令和 元年度 卒業論文概要			
所　　属	機械情報工学科	指導教員	光來 健一
学生番号	16237046	学生氏名	高橋孝汰
論文題目	複数ホストにまたがる仮想マシンのデータ暗号化の最適化		

1　はじめに

近年，大容量のメモリを持つ仮想マシン（VM）が利用されるようになってきている．例えば，Amazon EC2 では 24TB のメモリを持つ VM が提供されており，ビッグデータの解析などに利用されている．VM はホストのメンテナンスや負荷分散の際に別のホストへマイグレーションされるが，大容量メモリを持つ VM の場合には移送先として十分な空きメモリを持つホストを常に確保できるとは限らない．そこで，VM のメモリを複数の小さなホストに分割して転送する分割マイグレーション [1] が提案されている．分割マイグレーション後にはリモートページングを行って VM が必要とするメモリをホスト間で転送し動作させる．しかし，マイグレーション環境によっては分割マイグレーションとリモートページングの際にメモリデータを盗聴・改ざんされる危険性がある．そのためメモリデータを保護する必要があるが，SSL 等の暗号通信を用いると送受信のたびに暗号化・復号化され，ホストに格納されるメモリデータは別途，暗号化が必要になる．また，暗号化が必要なデータかどうかは考慮されず，すべてのメモリデータが一律に暗号化される．

本研究では，分割マイグレーションとリモートページングにおいてメモリデータの暗号化を最適化する SEmigrate を提案する．

2　複数ホストにまたがる VM の盗聴の危険性

VM が動作しているホストをメンテナンスする場合などには，マイグレーションにより VM を別のホストに移動させることで実行を継続することができる．VM マイグレーションは VM のメモリを移送先ホストに転送し，転送中に VM が更新したメモリを再送した後，移送先ホストで VM を再開する．このようにマイグレーションを行うには移送先に VM のメモリよりも大きな空きメモリが必要となる．大容量メモリを持つ VM をマイグレーションする場合，移送先として十分な空きメモリを持つホストを常に確保できるとはかぎらない．適切な移送先ホストがなければメンテナンス中は VM 上のサービスが長時間停止することになる．

そこで，図 1 のように複数のホストへ VM を分割して転送する分割マイグレーション [1] が提案されている．分割マイグレーション

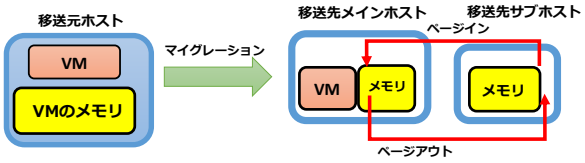


図 1　分割マイグレーション

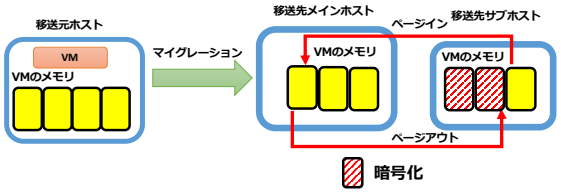


図 2　SEmigrate の暗号化マイグレーション

レーションは仮想 CPU や仮想デバイスなどの VM の核となる状態と，VM がアクセスすることが予測されるメモリデータをメインホストへ転送する．一方，メインホストへ入りきらないメモリデータはサブホストへ転送する．マイグレーション後はメインホスト上で VM 本体が動作し，サブホストはその VM にメモリを提供する．VM はメインホスト上のメモリに直接アクセスすることができるが，サブホスト上のメモリにはリモートページングを行ってアクセスする．VM がサブホストに存在するメモリデータを必要とした際には，そのデータをメインホストへ転送（ページイン）する．同時に，今後アクセスされる可能性が最も低いと予測されるメインホスト上のメモリデータをサブホストへ転送（ページアウト）する．

しかし，マイグレーション環境によっては分割マイグレーションとリモートページングの際に VM のメモリデータを盗聴される危険性がある．例えば，データセンタ間やクラウド間でインターネット経由でメモリデータを転送する場合や，メインホストと管理者が異なるサブホストを利用する場合などである．情報漏洩を防ぐためにはメモリデータを保護する必要があるが，SSL 等の暗号通信を用いると送信のたびにデータが暗号化され，受信のたびに復号される．そのため，サブホスト上のメモリデータは別途，暗号化が必要になり，オーバヘッドが大きくなる．また，暗号化が必要な機密情報かどうかは考慮されず，すべてのメモリデータが一律に暗号化される．

3　SEmigrate

本研究では，分割マイグレーションとリモートページングにおいてメモリデータの暗号化を最適化することでオーバヘッドを削減する SEmigrate を提案する．SEmigrate では，サブホストにおいてメモリデータの復号を行わないようにすることでオーバヘッドを削減し，サブホストにおける情報漏洩も防ぐ．また，メモリに格納されているデータの重要度に応じて選択的に暗号化を行うようにすることでもオーバヘッドを削減する．

3.1　分割マイグレーションとリモートページングの暗号化

SEmigrate は図 2 のように，分割マイグレーションの際に移送元ホストでメモリデータを暗号化し，移送先ホストに転送する．移送先メインホストでは，受信したメモリデータを

VM がアクセスできるように復号して VM のメモリに格納する．一方，サブホスト上のメモリには VM が直接アクセスしないため，受信したメモリデータを復号せずに格納することで復号化のオーバヘッドを削減する．これにより，サブホストでの情報漏洩を防ぐために復号したデータを再暗号化する必要がなくなる．復号して再暗号化するまでの間に情報が漏洩する恐れもない．

