平成 29 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	14237065	学生氏名	大和 功輝
論文題目	複数ホストに分割されたメモリを用いる仮想マシンの監視機構		

1　はじめに

近年，仮想マシン（VM）をユーザに提供する IaaS 型クラウドが普及している．IaaS 型クラウドでは大容量のメモリを持つ VM も提供されており，Amazon EC2 では 4TB のメモリを持つ VM が利用可能である．このような VM はビッグデータの解析などに必要とされている．一方，ホストのメンテナンス時などに VM を別のホストにマイグレーションする際には，移送先として十分な空きメモリを持つホストの確保が困難になる．そこで，複数のホストに VM のメモリ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．

しかし，VM のメモリが複数ホストに分割されていると，VM を外部から監視するのが難しくなる．従来，侵入検知システム（IDS）を VM 内で動作させて攻撃を検知していたが，攻撃者の侵入時に IDS が無効化される恐れがあった．この問題を解決するために，IDS を VM の外側で安全に動作させる IDS オフロードと呼ばれる手法が提案されている．オフロードされた IDS は VM のメモリ上の情報を基に攻撃を検知するが，メモリが一つのホスト上にないと必要な情報を取得するのが困難になる．

本研究では，複数ホストに分割されたメモリを用いる VM に対して IDS オフロードを可能にするシステム VMemTrans を提案する．

2　分割マイグレーション後の IDS オフロード

VM が動作しているホストをメンテナンスする際や負荷分散を行う際には，VM マイグレーションが行われる．マイグレーションは，VM を停止させずに他のホストに移送する技術である．マイグレーションの際には，移送元ホストの VM のメモリ上のデータがネットワーク経由で移送先ホストに転送される．そのため，移送先ホストに VM のメモリよりも大きな空きメモリが必要となる．しかし，大容量メモリを持つ VM の場合には，マイグレーションのために十分な空きメモリを持つホストを確保しておくのはコストの面などから困難である．

そこで，VM を分割してメモリを複数のホストへ転送することのでき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では，VM の核となる情報とアクセスが予測されるメモリのデータをメインホストへ転送し，メインホストに入り切らないメモリのデータをサブホスト群へ転送する．マイグレーション後には，VM がメインホスト上で実行され，VM がサブホスト上にあるメモリを必要とした際には，リモートページングと呼ばれる機構を用いてサブホストから必要なメモリデータを取得する．

しかし，分割マイグレーションを行うと IDS オフロードと呼ばれる手法を用いるのが難しくなる．IDS オフロードは従来，VM 内で動作していた IDS を VM の外側で安全に動作させるための手法である．IDS はアプリケーションの状態やディスク，ネットワークを監視することで攻撃を検出する．この IDS を VM の外側で動作させることにより，攻撃者が VM 内に侵入したとしても，IDS が無効化されるのを防ぐことができる．一方，オフロードされた IDS は VM のメモリを解析することで必要な情報を取得するが，メインホスト上にオフロードされた IDS はサブホスト上にある VM のメモリにはアクセスすることができない．逆に，IDS をサブホスト上にオフロードすると，メインホスト上にある VM のメモリにアクセスすることができない．

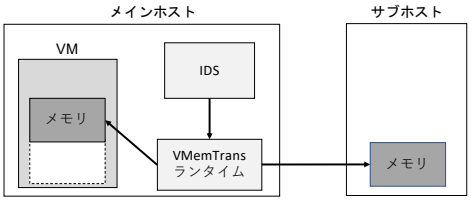


図 1: VMemTrans のシステム構成

3　VMemTrans

本研究では，複数ホストに分割されたメモリを用いる VM に対して IDS オフロードを可能にするシステム VMemTrans を提案する．VMemTrans では図 1 のように，VM が動作するメインホスト上に IDS をオフロードする．IDS は VMemTrans ランタイムを用いて動作し，サブホスト上にある VM のメモリにも透過的にアクセスすることができる．これにより，IDS は VM のメモリが分割されていないかのように VM の監視を行うことができる．

3.1　メインホスト上のメモリの共有

オフロードした IDS が VM のメモリにアクセスできるようにするために，VMemTrans ランタイムはメインホスト上にある VM のメモリを共有する．そのために，VM はメモリファイルと呼ばれるファイルにメモリのデータを格納し，ファイルとしてメモリを共有できるようにする．メモリファイルの内容はメモリ上に置かれるため，VM のメモリアクセス性能は低下しない．VMemTrans ランタイムはこのメモリファイルにアクセスすることにより，格納されたメモリデータを取得して IDS に提供する．メモリファイルはスパースファイルと呼ばれる特殊なファイルであり，メインホスト上にないメモリについてはファイル中のデータも存在しない．

