

仮想マシン移送における移送コストの定量的調査

古藤明音¹ 山田浩史^{2,3} 河野健二^{1,3}

概要: 近年, クラウドコンピューティングにおいてオペレーティングシステム (OS) の仮想化技術が広く利用されている. 特に OS の仮想化技術の一つである仮想マシン (VM) の Live Migration を用いることで, クラウド環境における大規模なリソースを効率良く管理することができる. 近年 Live Migration の研究は広く行われており, 様々な移送手法が提案されているが, いずれの手法でも Live Migration の実行により VM のパフォーマンスが低下する問題がある. 本研究ではこのような移送による VM のパフォーマンス低下を Migration Noise と定義する. Migration Noise の種類や度合いは移送方式やシステムの稼働状況により異なるため, 移送の実行を管理する側が状況に応じて Migration Noise を最も緩和する適切な Live Migration を実行する必要がある. 本研究では, 状況に応じて適切な移送方式を選択するため, 移送方式や仮想化環境により発生する Migration Noise の違いについて定量的に調査する. 調査の結果, Migration Noise の発生の方は同じ移送方式でも移送 VM のワークロードによって変化し, また同じワークロードでも移送方式により結果は異なった. また, 調査結果から移送方式ごとに発生する Migration Noise をいくつかの指標を用いることで定量的に比較できることを示した.

キーワード: 仮想化, Live Migration, サーバコンソリデーション

1. はじめに

近年, クラウドコンピューティングやサーバのクラスタリングにおいて, オペレーティングシステム (OS) の仮想化技術が広く利用されている. OS の仮想化技術とは, 一台の物理マシン上で複数の仮想マシン (VM) を稼働させる技術のことである. OS の仮想化技術を基盤とするクラウド環境では, クラウド上のサービスを仮想マシン (VM) を用いて提供することで, ユーザの要求合わせたリソースの伸縮がおこなえる. 例えば, Amazon EC2 [1] は 10 ヶ所以上のデータセンター上で, ユーザの要求に合わせた様々な種類の VM を提供している. こうした OS の仮想化技術の一つに Live Migration がある. Live Migration とは, ネットワークのコネクションやアプリケーションの実行を維持したまま VM を移送する技術である. 特にクラウド環境では Live Migration を利用することでより柔軟にリソースの伸縮をおこなうことができる. 例えば, VM 上で稼働しているサービスを停止させることなく物理サーバのメンテナンスを行ったり, サーバ上で発生した突発的な負

荷に追従した負荷分散をおこなえる. 近年 Live Migration の研究は広く行われており, 様々な移送手法が提案されている [2], [3].

しかしいずれの Live Migration 手法でも移送の実行により VM のパフォーマンスが低下する問題がある. こうした VM のパフォーマンス低下は Live Migration 実行時の計算リソースの消費や Live Migration の仕様が原因で発生する. Live Migration の実行には主に CPU 時間やネットワーク帯域を要する. そのため, 移送ホストで突発的に負荷が増大した際に Live Migration によって負荷を緩和させようとしても, 移送の実行中は負荷がさらに増大したり, 移送に時間がかかり急激な負荷に追従する負荷分散をおこなえない. VM を最適なホストに配置する研究 [4], [5], [6], [7] も広くおこなわれているが, こうした移送による VM のパフォーマンス低下は VM を最適なタイミングで配置する妨げとなり, モデル通りに配置することができない原因になる. 本研究ではこのような移送による VM のパフォーマンス低下を Migration Noise と定義する. Migration Noise が発生する要因は様々であり, 移送する VM だけでなく移送ホスト上で稼働するその他の VM (Co-located VM) のパフォーマンスも低下させる. その結果, Live Migration の実行により移送ホストを含むシステム全体のスループットが低下してしまう. クラウド

¹ 慶応義塾大学
Keio University

² 東京農工大学
Tokyo university of Agriculture and Technology

³ JST CREST

環境やクラスタ環境など実際の環境で発生する Migration Noise の種類や度合いは移送方式やシステムの稼働状況により異なるため、移送の実行を管理する者が状況に応じて Migration Noise を最も緩和する適切な Live Migration 手法を選択する必要がある。そこで本研究では状況に応じた最適な移送手法について調査するため、様々な状況で発生する Migration Noise について定量的に調査した。調査には典型的な Live Migration 手法である Xen Pre-Copy, KVM Pre-Copy, KVM Post-Copy を用い、CPU リソースの競合が引き起こす Migration Noise について調査した。調査の結果、Migration Noise は VM のワークロードや移送手法によって発生した場所や大きさが異なった。また、得られた結果をいくつかの軸にそって評価することで、3 種類の移送方式で発生した Migration Noise を定量的に比較することができた。

本論文の構成を以下に示す。2 章では Migration Noise がどのような原因で発生するのか、また移送方式によってどのような Migration Noise が発生し得るのかについて述べる。3 章では Migration Noise の評価方法について説明する。4 章では調査の概要を示す。5 章では調査結果とその結果についての考察を述べる。6 章では関連研究について説明する。7 章ではまとめを述べる。

