

平成 25 年度 卒業論文概要			
所 属	機械情報工学科	指導教員	光来 健一
学生番号	10237036	学生氏名	三宮 浩太
論文題目	クラウドにおける仮想マシン構成のアプリケーション単位での最適化		

## 1 はじめに

近年、クラウドコンピューティングの利用が広まっており、そのサービスのひとつである IaaS 型クラウドはユーザに仮想マシン (VM) を提供する。IaaS 型クラウドでは、必要に応じて VM の台数を増やすことで負荷の増大に迅速に対応することができる。一方、負荷が低下した時には VM の台数を減らして最適化することで、台数に応じて決まるコストを削減することができる。

しかし、VM の台数の増減では VM が 1 台になった後、台数をそれ以上減らすことができない。さらにコストを削減するには、性能の低い VM に切り替えたり、複数のアプリケーションを 1 つの VM に統合したりする必要がある。この際に、VM 内のアプリケーションを別の VM に移動させる必要があり、サービスのダウンタイムの原因となる。また、複数のアプリケーションを同じ VM 上で動かすとセキュリティが低下する恐れがある。

本研究では、個々のアプリケーションを軽量な VM 内で動作させることにより、アプリケーション単位で VM 構成の最適化を行うことが可能なシステム FlexCapsule を提案する。

## 2 VM 構成の最適化手法

IaaS 型クラウドにおける VM 構成の最適化は、VM 内のシステムの CPU 使用率やメモリ使用量などに応じて行われる。最適化手法として、図 1 のような 3 つの手法が挙げられる。最も一般的な最適化手法は VM の数を増減させるスケールアウト・スケールインである。システムが高負荷になるとスケールアウトによって、同じアプリケーションが動く VM の台数を増やし、負荷を分散させる。逆に、システムが低負荷になるとスケールインによって VM の台数を減らし、VM のコストを削減する。しかし、VM が 1 台の状態ではシステムが低負荷になっても、それ以上 VM の数を減らすことはできない。そのため、アプリケーションがほとんど動いていない時のコストの削減には限界がある。

一方、1 台の VM に対する最適化手法として、VM の性能を上下させるスケールアップ・スケールダウンがある。システムが高負荷になるとスケールアップによって、VM に CPU やメモリを追加することで性能を上げる。逆に、システムが低負荷になるとスケールダウンによって VM の性能を下げ、VM のコストを削減する。既存の多くのクラウドでは実行中の VM の性能を変更することはできないため、提供されている何種類かの性能の VM を切り替えて使うことでスケールアップ・スケールダウンを実現する。そのため、アプリケーションを元の VM で停止させてから、そのデータ等を最適な VM に移動して、起動し直す必要がある。この期間はアプリケーションが

サービスを提供できないダウンタイムとなる。また、提供されている最低性能の VM よりも性能を下げてもコストを削減することはできない。

さらなる最適化を行うためには、複数のアプリケーションを 1 つの VM に統合する方法が考えられる。例えば、2 台の VM で別々の低負荷なアプリケーションが動作している時、それらを 1 つの VM に集約することによって VM を 1 台に減らすことができる。その後でシステムの負荷が高くなった場合には、アプリケーションを再び分離して、別の VM で動作させることで負荷を分散させることができる。しかし、スケールアップ・スケールダウンの場合と同様に、アプリケーションを VM 間で移動させる際にダウンタイムが発生する。また、同一 VM 上で複数のアプリケーションを動作させることになるため、アプリケーション間の隔離が弱くなるというセキュリティ上の問題も生じる。

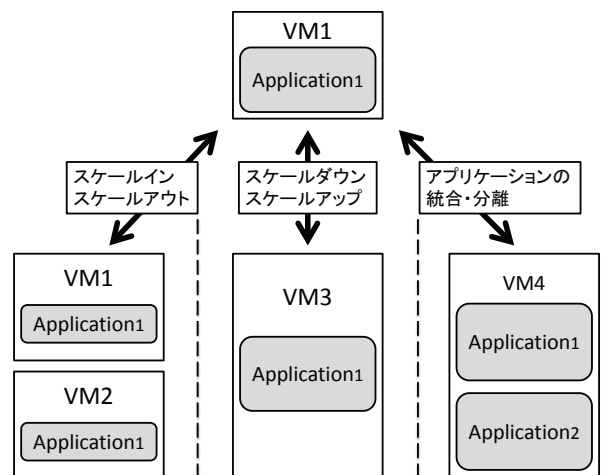


図 1 VM 構成の最適化

## 3 FlexCapsule

本研究では、個々のアプリケーションを軽量な VM の中で動作させることにより、アプリケーション単位での VM 構成の最適化を実現するシステム FlexCapsule を提案する。FlexCapsule ではクラウドによって提供される VM (クラウド VM) の中でさらにアプリケーション用の VM (アプリケーション VM) を動作させる。そのためにネストした仮想化 [1] を用いる。ネストした仮想化により、図 2 のようにクラウド VM 内でハイパーバイザと呼ばれるソフトウェアを動作させ、その上でアプリケーション VM を実行する。アプリケーション VM 内ではライブラリ OS と呼ばれる軽量 OS を用いる。また、アプリケーション VM を管理するために、管理 VM と呼ばれる VM を動作させる。

