ライブラリOSを用いた異種TEE間マイグレーション機構

近藤　瑛[1]　光来　健一[1]

1. はじめに

近年，クラウドコンピューティングの普及により，クラウド上で機密情報を扱う機会が増加している．その一方，悪意を持ったクラウド管理者などによって機密情報が盗まれることが懸念されている．そこで，CPU の隔離実行環境（TEE）を用いてアプリケーションや仮想マシン（VM）を信頼できないクラウドの OS やハイパーバイザから保護する機密コンピューティングが活用されている．TEE には，プロセス単位での保護を提供する TEE と，VM 単位での保護を提供する TEE が存在する．

クラウド環境では，ホストの保守作業や障害対応などのために，実行中のアプリケーションや VM を別のホストへ移動させるマイグレーションが必要となる．同種の TEE 間ではこれまでにマイグレーション手法がいくつか提案されている．例えば，SGX のエンクレイヴ間のマイグレーションを実現する手法 [1], [2] や，TDX を用いた機密 VM 間のマイグレーションを実現する手法 [3] が提案されている．一方，異種 TEE 間では TEE ごとに移動すべき状態が異なるため，既存の手法を適用することは困難である．

本稿では，TEE 内で共通のライブラリ OS を用いることで，異種 TEE 間のマイグレーションを可能にする MigTEE を提案する．

2. MigTEE

MigTEE は，TEE 内で共通のライブラリ OS を用いてアプリケーションを実行し，ライブラリ OS が管理するアプリケーションの状態を移動させることでマイグレーションを実現する．ライブラリ OS は，OS の機能をライブラリとしてアプリケーションに提供するため，汎用 OS を動かせないプロセスベースの TEE と汎用 OS を動かせる VM ベースの TEE の両方で利用することができる．MigTEE は図 1 のように，移送元 TEE 内でアプリケーションの状態を保存し，保存した状態を移送先ホストに転送して，移送先 TEE 内に復元する．

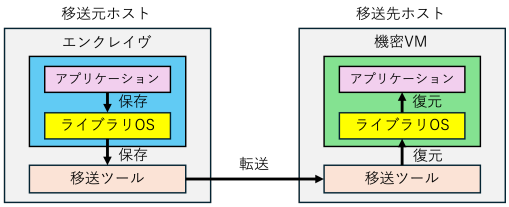


図 1: MigTEE のシステム構成

移送元では，マイグレーション開始時にアプリケーションが状態保存システムコールを実行する．その際に，ライブラリ OS がアプリケーションの状態およびライブラリ OS 自身の状態を取得し，チェックポイントとしてファイルに保存する．保存対象には，メモリ情報やプロセスの実行時情報，ファイルシステム情報などが含まれる．メモリ情報としては，各メモリ領域のアドレスやサイズなどの情報と，実際に格納されているメモリデータを保存する．また，プロセス情報としてプロセス ID や実行環境に関する情報，ファイルシステム情報としてルートディレクトリやマウント情報，システム情報として CPU およびメモリに関する情報や DNS 設定などを保存する．

移送先ではまず，そのホストの CPU を用いて作成可能な TEE を新たに作成する．次に，移送元で保存したチェックポイントファイルを読み込み，保存された状態を取り出す．そして，移送元 TEE と同じメモリ領域を移送先 TEE 内に確保し，その領域に保存されたメモリデータを書き込むことで，アプリケーションのメモリ状態を復元する．さらに，保存されたプロセスの実行時情報やファイルシステム情報などをライブラリ OS に設定することで，アプリケーションの状態を復元する．

3. 実験

ライブラリ OS として Intel SGX と Intel TDX の 2 つの TEE に対応している Gramine [4] を用いて MigTEE を実

[1] 九州工業大学
Kyushu Institute of Technology

© 1959 Information Processing Society of Japan

装した．まず，MigTEE を用いて実行中のアプリケーションの状態が正常に保存・復元できるかを確認する実験を行った．現時点で MigTEE は TEE に対応できていないため，本実験では TEE を使用せず，通常の Linux 上でアプリケーションを動作させた．実験の結果，すべてのアプリケーションで状態を正常に保存できたが，復元後は正常に動作を継続できなかった．

次に，状態の保存・復元時間とチェックポイントのファイルサイズを測定する実験を行った．保存時間を図 2，チェックポイントのファイルサイズを図 3，復元時間を図 4 に示す．保存時間と復元時間はアプリケーションによって大きく異なっているが，各アプリケーションにおける測定値のばらつきは小さいことを確認した．また，チェックポイントのファイルサイズもアプリケーションによって異なっており，メモリデータ量が多いほどファイルサイズが大きくなる傾向が見られた．

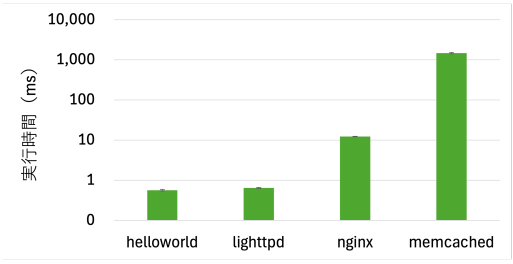


図 2: 状態保存にかかる時間

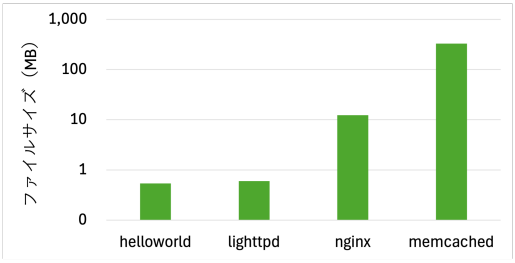


図 3: チェックポイントファイルのサイズ

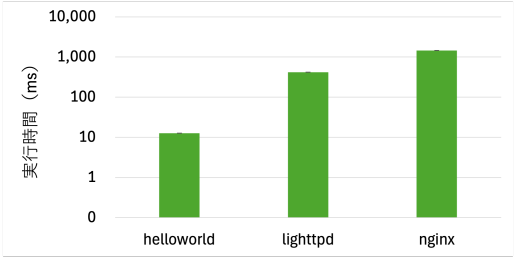


図 4: 状態復元にかかる時間

4. まとめ

本稿では，TEE 内で共通のライブラリ OS を用いることで，異なる種類の TEE 間においてもアプリケーションのマイグレーションを可能にする MigTEE を提案した．今後の課題は，移送元 TEE で保存した状態を基に移送先 TEE で正常にアプリケーションの復元ができるようにすることである．

謝辞 本研究は，JST 経済安全保障重要技術育成プログラム【JPMJKP24U4】の支援を受けたものである．

参考文献

[1] Gu, J., Hua, Z., Xia, Y., Chen, H., Zang, B., Guan, H. and Li, J.: Secure Live Migration of SGX Enclaves on Untrusted Cloud, *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (2017).
[2] Nakashima, K. and Kourai, K.: MigSGX: A Migration Mechanism for Containers Including SGX Applications, *Proceedings of the 14th IEEE International Conference on Cloud Computing (UCC)* (2021).
[3] Intel Inc.: Intel TDX Migration TD Design Guide (2023).
[4] Gramine Project: Gramine, `https://gramineproject.io`.

© 1959 Information Processing Society of Japan