# 複数ホストにまたがるVMのデータ暗号化の最適化

# 複数ホストにまたがるVMのデータ暗号化の最適化

堀尾 周平[1]　高橋 孝汰[1]　光来 健一[1]　Lukman Ab. Rahim[2]

## 1. はじめに

近年，大容量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 が提供されている．VM はマイグレーションと呼ばれる技術によりホストのメンテナンス時などに別のホストに移動させることができ，サービスを提供し続けることができる．しかし，大容量メモリを持つ VM の場合は十分なメモリを持つ移送先ホストを常に確保できるとは限らない．そこで，VM のメモリを分割して小さなメインホストとサブホストに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これらのホスト間で必要に応じてメモリデータをやりとりするリモートページングを行いながら VM が動作する．しかし，実行環境によってはネットワーク上やサブホストにおいてメモリデータを盗聴される危険性がある．メモリデータは暗号化によって保護することができるが，そのオーバヘッドにより性能が低下する．

本稿では，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 の暗号化

分割マイグレーションは図 1 のように，VM の大容量メモリを分割して複数の小さなホストに転送するマイグレーション手法である．分割マイグレーションはメインホストに今後アクセスされそうなメモリデータと仮想 CPU などの VM コアの状態を転送する．サブホストにはそれ以外のメモリを転送する．分割マイグレーション後は VM コアがメインホスト上で動作し，サブホストはその VM にメモリを提供する．VM がサブホスト上のメモリにアクセスした時にはリモートページングを行い，サブホストに存在するメモリデータをメインホストへ転送（ページイン）する．代わりに，メインホスト上の不要なメモリデータをサブホストへ転送（ページアウト）する．

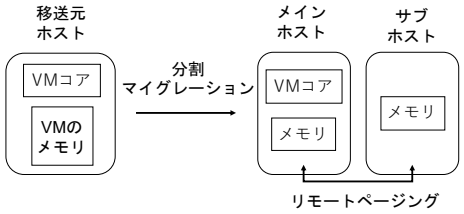



**図 1**　分割マイグレーション

しかし，実行環境によっては分割マイグレーションやリモートページングの際にメモリデータを盗聴される危険性がある．例えば，メモリデータが安全とは限らないネットワーク経由で転送される場合には盗聴されやすくなる．また，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にも盗聴されるリスクが高まる．情報漏洩を防ぐためにはメモリデータを暗号化すればよいが，SSL 等の暗号通信を用いるとメモリデータを転送するたびに暗号化・復号化が行われるため，性能が大きく低下する．また，サブホストに転送されたメモリデータは復号した後に再暗号化して安全に保持するが，再暗号化するまでに盗聴される可能性や悪意のある管理者によって暗号鍵を盗まれる可能性がある．一方，メインホストに転送されたメモリデータは VM のメモリ保護機構 [2] により安全に保持することができる．

## 3. SEmigrate

SEmigrate はサブホストにおいて VM のメモリデータを復号しないようにすることで暗号化のオーバヘッドを削減する．分割マイグレーション時には図 2 のように，移送元ホストで暗号化したメモリデータを移送先メインホストでのみ復号する．移送先サブホストではメモリデータを復号せずに保持する．リモートページング時には，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．サブホス

[1] 九州工業大学
Kyushu Institute of Technology
[2] Universiti Teknologi Petronas

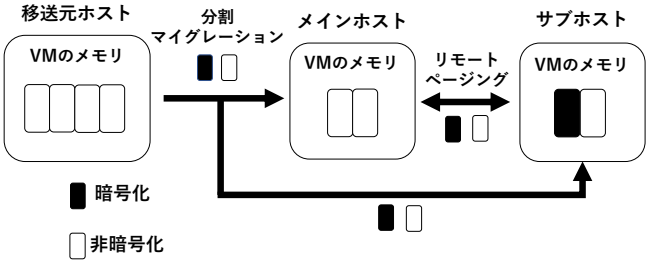



**図 2** SEmigrate による暗号化の最適化

トにあるメモリデータはそのままメインホストに転送して復号し，メインホストにあるメモリデータは暗号化してからサブホストに転送してそのまま保持する．これにより，サブホストでの暗号化のオーバヘッドを削減し，情報漏洩を完全に防ぐ．

