

仮想環境構築・運用自動化技術

小笠原大治* 堀口真理子*
河野義哉* 金木佑介**
遠藤 司*

Automation Technology for Virtual Machine Construction and Operation Process

Daiji Ogasawara, Yoshiya Kono, Tsukasa Endo, Mariko Horiguchi, Yusuke Kaneki

要 旨

多くの企業でサーバの仮想化が進むにつれ、大量にVM (Virtual Machine)を構築する要求が増えてきている。そこで三菱電機インフォメーションシステムズ株式会社(MDIS)は、大量のVMを迅速に構築する“仮想環境構築・運用自動化ソフトウェア”を開発した。

このソフトウェアは、大量のVM生成とOS設定、アプリケーションの導入・設定といった従来人手で行っていた作業を、あらかじめ定義した情報を基に自動化する。

このソフトウェアの主な特長は、次の3つである。

(1) 大量のサーバの構築自動化

1台目のVMを雛形(ひながた)として2台目以降のVMを自動生成する機能と複数台のサーバを並列に構築する機能によって、類似したサーバを大量に構築する際の工数と

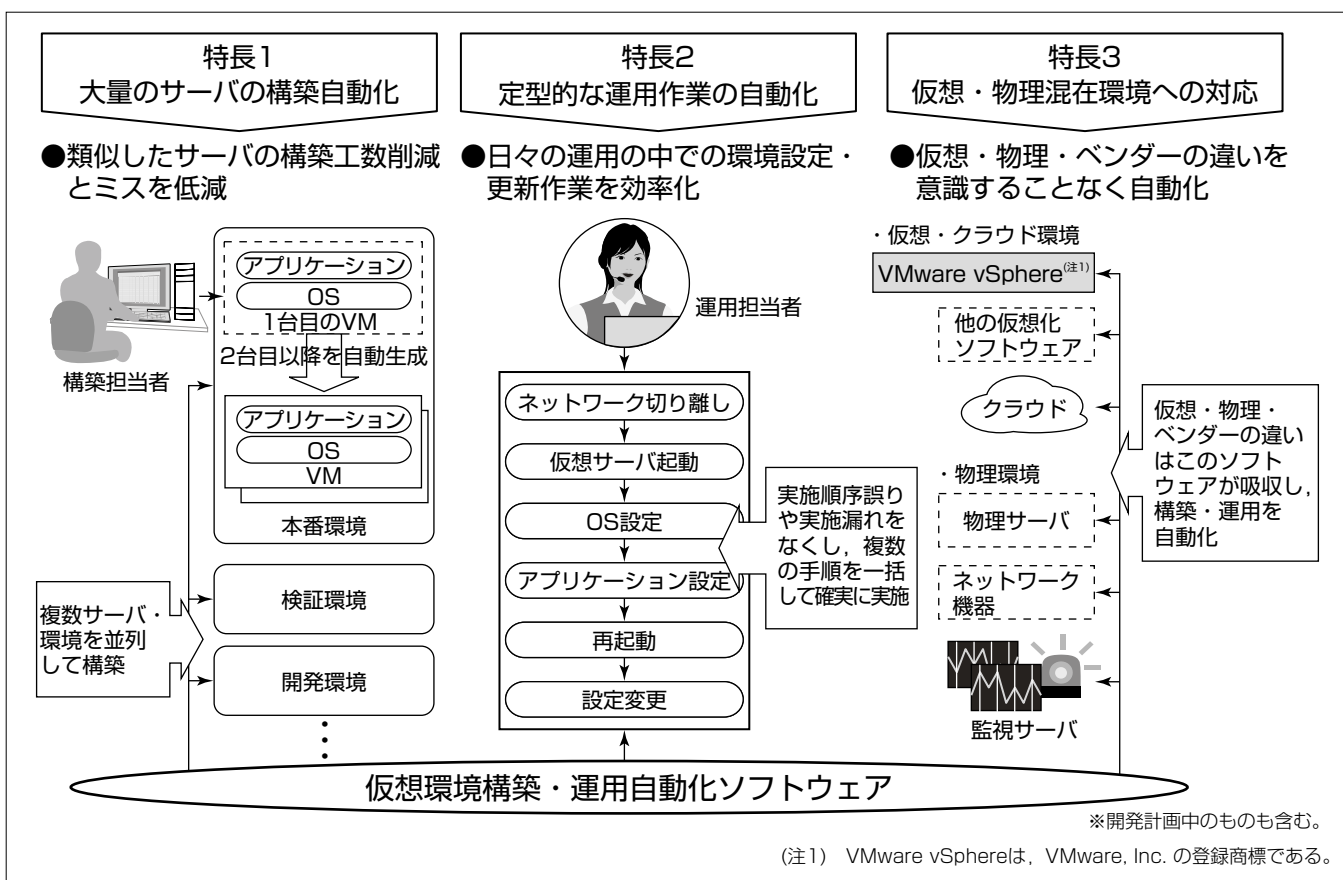
所要時間を大幅に削減し、OSとアプリケーションの設定ミスを低減する。

(2) 定型的な運用作業の自動化

大量のサーバを構築した後の、日々の運用の中でのOS・アプリケーション環境の設定・更新作業を自動化し、運用作業の効率化と手順誤り・漏れをなくす。

(3) 仮想・物理混在環境への対応

仮想・物理混在環境での構築・運用を自動化する。各々の環境を制御するために各ベンダーが提供しているAPI (Application Programming Interface)の違いをこのソフトウェアで吸収しているため、マルチベンダー環境にも対応できる。



仮想環境構築・運用自動化ソフトウェアの特長

大量のサーバの構築自動化、定型的な運用作業の自動化、仮想・物理混在環境への対応の3つの特長を持つ。アプリケーション導入手順や設定ファイル等のテンプレート情報を、対象サーバに応じた処理情報に整形することで、複数台の対象サーバの環境構築と運用を効率的に行う。なお、このソフトウェアの技術は現在特許出願中である(出願番号:特願2013-30940)。

1. ま え が き

多くの企業でサーバの仮想化が進み、構築・運用すべき仮想マシン数が大幅に増えてきている。しかし、その運用管理費用は横ばいのため、数年前と比べ同じ人数(費用)で2倍近い台数のサーバをミスなく構築・運用することが必要になってきた⁽¹⁾。そこでMDISは、大量のサーバの構築・運用の工数削減とミス低減を狙い、“仮想環境構築・運用自動化ソフトウェア”を開発した。

