

AWS と Azure での高度なオートスケーリングの実現と比較評価

1220333 近藤 雅文 【分散処理 OS 研究室】

1 はじめに

パブリッククラウドコンピューティングのサービスの一つである仮想コンピューティング環境ではオートスケーリング機能が提供されている。AWS における先行研究 [1] では、急激なアクセス負荷に対応するため、SDK を用いた高度なスケーリングポリシーを実装することで問題に対応した。しかし、高度なスケーリングポリシーを実装するためには、どの仮想コンピューティング環境が優れているのかが明確になっていない。本研究では先行研究 [1] で AWS に実装された高度なスケーリングポリシーを Azure に実装し AWS と比較評価する。

2 システム構成

Azure に実装したシステムの構成を図 1 に示す。ロードバランサ経由で VM に負荷が掛けられ、CPU 使用率を監視していた Monitor による警告で Function App が呼び出される。Function App が SDK で API を操作し、Monitor からメトリック取得、計算、VM Scale Sets に対するスケーリングを実行する。スケールアウトの際に起動する VM には docker コンテナ上アプリを立ち上げる。これらは先行研究で AWS に実装されたものと同様になった。しかし、Function App にスケーリングポリシーを実装する際の言語、SDK、API にはそれぞれ特徴がみられた。

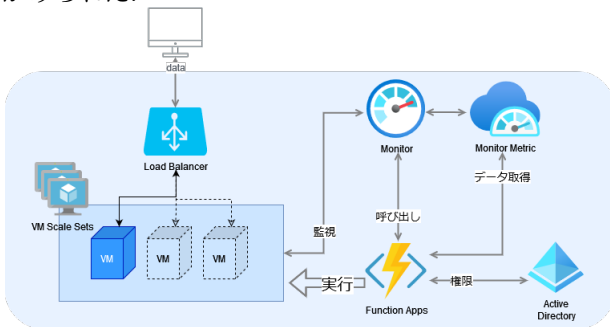


図 1 Azure の構成

3 評価

AWS と Azure におけるスケーリングポリシー実装の特徴を比較評価する。また、Apache Bench を用いて負荷をかけ、スケーリングについて比較評価する。負荷モデルとして、図 2 に示す先行研究で用いられていたアクセスデータを用いる。

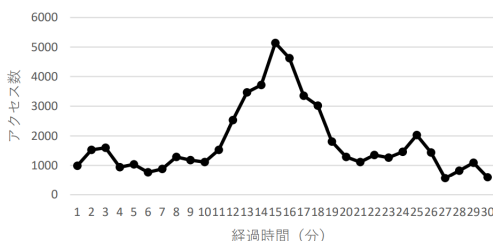


図 2 アクセスデータ

(1) スケーリングポリシーの実装

スケーリングポリシーの実装では次のような特徴がみられた。AWS と Azure で共通に SDK が準備されている言語は Java, C#, Python, JavaScript, PowerShell, Go, Rust が挙げられ、加えて AWS では Ruby を利用することができる。しかし、AWS の SDK は正式版であるが、Azure はベータ版である。また、API より仮想マシン群の平均 CPU 使用率を取得した際のデータが Azure では時系列であったが、AWS では時系列ではないため時系列でデータを利用したい場合、利用者側がソートを行う必要がある。データの構造は言語に依存するが、AWS も Azure もジャグ配列のような構造になっている。

(2) 負荷テスト

また、負荷実験結果を図 3 に示す。AWS ではアクセス数に対応してスケーリングを行えているが、Azure では行えていない。そして負荷テスト時に Azure では仮想マシン上のアプリにアクセスできない時間がほとんどであった。これはスケーリングポリシーのパラメータが AWS に適したものであったからと考える。

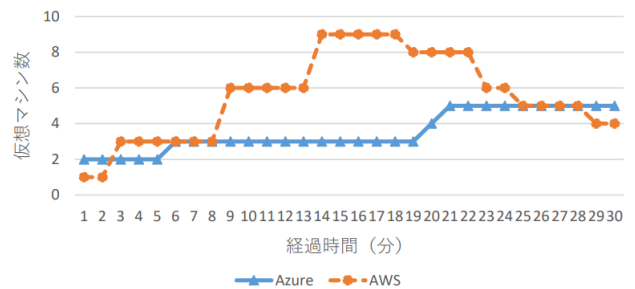


図 3 負荷テスト時の仮想マシン数の遷移

以上のことより、AWS と Azure には類似したサービスがあるため、SDK を使用した高度なスケーリングポリシーを同じシステム構成で実現できることが明確になった。しかし、AWS と Azure ではそれぞれに適したスケーリングポリシーを実装する必要があることが判明した。

4 まとめ

本研究では、Azure に先行研究 [1] と同様の高度なスケーリングポリシーを実装し、AWS と Azure を比較評価した。

参考文献

[1] 池上 昌志, "AWS AutoScaling を用いたアプリケーションスケールリングの実装と評価", 令和2年度高知工科大学学士學位論文 (2021).