

サービス指向の IaaS クラウドシステム環境の構築

大山 裕泰, 杉木 章義, 加藤 和彦

筑波大学大学院 システム情報工学研究科

概要

近年の IaaS クラウドサービスの成長とともに、これを実現するシステム環境は複雑化している。こうした大規模なシステムの構築には、従来は仮想計算機 (VM : Virtual Machine) とネットワークをそれぞれが用意する API を利用していたが、これらを別々に設定することによって管理コストが増大する点で問題がある。本研究では、VM やネットワーク環境において運用されるアプリケーションサービスを中心としたシステム環境の構築手法を提案することで、システム環境の構築に伴う記述を簡略化する。

1. 背景

クラウドと呼ばれるインターネットを介したシステムを提供するサービスが成長している。クラウドサービスのうち IaaS (infrastructure-as-a-service) 型クラウドでは、仮想計算機 (VM : Virtual Machine) をサービスとして提供する。

また、こうした IaaS 用クラウド基盤環境の構築・運用・管理を行うためのソフトウェアがいくつか [1, 2, 5] 開発されている。

従来の IaaS 用のクラウド基盤ソフトウェアや、Amazon EC2 [3] などの IaaS サービスでは、API を通じて VM 及びネットワークを設定する機能を提供している。例として OpenStack では、当該クラウド基盤が用意する VM 用の API (Nova API) によって VM の生成・起動を行い、また別の仮想ネットワーク用の API (Quantum API) により仮想ネットワーク及び仮想ポートの設定を行う。また CloudStack や Amazon EC2 でも、同じような API を提供している。

ただこうした VM とネットワークの設定を別々に設定することにより、システム環境の構築に伴う設定が煩雑になる。また、VM の追加や削除といったシステム環境の変更を行う場合にも、これらの API を介した再設定が必要になり、システム規模が大きくなるほど、設定の保守に大きなコストが掛かってしまう。

こうした手法の背景には、クラウドコンピューティングにおいて、OS とネットワークが別のコミュニティで研究され発展してきたことが考えられる。従来 OS のコミュニティでは、VM に注目し、仮想インターフェイスよりも上位レベルの研究が主に行われてきた。ネットワークのコミュニティでは、仮想インターフェイスよりも先のネットワークの管理に関心があった。

しかし、クラウドコンピューティングの本来の目的はサービスの提供であり、両者が協力してと取り組んでゆく必要が

ある。その一環として、クラウド基盤におけるシステム環境の構築を、これまでの VM やネットワークといった個別の要素に対する設定を別々の手法で行うのではなく、サービスを中心とした VM 及びネットワークを扱う手法について考える必要がある。

2. 提案手法

本研究では、ユーザが本来実現したサービスを中心に設定をすることで、これに基づき実現させたいサービス環境に適した VM 及びネットワークの設定を行ったシステム環境を構築する手法を提案する。

ここでサービスとは、クラウド基盤上で運用するインターネットサービスを構成する、個々のアプリケーションサービスの実行環境のグループを表す。サービスは、実際にアプリケーションサービスを提供する実行環境である 1 つ以上の VM から構成される。また、サービスの概念的な接続関係を表すものをサービスパスと定義する。また、サービスとサービス間の接続関係を記述したファイルをサービスポリシーと定義する。

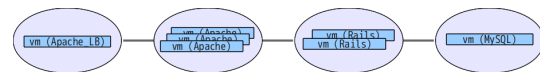


図 1. サービスとサービスパス

図 1 は、web アプリケーションサービスを提供するシステム環境における例を示す。それぞれの丸で囲まれたものがサービスを表し、各サービスはサービスパスによって接続される。サービスパスは、サービスに所属する各 VM の接続範囲を決定するもので、VM は所属するサービスのサービスパスが接続するサービスに属する VM との間のみ接続することができる。

我々は、ユーザが記述したサービスポリシーに基づき VM 及び仮想ネットワークを設定することでサービスの提供に適したシステム環境の構築を行う。また構築したシステム環境において、ポリシーに基づいた VM 間通信が行えるようネットワークの制御を行う。