本稿では、このソフトウェアの主な特長と適用事例・効果について述べる。

2. 仮想環境構築・運用時の課題

2.1 大量サーバ構築時の工数削減とミス低減

システム構築作業では、本番システムだけでなくアプリケーションやシステムの開発環境と検証環境も構築するのが一般的である。さらに、BCP(Business Continuity Planning)用に本番とほぼ同じ構成の災害対策システム(BCP環境)を構築することが増えている。

人手で類似した構成のサーバを多数構築する場合、設定誤りを引き起こしやすい。100台以上のサーバの構築設定を手作業で行うと、IPアドレスを誤って設定してしまうことは、十分起こり得る⁽²⁾。

構築作業の自動化を実現すれば、構築作業全体の工数削減とミス低減を狙うことができる。

2.2 運用の効率化と運用容易性の向上

大量にサーバを持っていると、日々の運用の中でのOS・アプリケーション環境の設定・更新作業も増えるため、その効率化が必要である⁽³⁾。

例えばインターネットからのアクセスを分散して処理しているWebサーバが10台あり、これらすべての環境設定を実施したい時、1台1台ネットワークから切り離し、手作業で設定を行わなければならない。大量のサーバをインターネット公開しているオフィシャルサイトの運用では、この設定作業を3日に1回実施しているケースもある。

このような定型的な作業を自動化すれば、時間や工数の削減を狙うことができる。

また、運用の中では非定型的な作業も発生する。非定型的な作業は、低頻度ではあるが作業項目が多岐にわたったり、担当者が作業に未習熟であったりするため、対応が容易ではない。

例えば被災時のBCP環境への切替え作業は非定型的な作業である。被災時には緊急性が求められ、環

境切替え方法を確認する時間が十分に確保できないことや、担当者が不在の可能性も考えられる。

非定型的な作業だからこそ、即座に判断・行動できる仕組みを用意すれば、運用容易性を高めることができる。

2.3 仮想・物理混在環境への対応

サーバの仮想化が進展している一方で、すべてのシステムを仮想環境へ移行しないケースもある。そのため仮想・物理サーバが混在した環境での運用管理が求められる。また、サーバOSの種類によって、OS・アプリケーション環境を設定するためのベンダー提供コマンド・APIが異なるため、環境ごとに多くの技術知識や経験を習得するなどの対応が必要である。そこで、構築・運用を自動化する際は、仮想・物理の混在及びOSの差異を意識させないことで利用者の利便性を更に高めることができる。

