Vol.2018-DPS-175 No.16
Vol.2018-MBL-87 No.16
Vol.2018-ITS-73 No.16
2018/5/24

サービスの停止時間を短縮するプログラム実行環境のプリコピー移送手法

黒木 勇作[1]　大西 史洋[1]　横山 和俊[1]　谷口 秀夫[2]

概要：クラウドコンピューティング環境を提供する広域分散システムでは、集中した負荷やデータを他の計算機に分散させるため、プログラム実行環境の移送が頻繁に行われる。代表的な手法である仮想マシンを用いる手法は、仮想マシンの全データを移送させるため、非常に時間がかかる問題がある。これに対し、移送対象のプログラム実行環境を特定し、それらを分割して転送する手法が提案されている。本稿では、移送対象の分割転送について、サービスの停止時間を短縮する手法を提案する。具体的には、シミュレーション評価により、ファイルへの更新頻度を用いて転送順序を決定する方法が、サービスの停止時間の短縮に効果的であることを示す。

キーワード：クラウドコンピューティング、仮想マシン移送、プリコピー

1. はじめに

近年、クラウドコンピューティングが急速に普及している。クラウドコンピューティング環境は、地理的に離れた計算機同士がネットワークにより接続される広域分散システムとして実現されていることが多い。広域分散システムでは、各計算機で分散処理を行い、各計算機にかかる負荷を分散させることが行われる。負荷が集中する計算機が存在する場合は、計算機上の AP とその実行環境を他の計算機に移送し、負荷を分散させる。このため、AP とその実行環境を効率よく移送することが重要となっている。

現在、広域分散システム上の移送技術として、仮想マシンを移送させる手法がある。この手法は仮想マシン上の全資源を移送するため、移送したい AP の資源だけでなく、他の AP の資源も移送してしまう。広域分散システムにおいて、ネットワークの転送速度は、データセンタ内では比較的高速であるが、地理的に離れたデータセンタ間では低速であることが多い。そのため、余分な資源の移送により、移送に時間がかかり、ネットワークへの負荷も増大する問題がある。この問題に対し、我々は、ファイルへのアクセス情報を用いたプログラム実行環境の移送手法を提案した [1]。提案手法は、AP のファイルへのアクセス情報を利用し、使用中のファイル等を特定し、必要最小限の資源のみを移送する。これにより、移送にかかる時間を短縮している。また、移送対象の資源を分割して転送するプリコピー移送手法を用いることで、プログラムの実行中断を抑制し、サービス停止時間を短くしている。本稿では、分割して転送するステップについて、ファイルの転送順序がプログラム実行の停止時間に与える影響について述べる。

[1] 高知工科大学情報学群
School of Information, Kochi University of Technology
[2] 岡山大学大学院
Graduate School of Natural Science and Technology, Okayama University

2. ファイルへのアクセス情報を用いたプログラム実行環境のプリコピー移送手法

我々は、システム上で稼働している AP がファイルにアクセスする情報を移送に利用する移送手法の研究を行っている。提案手法は移送対象を特定する「特定ステップ」と、特定した対象を移送する「転送ステップ」の 2 ステップで移送を行う。この章ではこれら 2 つのステップについて説明する。

2.1 特定ステップ

特定ステップでは、計算機上から移送したい資源の特定を行う [2]。移送対象を特定するために、各 AP が発行する open システムコールを利用する。特定機構の概要図を図 1 に示す。AP からファイルへとアクセスする際に AP が発行する open システムコールは、システムコールライブラリを経由してカーネルでの open 処理に移行する。提案手法では、システム監視機構をシステムコールライブラリで実現している。、システムコール監視機構は、どの AP がど