こうしたサービスを中心としたシステム環境の構築により、各サービスを提供する VM やサービス間の仮想ネットワークの設定を各 VM 毎に行う設定を抽象化でき、システム環境を構築するための設定の記述が簡略化される。また、運用環境に適した VM の数をシステム側で自動調節することにより、トラフィックに応じた VM の追加や削除に伴う設定を行う必要がなくなる。

3. 実装

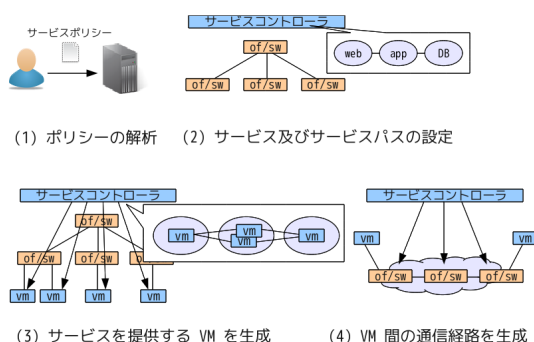


図 2. 提案手法におけるシステム構築の流れ

本研究室が開発を行なっているクラウド基盤ミドルウェア Kumoi [5] を利用することで、本提案システムを実現する。

Kumoi は Scala 言語で実装されたクラウド基盤ミドルウェアであり、主な特徴としては VM やネットワークといったリソースが静的に型付けされた Scala オブジェクトに抽象化し、それらを CRUD (Create, Read, Update, Delete) の統一したインターフェイスから操作することができる。さらに、Scala が持つ map() や filter() といった関数型演算を利用することで、ある VM リストに対して特定の条件の VM に対して処理を行うといったことが簡単に記述することができる。また、クラウド基盤におけるシステム環境の構築・管理を行うためのシェル環境を提供しており、クラウド基盤を提供する事業者は、Kumoi が提供するシェル環境から VM ・ネットワークそれぞれに対して柔軟な制御を行うことができる。

また、クラウド基盤環境における仮想ネットワークの制御では OpenFlow [4] をもとに実装されており、VM が接続する仮想 OpenFlow スイッチと、OpenFlow コントローラのオブジェクトをそれぞれ提供している。これにより、システム環境における動的な VM 間の通信や、各サービスで稼動す

る VM の数の動的な伸縮といった柔軟で拡張性のあるネットワークの制御を行うことが可能となる。

本提案システムはこうした Kumoi シェル環境を利用し、サービスポリシーからシステム環境を構築する。

図 2 は、本提案でのサービスポリシーからクラウド基盤環境においてシステムを自動的に構築し、サービス間での通信を行うまでの処理について示したものである。

図中のサービスコントローラとは、VM とネットワークを管理するためのオブジェクトである。(1) サービスコントローラは、まずポリシーを解析し、(2) サービス及びサービスパスオブジェクトを生成する。そして、各サービスオブジェクトに対して、(3) アプリケーションサービスの実行環境である VM を生成し、それらを接続する(4) 仮想ネットワークを、サービスパスオブジェクトに基づいて設定する。また(4) では、サービスのリソースの状況に応じて、ポリシーに基づいた VM の追加・削除を動的に行う。

さらに、以上のネットワーク制御は OpenFlow を用いて行う。

4. 現状と今後の予定

現在、ポリシーに即したサービス間通信を実現する OpenFlow ネットワークにおける経路設定の実装を終えている。今後は残りの実装を行うとともに、実験及び評価をおこなってゆく予定である。

参考文献

- 1) OpenStack. <http://www.openstack.org/>.
- 2) CloudStack. <http://cloudstack.org/>.
- 3) Amazon EC2. <http://aws.amazon.com/ec2/>.
- 4) Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker and Jonathan Turner, *OpenFlow: Enabling Innovation in Campus Networks*, ACM SIGCOMM Computer Communication Review, April 2008.
- 5) Akiyoshi Sugiki, Kazuhiko Kato, Yoshiaki Ishii, Hiroki Taniguchi and Nobuyuki Hirooka, *Kumoi: A High-Level Scripting Environment for Collective Virtual Machines*, IEEE International Conference on Parallel and Distributed Systems, December 2010.
- 6) Theophilus Benson, Aditya Akella, Anees Shaikh and Sambit Sahu, *CloudNaaS: a Cloud Networking Platform for Enterprise Applications*, ACM Symposium on Cloud Computing, October 2011.