

## コンテナ環境における並列アプリケーション性能に関する考察

## A Study on Performance of Applications in Container

梅澤 綾果<sup>†</sup>      山口 実靖<sup>†</sup>  
Ayaka Umezawa      Saneyasu Yamaguchi

## 1. はじめに

情報サービスの普及に伴いサーバの設置スペースや消費電力の肥大化が問題となり、仮想化技術を用いたサーバ集約やそれを用いたクラウドコンピューティングの重要性が高まっている。サーバの集約を高度に行うには、多数のサーバが集約された環境でも低くない性能を提供することが必要となる。仮想化環境にてより高い性能を提供できる仮想化技術として、コンテナ型仮想化技術があり、注目を集めている。

多数のサーバサイドアプリケーションを運用する手法として、アプリケーション群が単一の OS を共有する手法と、各アプリケーションが独占的に OS を有する手法が考えられる。前者の方が効率的に計算資源を使用できると予想されるが、後者の方が独立性が高く他のアプリケーションとの運用上の衝突(使用するポート番号やソフトウェアのバージョンなど)を回避できると期待できる。よって、より高い性能で OS を独占的に占有できる環境の提供が重要であると考えられる。また、Web アプリケーションや Web サイトなどのネットワーク上のサービスには、負荷は極めて低いが停止することができないものも多く、これらは高い集約度で集約されることが好ましいと考えられる。

本稿では、Docker を用いるコンテナ型仮想化環境に着目し、その基本的な性能と、高集約環境における性能についての考察を行う。

## 2. コンテナ型仮想化技術

コンテナ型仮想化技術は、ホスト OS 上にコンテナと呼ばれる仮想的なユーザー空間を複数構築する手法である。ゲスト OS は存在せず、カーネルをホストとコンテナで共有する。ゲスト OS を稼働させる仮想計算機型の仮想化手法と比べ、使用する計算機資源やオーバーヘッドが小さくより高い性能を提供できると期待されている。

コンテナ型仮想化環境を提供できる著名なシステムとして Docker がある。

## 3. 関連研究

高集約環境の性能に関する研究として、仮想計算機型仮想化技術を用いた高集約環境の性能について考察[1]がある。当該研究では、単一の物理計算機上に多数の仮想計算機を稼働させ、それぞれの仮想計算機上に低負荷の Web アプリケーションを稼働させたときの応答性能などの評価や考察を行っている。そして、評価結果により多 VM 環境では Web アプリケーションの負荷が低い状態であっても各仮想計算機上で稼働しているゲスト OS 群の負荷が大きくなり、大幅な応答性能の劣化が生じることが示されている。また、ゲスト OS カーネルタイマー周波数を低減させるなどゲスト OS の負荷を軽減することにより応答時間の改善が行えることが示されている。

上記文献にて、仮想計算機型仮想化技術を用いて高集約環境を構築する上では、各 VM 上で動作するゲスト OS の負荷が大きな課題であることが示されている。よって、ゲスト OS の稼働を要しないコンテナ型仮想化技術を用いることにより高集約環境の性能の大幅な改善ができると期待できる。

## 4. 性能評価

本章において、コンテナ型仮想化環境の性能の評価を行う。実験環境は、CPU Intel Celeron CPU G530 2.40GHz, 2 コア, メモリ 1GB, ホスト OS CentOS 7 Linux 3.10.0, Docker 1.11.2 である。

## 4.1 演算処理性能

ホスト OS とコンテナ内にて CPU ベンチマーク姫野ベンチを実行し、演算処理性能を評価した。測定結果を図 1 に示す。図内の“Host”と“Container”は、各環境にてベンチマークを非並列で実行した場合の性能であり、残りの 4 種は