3. 仮想環境構築・運用自動化ソフトウェア

“仮想環境構築・運用自動化ソフトウェア”は、あらかじめ定義した情報を基に、大量のVM生成とOS設定、アプリケーションの導入・設定といった、従来人手で行っていた作業を自動化するソフトウェアである。

このソフトウェアを利用して、類似したサーバを複数構築・運用する場合の使い方を図1に示す。

(1) VMテンプレート作成

元となるVMと、VM作成後に配置したいアプリケーション・設定ファイルの雛形を作成し、テンプレート領域へ配置する。

(2) パラメータ定義

複数のサーバで共通なパラメータ(VMテンプレート名など)と、サーバ固有のパラメータ(IPアドレス、ホスト名等)を登録する。

(3) 処理フローの作成

サーバの電源入・電源切・VMへのファイルコピー・運用コマンド実行等、このソフトウェアが提供する操作を組み合わせて、従来人手で行っていた作業を登録する。

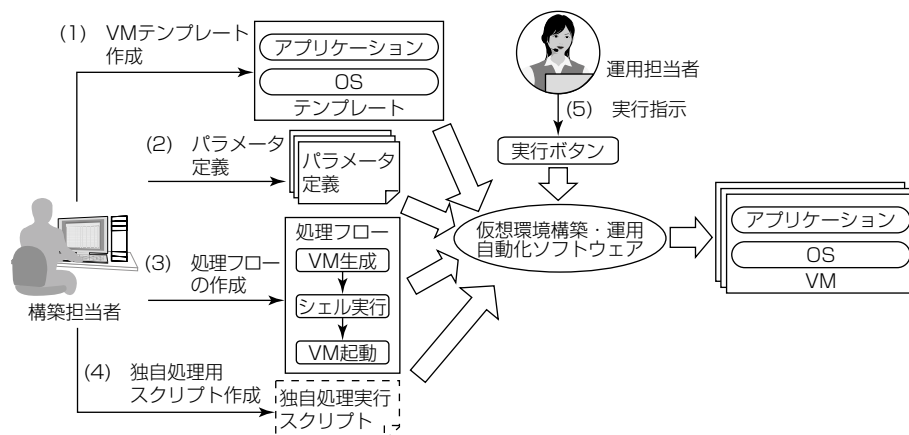


図1. 仮想環境構築・運用自動化ソフトウェアの使い方

(4) 独自処理用スクリプト作成

利用者が作成したアプリケーションを導入したい場合は、個別に独自処理を実行するスクリプトを作成する。

(5) 実行指示

このソフトウェアの管理画面から生成対象サーバを選択し、実行を指示する。

4. 特長

このソフトウェアの3つの特長を次に述べる。

4.1 大量のサーバの並列構築を自動化

4.1.1 VM・OS・アプリケーション構築自動化機能

VM・OSからアプリケーションまで、システム構築手順を自動実行する、このソフトウェアの中心機能である。具体的には元となるVMを複製し、複製したVMに対しMAC(Media Access Control)アドレスとネットワークへの接続設定を行い、VMとOSを起動(電源入)する。VMの起動後、OSに対しIPアドレス・サブネット・ゲートウェイ等のネットワークカードごとの設定と、ホスト名変更を行い、適宜OSを再起動して、設定を反映させる。OSの設定終了後、アプリケーションを導入するためのファイル群を各VMへコピーし、アプリケーションを導入するためのスクリプトを実行する。

この機能によって、ホスト名とIPアドレスだけが異なるVMを多数構築する場合、2台目以降の構築はほぼ全自動で実施できる。

4.1.2 並列実行機能

複数のサーバに対する構築処理を並列で実行する機能である。これによって、1台1台順に構築する場合と比べ、構築にかかる時間を大幅に短縮できる。並列処理数を手動調整することも可能なため、仮想化ソフトウェアの並列処理数の限界値に合わせて、可能な限り並列処理を実行することで、構築時間を短縮することができる。