リモートページングの際には，メインホストでのみメモリデータの暗号化・復号化を行うことでオーバヘッドを削減する．ページイン時には転送元のサブホストではメモリデータを復号せず，転送先のメインホストで復号して VM のメモリに格納する．ページアウト時には転送元のメインホストでメモリデータを暗号化し，転送先のサブホストでは復号せずに格納する．

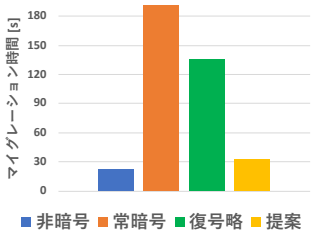
3.2 選択的なメモリ暗号化

SEmigrate 未使用メモリやコード領域など機密情報が格納されていないメモリデータは暗号化せずに転送し，暗号化・復号化のオーバヘッドを削減する．このようなメモリ属性の多くは VM 内の OS が管理しているため，VM イントロスペクションと呼ばれる技術を用いて OS を改変せずに必要な情報を取得する．そして，その情報を基に機密情報が格納されうるメモリデータだけを暗号化する．メモリ属性の取得には，OS のソースコードを利用して VM 内の OS の情報を取得するためのフレームワークである LLView [2] を用いる．SEmigrate は OS のメモリページの参照カウンタが 0 であれば未使用領域，ページの属性が実行可能であればコード領域と判定する．

マイグレーション時には，各メモリページごとにメモリ属性を調べ，機密情報が含まれていなければ暗号化せずに転送する．サブホストは暗号化されていないメモリデータを受信した場合でも暗号化を行わない．同時に，メインホストに非暗号化フラグを送信し，フラグを基に暗号化ビットマップを作成する．暗号化ビットマップは各ページが暗号化されているかどうかを表す．ページイン時には作成したビットマップを確認して，ビットが立っている場合のみ，転送先のメインホストでメモリデータを復号する．ページアウト時には再度，メモリ属性を取得する必要があるが，現在の実装では暗号化ビットマップに基づいてメモリデータの暗号化を行う．

3.3 データの整合性の検査

SEmigrate はマイグレーション時とリモートページング時に転送されるメモリデータが改ざんされていないことをハッシュ値を用いて確認する。マイグレーション時には，暗号化を行う前にメモリデータのハッシュ値を算出し，メモリデータとともに移送先ホストへ転送する．メインホストでは受信したメモリデータを復号した後でハッシュ値を算出し，受信したハッシュ値と比較して改ざんされていないことを確認する．サブホストではメモリデータを復号しないため，受信したハッシュ値の確認は行わず，そのまま保持しておく．そして，ページイン時にメインホストへメモリデータとともにハッシュ値を転送し，メモリデータを復号した後でハッシュ値の検査を行う．ページアウト時には，メインホストで転送するメモリデータのハッシュ値を計算してサブホストに転送する．



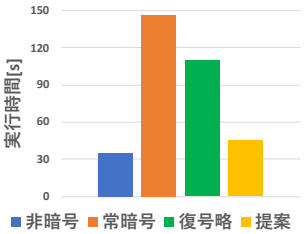


図 3 マイグレーション時間　図 4 ベンチマーク実行時間

4 実験

SEmigrate によるデータ暗号化の最適化の効果を調べる実験を行った．比較として，通信時に常に暗号化・復号化を行う場合（常暗号），サブホストでは復号しない場合（復号略），および暗号化を行わない従来システム（非暗号）を用いた．実験には，移送元ホストと移送先メインホストとして Intel Core i7-8700 の CPU，32GB のメモリを搭載したマシン，移送先サブホストとして Intel Xeon E3-1226 v3，16GB のメモリを搭載したマシンを用い，10 ギガビットイーサネットで接続した．これらのマシンでは Linux 4.18.17 を動作させ，仮想化ソフトウェアには QEMU-KVM 2.11.2 を用いた．VM には 20GB のメモリを割り当て，分割マイグレーションで 10GB ずつに分割した．

分割マイグレーションにかかる時間を測定した結果を図 3 に示す．データ転送時に常に暗号化・復号化を行った場合と比較して，サブホストで復号しないようにした場合には 30 %高速化した．選択的な暗号化も行う SEmigrate では未使用メモリが大量に存在したため 80 %高速化できた．ただし，まったく暗号化を行わない場合と比べると 1.5 倍の時間がかかった．

次に，リモートページング性能を調べるために，分割マイグレーション後に VM 内で大量のメモリにアクセスするベンチマークの実行時間を測定した．図 4 に示すように，ページイン・ページアウト時に常に暗号化・復号化を行った場合と比べて，サブホストで復号しないと 25 %高速化した．SEmigtate では 70 %高速化できたが，まったく暗号化を行わない場合と比べると 1.3 倍の時間がかかった。

5 まとめ

本研究では，分割マイグレーションとリモートページングにおいて，メモリデータの暗号化を最適化することでオーバヘッドを削減する SEmigrate を提案した．SEmigrate では，移送先サブホストにおいてメモリデータの暗号化・復号化を行わない．さらに，メモリ属性に基づいて VM のメモリデータを選択的に暗号化する．今後の課題は，VM 内で動作するアプリケーションのメモリ情報なども活用して，選択的な暗号化の適用範囲を拡げていくことである．

参考文献

[1] M. Suetake, T. Kashiwagi, H. Kizu, and K. Kourai. S-memV: Split Migration of Large-Memory Virtual Machines in IaaS Clouds. *CLOUD 2018.*

[2] Y. Ozaki, S. Kanamoto, H. Yamamoto, and K. Kourai. Detecting System Failures with GPUs and LLVM. *APSys 2019.*