

# VM スタンバイシステムを用いたプロビジョニングの高速化

## High Performance Provisioning Using a VM Standby System

町田 文雄<sup>†</sup> 川戸 正裕<sup>†</sup> 前野 義晴<sup>†</sup>  
 Fumio MACHIDA Masahiro KAWATO Yoshiharu MAENO

### 1. はじめに

プロビジョニングは IT システムの負荷変動や障害に備えて計算機資源を自動で配備する技術である。近年データセンターや企業システムの計算機資源を有効活用するため、サーバのプロビジョニング技術への期待が高まっている [1]。しかし、サーバのプロビジョニングには数十分から数時間かかるため、突発的な負荷変動や障害に迅速に対応できないという課題がある。本稿では、時間のかかるサーバのプロビジョニング処理を高速化するため、仮想マシン (VM) を使ったスタンバイ制御技術を提案する。提案する手法は、プロビジョニングの対象となるプールサーバ上で VM によるスタンバイ領域を作成し、スタンバイ領域上でプロビジョニングを事前に実行する。プロビジョニングが要求されてから、スタンバイ領域への資源割当量を増加させ、ネットワーク構成を変えることで迅速にサービス提供可能な状態にする。この手法は、プロビジョニングの要求が発生する前にプロビジョニング処理を開始する必要があるため、プロビジョニング要求を事前に予測する技術が必要となる。本稿では、プロビジョニング予測技術の基本設計と実験システムによる評価結果を示す。

### 2. VM スタンバイシステム

提案する VM スタンバイシステムの構成を示す (図 1 参照)。VM スタンバイシステムは Xen [2] や VMware [3] などの VM モニタ (VMM) によって仕切られたアクティブ領域とスタンバイ領域からなる。アクティブ領域では、運用中のアプリケーションを VM 上で稼働させ、アプリケーションを実行するために十分な資源 (CPU、メモリ、I/O 帯域) を割り当てる。一方、スタンバイ領域では、近い将来利用される可能性のあるアプリケーションを VM 上で稼働させ、必要最低限な資源を割り当てる。

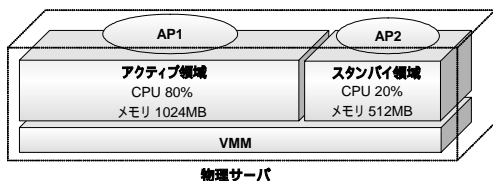


図 1 VM スタンバイシステム

<sup>†</sup> 日本電気株式会社 インターネットシステム研究所  
 Internet Systems Research Laboratories, NEC Corporation

VM スタンバイシステム上では、Active/Standby スイッチ (A/S スイッチ) 機能によりプロビジョニングを実行する。A/S スイッチ機能とは、アクティブ領域とスタンバイ領域のリソース割り当て量を動的に入れ替える機能である。A/S スイッチにより、スタンバイ領域で稼働していたアプリケーションは実運用のための十分な資源を割り当てられ、運用開始可能な状態となる。プロビジョニング処理ステップのいくつかを事前に実行してスタンバイ状態を作ることができるため、プロビジョニングが要求されてからの処理時間は大幅な短縮が見込める。

### 3. スタンバイ状態の決定法

VM スタンバイシステムを使ってプロビジョニング時間を短縮化するためには、プロビジョニングが要求されるアプリケーションを事前に予測し、スタンバイ領域に準備しておく必要がある。本研究では、現在の負荷変動状態と過去の負荷情報の統計データから、各アプリケーションに対するプロビジョニング要求を予測することで、プロビジョニング時間を最短にするスタンバイ状態を決定する。

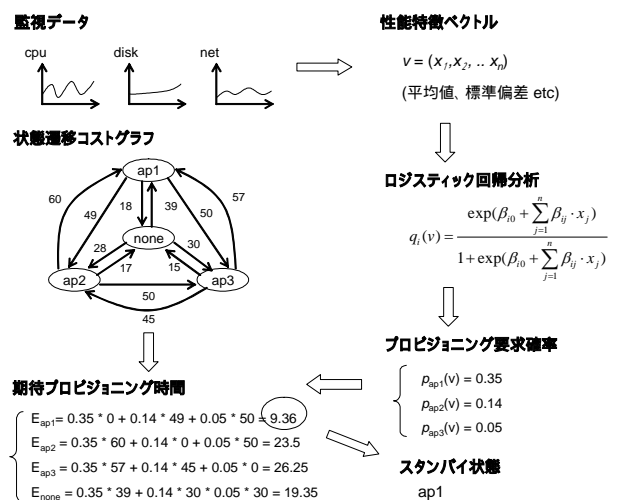


図 2 スタンバイ状態決定手順

スタンバイ状態の決定手順を図 2 に示す。はじめに、現在の監視データに基づいて統計値を求め、性能特徴ベクトルを生成する。次に、性能特徴ベクトルを入力としてロジスティック回帰分析を行い、各アプリケーションのプロビジョニング要求確率を予測する。ロジスティック回帰分析は、複数の説明変数を用いて 2 値データや比率の目的変数を導出する分析手法である [4]。ロジスティック関数の回

