



オーケストレーション

佐藤陽一 ((株)オープンシステムズラボラトリ)

ysato@opensystemslab.net

1. オーケストレーションとは

オーケストレーションとは、複数のネットワークやサーバ、ソフトウェア等を組み合わせて目的とするサービスや機能を、生成し、運用管理し、最終的には消去するまでのライフサイクルを制御・管理する仕組みである。図1にオーケストレータを含むサービス管理の概念図を示す。

オーケストレータはOSS/BSS (Operation Support Systems/Business Support Systems) と連携しながら、ネットワークやクラウドシステム、アプリケーションを連携させ、サービス等の提供を行う。具体的には、OSS/BSSからの指示により、事前に指定された手順(ワークフロー)に従い、ネットワーク装置の設定、サーバのOSのインストールや設定、アプリケーションのインストールや設定をソフトウェアが自動で行う。装置の設定やアプリケーションのインストール等は各種コントローラを介して行うのが一般的である。オーケストレータがないと、ネットワーク機能、クラウド機能やアプリケーション等の特性の異なる機能の連携といった複雑な運用を人手で行うことになり、高度なスキルを有する人材の確保や人為ミスの発生の懸念などが課題となる。それに対しオーケストレーションを用いる運用システムにおいては、自動化による運用の簡易化、人為ミスの軽減等の効果が期待できる。

2. オーケストレーションの歴史的背景

オーケストレーションの概念がネットワークやサービスの運用技術に導入された契機は、SDN (Software Defined Networking) の考え方がネットワーク運用にも浸透したことが大きい。

SDN以前は、伝送装置等の通信装置においてはGUIを介して人手による運用が一般的であった。直感的に分かりやすいHMI (Human Machine Interface) により、操作性向上や人為ミスを低減する方向で技術検討が行われていた。それに対し、ルータやスイッチ等の操作はCLI (Command Line Interface) が一般的で、ネットワーク全体を俯瞰する視認性は劣るものの、大量の設定をミスなく行うために、スクリプトによる一括設定を行うという工夫が行われていた。

例えば、汎用的なスプレッドシートのマクロ機能を用いて、設定に必要なパラメータを投入すれば、それに対応した一連のコマンドラインが自動生成されるといったツールを用いて、ミスなく大量のコンフィグ投入が行われていた。

一方、クラウドコンピューティングの運用では、数百、数千以上に及ぶ大量のサーバやソフトウェア、更にネットワーク機器の設定を行う必要があり、商用導入の初期段階からソフトウェアによる設定、監視、運用の自動化が行われていた。それゆえ、少人数で大規模なクラウドプラットフォームの

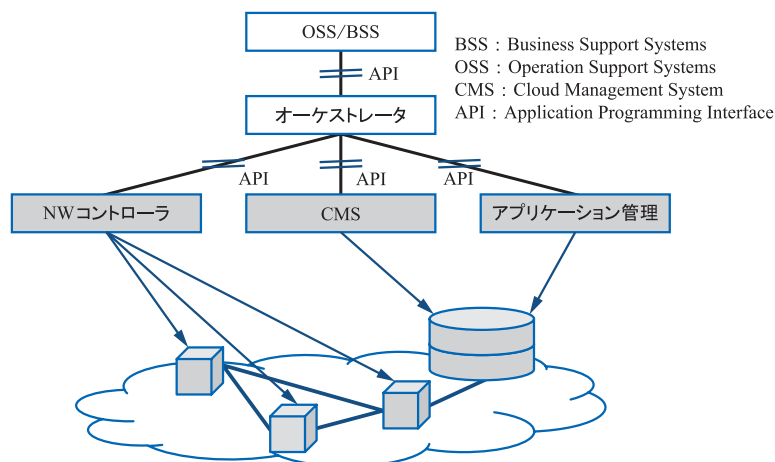


図1 サービス管理の概念図

構築，運用が可能となっている。

SDN はデータプレーンのプログラマビリティの実現を主目的として考案されたが，データプレーンを制御するコントローラが上位の管理層に対し API (Application Programming Interface) を提供する実装が一般化したため，外部からのソフトウェアによるネットワーク機能の制御が，HMI や CLI と比べて容易になるという効果がある。

