複数クラウド間でスケールアウトやディザスタリカバリを実現するクラウド連携マネージャの設計と実装

武田健太郎† 伊藤雅典† 山中顕次郎† 村上明彦†
†株式会社NTTデータ 技術開発本部

1. はじめに

システムの基盤として、仮想マシンを貸し出すクラウドサービス(IaaS)の利用が拡大している。クラウドは多くのシステムの統一基盤であり、クラウドにおける障害やリソース不足の問題は、そのクラウド上で動作する多数のシステムに影響を及ぼす。しかし、各クラウドは独立に提供されており、あるクラウドで発生した問題を他クラウドの補助により解決するクラウド連携の仕組みは実現されていない。

筆者らは、クラウド同士を自律的に連携させリソースを融通させるクラウド連携マネージャを設計しプロトタイプを開発した。クラウド連携マネージャは自クラウドのリソースや自クラウド上で動作しているシステムの構成情報を管理し、他のクラウド連携マネージャと交換し合う。交換した構成情報を元にシステムの構成を制御することで、クラウドを跨いだスケールアウトやディザスタリカバリを実現する。

2. システムモデル

2.1. 上位システム

クラウド上で動作するシステムを特に上位システムと呼称する。上位システムはクラウドで起動した仮想マシン上のサービス群の協調動作によって実現される(図 1)。クラウド上で動作するシステムであればいずれも上位システムと呼称するが、本研究では特に、各層が冗長化されたWeb3 層システムとHadoopによる分散処理システムの2種類を典型例とする。

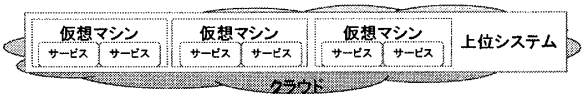

図 1 上位システム

The design and implementation of "Cloud Federation Manager", which enables scale-out and disaster recovery between cloud systems.
Kentaro Takeda†, Masanori Itoh†, Kenjiro Yamanaka†, Akihiko Murakami†
†Research and Development Headquarters, NTT DATA CORPORATION

2.2. 前提条件

上位システムを構成するデータは、仮想マシンのディスクイメージや仮想ストレージに含まれるユーザデータと、構成情報などのメタデータに分類できる。クラウド連携のためには両方のデータをクラウド間で同期しておく必要がある。本稿ではメタデータを動的な同期対象として扱い、ユーザデータはスケールアウト・ディザスタリカバリ等のイベント発生前にあらかじめ同期が完了しているものとする。

3. 課題

3.1. 上位システムの構成情報の記述

複数クラウド間で上位システムの情報をやりとりするために、上位システムの構成情報の統一した記述手法が必要である。構成情報には、サービスの仮想マシンへの配置、サービスの基本的な設定情報や依存関係、仮想マシンの必要スペック、主要なネットワーク設定などの情報が記述できる必要がある。

3.2. 上位システム固有の手順の記述

上位システムの起動・停止・スケールアウト・ディザスタリカバリなどのイベントを実現するには、構成情報に記述した静的構造に加え、その上位システムに固有の手順を記述できる必要がある。手順には、仮想マシンやサービスの起動・停止、ネットワークの設定、サービスの設定値の動的な更新、構成情報の参照・更新などを記述する。手順の記述を容易にするために、クラウドや仮想マシンに対する操作を抽象度の高いAPIで記述できる必要がある。

3.3. 構成情報と手順のクラウド間交換

複数クラウドで上位システムの構成を制御するには、構成情報や手順の実行状態などの上位システムに関する情報を、関連するすべてのクラウド連携マネージャ間で共有する必要がある。3 つ以上のクラウドでの連携や、連携するクラウドの増減に対応するため、特定の上位システムの情報を任意の数のクラウド連携マネージャで漏れなく共有できる、柔軟な情報交換の仕組みが必要である。

4. 設計と実装

クラウド連携マネージャの全体アーキテクチャを図 2に示す。クラウド連携マネージャはいくつかのインタフェースと構成管理・構成制御の 2 つのコンポーネントで構成される。