2. Migration Noise

2.1 Migration Noise とは

Migration Noise は Live Migration の実行により移送 VM や Co-located VM のパフォーマンスが低下する現象である。Migration Noise の主な原因は、Live Migration を実行した際の移送プロセスによる計算リソースの消費や Live Migration の仕様で実行される動作による影響である。以下に Migration Noise を引き起こす要因の例を示す。これらはあくまで一例であり、クラウド環境などでは Live Migration 実行時の環境によって以下の例にない Migration Noise が発生する場合もある。また、これらの Migration Noise は必ずしも単一で発生するとは限らず、複数の Migration Noise が同時に発生する場合もある。

- **CPU 競合:** 移送プロセスの CPU リソースの消費により発生する CPU 競合。Live Migration 中、移送の実行は移送元と移送先のそれぞれで稼働する移送プロセスによりおこなわれる。移送元で稼働する移送プロセスは、移送 VM のメモリページをトラッキングしたりメモリページを移送先へ転送する。一方、移送先ではメモリページを受信し、そのメモリページから VM を再構成する。これらの処理は Live Migration 中常におこなわれているため、必要とされる CPU リソースは無視できないほど大きい。
- **ネットワーク I/O 競合:** 移送プロセスがネットワーク帯域を占有するために発生するネットワーク競合。

Live Migration 中は最低でも移送 VM のサイズ分のメモリページを転送する必要があるため、大きなネットワーク帯域を要する。そのため、移送と同じネットワークを使用している VM は Live Migration 中に処理速度が低下する。

- **メモリ解放遅延:** 移送プロセスが Live Migration を完了するまで移送 VM のメモリを保持するために発生するメモリの解放遅延。これは Live Migration が移送中は移送元と移送先で転送しているメモリページの一貫性を保持するという仕様であるために発生する。そのため、クラウド環境などで突発的なメモリ負荷を分散するために Live Migration を実行しても、転送したメモリページから順にメモリを解放することができないため即座に負荷を分散することができない。
- **ページフォールト:** 移送プロセスは Live Migration 中、移送したいメモリページを探索するためにページフォールトを用いる。しかし、ページフォールトが頻繁に発生すると移送 VM の稼働が頻繁に中断される。また、ページフォールトの処理にも CPU 時間が必要になるため、CPU 競合による Migration Noise も発生する。

移送プロセスによるリソース使用を削減する研究は広くおこなわれているが、いずれも移送中のダウンタイムや移送時間の削減に着目しており、Migration Noise について定量的な評価はおこなっていない。しかし、Migration Noise は Live Migration を実行時の環境がどのようなものであっても発生する。特に Live Migration の挙動はワークロードによって多様に変化するため、ダウンタイムや移送時間が大きくなると Migration Noise も無視できないほど大きくなる。そのため、システムの稼働状況を考慮して Live Migration を実行するためには Migration Noise を定量的に調査することが重要である。本研究では仮想化の実装、移送方式による Migration Noise の違いを調査し、ダウンタイムや移送時間などの移送全体の挙動から得られる性能と Migration Noise の関係性についても述べる。本研究は Migration Noise を定量的に評価するための指標を示し、システムの状態に応じて適切な移送手法を決定する手がかりになると予想される。

2.2 移送方式による Migration Noise の特徴

Migration Noise の種類や大きさ、発生する場所などは移送ホスト上で稼働する VM のワークロードだけでなく、移送方式によっても変化する。本研究では、代表的な移送方式である Pre-Copy Live Migration と Post-Copy Live Migration を用いて調査をおこなった。図 1 にそれぞれの移送手法の動作の流れを示す。それぞれの移送手法において発生する Migration Noise の違いは以下の通りである。

Pre-Copy Live Migration: Pre-Copy (図 1(a)) は移

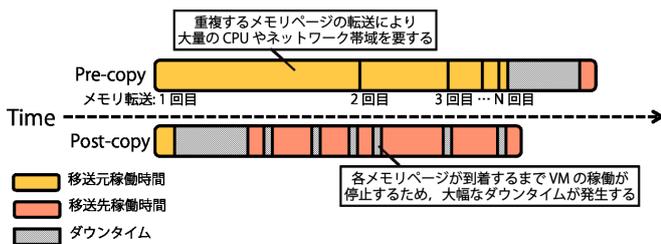


図 1 移送方式ごとの動作の流れ

送 VM のダウンタイムを考慮した移送手法である [2]。この手法では、移送元ホスト上で稼働する VM が Migration Noise の影響を受けやすい。それは、移送 VM が Pre-Copy を実行開始直後から完了する直前まで移送元ホストで稼働するためである。例えば、移送元ホストで CPU 競合が発生している際に Pre-Copy を実行すると、移送元ホストでは移送 VM と Co-located VM によるリソース消費に加えて移送プロセスがリソースを必要とするため、CPU 競合が深刻化する。また、Pre-Copy では移送 VM がサスペンド中に転送するメモリページを削減するために移送中に内容が更新されたメモリページを繰り返し転送する。これにより、Live Migration 実行中は常に多くのメモリ転送が発生し、ネットワーク I/O 競合による Migration Noise も発生しやすい。

