

軽量実行環境におけるアプリケーションの実行性能比較

宮澤 元[†]南山大学 理工学部 ソフトウェア工学科[†]

1. はじめに

マルチテナントの IaaS (Infrastructure-as-a-Service) 環境を提供するクラウドコンピューティング (クラウド) のアプリケーション実行基盤として、仮想マシンが広く利用されてきた。仮想マシンを利用することによって、実行環境同士の隔離が実現できる。また、複数の実行環境を 1 台の物理的な計算機に集約することも可能である。しかし、仮想マシンを実現するためには、計算機ハードウェアを仮想化するオーバーヘッドが大きいという問題がある。また、仮想マシン内では汎用オペレーティングシステム (OS) をゲスト OS として動作させるのが普通なので、仮想マシンを実現するための仮想マシンモニタにおける仮想 CPU やメモリの管理が、ゲスト OS でのプロセスや仮想メモリの管理と重複してしまう問題もある。

そこで現在では、ホスト OS カーネルが提供する名前空間を用いてコンテナと呼ばれる実行環境を互いに隔離できるコンテナ仮想化技術が普及している。コンテナ仮想化では仮想マシンモニタが提供するようなハードウェアの仮想化を行わない。また、コンテナはアプリケーションプロセスを隔離するものであり、ゲスト OS が必要ないので、効率的な IaaS クラウド基盤を実現できる。

一方、従来の仮想マシンのオーバーヘッドを低減するための研究も行われている。その代表的なものに Unikernels がある [1]。Unikernels では、ユーザコードにライブラリ OS をリンクして単一のアプリケーションバイナリとしたものを仮想マシン内で動作させる。これにより、仮想マシンのゲスト OS として汎用 OS を利用する場合と比べ、ゲスト OS に起因するオーバーヘッドを低減できるので性能向上が見込まれる。さらに、仮想マシンモニタの構成を改善することによりコンテナの生成・起動と同等以上に高速な仮想マシンの生成・起動を実現する研究も行われている [2]。

しかし、コンテナ仮想化環境と Unikernels 環境の軽量仮想実行環境のうち、どちらがアプリケーションの実行により適しているかは、状況により異なる。従来型の仮想マシンも含めたアプリケーションの実行環境の性能評価に関する研究も行われているが、使用する仮想マ

シンモニタの種類やアプリケーションの特性などによっても結果は異なる [3, 4, 5]。例えば Unikernels 環境では、アプリケーションはライブラリ OS と一体化した単一のバイナリとして動作するので、汎用のホスト OS 上でプロセスとして動作するコンテナ仮想化環境よりコンテキストスイッチが少ない分、性能的に有利だと考えられる。その反面、Unikernels においても仮想化のオーバーヘッドは依然として存在する。特に入出力仮想化のオーバーヘッドは一般に大きく、ネットワーク通信などの入出力を多く行うアプリケーションではコンテナ仮想化が有利となってもおかしくない。

我々は、コンテナ仮想化環境や Unikernels 環境をクラウドにおけるアプリケーションの軽量実行環境と捉え、アプリケーションの動作状況に合わせて適切な実行基盤を与えられるような軽量実行環境の実現を目指している。そのためには、具体的なアプリケーションの動作状況に対して、既存の軽量実行環境がどのように振る舞うかの詳細なデータの把握が不可欠である。本稿では、軽量実行環境におけるアプリケーションの実行性能について報告する。ごく単純な数値計算アプリケーションを複数の軽量実行環境上で動作させて性能測定実験を行い、比較を行う。

2. 軽量実行環境の性能比較

コンテナ仮想化環境や Unikernel 環境など、軽量実行環境についての性能比較は数多くなされている。起動時間やメモリフットプリントといった軽量実行環境そのものの性能比較や、いくつかの軽量実行環境におけるアプリケーションの性能比較については様々な報告がある。

一方、複数のアプリケーションを動作させた状態でのプラットフォーム全体での性能比較はほとんどなされていない。マルチテナントの IaaS 環境などでは、同一の物理マシン上で複数の仮想マシンやコンテナが動作しており、このような状況における軽量仮想実行環境ごとのアプリケーションの性能特性を明らかにすることは重要だと思われる。

3. 実験

軽量実行環境として代表的な Unikernels 環境の一つである OSv (version 0.52.0) と、広く利用されているコンテナ仮想化環境の Docker (version 18.09.0) を利用して、単純な数値計算アプリケーションを動作させ、実行時間を計

Performance Comparison of Applications on Lightweight Execution Environments

[†] Department of Software Engineering, Faculty of Science and Engineering, Nanzan University

