# 柔軟で動的な実行環境を提供するための コンテナ型仮想化基盤アーキテクチャの検討

2110525 平地浩一 矢崎研究室

# 1 はじめに

近年,仮想化技術の発展により,物理的な計算資源を仮想化し,1台の物理マシンで複数の仮想マシンを実行することができるようになった.仮想化される主な資源としてCPU,メモリ,ネットワーク,ストレージなどがある.仮想化技術の中でも,コンテナ仮想化は従来の仮想マシンと比べてオーバーヘッドが小さく,特にシステム開発や運用においては,柔軟性,可搬性,再現性に優れるなど多くの利点がある.また,物理的なサーバリソースを効率的に運用したり,拡張性やセキュリティに優れたシステムを構築することも可能であり,今後より多くの分野での活用が期待される.

その一方で、大規模にコンテナを管理する仕組みである Kubernetes など、コンテナ型仮想化技術およびその管理 機構は複雑化しており、多くの開発者や研究者にとって活用のハードルが高い。このことから、広く様々な分野でコンテナ方仮想化技術を応用するためには、利用者の視点からより手軽に扱えるコンテナ型仮想化基盤の開発が求められている。

# 2 目的

本研究では、利用者が自分自身で様々なコンピューティング環境をオンデマンドに使用できるコンテナ型仮想化基盤を提案・構築する. 提案システムは、ユーザからの接続要求に対して、その時にユーザが必要としているコンピューティング環境を最適な計算機資源量で、柔軟かつ動的に提供することを目指す.

# 3 提案するシステムの構成

本研究で提案するシステムの構成を図 1 に示す。本研究では主に、図 1 中における Controller(コントローラ) に該当する機能を実装する。 図ではシステム応用の一例とし

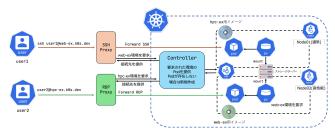


図1 提案システムの構成

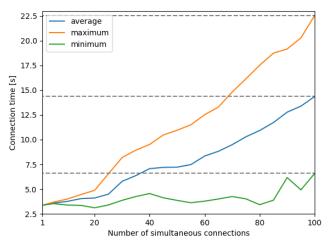


図 2 並列実行数と接続時間の関係

て、大学等の授業において学習環境管理として利用することを想定した構成を示す. 授業では、その内容に応じて異なるアプリケーションやパッケージ、OSを使用する.

授業担当者はあらかじめ、それぞれの環境をコンテナイメージとして定義しておく、授業担当者は、授業環境ごとのエンドポイントおよび、SSHやRDP等のプロトコルを指定し、ユーザに伝える、ユーザは、指示されたエンドポイントとプロトコルでシステムにアクセスすることで、自分専用の演習環境を利用することができる.

### 4 既存システムのレイテンシ測定実験

提案するコントローラを実装するにあたって、要求される時間的な制約を確認するために実験を行った。実験ではContainerSSHを用いて複数人が同時にアクセスした際のレイテンシを測定した。

実験では、backend に Docker を用いた。検証環境において ContainerSSH と Docker を実行し、ssh コマンドで同時接続を行なった際の接続時間を計測した.

# 4.1 クライアント側から SSH 接続時のターンアラウンド タイムの計測

まず初めに複数のユーザが同時に接続した際に,ユーザが SSH 接続をしてからコマンドを実行して終了するまでのターンアラウンドタイムの変化を計測した. スクリプトを用いて同時に接続する数を 1 から 10 づつ 100 まで増やしていき,接続数ごとの処理時間を記録した. 結果を図 2 に示す.

グラフの横軸は同時接続数を表している. 縦軸は SSH

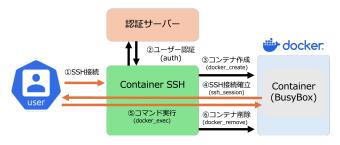


図3 ContainerSSH における内部処理

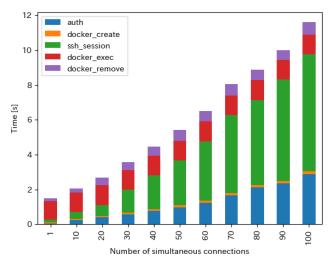


図 4 並列実行数と主な処理時間の関係

コマンドを実行してから接続が成功し、date コマンドを 実行したのち、ssh 接続を切断するまでの時間を、クライ アント側から time コマンドで測定した時間を秒で表して いる.

### 4.2 サーバ側における各部分処理にかかる時間の測定

前の実験より、ユーザが SSH コマンドを実行してから の応答時間は同時接続数に応じて長くなることがわかっ た.ここでは、どの処理によって接続時間が増加するのか を検証した.

実験では、並列に接続するクライアント数を 1 から 100 まで変化させ、それぞれの場合で各処理にかかる平均の時間を算出した.結果を図 4 に示す. グラフの横軸は同時接続数、縦軸は処理に要した時間を示している.また、図 3 に示したそれぞれの処理ごとの時間を積み上げて示している.

# 5 考察

図2より、ContainerSSHでは同時接続数が多くなるほど1つの接続にかかる時間が増加していくことが確認できた. 同時接続数が100程度の場合、最長で22秒ほどの時間がかかっており、ユーザがストレスなく現実的に利用できるとは言えない。平均の接続時間に関しても100接続であっても約14秒程度かかっている。より接続数が増えた場合、これらの時間はさらに増加していくと考える。提

案システムが想定している 200~300 名程度の同時利用を考えると,ユーザ数に応じて顕著に処理時間が増加する部分においては対策が必要である.

図4を参照すると、ContainerSSHの内部処理のうち、認証 (auth)とSSHセッション管理 (ssh\_session)の各処理は同時接続数に応じて特に増加していくことが分かった.認証とSSHセッション管理に関しては、ユーザの待ち時間への影響が大きいため、提案手法ではこの時間を短縮する実装上の工夫が必要であると考える。コンテナ削除処理も接続数に対して若干増加しているが、コンテナ削除処理はユーザが環境を利用し終わった後に行われるため接続時間に大きな影響はないと考える。

認証部分に関しては、同時接続数が増えるほど認証サーバの負荷が増えて応答が遅くなっていると考える。現在の認証サーバは ContainerSSH が提供する簡易的なものを利用している。今後 OpenID Connect や LDAP 等のより高度な認証機能を追加していくとより大きなボトルネックになると予想する。そのため、1 度認証したクライアントに対して一時的なトークンを払い出し、以降は一定時間認証サーバにアクセスする必要をなくといったような仕組み等を検討している。

SSH セッション管理部分に関しては、接続数ごとに SSH の中継をするプロセスが増えていき負荷になっていると想定する.このため、負荷分散をするための仕組みが必要になると考える.

#### 6 まとめ

本論文では、利用者の様々な要求に対して柔軟なコンピューティング環境を動的に提供するためのコンテナ型仮想化環境を提案し、これを実現するためのアーキテクチャについて、既存実装を踏まえて検討および検証した.

今後、特に時間を要する認証と SSH の Proxy 部分に関して認証サーバへの負荷が少ない認証の仕組み作りや、SSH の Proxy を負荷分散するなどの対策を行なっていく予定である.

# 参考文献

- Cloud Native Foundation. Kuberntes. https://kubernetes. io/.
- [2] Oscar Eriksson, Henrik Bjorklund, Bj Bjorklund, and Michael Minock. An evaluation of honeypots with compliant kubernetes, 2023
- [3] Dan Fulton, Laurie Stephey, R Shane Canon, Brandon Cook, and Adam Lavely. Containers everywhere: Towards a fully containerized hpc platform. 2023.