

異機種混合環境下でのシステム構築における サーバー・プロビジョニング機能導入によるコストと構築時間の削減

小林 誠士 佐貫 俊幸

Cost and Time Reduction to Configure Application Environment with Heterogeneous Platforms by Deploying Server Provisioning

Seiji Kobayashi Toshiyuki Sanuki

異機種混合のシステム環境下では、構成変更やソフトウェア導入、運用管理作業に多大な知識と労力を要する。そのため、ベストプラクティスに基づきサーバーやストレージ、ネットワーク機器の導入設定、構成を自動で行うことができるプロビジョニング機能を導入することで、労力を削減し、より効率的かつ安定的に環境構築・運用が行えるものと期待される。本論文では、サーバーを対象にプロビジョニング機能を実現するための環境を構築し、評価実験の結果をもとにプロビジョニング機能導入の効果について考察する。また、サーバー・プロビジョニングの費用対効果についても合わせて考察する。

It becomes more complex and time consuming that people change system configuration, install software and manage the systems under heterogeneous environment. In addition, a wide range of knowledge and highly-qualified skills are also required. Provisioning capability enables automated configuration of IT resources such as server, storage, network device and so on. Therefore it is expected to reduce workload and achieve more efficient and reliable system configuration and management by deploying provisioning function. In this paper, we introduce the overview of the server provisioning system which was implemented for evaluations and discuss some effects of deploying provisioning functions based on the results of the evaluation experiments. In addition, we also examine the cost effectiveness of the server provisioning deployment.

Key Words & Phrases : オートノミック・コンピューティング , オンデマンド・オペレーティング環境 ,
サーバー・プロビジョニング , TIO
autonomic computing, on demand operating environment,
server provisioning, Tivoli Intelligent Orchestrator

1. はじめに

異機種混合で構成されるシステムでは、その複雑化に伴い、環境構築や構成変更作業に多大な知識と労力を要する。一方で、ITの急速な需要拡大に伴い、国内外でIT技術者が不足していることが報告されている[1][2]。そのため、これら複雑なシステムの構築作業や運用管理を、自動的にまた柔軟に行うことができるオンデマンド・オペレーティング環境(On Demand Operating Environment: odOE [3][4])の実現が強く望まれている。ここで、odOEを実現する上で重要な技術のひとつにプロビジョニング機能(第2章参照)が挙げられる。プロビジョニング機能では、

ベストプラクティスに基づいて必要な環境を自動的に構築するため、複雑なシステムを構築する際の作業効率や品質の向上を図れるものとして期待されている。特に、多数のサーバーを保有するデータセンターや開発・テストセンターでは、提供するサービスの保守運用や検証内容に応じ、検証環境の構築を繰り返し行う必要がある。そのため、そのような環境構築作業をプロビジョニング機能により自動化できることのメリットは大きい。

今回筆者らは、お客様に対しIBMミドルウェアのソリューション稼働検証やデモンストレーションの場を提供する渋谷ソフトウェア・コンピテンシーセンター(SWCOC [5])において、Tivoli® Intelligent Orchestrator [6][7][8]を用いてサーバー・プロビジョニング機能を提供できるサーバー・プロビジョニング環境を