表 1: 実験に利用した PC の仕様

CPU	Intel Core i7-8700K 3.7GHz
コア数	6 コア 12 スレッド
メモリ	DDR4-2666 DRAM 32G バイト
ストレージ	Plextor SSD PX512M9PEG
OS	Ubuntu Server 16.04.5 Kernel 4.4.0

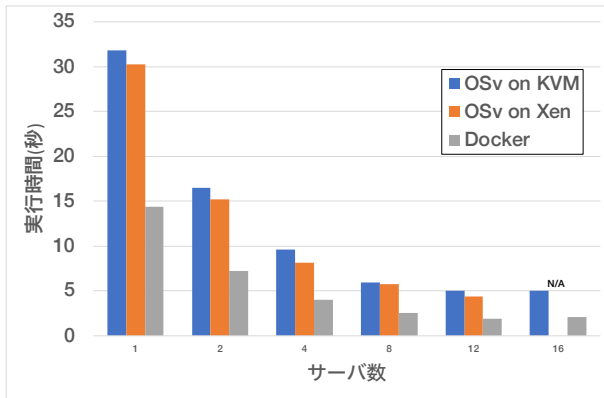


図 1: 並列計算の実験結果

測する実験を行った。OSv については、QEMU(version 2.5.0) と Xen(version 4.6.5) の両方で動作させた。

3.1 実験環境

実験には表 1 に示す仕様の PC を用いた。また、軽量実行環境には、それぞれ 2G バイトのメモリを割り当てた。

3.2 並列計算実験

単純な数値計算アプリケーションを作成し、このアプリケーションの実行時間を計測する実験を行なった。数値計算の内容は、モンテカルロ法を用いた円周率の近似値の並列計算であり、プログラムは C 言語で作成した。利用するサーバ全体であらかじめ定めたサンプルサイズ分の座標を生成するものとし、各サーバはそれぞれ均等に割り当てられた数の座標をランダムに生成する。サーバ毎にカウントした単位円内の座標の数を集計し、円周率の近似値を計算する。

実験では、利用するサーバの台数を 1 から 16 まで変化させ、各設定の実験を 10 回ずつ行なった平均値の結果とした。実験結果を図 1 に示す。

ほぼ CPU バウンズの単純なアプリケーションであるにも関わらず、Docker 上で動作させた方が、OSv よりおよそ 2 倍近く速い結果となった。OSv に関しては、仮想化方式の違いはアプリケーションの実行性能に大きな影響を与えていないことがわかる*。

*サーバ数 16 台のとき、OSv on Xen はメモリ不足で動作しなかった。

田尻による Apache Spark を用いた類似の実験で、OSv 上のアプリケーションが Docker 上のものより高速な結果を出すことが報告されており [6]、今回の結果とは異なる。仮想化のオーバーヘッドを考慮しても説明しにくい差がついており、原因を調査する必要がある。

4. まとめ

いくつかの軽量実行環境を利用して単純な数値計算アプリケーションを実行させ、実行時間を計測する実験を行なった。アプリケーションをコンテナ仮想化環境で動作させる方が、Unikernels 環境よりも大幅に実行時間が短い結果となった。

さまざまな状況において、アプリケーションの実行にどのような軽量実行環境が適しているかを調べるためには、今回の実験では全く不十分である。今後は、まず I/O バウンドアプリケーションについて今回と同様の実験を行う。また、単純なベンチマークだけではなく、より現実的なワークロードにおけるアプリケーションの振る舞いについても調べる。

参考文献

- [1] Anil Madhavapeddy, et al., “Unikernels: Library Operating Systems for the Cloud,” in *Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'13)*, pp.461–472, 2013.
- [2] Filipe Manco, et al., “My VM is Lighter (and Safer) than your Container,” in *Proceedings of the 26th Symposium of Operating Systems Principles*, pp.218–233, 2017.
- [3] M. G. Xavier, et al., “Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments,” in *Proceedings of the 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2013)*, pp.233–240, 2013.
- [4] S. Gupta and D. Gera, “A Comparison of LXD, Docker and Virtual Machine,” in *International Journal of Scientific & Engineering Research*, Volume. 7, Issue 9, pp.1414–1417, 2016.
- [5] B. Xavier, et al., “Time Provisioning Evaluation of KVM, Docker and Unikernels in a Cloud Platform,” in *Proceedings of the 16th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, pp.277–280, 2016.
- [6] 田尻翔太, “IaaS 環境におけるマルチプロセスアプリケーションを考慮した仮想化基盤の動的構成管理に関する研究,” 南山大学大学院理工学研究科修士論文, 2017.