "Performance and energy modeling for live migration of virtual machines," *Cluster computing*, Vol. 16, No. 2, pp. 249–264 (2013)
- (6) S. Akoush, R. Sohan, A. Rice, A.W. Moore, and A. Hopper, "Predicting the performance of virtual machine migration," In *Proc. of IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 37–46 (2010)
- (7) W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds," *A performance evaluation*, pp. 254–265 (2009)
- (8) N. Bansal, C. Alberto, and S. Maxim, "Improved approximation algorithms for multidimensional bin packing problems," *IEEE 4th Annual Symposium on Foundations of Computer Science (FOCS'06)*, pp. 697–708 (2006)
- (9) S. S. Seiden, "On the online bin packing problem," *Journal of the ACM (JACM)*, pp. 640–67 (2002)
- (10) 網代育大:「仮想マシン配置のためのパッキングアルゴリズム」, *情報処理学会 研究報告 システム評価*, No. 2010-EVA-33, Vol. 4, pp.1-7 (2010)
- (11) 高橋里司・竹房あつ子・繁野麻衣子・中田秀基・工藤知宏・吉瀬章子:「省電力化のためのマッチングに基づく仮想計算機パッキングアルゴリズム」, *情報処理学会論文誌, コンピューティングシステム*, Vol. 5, No. 5, pp. 33–42 (2012)
- (12) H.H. Greenberg, "A branch-bound solution to the general scheduling problem," *Operations Research*, Vol. 16, No. 2, pp. 353–361 (1968)
- (13) R. Kobayashi, K. Sato, M. Samejima, and N. Komoda, "A Prediction-based Scheduling Method for Dynamic Placement of Virtual Machines in Data Center," In *Proc. of International Conference on Applications of Computer Engineering (ACE)*, pp. 19–24 (2014)
- (14) 小林良至・佐藤勝紀・鮫島正樹・薦田憲久:「分枝限定法によるリソース予測使用量に基づく仮想マシン動的配置高速計画方式」, *平成 26 年電気学会電子・情報・システム部門大会講演論文集*, pp. 1791–1792 (2014)
- (15) G. Kitagawa and W. Gersch, "A smoothness priors long AR model method for spectral estimation," *IEEE Transactions on Automatic Control*, Vol. 30, No. 1, pp. 3-8 (1985)
- (16) B. E Hansen, "The grid bootstrap and the autoregressive model," *Review of Economics and Statistics*, pp. 594–607 (1999)
- (17) P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," In *Proc. of Computer-Supported Cooperative Work (CSCW)*, pp. 175–186 (1994)
- (18) U. Guido, G. Pierre, and M.V. Steen, "Wikipedia workload analysis for decentralized hosting," *Computer Networks*, Vol. 53, No. 11, pp. 1830–1845 (2009)
- (19) J. Allspaw, 「キャパシティプランニング - リソースを最大限に活かすサイト分析・予測・配置」, オーム社 (2014)