Post-Copy Live Migration: Post-Copy (図 1(b)) は移送時間を考慮した移送手法である [3]。この手法では、移送先ホスト上で稼働する VM が Migration Noise の影響を受けやすい。これは、移送 VM が移送開始後すぐに移送先で稼働を再開するためである。Pre-Copy とは反対に、Post-Copy では移送先ホストで移送 VM、Co-located VM、移送プロセスによるリソース消費が発生するため、移送先ホスト上で CPU 競合による Migration Noise が発生しやすい。また、Post-Copy では移送 VM のメモリページは移送先での VM の稼働に応じ、必要なものから順に転送される。この時、移送 VM からのメモリページの要求はページフォールトを介しておこなわれるため、ページフォールトによる Migration Noise が発生しやすい。

3. 評価方法

本研究では、2.1 で列挙した Migration Noise のうち、CPU 競合により発生する Migration Noise について調査した。Migration Noise は VM が移送を実行していない時の平均 CPU 使用量と移送中に VM に割当てられた単位時間あたりの CPU 使用量の差分とする。また、定量的な評価をおこなうために、Migration Noise を以下の特徴に着目して調査した。

- **Migration Noise の総量:** Migration Noise を時間で積分した値。移送時間や単位時間あたりに発生する Migration Noise の影響が含まれており、Migration

Noise の大まかな特徴を捉えるための手がかりとなる。

- **最悪性能:** Migration Noise の最大値。Live Migration 実行時、VM に最低限保証される性能を示す。特に移送 VM では移送中に必ずダウンタイムが発生するため、ダウンタイムが単位時間を超えて大きくなると Migration Noise の最大値も大きくなる。
- **単位時間あたりの性能低下:** Migration Noise の平均値。Live Migration を実行中の VM の平均的なパフォーマンス低下を示す。
- **性能低下の安定性:** Migration Noise の標準偏差。単位時間あたりに発生する Migration Noise の動向を知る手がかりとなる。この値が小さいほど移送中の Migration Noise の変化は小さく、性能低下が安定していることを示す。

4. 調査概要

4.1 実験環境

調査には 2 台の移送ホスト用マシンと 1 台の NFS を用いた。VM は 2 台の Co-located VM を移送元ホスト、移送先ホストに 1 台ずつ配置し、1 台の移送 VM をそれらのホスト上で Live Migration した。3 台の物理マシンは Xeon 2.8GHz クアッドコア、32GB のメモリを搭載する Dell PowerEdge T610 を用いた。これらのマシンは Cisco Catalyst 3750G ネットワークスイッチとギガビットイーサネットを用いて接続した。VM はいずれも 1.5 GB のメモリと VCPU 1 つを搭載している。全てのホスト、VM 上では Linux 3.0.4 カーネルを稼働させた。仮想化環境には Xen と KVM を用い、Xen はバージョン 4.1.0 を、KVM は Linux 3.0.4 に内蔵しているものと QEMU 1.1-rc1 を組み合わせて用いた。調査をおこなった移送手法は Xen Pre-Copy、KVM Pre-Copy、KVM Post-Copy の 3 手法であり、Xen Pre-Copy と KVM Post-Copy は仮想化環境に既に実装されているものを用いた。KVM Post-Copy は、QEMU 1.1-rc1 に実装されているもの [8] を用意した。

4.2 ワークロード

移送 VM では Working set と Dirty-rate を調節できるワークロードを稼働させた。Working set とは、頻繁に更新されるメモリページの集合、Dirty-rate とはメモリの更新頻度のことである。移送の挙動は Working set と Dirty-rate によって変化する。Pre-Copy では直前の転送中に内容が更新されたメモリページを繰り返し転送するため、Working set や Dirty-rate の値が大きいほど繰り返し転送されるメモリページが増える。一方、Post-Copy では移送 VM が要求したメモリページがページフォールトを介して取り寄せられるため、Working set や Dirty-rate の値が大きいほどページフォールトが頻繁に発生する。本研究では移送 VM の Working set と Dirty-rate を変化によ