Vol.2018-DPS-175 No.16
Vol.2018-MBL-87 No.16
Vol.2018-ITS-73 No.16
2018/5/24

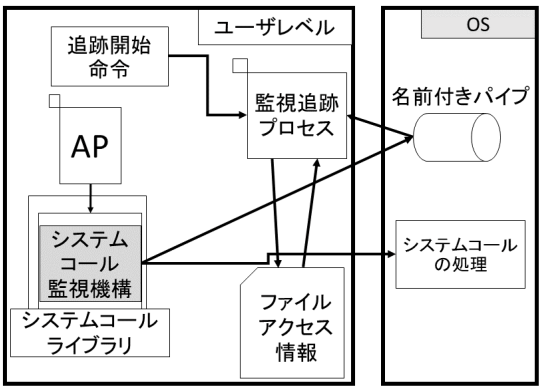

図 1　特定機構の概要図

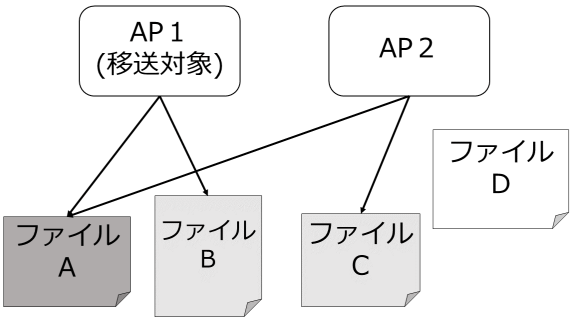

図 2　共有関係の特定

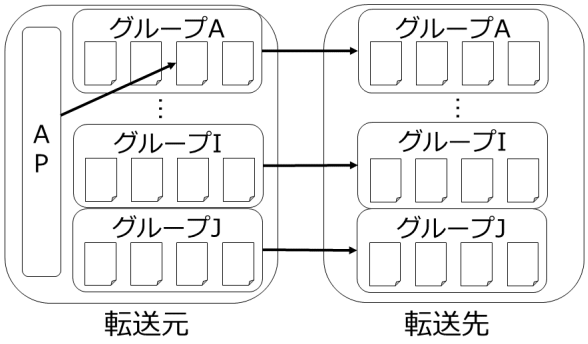

図 3　プリコピーフェーズ

のファイルを open し利用したのかを監視し、それらの情報を名前付きパイプを通して、追跡プロセスに送信する。追跡プロセスは、それらのアクセス情報を、定期的に二次記憶装置等に記録する。ここで、特定する資源に漏れがないように、十分な時間監視を行い、記録した情報から移送対象となる AP が利用したファイル資源をまとめて移送対象とする。

外部から追跡開始命令が追跡プロセスに与えられると、追跡プロセスは二次記憶装置からアクセス情報を読み出し、ファイルの共有関係を考慮して移送対象を特定する。ファイルの共有関係を考慮した追跡の様子を図 2 を用いて説明する。図 2 に示すように、AP として AP1 と AP2 が実行されている。また、ファイル資源として、ファイル A、ファイル B、ファイル C、ファイル D がある。ここで AP1 が利用するファイルは、ファイル A とファイル B であり、AP2 が利用するファイルは、ファイル A とファイル C である。すなわちファイル A が 2 つの AP によって共有されている。ここで AP1 を移送する場合を考える。AP1 がファイル A に対し、何らかの書き込み処理を行う場合、AP2 がその情報に依存して動作している場合がある。そのため AP1 とファイル A、ファイル B のみを移送した場合、AP2 が AP1 からの情報を受け取ることができず、意図しない動作をする可能性がある。この可能性を考慮し、AP1 を移送する場合、追跡プロセスが、AP1、ファイル A、ファイル B に加え、ファイル A を共有する AP2 と、AP2 が利用しているファイル C も特定し移送対象とする。

2.2　転送ステップ

転送ステップでは、特定ステップで決定した移送対象の資源を転送する。このとき、移送対象をいくつかのグループに分割して転送する。本ステップでは、AP を動作させながら転送を行う「プリコピーフェーズ」と、AP を停止させて転送を行う「最終コピーフェーズ」の 2 段階のフェーズで全資源の転送を行う。これら 2 つのフェーズについて以下で説明する。