FlexCapsule では、アプリケーション VM を用いることに

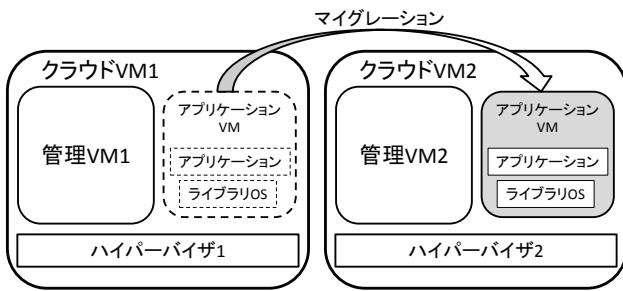


図2 FlexCapsule のシステム構成

より、VMのマイグレーション技術を利用してアプリケーションを停止させずに移動することができる。また、ライブラリOSを用いてVMのサイズを小さく抑えることで、マイグレーションを高速に行うことができる。これにより、クラウドVMのスケールアップ・スケールダウンやアプリケーションの統合・分離を容易に行えるようになる。また、1つのクラウドVM内で複数のアプリケーションVMを実行しても、VM間の強い隔離によりセキュリティの低下を防ぐことができる。

### 3.1 FlexCapsule OS

FlexCapsule OSは軽量なライブラリOSである。ライブラリOSはOSの機能を提供するライブラリであり、アプリケーションにリンクして用いられる。アプリケーションはOSの機能を利用する際にこのライブラリの関数を呼び出す。ライブラリOSを用いることにより、アプリケーションごとに利用する機能だけを残して軽量化を図ることができる。そのため、汎用OSを用いるよりもアプリケーションVMが必要とするメモリを少なくすることができる。FlexCapsule OSは仮想化ソフトウェアXenが提供するMini-OSをベースとして実装した。Mini-OSはアプリケーションに標準Cライブラリとほぼ同じインタフェースを提供するが、プロセス制御やファイル関連の一部の関数はサポートされない。

### 3.2 マイグレーションのサポート

FlexCapsule OSは高速化および軽量化のために準仮想化VMで動作する。準仮想化VMは仮想化に対応したOSしか動作させることができないため、マイグレーションを行うには、FlexCapsule OS自身がマイグレーションに対応する必要がある。そこで、マイグレーション時のサスペンド・レジューム処理を行うシャットダウンハンドラを実装した。VMのマイグレーション時にはVMを停止させるためのサスペンド要求が送られる。シャットダウンハンドラはXenStoreと呼ばれるデータベースを監視することでこの要求を受け取る。サスペンド要求を受け取ると、マイグレーション後に再開できるように、FlexCapsule OS内の各機能をサスペンド状態にする。サスペンド処理が完了したらハイパーバイザを呼び出してVM全体をサスペンド状態に移行させる。

VMのマイグレーションが行われた後、シャットダウンハンドラはハイパーバイザを最後に呼び出した箇所から再開され、FlexCapsule OSのレジューム処理を行う。レジューム処理では、サスペンド状態になっている各機能をマイグレーション先の実行環境に合わせて利用可能な状態にする。例えば、アプリケーションVMのCPUにマイグレーション先のクラウドVMのCPUを割り当て直したり、アプリケーションVMへのメモリ割り当ての管理テーブルを構築し直したりする。

### 3.3 アプリケーションVMの管理

FlexCapsuleはアプリケーションVMを従来のOSのプロセスと同じように管理できるようにするための実行環境を提供する。この実行環境はクラウドVM内の管理VM上で動作する。例えば、アプリケーションを実行する時にプロセスを作成する代わりに、アプリケーションVMを作成して起動する。また、プロセスの一覧を表示するpsコマンドを実行すると、アプリケーションVMの情報を表示する。

## 4 実験

FlexCapsuleを用いてアプリケーションVMをマイグレーションし、マイグレーションにかかる時間とダウンタイムを計測した。実験にはIntel Xeon E3-1290 3.70 GHzのCPU、8GBのメモリを搭載したマシンを使用した。システム全体の仮想化とクラウドVM内の仮想化のためにどちらもXen 4.2.2を用い、クラウドVMを管理するための管理VMではLinux 3.5.0を動かした。クラウドVMには2CPUと1GBのメモリを割り当て、クラウドVM内の管理VMではLinux 3.8.0を動かした。

アプリケーションVMのメモリサイズを変化させてマイグレーション時間とダウンタイムをそれぞれ10回ずつ計測した平均値を図3に示す。この結果より、マイグレーション時間はメモリサイズに比例していることが分かる。一方、ダウンタイムは常に0.2秒程度となり、メモリサイズの影響は受けなかったことが分かった。

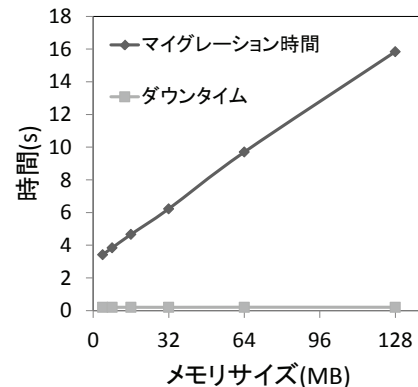


図3 アプリケーションVMのマイグレーション性能

## 5 まとめ

本研究では、クラウドにおいて個々のアプリケーションを軽量なVMの中で動作させることにより、アプリケーション単位でのVM構成の最適化を実現するシステムFlexCapsuleを提案した。FlexCapsuleはライブラリOSを用いたアプリケーションVMのマイグレーション、および、アプリケーションVMを管理するための実行環境を提供する。今後の課題は、従来のアプリケーションで使用されているプロセス制御やファイル関連の機能のサポート、および、より高機能なアプリケーションVMの実行環境の提供である。

### 参考文献

[1] M. Ben-Yehuda et al. The Turtles Project: Design and Implementation of Nested Virtualization, In Proc. OSDI 2010, pp.423-436, 2010.