3.2　サブホスト上のメモリへのアクセス

VMemTrans はサブホスト上の VM のメモリにアクセスするために 2 種類の手法を提供する．

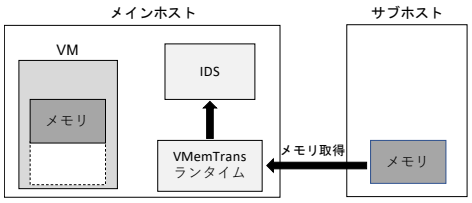


図 2: VMemTrans ランタイムによるメモリの取得

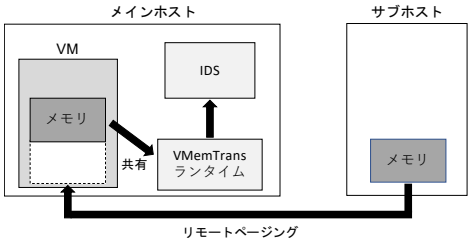


図 3: VM のリモートページングを利用したメモリの取得

3.2.1 VMemTrans ランタイムによるメモリ取得

1 つ目の手法は，図 2 のように VMemTrans ランタイム自身がサブホストからメモリを取得する手法である．IDS がメインホスト上に存在しないメモリにアクセスしたことを検出すると，VMemTrans ランタイムはサブホストにそのメモリのデータを要求する．サブホストからメモリデータを受信すると，それを IDS に提供する．この手法は VM のリモートページングとは独立してサブホスト上のメモリデータを取得可能であるため，VM のメモリアクセス性能に影響を与えない．一方で，取得したメモリを VMemTrans ランタイムが保持し続けると，メインホストのメモリ容量を圧迫する上，IDS に提供するデータが最新ではなくなる可能性がある．そこで，VMemTrans では最小限のメモリデータだけを保持する．

3.2.2 VM によるリモートページングの利用

もう一つの手法は，図 3 のようにリモートページングを用いて VM にサブホストのメモリを取得させる手法である．IDS によるメインホスト上に存在しないメモリへのアクセスを検出すると，VMemTrans ランタイムは VM にリモートページングを依頼するコマンドを送信する．コマンドを受け取った VM はリモートページングを行い，取得したサブホストのメモリデータをメモリファイルに書き込むことで VMemTrans ランタイムと共有する．この手法では，IDS が常に最新のメモリデータを監視することが可能であり，IDS が頻繁にアクセスするメモリデータはメインホスト上に保持され続ける可能性が高い．一方で，VM にコマンドを送信するオーバヘッドや，不要なメモリデータをサブホストに転送するオーバヘッドがあり，VM のメモリアクセス性能に影響を与える可能性がある．

3.3 Transcall との統合

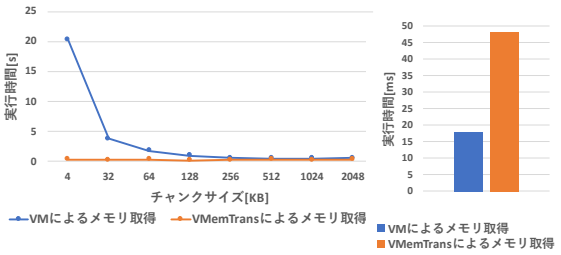
既存の IDS に修正を加えることなく IDS オフロードを可能にするために，Transcall [2] と VMemTrans を統合した．Transcall は VM 内のプロセスやネットワークなどの情報を取得してシャドウ proc ファイルシステムを構築し，既存の IDS に提供する．

4 実験

メモリが分割された VM に対して，VM 内の情報を取得してシャドウ proc ファイルシステムを構築するのに必要な時間を測定した．この実験では，一度に取得するデータ量 (チャンクサイズ) を変更しながら構築時間を測定し，3.2 節の 2 つのメモリ取得手法を用いた場合について比較を行った．また，シャドウ proc ファイルシステムの構築後に，VM 内での ps コマンドの実行時間を測定し，VM のメモリアクセス性能への影響を調べた．メインホストとサブホストとして，Intel Xeon E3-1225 v5 の CPU，8GB のメモリを搭載した 2 台のマシンを使用し，ギガビットイーサネットで接続した．各ホストでは分割マイグレーションを実装した Linux 4.11 および QEMU-KVM 2.4.1 を動作させた．VM には 2GB のメモリを割り当て，1GB ずつに分割した．

図 4 に実験結果を示す．図 4(a) より，VMemTrans ランタイムがメモリ取得を行う手法のほうが構築時間が短いことが分かる．VM がメモリ取得を行う手法では，チャンクサイズが小さい場合に構築時間が大幅に長くなった．これは，VM へのコマンド送信やリモートページングの回数が多くなるためである．チャンクサイズを大きくするとこれらのオーバヘッドは減り，VMemTrans ランタイムによるメモリ取得と比べて構築時間は約 1.6 倍となった．VMemTrans ランタイムがメモリ取得を行う手法では，VM へのコマンド送信を行わないためチャンクサイズの影響が小さいと考えられる．

一方，図 4(b) より，VM がメモリ取得を行った後のほうが ps コマンドの実行時間が短いことが分かる．これは，シャドウ proc ファイルシステムの構築に使用したデータがメインホスト上に保持されている可能性が高いためである．その場合には，VM 内で ps コマンドを実行した時に再度リモートページングを行う必要がない．



(a) シャドウ proc ファイルシステムの構築時間　(b) ps コマンドの実行時間

図 4: 実験結果

5 まとめ

本研究では，複数ホストに分割されたメモリを用いる VM に対して IDS オフロードを可能にするシステム VMemTrans を提案した．VMemTrans では，IDS は VM のメモリが分割されていることを意識することなく，監視を行うことができる．今後の課題は，IDS をメインホスト以外のホストへオフロードできるようにするなど，様々な監視方法に対応することである．

参考文献

[1] M. Suetake, H. Kizu, and K. Kourai. Split migration of large memory virtual machines, Proc. APSys 2016.

[2] 飯田貴大，光来健一．仮想マシンを用いた既存 IDS のオフロード, SWoPP 金沢 2010.