2.2.1　プリコピーフェーズ

プリコピーフェーズにおける具体的な転送例を図 3 に示す。図 3 のように、資源をグループ A からグループ J までの 10 個のグループに分割する。プリコピーフェーズでは、グループ単位で逐次転送する。プリコピーフェーズでは AP を動作させながらファイルの転送を行うので、サービスを停止させずに転送が可能である。一方で、動作している AP が移送対象となるファイルに対し更新処理を行う場合があり、その場合は、更新された該当ファイルを再送しなければならない。再送が増加すると全体の転送量が増加するため、ネットワークへの負荷と移送時間の増加の原因となる。そのため、プリコピーフェーズにおいては、この再送を可能な限り抑制する必要がある。

2.2.2　最終コピーフェーズ

最終コピーフェーズにおける転送例を図 4 に示す。図 4 は、プリコピーフェーズの終了直後である。図 4 に示すように、プリコピーフェーズで、グループ A からグループ J までのグループにおいて、一部のファイルを除くファイルが転送済である。残った未転送であるファイルは、グループ A からグループ J のファイルのうち、プリコピーフェー

Vol.2018-DPS-175 No.16
Vol.2018-MBL-87 No.16
Vol.2018-ITS-73 No.16
2018/5/24

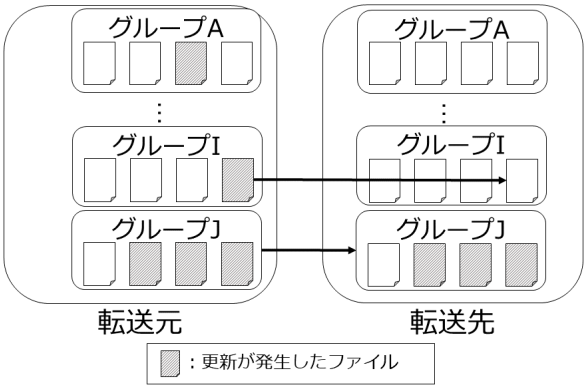

図 4　最終コピーフェーズ

ズ中に更新が発生し、再送が必要となったファイルである。図 4 ではプリコピーフェーズ中に更新されたファイルを網かけして示している。最終コピーフェーズでは AP を停止させて未転送である残りのファイルの転送を行う。つまり、AP を停止させるため、最終コピーフェーズが長期化するとサービスの停止時間が増加する。そのため、最終コピーフェーズは可能な限り短時間で終了させる必要がある。

3. 関連研究とこれまでの取組み

この章では、関連研究とこれまでの我々の取組みを説明する。

3.1 関連研究

プリコピー型ライブマイグレーションに関する関連研究として、文献 [3] に示す研究がある。文献 [3] は、メモリを対象としたプリコピー型ライブマイグレーションを実現したものである。一般的なプリコピーマイグレーションでは、物理メモリの先頭から順に移送先にコピーし転送を行う。文献 [3] では、移送に移行する前に、各メモリページの更新頻度を基に更新頻度の低いものを優先して転送するように転送の順序を操作することで、総ページ転送量を増加させることなく、プリコピーの効率化を実現している。同時にマイグレーションにおけるマシンの停止時間を短縮している。この手法は固定サイズのメモリを対象としているため、可変サイズのファイルに対しての適用が難しいことが問題となる。

3.2 これまでの取組み

我々はこれまでに、期待値を用いたプリコピーデータ量の削減手法を提案している。この手法では、ファイルのサイズとファイルの更新頻度を用いる。ファイルが持つ期待値は、ファイルサイズと更新頻度を乗算した値である。この期待値が小さい順に転送を行うことで、プリコピーデータ量の削減を実現している。