帰係数は過去の統計データを基に事前に求めておく。最後に、計算された要求確率に基づいて期待プロビジョニング時間を最小にするスタンバイ状態を決定する。期待プロビジョニング時間は、事前に評価した状態遷移コストと各アプリケーションに対する要求確率の積和で計算する。

#### 4. 評価

3つの異なる Web アプリケーションシステムで VM スタンバイシステムを共有する実験システムを構築し、提案手法によるプロビジョニング高速化の効果を評価した。各アプリケーションのソフトウェア構成、プロビジョニングを要求する CPU 負荷の閾値、及び起動時間と停止時間を表 1 に示す。

表 1 アプリケーション構成

アプリケーション	Web portal	Blog server	Online shop
ソフトウェア構成	Apache	Apache+MySQL	JBoss+MySQL
CPU 負荷閾値(%)	25	85	50
起動時間(sec)	50.7	52.6	52.3
停止時間(sec)	25.5	24.9	25.5

はじめに、ロジスティック関数の回帰係数を求めるため、各アプリケーションに負荷を与え、負荷変動とプロビジョニング要求との相関ルールを抽出した。性能指標として CPU 利用率 (vmstat の usr 値) とブロックデバイスから受信したデータ量 (vmstat の bo 値) を監視し、5 秒間の平均値と標準偏差を求めて性能特徴ベクトルとした。性能特徴ベクトルを観測後、30 秒後から 35 秒後の期間の CPU 利用率の最大値を求め、閾値と比較してプロビジョニングが要求される可能性を判定した。抽出された相関ルールの例を図 3 に示す。

	Web portal	Blog server	Online shop	
VM	平均	$x_1$	$x_5$	$x_9$
	標準偏差	$x_2$	$x_6$	$x_{10}$
CPU	平均	$x_3$	$x_7$	$x_{11}$
	標準偏差	$x_4$	$x_8$	$x_{12}$

**相関ルール**

$v=(10.2,1.6,176,12.39,13.8,3.31,56.8,71.48,35.8,1.93,25.6,51.2)$   
 30 - 35 sec → "Online shop"

図 3 特徴ベクトルとプロビジョニング要求の相関ルール

抽出された相関ルールを用いて各アプリケーションに対するロジスティック関数の回帰係数を導出した。一例として、Web portal のロジスティック関数を以下に示す。

$$q_{web} = \frac{\exp(-9.44 + 0.79x_3 - 0.05x_5)}{1 + \exp(-9.44 + 0.79x_3 - 0.05x_5)}$$

次に、導出したロジスティック関数によってプロビジョニング要求の予測が行えることを確かめるため、再度各アプリケーションに負荷をかけ、スタンバイ状態決定

プロセスをシミュレートした。相関ルールを抽出する際に利用した負荷にランダムな負荷ノイズを加えて評価した。プロビジョニング要求は CPU 負荷が初めて閾値を越えた時点で発生するものとした。

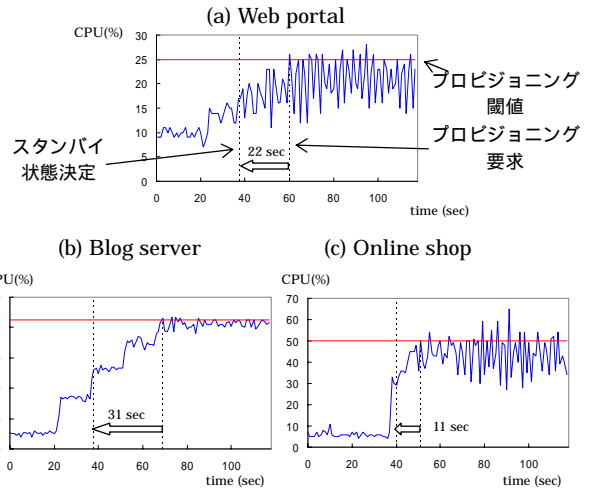


図 4 プロビジョニング予測の評価結果

何れのアプリケーションのプロビジョニングが要求される場合においても、プロビジョニング要求を予測してスタンバイ状態を決定できることが確認できた。シミュレーション上では、プロビジョニング要求が発生する 11 秒前から 31 秒前にスタンバイ状態を決定し、プロビジョニングを開始できる。

#### 5. まとめと考察

プロビジョニング時間を短縮化するため、VM スタンバイシステムを用いて事前にプロビジョニングを実行する手法を提案した。提案手法はシステムの性能監視情報からロジスティック回帰分析によってプロビジョニングの要求確率を予測する。実験システムで評価した結果、30 秒先のプロビジョニング要求を予測できた。

実際は、数十分から数時間先のプロビジョニング要求を予測できなければ、プロビジョニング高速化の効果は小さい。提案手法を用いて数十分から数時間後のプロビジョニング要求を予測可能かどうかについて今後検証を進める予定である。

#### 参考文献

- [1] 町田文雄, 小倉章嗣, 西村祥治, 川戸正裕, 前野義晴: 自律運用管理ミドルウェアを高速化する資源情報キャッシュ制御方式, IPSJ, DSM シンポジウム 2005, 2005.
- [2] Xen: <http://www.xen-source.com/>
- [3] VMware: <http://www.vmware.com/>
- [4] D. W. Hosmer and S. Lemeshow, Applied logistic regression, 2nd ed, Wiley, 2000.