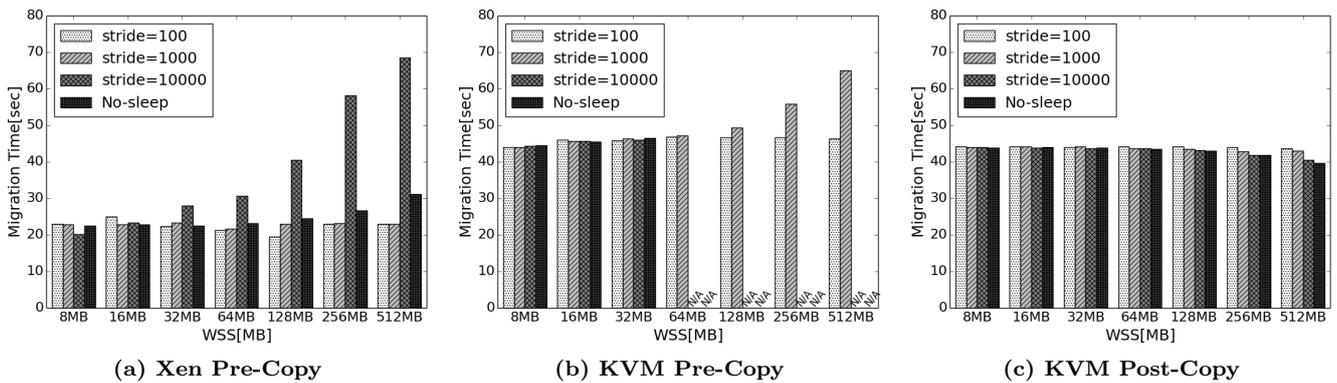


図 2 移送 VM のワークロードの変化による移送時間の変化

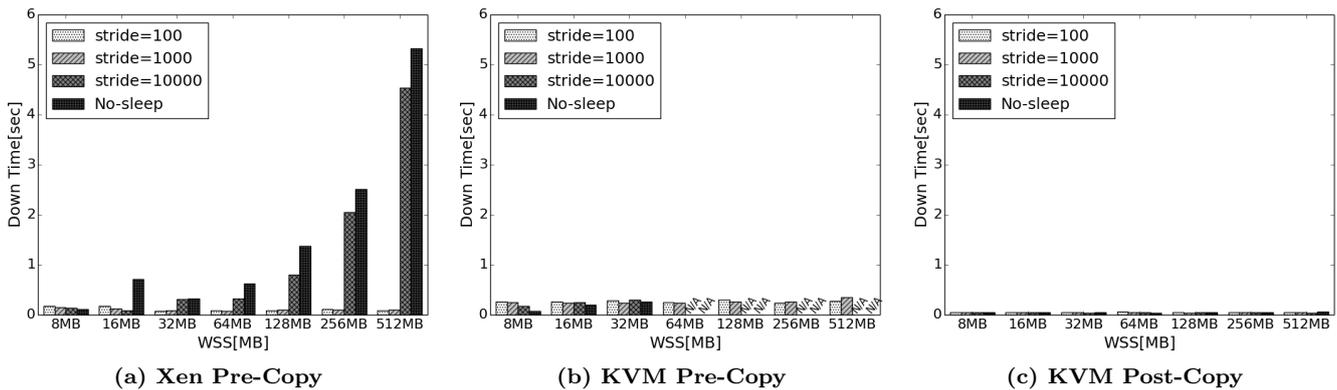


図 3 移送 VM のワークロードの変化によるダウンタイムの変化

る Migration Noise の変化を調査するため、*Workingset*、*Stride*、*Sleep* の 3 つのパラメータを与える C プログラムを用いて実験をおこなった。*Workingset* はメモリの更新範囲である Working set を設定するために、*Stride* と *Sleep* は Dirty-rate を設定するためのパラメータである。移送 VM では最初に *Workingset* 分のメモリページを取得し、それらのメモリページを *Stride* 個更新する度に *Sleep* 回、空の無限ループを周ることで一定頻度のメモリ更新をおこなう。今回は *Sleep* を 50 万回に固定して実験をおこなった。

Co-located VM では、CPU リソースへの影響のみを調査するため、空の無限ループを回る簡単な C プログラムを稼働させた。

本研究では、上記ワークロードを用いて移送元ホストと移送先ホスト上で 1 台ずつ Co-located VM を稼働させ、両ホスト上で CPU 競合が発生している状態にする。その状態で移送 VM を Live Migration した際の Migration Noise を調査した。調査は移送 VM と移送元、移送先の Co-located VM で発生する Migration Noise を対象におこなった。