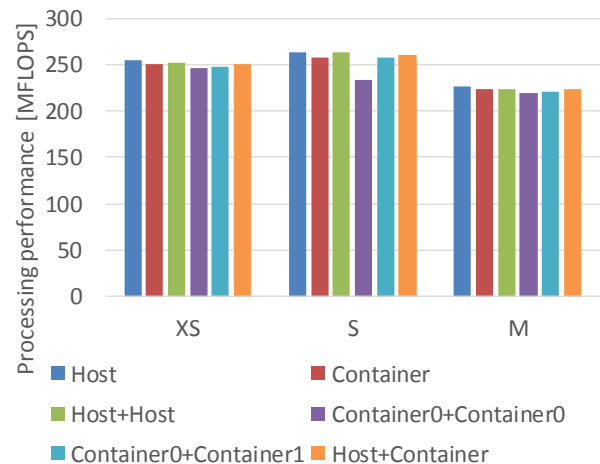


図 2 CPU 演算性能

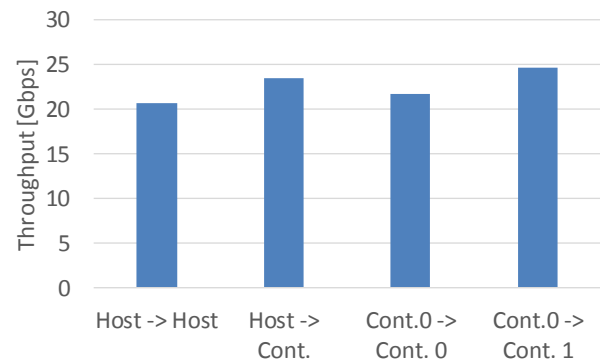


図 2 通信性能

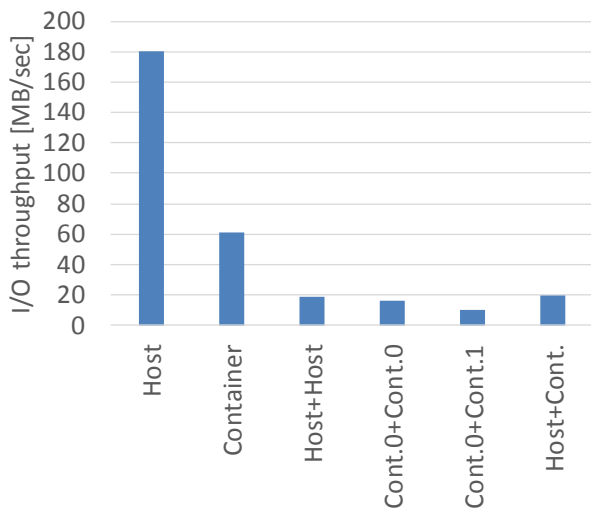


図 4 I/O 性能

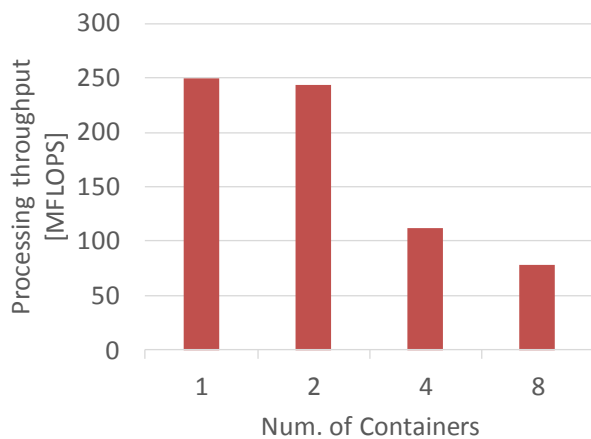


図 4 CPU 演算性能 (高集約環境)

表記の 2 カ所でベンチマークを並列で実行した場合の性能である。“Container0 + Container0”は単一のコンテナ上で 2 個のベンチマークを実行した状況であり，“Container0 + Container1”は異なる 2 コンテナを起動しそれぞれでベンチマークを起動した状況である。横軸はベンチマークサイズ、縦軸は演算性能である。図より、コンテナ環境においてもホスト OS 環境と近い演算性能が得られていることが分かる。

#### 4.2 通信性能

iperf を用いて通信性能の評価を行った。通信はホスト OS-ホスト OS 間、ホスト OS-コンテナ間(ホストからコンテナへの送信)、同一コンテナ間、同一ホスト OS 上の異なる 2 コンテナ間で行った。評価結果を図 2 に示す。図より、ホスト OS 間と同一コンテナ間でほぼ同等の性能が得られ、コンテナ使用のオーバーヘッドが小さいことが分かる。

#### 4.3 I/O 性能

ホスト OS 上とコンテナ上で I/O ベンチマーク FFSB を実行し、I/O 性能を評価した。測定は 1MB バイトのファイル

を 4096 個作成し、1MB の読み込みを 5 分間繰り返すことにより行った。ベンチマークは、ホスト OS 上で単独実行、コンテナ内で単独実行、ホスト OS 上で 2 プロセス並列実行、コンテナ内で 2 プロセス並列実行、ホスト OS とコンテナ上で 1 プロセスずつ並列実行した。測定結果を図 3 に示す。図より I/O 性能においてはホスト上とコンテナ内で差が見られることが分かる。

#### 4.4 多コンテナ環境における性能

1 台のホスト OS 上に 1 台から 8 台のコンテナを起動し、CPU ベンチマーク姫野ベンチを各コンテナ内で並列実行した。そのときに各コンテナで得られた性能を図 4 に示す。図より、単独実行時と 2 並列実行時に各コンテナで得られている性能がほぼ同等であり、合計で比較するとほぼ 2 倍の性能となっていることが分かる。これは、使用した計算機の CPU コアの数 2 個であるからである。並列数 2 以上に着目すると、性能はコンテナ数にほぼ反比例しており、CPU 資源は主にコンテナ内のアプリケーションに割り当てられており、コンテナ起動のオーバーヘッドは大きくないことが分かる。8 コンテナ環境にて反比例以上の性能が得られている理由は、ベンチマークの実行時間が完全に同一とならないため一部のベンチマークが先に終了している状況を含んでいるためである。これにより、8 コンテナ環境においてはコンテナ間の性能の差も確認され、最大で 1.9 倍程度の差が確認された。

#### 5. 考察

ゲスト OS を稼働させる仮想計算機型仮想化手法を用いた既存の研究[1]においては、VM 数を増加させるに従いコンテキストスイッチ数などが急激に増加し、システム性能の大幅な低下が確認された。しかし、前章の評価で得られたコンテナを用いる高集約環境における性能においてはコンテナ数の増加に対する大幅に性能劣化は見られず、コンテナ起動に起因するオーバーヘッドは VM 起動と比較して小さいと期待できる。

#### 6. おわりに

本稿では、コンテナ型の仮想化環境に着目し、その基礎性能と高集約環境における性能の評価を行った。そして、ゲスト OS を稼働させないコンテナ型仮想化手法の特徴についての考察を行った。

今後は、低負荷の Web アプリケーションサービスを個別に持つコンテナを多数集約した環境における性能についての考察を行っていく予定である。

#### 謝辞

本研究は JSPS 科研費 25280022, 26730040, 15H02696 の助成を受けたものである。

本研究は、JST、CREST の支援を受けたものである。

#### 参考文献

- [1] 越智 俊介, 山口 実靖, “高集約サーバ統合環境における Web アプリケーションの性能に関する考察”, 研究報告マルチメディア通信と分散処理 (DPS) 2011-DPS-146(50), 1-7, 2011-03-03 (2011).