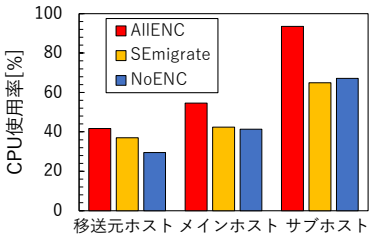
さらに，SEmigrate は機密情報が含まれるメモリデータのみを選択的に暗号化することで暗号化のオーバヘッドを削減する．分割マイグレーション時には，移送元ホストは機密情報を含むメモリデータだけを暗号化する．移送先メインホストでは暗号化されている場合だけメモリデータを復号し，移送先サブホストでは受信したメモリデータをそのまま保持する．リモートページング時には，サブホストにあるメモリデータをそのままメインホストに転送し，暗号化されていれば復号する．メインホストにあるメモリデータは機密情報を含む場合のみ暗号化してサブホストに転送する．

機密情報の有無を判定するために，SEmigrate は VM 内の OS のメモリを解析し，メモリ属性やプロセス情報を利用する．例えば，VM 内で使用されていない空きメモリには機密情報が含まれないため，転送するメモリデータが格納されているメモリ領域の属性を取得して空きメモリかどうかを調べる．また，暗号データしか扱わないアプリケーションのメモリは暗号化する必要がないため，転送するメモリデータを所有しているプロセスの名前が指定されたものと一致するかを調べる．

## 4. 実験

我々は分割マイグレーションとリモートページングをサポートした QEMU-KVM 2.11.2 に SEmigrate を実装した．選択的暗号化のために LLView[3] を用いて VM イントロスペクションを適用し，その対象となるゲスト OS として Linux 4.18 を用いた．メモリデータの暗号化には AES-NI を有効にした OpenSSL の AES-ECB を用い，鍵長は 256 ビットとした．

SEmigrate を用いて，分割マイグレーション中のメモリデータ暗号化による CPU 負荷を調べる実験を行った．この実験では，VM 内で 10GB のメモリを使用するアプリケーションを動かし，このプロセスのメモリを選択的に暗号化しないようにした．比較として，常に暗号化を行う場合（AllENC）と暗号化を行わない場合（NoENC）についても測定した．分割マイグレーション中の CPU 使用率は図 3 のようになった．SEmigrate は常に暗号化を行う場合に比べて CPU 使用率を 4〜27%削減できた．サブホストでの暗号化の最適化による削減量が大きかった．これはサブホストにて暗号化・復号化のオーバヘッドがすべて削減されたためである．しかし，SEmigrate は暗号化を行わない場合と比べると，メインホスト，サブホストでは CPU 使用率がほぼ変わらなかった．一方，移送元ホストにおいては 7%差が出ており，選択的に暗号化するためのオーバヘッドは小さくはないと言える．




| AllENCからの削減 | |
|---|---|
| 移送元ホスト | 4% |
| メインホスト | 19% |
| サブホスト | 27% |

**図 3** 分割マイグレーション時の VM の実行性能

## 5. まとめ

本稿では，分割マイグレーションおよびリモートページングにおいてメモリ暗号化を最適化する *SEmigrate* を提案した．SEmigrate を実装し，分割マイグレーション時の VM の性能向上を調べる実験を行った結果，CPU 使用率を最大で 27%削減できた．

今後の課題は，メモリデータの整合性検査を最適化することと実アプリケーションに SEmigrate を適用し，その内部情報を用いて選択的なメモリ暗号化が行えるようにすることである．

**謝辞** 本研究の一部は，JST，CREST，JPMJCR21M4 の支援を受けたものである．また，本研究成果の一部は，国立研究開発法人情報通信研究機構の委託研究により得られたものである．

### 参考文献

[1] Suetake, M., Kizu, H. and Kourai, K.: S-memV: Split Migration of Large-memory Virtual Machines in IaaS Clouds, *In Proc. Int. Conf. Cloud Computing, 2018.*
[2] Li, C., Raghunathan, A. and Jha, N. K.: Secure Virtual Machine Execution under an Untrusted Management OS, *In Proc. Int. Conf. Cloud Computing, 2010.*
[3] Ozaki, Y., Kanamoto, S., Yamamoto, H. and Kourai, K.: Detecting System Failures with GPUs and LLVM, *In Proc. Asia-Pasific Workshop on System, 2019.*