クラウドで実績を積んだ自動化技術，SDN の出現で，ネットワークの設定，運用の自動化の道具立てがそろったと言える。

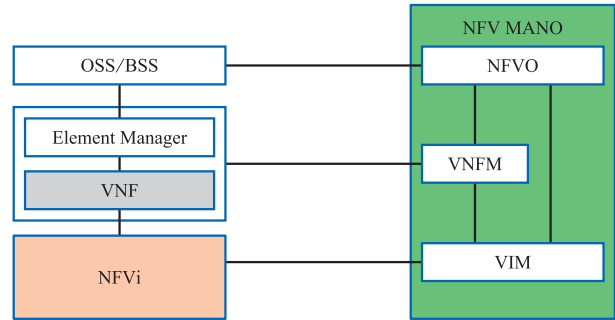
SDN やクラウドコンピューティングのメリットとしてネットワークや計算機リソースを仮想化するという機能がある。しかしながら，仮想化されたインスタンス (実体) を用いてサービスを構成する場合，その設計・運用は複雑化するというデメリットも生じる。更に，OS 上に構築された仮想計算機 (VM : Virtual Machine) に対し，コンテナによる仮想インスタンスを階層的に構築するという利用形態も現れ，人手に頼った運用はますます困難になったと言える。

以上述べたようにサービスを提供するという観点からは，ネットワーク単体の機能をサービスとして提供するシンプルな提供形態から，ネットワーク機能にクラウド機能，アプリケーション等を付加してサービス提供する形態に変化してきている。特性の異なる複数の機能要素を組み合わせるサービス提供するという一方で，高付加価値サービスを訴求できるというメリットがある反面，設計や運用が複雑化するという課題に直面していた。

3. オーケストレーションの例

オーケストレーション機能を管理システムのアーキテクチャに組み込んだ例として NFV (Network Functions Virtualisation) が挙げられる。図 2 に NFV のアーキテクチャを示す⁽¹⁾。NFV においては，仮想サーバ，仮想ネットワーク，更にはアプリケーションを連携動作させて要求された機能の提供を行う必要があり，そのため，NFV MANO (Management and Orchestration) に NFVO (NFV Orchestrator) というオーケストレーション機能が，VNFM (VNF Manager)，VIM (Virtualised Infrastructure Manager) を介してネットワーク機能，仮想サーバを制御するというアーキテクチャを採用している。

オープンソースで提供されているクラウド管理システムである OpenStack において，Heat⁽²⁾ と呼ばれるモジュールがオーケストレーション機能の位置付けとなっている。単に VM を生成，消去するのであれば静的なスクリプトでも実現可能であるが，システムの負荷状況に応じて動的にインスタンスを増減し，自動でスケールイン，スケールアウトを実現するような場合に Heat を用いることがある。負荷が増加し，VM を追加して負荷分散を行うようなケースにおいて，VM に割り振られる IP アドレスが VM 生成後に決まるよう



BSS : Business Support Systems
NFVi : Network Function Virtualised Infrastructure
NFVO : NFV Orchestrator
OSS : Operation Support Systems
VIM : Virtualised Infrastructure Manager
VNF : Virtualised Network Function
VNFM : VNF Manager

図 2 NFV のアーキテクチャ

な場合，負荷分散する VM の IP アドレスを，事前にスクリプトに記述することは難しいが，Heat を用いることで VM 生成後の IP アドレスを知り，その値を負荷分散の IP アドレスとして指示することが可能となる。このようにオーケストレーション機能は動的なシステム運用において，真価を発揮することができる。

4. まとめ

ネットワーク技術の進化に加え，仮想化されたネットワーク，サーバ，更にアプリケーションを組み合わせることで，高機能で使い勝手の良いサービスの提供が実現できる。その反面，それを支えるシステム全体の設計，運用，管理は複雑化していく。運用の現場では複雑化するシステムに対し，様々な工夫で乗り越えてきたが，本稿ではオーケストレーションという切り口で概要を説明した。今後は自動化を更に進化させた自律化の研究や，AI/ML の導入も行われていくと思われる。最後にネットワークの運用管理技術を体系的に学ぶ上で役立つ参考書^{(3),(4)}を紹介してこの小文の終わりにしたい。

文 献

- (1) NFV.
<https://www.etsi.org/technologies/nfv>
- (2) OpenStack Heat.
<https://wiki.openstack.org/wiki/Heat>
- (3) J. Edelman, S.S. Lowe, and M. Oswalt, Network Programmability and Automation : Skills for the Next-Generation Network Engineer, O'Reilly, 2018.
- (4) Y. Zhang, Network Function Virtualization : Concepts and Applicability in 5G Networks, IEEE Press, 2017.

(2023 年 2 月 22 日受付 2023 年 5 月 30 日最終受付)