文献 [3] に示した既存手法では、メモリページの情報によりマイグレーション開始前に転送順序を決定し、プリコピー中に転送順序を変更しない。そのため、移送中の更新のバラつきが、移送結果に影響を及ぼす場合がある。これに対し提案手法では、周期単位で転送順序を計算し決定する。AP が発行する write システムコールが一定回数実行される期間を、「プリコピーフェーズ 1 周期」とする。1 周期当たりの write システムコール実行回数が小さい場合、転送順序を短期間で再構成するためより正確に転送を行うことができる。

更新期待値を用いる手法は、更新頻度の高いファイルやサイズの大きなファイルが再送されることによる再送量の増加を防止することを目的としている。そのため、ファイルサイズの小さいファイルや、更新頻度の小さいファイルを優先して転送することにより、プリコピーフェーズの再送量を削減している。具体的には、既存手法である文献 [3] と比較して最大 93 %のプリコピーデータ量の削減に成功している。一方、最終コピーデータ量も既存方式と比較して削減に成功しているが、最大 19 %の削減となっている。この手法ではプリコピーフェーズの転送効率化のため、サイズの大きなファイルの転送を後回しにするので、最終コピーデータ量が大きくなる場合もある。また、最終コピーデータ量はプリコピーフェーズでの再送量に完全に依存しているため、最終コピーフェーズでの AP の停止時間が一定ではないという問題がある。

4. サービスの停止時間を短縮する転送手法

4.1 提案手法の概要

本稿では、最終コピーフェーズでの AP 実行の中断を短縮するため、最終コピーデータ量を削減する転送方法を検討する。基本的な動作は以下の通りである。

(1) 最終コピーフェーズに移行する条件として、最終コピーデータ量の上限値を指定する、例えば上限値を X と指定する。

(2) 最終コピーデータ量が X 以下になるまでプリコピーを繰り返す。どのファイルを優先して転送するかは、後述の転送順序を用いる。

(3) 1 周期のプリコピーが終了後、未転送のファイルの総サイズを調べ、X 以下であれば、最終コピーフェーズに移行する。

4.2 転送順序

ファイルの転送順序を変化させ、最も早くプリコピーフェーズが終了する転送順序、すなわち、プリコピーデータ量が最も小さい転送順序を調査する。具体的には、以下の情報を利用する。

4.2.1 アクセス種別

ファイルが AP からアクセスされる場合、読み取り専用である Read-Only ファイルと、書き込み更新の可能性がある Read-Write ファイルの 2 種類に大別できる。アクセス種別が Read-Only の場合、AP の書き込みによって更新される可能性はないため、Read-Write ファイルよりも優先して転送することができる。つまり、Read-Only ファイルを全て転送した後、Read-Write ファイルの転送を開始する。

4.2.2 ファイルサイズと更新頻度

ファイルの更新頻度は単位時間当たりに、AP が特定のファイルを更新する回数を表わす。転送対象となるファイルは各々異なるサイズを持っており、更新頻度も異なる。そこで、ファイルサイズの大きい順 (サイズ降順)、ファイルサイズの小さい順 (サイズ昇順)、および更新頻度の小さい順 (更新頻度昇順) の 3 種類の転送順序における転送を評価する。以下で、各転送順序からどのような結果が期待できるのかを説明する。

(1) サイズ降順では、ファイルサイズの大きいものを優先して転送する。1 ファイル当たりのサイズが大きいため、転送するファイルの数を少なく抑えつつ、より多くのデータを転送することができるため、早い段階で一定の最終コピーデータ量へ到達することが期待できる。一方で転送済ファイルはサイズが大きいため、再送が発生した場合再送量が増大しやすいという問題がある。

(2) サイズ昇順では、ファイルサイズの小さいものを優先して転送する。サイズの小さいファイルを優先して転送することで、同データ量でサイズ降順で比較した場合、転送済ファイルの数は多くなるため、再送の可能性は上昇する点は問題となる。一方で、再送が発生しても再送量は増大しにくいため、ある程度更新頻度の高いファイルを転送した場合でも、再送の頻繁によって未転送ファイルの転送が遅れる可能性が小さくなり、ファイル全体に対して効率よく転送を行えることが期待できる。