提出日: 2004年08月31日 再提出日: 2005年8月30日

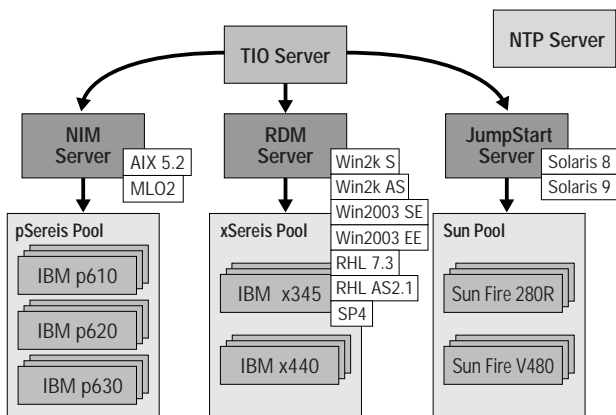


図1. プロビジョニングシステム概要

を有していないため、異機種混合環境をサポートするために、別途OS導入用の独立したブートサーバーを構成した。この中にはNIM(Network Installation Management)サーバー、RDM(Remote Deployment Management)サーバー、JumpStartサーバーの3種類が含まれる。このような構成をとることで、異機種混合の環境においても、サーバー・プロビジョニング機能からは等価な処理で各サーバーのセットアップ作業を自動で行うことができるようになる。現在、このサーバー・プロビジョニング環境で自動導入できる各種OS、パッチは、図1中の各ブートサーバー上に記したものである(OS 9種類、パッチ2種類)。また、ミドルウェアについては独自のワークフローを作成することでWebSphere Application Server (WAS)5.01、DB2® 8.1の導入を可能とした。RAID構成(xSeries®のみ)、論理区画・ファイルシステムの作成(pSeries®のみ)といったディスク操作を行うためのワークフローも同時に開発し、サーバーのローカルストレージの制御も可能にした。サーバーの設定直後にはNTPサーバーと連携し、自動的に時間調整を行う。

TIOを用いてインストール作業を行う場合には、図2にあるようにWebブラウザ上のメニューより、あらか

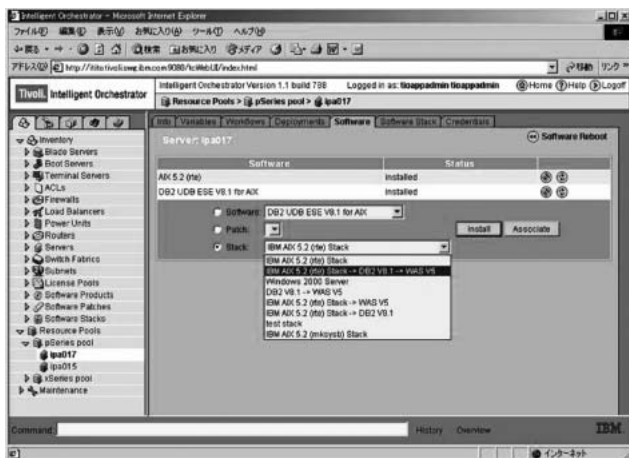
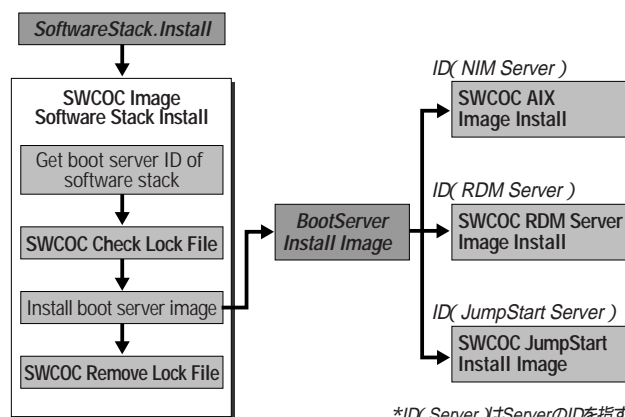


図2. Webブラウザ経由での導入作業

じめ登録しておいたOSやミドルウェアを選択するだけで、実際の導入作業を行うことができる。

4.2 ワークフロー概要

TIOでは、OSやミドルウェアの導入手順をワークフローとして記述している。ワークフローでは、サーバーやスイッチ、OS、パッチ、ミドルウェアといったTIOの管理対象となるリソースへの処理をロジカル・オペレーションという抽象化された処理で記述する。これにより、ワークフローのポータビリティの向上を図っている。図3は、TIOにおけるOSインストール用ワークフローの構成を模式的に表している。このワークフローでは、最上位においてSoftware Stack Installというロジカル・オペレーションを呼び出し、その実体よりBootServer.Install Imageというロジカル・オペレーションを呼び出している。これは、「ある選択されたサーバーに対し、ブートサーバー上のイメージをインストールせよ」という意味の記述であり、導入するOSが変わっても同一の記述となる。一方、OS毎に異なる具体的な導入手順は、それぞれ別のワークフローとして定義しておき、その内のどのワークフローが実際に呼び出されるかは、BootServer.Install Imageに渡されるブートサーバーのIDによって決定される。なお、ブートサーバーのIDはTIOサーバー内で管理されている。