5. 調査結果

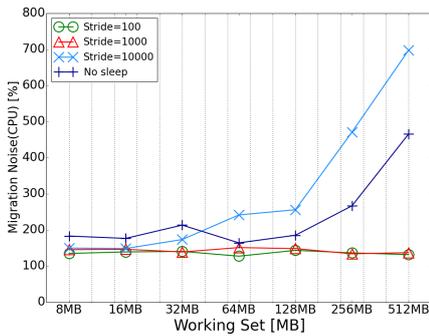
5.1 移送全体の挙動

移送時間とダウンタイムの調査結果をそれぞれ図 2、3 に示す。これらの結果から、それぞれの Live Migration

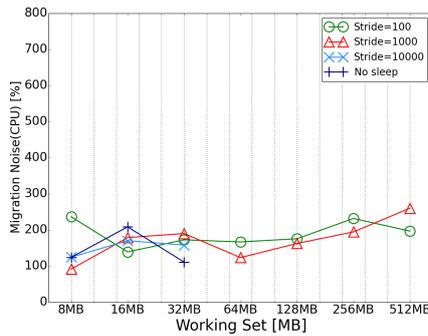
手法についていくつかの特徴を得ることができた。Xen Pre-Copy (図 2(a), 図 3(a)) では *Workingset*、*Dirty-rate* の値が大きくなると移送時間もダウンタイムも大きくなった。しかし、移送時間とダウンタイムの大きさにはワークロードに関係なく一貫した関係性は見られなかった。例えば *Workingset* が 512MB の時、ダウンタイムは *Dirty-rate* が大きいほど増加したが、移送時間は *Stride* が 10000 の時が最も大きくなった。KVM Pre-Copy では *Workingset*、*Dirty-rate* の値が大きくなると移送に失敗した。これは KVM Pre-Copy では移送専用のパラメータを Live Migration 実行時に設定しなければならないためである。特に、KVM Pre-Copy には移送 VM のダウンタイムの上限を設定するパラメータがある。これは、移送プロセスが移送 VM をサスペンドするタイミングを示すパラメータであり、移送中にメモリの転送状況から予測した VM のサスペンド時間がこの値を下回らなければ移送 VM はサスペンドされず移送を完了することができない。このパラメータは初期値がとて小さく、ワークロードに合わせて適切に値を設定しなければ移送に失敗してしまう。KVM Post-Copy では移送 VM のワークロードに関係なく移送時間、ダウンタイムは一定であった。

5.2 移送中の性能変化

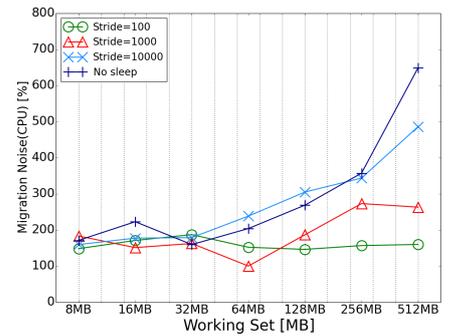
移送 VM、Co-located VM における Migration Noise を調査した結果、特に Migration Noise の総量、最大値、に



(a) Xen Pre-Copy (移送元)

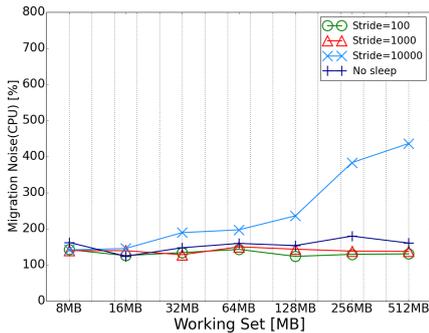


(b) KVM Pre-Copy (移送元)

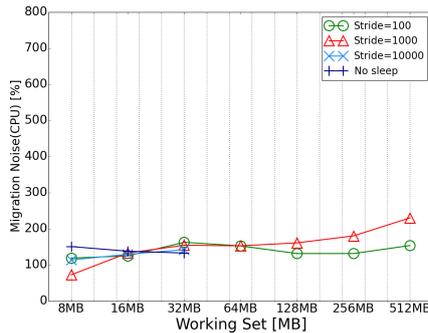


(b) KVM Post-Copy (移送先)

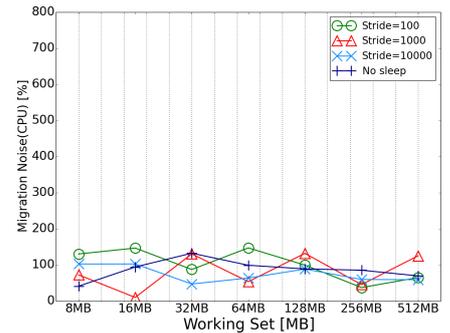
図 4 移送 VM における Migration Noise の総量



(a) Xen Pre-Copy (移送元)



(b) KVM Pre-Copy (移送元)



(b) KVM Post-Copy (移送先)

図 5 Co-located VM における Migration Noise の総量

おいて移送方式ごとに様々な結果が得られた。Migration Noise の総量の結果を図 4, 5 に示す。横軸は Working set の大きさ、縦軸は移送中の単位時間あたりの CPU リソースの低下量 (%) を積分した値を示す。2.2 でも述べたように、Pre-Copy は移送元ホスト、Post-Copy では移送先ホストでそれぞれ Migration Noise の影響を受けやすい。この特徴は調査結果においても現れ、移送 VM、Co-located VM 共に Pre-Copy 実行時の移送先ホスト、Post-Copy 実行時の移送元ホストではいずれも発生した Migration Noise は 10% 程度であった。よって、ここでは特に Pre-Copy を実行時に移送元ホスト上で稼働する VM と Post-Copy を実行時に移送先ホスト上で稼働する VM に着目し調査結果を述べる。