(3) 更新頻度昇順では、ファイルの更新が小さいファイルを優先して転送する。再送ファイルの発生は、転送済ファイルの AP による更新が原因であるので、更新頻度の高いファイルを転送すると、再送も発生しやすくなる。そのため、周期ごとの転送順序の決定段階で更新頻度の高いファイルは、その後も更新されやすいと判断し、転送を後回しにすることで転送後の更新の発生を抑制する。

5. 評価と考察

5.1 評価内容

前章で説明した転送順序の評価を行った。既存手法として文献 [3] の手法を後述するデータセットに対応させたものを用いる。

5.1.1 評価に用いるデータセット・パラメータ

評価に用いたデータセットを表 1 に、設定するパラメータを表 2 に示す。表 1 中の R-O は Read-Only、R-W は Read-Write をそれぞれ省略したものである。表 1 に示すように、200KB、400KB、800KB、1600KB、3200KB の 5 種類のサイズのファイルを用意した。Read-Only ファイルは合計 1000 個、Read-Write ファイルは合計 10000 個になるように設定している。更新の頻度は Read-Write ファイル全体に対し、完全に均等にアクセスが発生するパターンと、アクセスを 1:10:10:79 に偏らせたパターンの 2 パターンで write アクセス、すなわち更新が発生すると仮定する。

また、表 2 に示すように、本評価では転送速度を 5Mbps とする。write システムコール 1 回にかかる時間は 0.03 秒とし、プリコピーフェーズ 1 周期はプリコピーが 500 回行われた時点とする。プリコピーフェーズは 500 周期行う。

5.1.2 プリコピーフェーズの終了条件

プリコピーフェーズの終了条件であり最終コピーデータ量の上限値は以下のように設定した。

(1) 更新の発生が均等の場合：5500MB

(2) 更新の発生を偏らせた場合：5000MB

評価では、既存手法、サイズ降順、サイズ昇順、更新頻度昇順のそれぞれについて、最終コピーフェーズに移行した時のプリコピーデータ量を比較する。

表 1 ファイルデータセット

サイズ	R-O ファイル数	R-W ファイル数
200KB	350	3,500
400KB	300	3,000
800KB	200	2,000
1,600KB	100	1,000
3,200KB	50	500
合計	1,000	10,000

表 2 シミュレーションで用いたパラメータ

転送速度	5Mbps
システムコールに要する時間	0.03s
1 周期あたりのプリコピー回数	500 回
1 周期あたりの最大転送量	75MB

5.2 評価結果と考察

5.2.1 更新の発生が均等の場合

図 5 はファイルの更新が均等に発生する場合の、プリコピーデータ量のグラフである。横軸は各転送順序、縦軸はプリコピーデータ量を表している。既存手法のプリコピーフェーズ周期は提案手法と異なり、全ファイルに対し転送を実行し、それを 10 回繰り返す。更新が均等に発生する場合では、提案手法の各転送順序はいずれも既存手法と比

較すると良い結果を示しているが、各転送順序で比較すると差が出ていない。更新が均等に発生するため、転送順序による差が出なかったことが理由である。

5.2.2 更新の発生を偏らせた場合

図 6 はファイルの更新が 1:10:10:79 と偏って発生する場合の、プリコピーデータ量のグラフである。図 5 と同様に、横軸は各転送順序、縦軸はプリコピーデータ量を表している。更新が偏って発生する場合では、サイズ昇順、すなわちファイルサイズの小さいものから転送することでより早く目的の最終コピーデータ量に達していることがわかる。ファイルサイズが大きいファイルを優先して転送すると、サイズの大きな再送が転送の大部分を占めてしまうことで、プリコピーデータ量が削減されない原因になったことが考えられる。また、更新頻度昇順、すなわち更新が発生する確率の小さいファイルを優先して転送した場合、最も早く目的の最終コピーデータ量に達していることがわかる。優先して転送したファイルの更新率が小さく、再送を抑制することで、プリコピーデータ量を増加させることができたと考えられる。

