

図3 ContainerSSH における内部処理

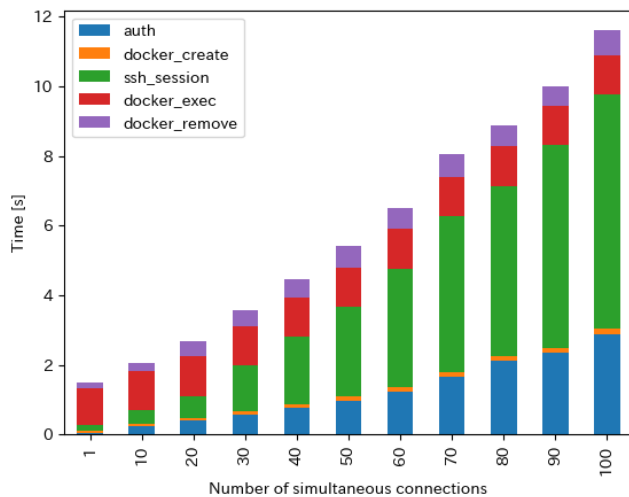


図4 並列実行数と主な処理時間の関係

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られる。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られる。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

- [1] Cloud Native Foundation. Kubernetes. <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 Lavery. Containers everywhere: Towards a fully containerized hpc platform. 2023.