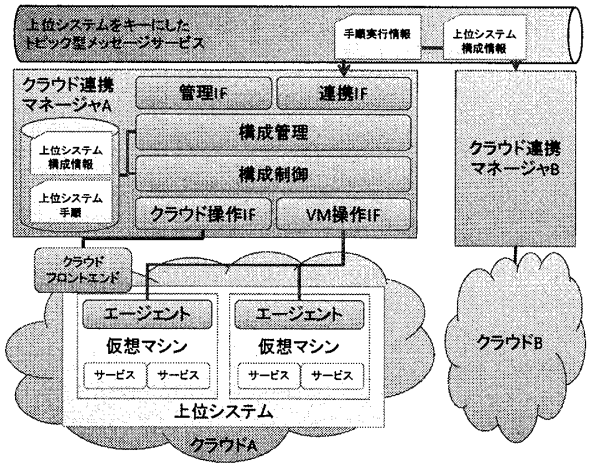

図 2 全体アーキテクチャ

構成管理は上位システム管理者が作成した構成情報や手順、連携インタフェース経由で取得した他クラウドでの上位システムの情報を読み込み、モデルオブジェクトに変換し DB に保存する。構成制御は構成管理により提供されたモデルオブジェクトを元に、手順実行エンジンを用いて上位システムの構成を制御する。手順の実行には多くの場合仮想マシン上での操作が必要になるため、仮想マシンには専用のエージェントをあらかじめ導入しておく。

プロトタイプの実装と検証には Eucalyptus 1.5.2 で構築した 2 つのクラウドを使用した。Eucalyptus は Amazon EC2 と API レベルで互換性を持つクラウド基盤 OSS である。

4.1. 構成情報の記述方式

構成情報の記述方式にはOVF[1]を使用する。OVFは仮想マシンの相互利用性の向上を目的として公開されている仮想マシンのパッケージング仕様であり、単一の仮想マシンだけでなく、複数の仮想マシンから構成されるシステムを記述できる。サービス間の依存関係や基本的な設定情報、設定ファイル等の外部参照情報も含めることができ、本研究で対象とする上位システムの構成情報の記述に適した仕様である。

[1] Open Virtualization Format, http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

4.2. 手順の記述方式

手順には基本的な制御構造に加え、表 1に示すようなモデルオブジェクトが利用できる。さらに、表 2に示すような手順APIが構成制御から提供されており、上位システムの起動・停止・スケールアウト・ディザスタリカバリの手順を抽象度高く記述できる。加えて、テンプレートエンジンの機能を利用することで、設定値を更新した設定ファイルをテンプレートから生成しサービスに反映する、という操作を手順中に記述できる。

表 1 モデルオブジェクトの一部

モデルオブジェクト名	説明
BusinessSystem	上位システム
BusinessSystemElement	サービスの雛型
BusinessSysmteElementInstance	サービスの実体
ConfigTemplate	設定ファイルテンプレート
VMClass	仮想マシンの雛型
VMInstance	仮想マシンの実体

表 2 手順 API の一部

API名	説明
run_vm	仮想マシンを起動する
attach_volume	仮想マシンにストレージを取り付ける
associate_address	仮想マシンにIPアドレスを割り当てる
run_agent_command	仮想マシン上でコマンドを実行する
put_file	仮想マシンにファイルを置く
async	手順ブロックを別スレッドで動作させる
wait	手順ブロックの終了を待つ
publish_cloud_event	上位システムに関する情報を出版する
wait_cloud_event	特定の情報が出版されるのを待つ

4.3. トピック型メッセージサービス

クラウド連携マネージャ間の情報交換には、トピック型のメッセージサービスを利用する。上位システムの識別子をキーとしてメッセージサービスに対し購読・出版することで、特定の上位システムに関する情報を複数クラウド連携マネージャ間で漏れなく共有できる。購読・出版の操作は、表 2に示したAPIを用いて手順中に記述できる。

5. まとめと残課題

プロトタイプの実装と評価により、設計したアーキテクチャでクラウドを跨いだスケールアウトやディザスタリカバリが実現できることを確認した。残課題としては、性能など非機能面での評価、仕様の異なるクラウド間での連携の実現、上位システムのユーザデータ同期の自動化などが挙げられる。

謝辞

本研究は総務省「セキュアクラウドネットワーキング技術の研究開発（クラウドサービス連携技術）」委託研究による研究成果です。