4.2 定型的な運用作業の自動化

4.2.1 一括実行・順序実行機能

定型的な運用作業を複数のサーバに対し一括、又は順序に従い実行する機能である。

一括実行機能を活用することで、各VMに対し時間がかかる運用作業をまとめて指示できる。実行指示した作業はバックグラウンドで処理されるため、実行指示した管理端末の電源を切り、翌朝実行結果を確認するという使い方もできる。

また、運用作業順序に依存関係がある場合は、順序付けを行うことで、前の作業が終了してから次の作業を自動で実行することができる。幾つかのサーバで処理が失敗した時は、失敗したサーバだけを処理対象とする機能も持つ。

これらの機能によって、大量のサーバに対する運用作業を効率的に行うことができる。

また、被災時のBCP環境への切替え作業のような、非定

常的だが、定型的な運用作業を自動化することで、運用容易性を高めることができる。

4.2.2 テンプレート化支援機能

定型的な運用作業を自動化するために、ユーザーが記述する個別スクリプトファイルとOS・アプリケーション用の設定ファイルのテンプレート化を支援する機能である。ファイル中の動的に値を変更したい箇所(IPアドレスなど)をパラメータとして定義し、テンプレート用のフォルダへ保存しておく。このソフトウェアがこのテンプレートを使った運用作業を実行する時に、パラメータに適切な情報をセットし、VMへのコピーと実行作業を自動で行う。

この機能ではパラメータに加え、条件分岐(if文)と繰り返し(foreach文)の制御構文も記述することができる。

図2はこの機能の動作例である。httpd.confという設定ファイルのテンプレートに対し、IPアドレスとドキュメントルート文字列が設定される。

4.2.3 管理画面開発用通信インターフェース

このソフトウェアは、定型的な運用作業の自動化を指示する画面を利用者が自由に開発できるよう、通信インターフェースを公開している。インターフェース形式は、インターネットでのサービス公開インターフェースとして一般的なREST API(REpresentational State Transfer API)を採用することで、特定の開発言語に縛られることなく自由に開発することができる。

このソフトウェアが標準で提供する管理画面(図3)もこのAPIを使っている。

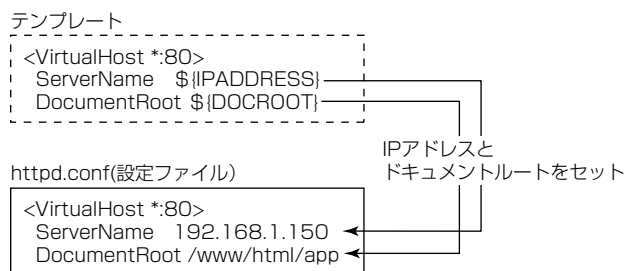


図2. テンプレート化支援機能

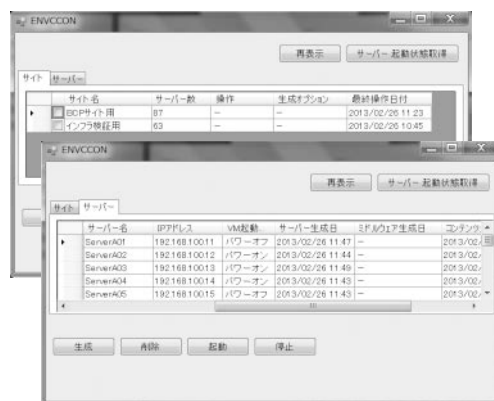


図3. 管理画面

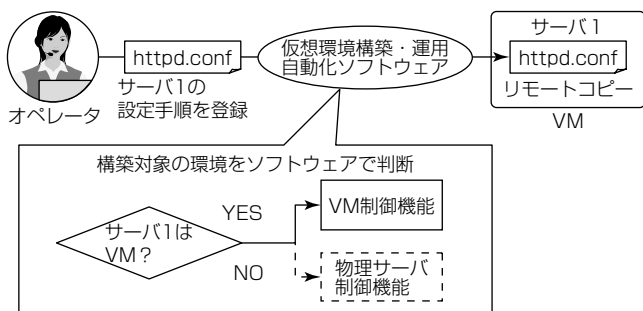


図 4. 仮想・物理混在環境での動作

4.3 仮想・物理混在環境への対応

4.3.1 VM・物理サーバ制御機能

このソフトウェアは、構築対象がVMか物理サーバかを意識することなくサーバを制御することができる。具体的にはVMの生成・削除・電源入・電源切・一覧取得とOSの再起動・ファイルのリモートコピー・リモート実行といった代表的な制御機能を抽象化し、ベンダー製品ごとの違いを知らなくともVM・物理サーバを制御できる。