図 4 を見ると、いずれの手法でも Migration Noise の総量は 100% 以上になっていることが分かる。これは、移送中に移送プロセスが消費した CPU 時間がホスト上のリソース競争を深刻化させたためである。また Workingset, Dirty-rate の値が大きくなると、Xen Pre-Copy, KVM Pre-Copy 実行時に稼働していた移送 VM では Migration Noise の総量が増加したことが分かる。Xen Pre-Copy における Migration Noise の総量の変化は移送時間とダウンタイムによるものである。図 2, 図 3 から分かるように、Xen Pre-Copy では移送時間、ダウンタイム共にワークロードによって結果が変化している。一方、KVM Post-Copy における Migration Noise の総量の変化は、移送 VM への

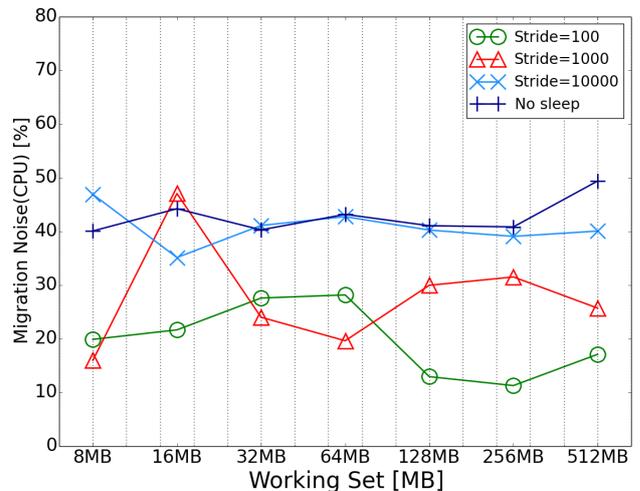


図 6 KVM Post-Copy 稼働時の移送 VM の最悪性能

リソースの割り当て遅延と、KVM のリソース割り当てのスケジューリング方法によるものである。2.2 でも述べたように、Post-Copy を実行すると、移送 VM は移送先での稼働に応じてメモリページを取り寄せる。そのため、Workingset や Dirty-rate の値が大きくなると要求が集中するため即座にリソースを割り当てるのが難しくなる。また、KVM では VM へのリソース割り当てのためのスケジューリングが安定せず、同一の実験をおこなっても結果にばらつきが生じることがあった。

Co-located VM で発生した Migration Noise の総量は、移送 VM の結果とは異なった。いずれの手法も、発生した

表 1 移送中の通常性能

移送方式	移送 VM		移送方式	Co-located VM(移送元)		Co-located VM(移送先)	
	最大値 - 最小値	標準偏差		最大値 - 最小値	標準偏差	最大値 - 最小値	標準偏差
Xen Pre-Copy (移送元)	14.54 - 5.55	1.83	Xen Pre-Copy	7.04 - 5.01	0.53	0.31 - 0.0	0.09
KVM Pre-Copy (移送元)	6.38 - 2.12	0.95	KVM Pre-Copy	3.72 - 1.72	0.44	0.00 - 0.00	0.00
KVM Post-Copy (移送先)	15.06 - 2.37	2.85	KVM Post-Copy	9.43 - 0.00	2.25	5.26 - 1.10	0.99

表 2 性能低下の安定性

移送方式	移送 VM		移送方式	Co-located VM(移送元)		Co-located VM(移送先)	
	最大値 - 最小値	標準偏差		最大値 - 最小値	標準偏差	最大値 - 最小値	標準偏差
Xen Pre-Copy (移送元)	16.38 - 2.37	3.35	Xen Pre-Copy	3.51 - 2.32	0.30	1.23 - 0.0	0.37
KVM Pre-Copy (移送元)	12.13 - 2.04	3.17	KVM Pre-Copy	5.81 - 0.96	1.47	0.00 - 0.00	0.00
KVM Post-Copy (移送先)	15.31 - 1.64	3.48	KVM Post-Copy	0.00 - 0.00	0.00	2.83 - 0.73	0.49

Migration Noise の総量はほとんど 200% 以内であった。Xen Pre-Copy においては、一部のワークロードにおいて値が大きく変化したが、これは移送時間によるものである。KVM Post-Copy ではワークロードに関係なく Migration Noise の総量にばらつきがあることが分かる。これは、移送 VM においても発生していた、KVM のリソーススケジューリングによるものである。

KVM のリソーススケジューリングの影響は移送 VM の最悪性能にも現れた。図 6 は KVM Post-Copy を実行した際の移送 VM における Migration Noise の最悪性能である。Workingset が 16MB の時の結果に注目すると、Stride が 1000 の時の移送 VM の性能低下の最大値が Stride が 1000 以上の場合の結果を下回っていることが分かる。これは Stride が 1000 の時、移送 VM が移送先で稼働を再開してから数秒後に急激にリソースを割当てられたためである。