*ID(Server)はServerのIDを指す

図3. ワークフローの構成概要

5. プロビジョニング機能導入効果の測定

5.1 導入効果測定方法

サーバー・プロビジョニング機能の導入効果を測定するために、同一構成のサーバーをセットアップするのに要する時間を、手作業で行う場合とサーバー・プロビジョニング機能を用いて行う場合とで測定し、その比較を行った。具体的には、評価ケースとしてIBM pSeries 610に対しAIX5.2Lを導入し、それに引き続きDB2 8.1の導入を行った。それぞれの複数被

BM2に比べて大幅に短縮されることになる。しかしながら、この場合、CDメディアの交換やパラメータ入力など、人手による操作を必要として導入作業が一時的に中断されるタイミングと、作業者が再び作業場に戻ってくるタイミングとでギャップが生じ、導入作業中にいわゆる“待ち”の状態が発生するため、全体として導入作業時間M1が延びてしまう結果となる。

一方、プロビジョニング機能を用いた場合には、いったん初期の導入作業を行えば、それ以降はワークフローに記された手順に従って一連の環境構築作業が自動的に行われる。そのため、途中で人手を介する操作は一切生じず、作業者の拘束時間を大幅に削減することができる(BA)。さらに、CDとネットワーク経由でのデータ転送速度の違いにより、“待ち”の時間を除いた導入作業そのものの所要時間も短縮できている。今回の実験環境(p610)では、CDの転送速度(1.2Mbps)が48倍速、ネットワークカードの転送速度が100Mbpsであり、実効で80Mbps程度の転送が実現されていると仮定すると、

$$\frac{80}{1.2 \times 4.8} \cong 1.39 \quad (1)$$

となり、データ転送の速度差だけで約1.4倍の導入時間短縮を達成できていると見込まれる。

表2は、今回の評価実験において、手作業によるAIX、DB2導入にかかった時間の平均と最短および、プロビジョニング機能を用いた場合の導入にかかった平均時間をまとめたものである。

表2. OS ,ミドルウェア導入時間の比較

導入時間	AIX(分)	DB2(分)
手動(平均)	88	49
手動(最短)	57	27
プロビジョニング	46	18

最短時間で作業を終了している場合には、導入作業中に“待ち”の時間がほとんどなかったと仮定すると、表2中のAIX、DB2それぞれの導入時間をみた場合

$$57 / 46 = 1.24 \quad (\text{AIXの場合}) \quad (2)$$

$$27 / 18 = 1.5 \quad (\text{DB2の場合}) \quad (3)$$

となり、導入時間の短縮効果が式(1)の見積もり(1.39倍)とほぼ合致することが分かる。

6.2 プロビジョニング機能導入の費用対効果

サーバー・プロビジョニング機能を導入するにあたっては、その初期導入コストを考慮する必要がある。すなわち、サーバーのセットアップ回数が極めて少ないに

も関わらず、プロビジョニング機能を導入するのはあまり効果的とは考えられない。そこで、ここではサーバー・プロビジョニング機能の導入コストをその環境構築に要する時間としてとらえ、実測結果をもとに、およそ何台程度のサーバーをセットアップすれば、プロビジョニング機能の初期導入コストを回収できるかについて検討する。

サーバー・プロビジョニング機能の実装としては、AIXマシン2台でTIOサーバーを構成し、OSインストール用のNIMサーバーも合わせ、計3台のマシンを準備するものとする(表3)。

表3. サーバー・プロビジョニング環境構築例

サーバー	導入ソフトウェア
TIOサーバー	AIX Tivoli Intelligent Orchestrator WebSphere Application Server DB2 LDAP Client Utility Software
TIO LDAPサーバー	AIX DB2 LDAP Server Utility Software
NIMサーバー	AIX NIM Server AIX Image