図 4 は、この機能の動作例である。

このソフトウェアは構築・運用対象の環境に応じた制御手順を選択することができる。そのため、図 4 の例のようにhttpd.confという設定ファイルをサーバ1へリモートコピーしたい時、オペレータはサーバ1がVMか物理サーバか知らなくとも、サーバ1へリモートコピーを実施できる。

5. 適用事例と効果

大手上場企業の商品紹介Webシステムでの適用事例について述べる。このシステムは本番環境とそれに類似した環境(検証環境, 開発環境, BCP環境)を複数運用している(図 5)。そのため大量サーバの構築工数削減とミス低減, 運用の効率化が課題であった。このソフトウェアを活用することで, 大量のサーバの構築や定型的な運用作業を自動化し, 課題を解決することができた。

(1) 大量のサーバの構築自動化

この適用事例では, 環境構築後に本番環境の設定変更が発生し, 対応するBCP環境の作り直しも数回発生した。BCP環境も大量のサーバを持つため, すべてを手動で作直すことは非常に手間がかかる。しかしこのソフトウェアを適用することで, BCP環境の作り直し作業をすべて自動でかつミスなく実施でき(図 5 の①), その結果, BCP環境の再構築工数を大幅に削減できた(図 6)。

(2) 運用の効率化と運用容易性の向上

適用事例では, 被災時のBCP環境への切替え作業を仮想化ソフトウェアやこのシステムの専門知識がないオペレータでも迅速・確実に実施できることが求められた。

そこでBCP環境への切替え手順をこのソフトウェアへ登録し, 画面開発用インタフェースを使って被災時専用画面

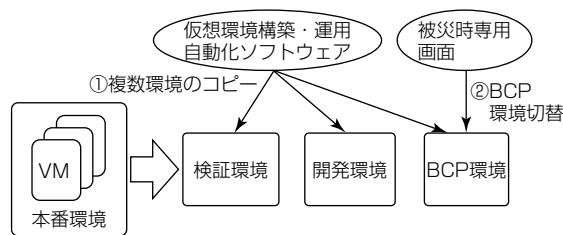


図 5. 適用事例



図 6. BCP環境の再構築工数削減効果イメージ

を提供した。その結果, オペレータは専用画面上のボタンを押すだけで切替え指示を確実に行うことができる(図 5 の②)。また, 幾つかのVMで処理が失敗した時は, “失敗したVMだけ再構築の処理対象にしたい”と指示するだけで, このソフトウェアが自動で対象を判断し切替え作業を再開することができるようになった。

さらに, 目標復旧時間(被災した後, 迅速にBCP環境へ切り替わるまでの目標時間)についても, このソフトウェアの並列実行機能を活用することで, 顧客の目標を達成した。

6. む す び

先に述べた適用事例を皮切りに, ITシステムの構築・運用の自動化を希望する顧客がMDISでも急速に増えてきている。

今後サーバ仮想化だけでなくネットワークやストレージの仮想化が進展すると, 仮想化ソフトウェアを使ったITシステムの構築・運用の自動化が今まで以上に可能となる。これは“Infrastructure as Code”⁽⁴⁾とも言われており, より大量のサーバの構築・運用が効率良く行えるようになる。MDISもこの潮流に追随するために, 更なる構築・運用自動化の機能拡充と対象プラットフォームの拡大を図っていく。

参考文献

- (1) IDC: Green IT: Where's the Competitive Advantage for CIO's?, Doc # DR2008_3MEW (2008)
- (2) 林 喜男: 人間信頼性工学-人間エラーの防止技術, 海文堂 (1984)
- (3) 一般社団法人 日本情報システム・ユーザー協会: ソフトウェアメトリクス調査2010 (2011)
- (4) Theo Sohlossnagle, et al.: Infrastructure as Code, InfoQ <http://www.infoq.com/presentations/infrastructure-as-code>