Xen Pre-Copy ではダウンタイムの影響が最悪性能に影響を与え、特に Stride が 10000 以上の場合では 50% 程度になった。一方、KVM Pre-Copy では最悪性能はいずれのワークロードでも移送に失敗しない限り 10 ~ 20% 程度にとどまった。

移送中の単位時間あたりの性能低下、移送低下の安定性についてはいずれも結果は横ばいであった。表 1 は移送中の単位時間あたりの性能低下の結果である。表内の結果は Working set と Dirty-rate パラメータの全ての組み合わせから得られた Migration Noise の平均値から最大のもの、最小のもの、平均値のばらつき (標準偏差) を示す。Xen Pre-Copy では、他の手法に比べ単位時間あたりの性能低下は大きいものの、ワークロードの変化による Migration Noise の平均値の変化はあまり見られなかった。KVM Pre-Copy では移送 VM、Co-located VM 共に移送中の単位時間あたりの性能低下はいずれの手法よりも小さく安定しているが、これは移送 VM のワークロードが軽い状況でしか移送に成功しなかったためだと言える。KVM Post-Copy では他の手法に比べ、移送中の平均的な性能低

下がワークロードによって大きく変化した。性能低下の平均が最も高くなる場合はいずれの手法よりも値が大きく、性能低下の平均が最も低い時には他の手法の結果よりも値が下回る時もあった。

表 2 は性能低下の安定性についての結果である。移送 VM で単位時間あたりに発生した Migration Noise の大きさは、いずれの手法でもそれほど移送中ほとんど一定であった。Co-located VM で発生した Migration Noise の動向は VM がどのホスト上で稼働しているかによって異なった。移送元では KVM Post-Copy を実行中に発生した Migration Noise の変化は最も小さかったが、移送先ホストでは KVM Post-Copy を実行中に発生した Migration Noise は移送中に最も変化した。これは KVM における移送プロセスのリソース消費量が小さいため、移送 VM が稼働していないホストではリソース競合による Migration Noise の影響をほとんど受けなかったと言える。

6. 関連研究

Live Migration 中のネットワークトラフィックを削減する手法が広く研究されている [9], [10], [11], [12], [13]。MECOM [9] は移送 VM のメモリページの特徴を考慮したメモリ圧縮機能を用いて転送するメモリの総量を削減した。Svard らは移送 VM のメモリページのうち、頻繁にアクセスされるメモリページを一つのまとまりとして扱い、それらを排他的論理和によって圧縮することでメモリの総量を削減した [10]。CR/RT-Motion [11] は移送 VM の動作を追跡したログのみを移送先へ転送し、移送先ではそのログを用いて VM を再現した。これらの研究は移送プロセスのリソース消費の削減に着目した研究であり、移送 VM や Co-located VM のパフォーマンス変化や移送中の性能変化を考慮していない。

Live Migration の実行をモデル化することで Migration Noise を緩和する研究がされている。Breitgand ら [14] は Pre-Copy Live Migration の実行に必要なネットワーク帯域を移送のステップごとに決定することで移送コストをモ

デル化した。Lim ら [15] は Live Migration の動作を移送元と移送先の移送プロセスの組として扱い、移送コストをモデル化した。これらの研究は代表的なワークロードを実行する際には効果的なモデルであるが、クラウド環境で発生する突発的な負荷増加などの予測が難しいワークロードについては考慮していない。

様々な移送手法を定量的に比較している研究がある。様々な Live Storage Migration 手法について定量的に調査した研究がある [16]。この研究では 3 種類の移送手法について移送コストや移送 VM のパフォーマンスへの影響について定量的に調査している。しかし、この研究は VM のストレージを転送する手法についての研究であり、本研究で対象としている VM のメモリの転送手法とは異なる。Hu ら [17] は Live Migration のパフォーマンスをいくつかの軸を用いて評価することで Live Migration の稼働を動的に評価するテスト用フレームワークを実装した。この研究では Co-located VM のパフォーマンスへの影響については考慮していない。本研究では移送 VM と Co-located VM の両方について定量的な調査をおこなっている。