ここで、導入作業を行う場合の作業時間を

$$T = T + (N) \quad (4)$$

$$(N) = kN^b \quad (5)$$

と見積もることとする。ここで、 T は導入作業完了までの時間、 T はファイル転送など導入完了までに最低限必要な時間を表している。また、 (N) は、導入作業を繰り返すことによる経験曲線効果[11]を表しており、 N は導入したサーバーの累計台数、 k 、 b は定数である。

ここで、サーバー・プロビジョニング環境構築の際に導入するミドルウェアや追加ソフトウェアの導入コストを、評価実験におけるDB2の手動導入コスト(T_{mw})と同等であると仮定する。また、NIMサーバー上にAIXのイメージを作成するコストを、手動でOSを導入する際に最低限必要なコスト(T_{os})と同等であると仮定する。すると表3より、サーバー・プロビジョニング環境を構築するために必要な初期コストは、

$$T_{os} = \sum_{i=1}^3 T_{osi} + \sum_{j=1}^9 T_{mjw} + T_{os} \quad (6)$$

と表すことができる。ここで、 T_{os} はOS導入コストである。簡単のため、構築したサーバー・プロビジョニング環境を利用してOSの導入のみを繰り返すとすると、プロビジョニング機能を利用してOS導入をする場合と、

のワークロード削減にも大きく貢献している。

謝辞

サーバー・プロビジョニング環境構築ならびに導入効果の測定にあたり、ご支援・ご協力いただいたIBM SWCOCスタッフ、特に渡邊宗考氏、吉田剛氏、Tivoli技術部 橋由記子氏、幕張技術サポート 松永泰彦氏に深謝いたします。

参考文献

- [1]経済産業省 平成16年版 情報通信白書
- [2]<http://www.nikkei.co.jp/news/main/20040802AT2M2902W01082004.html>, 2004.8.31
- [3]B.Jacob, et al., "On demand Operating Environment: Creating Business Flexibility", IBM Redbook SG246633, 2004.4.27
- [4]B.Jacob, et al., "On demand Operating Environment: Managing the infrastructure First Edition", IBM Redbook SG246634, 2004.4.27
- [5]Software Center of Competency, <http://www-6.ibm.com/jp/software/swcoc/>, 2004.8.31
- [6]IBM Tivoli Intelligent Orchestrator Operator's Guide Version 1.1.0, IBM, 2003
- [7]E.Manoel, et al., "Provisioning On Demand - Introducing IBM Tivoli Intelligent ThinkDynamic Orchestrator", SG248888, 2003.12.21
- [8]E.Manoel, et al., "IBM Tivoli Intelligent ThinkDynamic Orchestrator - Pre Proof-of-Concept Cookbook for Business Partners", IBM Redbook REDP3830, 2004.2.10
- [9]オートノミック・コンピューティング 日本IBM社内事例, <http://www-6.ibm.com/jp/autonomic/casestudies/swcoc.shtml>, 2004.8.31
- [10]SWCOCシステム環境, <http://www-6.ibm.com/jp/software/swcoc/platform/>, 2004.8.31
- [11]B.Henderson, "The Experience Curve", http://www.bcg.com/this_is_bcg/mission/experience_curve.jsp, 2004.8.31



日本アイ・ビー・エム株式会社
大和ソフトウェア開発研究所
専任開発技術担当部長

小林 誠士 Seiji Kobayashi

[プロフィール]

1996年日本IBM入社。基礎研究所にて著作権保護技術の研究・開発に従事。2002年よりエマージングビジネス(EBO)にてGrid/Autonomic Computingを用いた先進ソリューションの開発を推進。現在は、ソフトウェア開発研究所にてAutonomic Computing技術開発・先進ソリューション開発を担当。情報処理学会会員。



日本アイ・ビー・エム株式会社
大和システム開発研究所
シニア・テクニカル・スタッフ・メンバー

佐貫 俊幸 Toshiyuki Sanuki

[プロフィール]

1983年日本IBM入社。プログラミング言語の研究、Video On Demand(VoD)やコンテンツ配信などのデジタルメディア・システムの開発・構築などに従事。1997年米国IBMに出向。Emerging Business(EBO)を推進。現在は次世代マイクロプロセッサ応用システムの企画、開発を担当。情報処理学会、IEEEの会員。