7. まとめ

VM の Live Migration はクラウド環境におけるコンピュータリソースを柔軟に管理するのに有用な技術である。Live Migration は広く研究されており様々な移送手法が存在するが、いずれの手法においても Live Migration の実行により移送 VM や移送ホスト上で稼働するその他の VM のスループットが低下する問題が起こる。本研究ではこのような移送による性能低下を Migration Noise と定義した。Migration Noise の大きさや種類は仮想化環境や移送方式、VM のワークロードなど、移送実行時の環境に応じて異なる。Live Migration に関する研究はいずれも移送手法の評価に移送時間やダウンタイムなど、移送全体を通して得られた結果を用いていることが多いが、Migration Noise はワークロードによらず必ず発生するものであり評価に加える必要がある。そこで本研究では Xen Pre-Copy, KVM Pre-Copy, KVM Post-Copy の 3 種類の Live Migration をそれぞれ実行した際、移送ホストの CPU 競合により発生する Migration Noise を定量的に調査した。調査の結果、Migration Noise をいくつかの軸を用いて評価することで、Migration Noise を定量的に比較することができた。特に、一度の移送実行で発生する Migration Noise の総量、移送中の VM の最悪性能は移送方式やワークロードによって異なることがわかった。本研究は Migration Noise を定量的に評価するための指標を示し、システム状況に応じて適切な移送手法を決定する手がかりになると予想される。

参考文献

- [1] Amazon.com: Amazon EC2: Amazon Elastic Compute Cloud, <http://aws.amazon.com/ec2/>.
- [2] Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I. and Warfield, A.: Live Migration of Virtual Machines, *Proc. of the 2nd USENIX Symp. on Networked Systems Design and Implementation*, pp. 273–286 (2005).
- [3] Hines, M. R. and Gopalan, K.: Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning, *Proc. of the 2009 ACM Int'l Conf. on Virtual Execution Environments*, pp. 51–60 (2009).
- [4] Govindan, S., Wang, D., Sivasubramaniam, A. and Uraganekar, B.: Leveraging Stored Energy for Handling Power Emergencies in Aggressively Provisioned Datacenters, *Proc. of the 17th ACM Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, pp. 75–86 (2012).
- [5] Hermenier, F., Lorca, X., Menaud, J.-M., Muller, G. and Lawall, J.: Entropy: a Consolidation Manager for Clusters, *Proc. of the 2009 ACM Int'l Conf. on Virtual Execution Environments*, pp. 41–50 (2009).
- [6] Verma, A., Ahuja, P. and Neogi, A.: pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems, *Proc. of the 9th ACM/IFIP/USENIX Int'l Conf. on Middleware*, pp. 243–264 (2008).
- [7] Wood, T., Venkataramani, A. and Yousif, M.: Black-box and Gray-box Strategies for Virtual Machine Migration, *Proc. of the 4th USENIX Symp. on Networked Systems Design and Implementation*, pp. 229–242 (2007).
- [8] Hirofuchi, T. and Yamahata, I.: Yabusame: Postcopy Live Migration for Qemu/KVM, *KVM Forum 2011* (2011).
- [9] Jin, H., Deng, L., Wu, S., Shi, X. and Pan, X.: Live Virtual Machine Migration with Adaptive Memory Compression, *Proc. of 2009 IEEE Int'l Conf. on Cluster Computing*, pp. 1–10 (2009).
- [10] Svård, P., Hudzia, B., Tordsson, J. and Elmroth, E.: Evaluation of Delta Compression Techniques for Efficient Live Migration of Large Virtual Machines, *Proc. of the 7th ACM Int'l Conf. on Virtual Execution Environments*, pp. 111–120 (2011).
- [11] Liu, H., Jin, H., Liao, X., Hu, L. and Yu, C.: Live Migration of Virtual Machine Based on Full System Trace and Replay, *Proc. of the 18th ACM Int'l Symp. on High Performance Distributed Computing*, pp. 101–110 (2009).
- [12] Jo, C., Gustafsson, E., Son, J. and Egger, B.: Efficient Live Migration of Virtual Machines Using Shared Storage, *Proceedings of the 9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pp. 41–50 (online), available from (<http://doi.acm.org/10.1145/2451512.2451524>) (2013).
- [13] Chiang, J.-H., Li, H.-L. and Chiueh, T.-c.: Introspection-based Memory De-duplication and Migration, *Proceedings of the 9th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pp. 51–62 (online), available from (<http://doi.acm.org/10.1145/2451512.2451525>) (2013).
- [14] Breitgand, D., Kutiel, G. and Raz, D.: Cost-Aware Live Migration of Services in the Cloud, *Proc. of Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services* (2011).

- [15] Lim, S.-H., Huh, J.-S., Kim, Y. and Das, C. R.: Migration, Assignment, and Scheduling of Jobs in Virtualized Environment, *Proc. of the 3rd USENIX Workshop on Hot Topics in Cloud Computing* (2011).
- [16] Mashtizadeh, A., Emré, Garfinkel, T. and Cai, M.: The Design and Evolution of Live Storage Migration in VMware ESX, *Proc. of the 2011 USENIX Conf. on USENIX Annual Technical Conf.* (2011).
- [17] Hu, W., Hicks, A., Zhang, L., Dow, E. M., Soni, V., Jiang, H., Bull, R. and Matthews, J. N.: A Quantitative Study of Virtual Machine Live Migration, *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*, pp. 11:1–11:10 (online), available from (<http://doi.acm.org/10.1145/2494621.2494622>) (2013).