5.3 最終コピーデータ量の推移

5.3.1 最終コピーデータ量の評価結果

これまでは、ファイルの転送順序を変化させ、最も早くプリコピーフェーズが終了する転送順序、すなわち、プリコピーデータ量が最も小さい転送順序を調査した。ここでは上記評価に加え、プリコピーフェーズを十分に行う場合、最終コピーデータ量が最大どの程度削減できるのかを調査した。使用するデータセットおよびパラメータは前述の評価と同様のものを用いるが、プリコピーフェーズは、最終コピーデータ量にかかわらず、300 周期行う。また、更新が均等の場合、上記評価より効果的な結果が得られないことが予想されるため、更新が偏っている場合のみの調査とした。

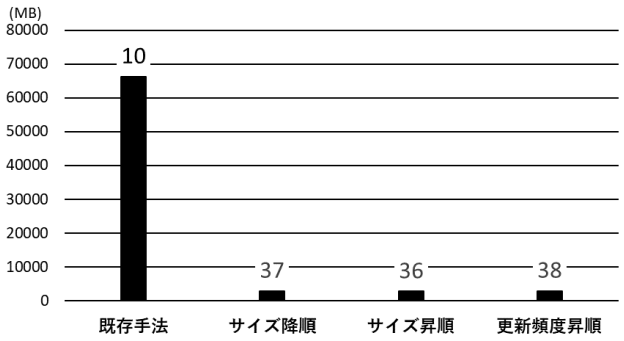

図 5 均等の場合のプリコピーデータ量

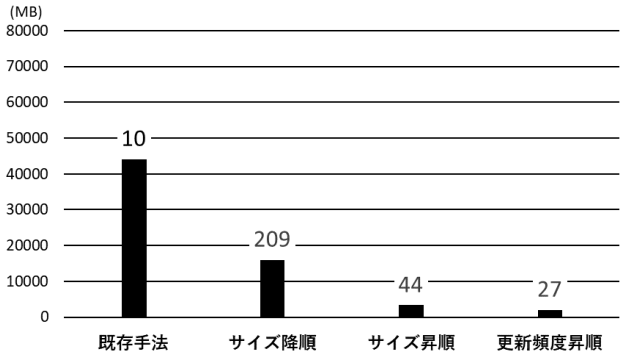

図 6 偏らせた場合のプリコピーデータ量

図 7 はファイルの更新が 1:10:10:79 と偏って発生する場合に、プリコピーフェーズを 300 周期行った場合における、最終コピーデータ量の推移を示したグラフである。横軸はプリコピーフェーズの周期、縦軸は最終コピーデータ量を表している。最も最終コピーデータ量を削減できた場合、既存手法では 5302MB、サイズ昇順では 4954MB、サイズ降順では 4748MB、更新頻度昇順では 1717MB という結果になった。

5.3.2 ファイルサイズによる転送に対する考察

ファイルサイズによる転送の場合、約 40 周期プリコピーフェーズを行うまでは、サイズ降順で転送を行った方が最終コピーデータ量が小さくなっているが、約 40 周期目から約 200 周期目までは、サイズ昇順で転送を行った方が最終コピーデータ量が少なくなっている。その後は、再びサイズ降順で転送を行った方が最終コピーデータ量が少なくなっている。

サイズ降順による転送では、約 40 周期ほどで、最終コピーデータ量がほとんど変化しなくなっている。再送ファイル数が増加した場合、1 ファイルあたりのサイズが大きいサイズ降順の転送では、同じファイルを何度も再送してしまう場合があり、再送量が未転送ファイルのデータ量を上回ってしまい、早い段階で転送が停滞してしまうことが考えられる。サイズ昇順における転送では、序盤にサイズの小さいファイルを転送済みであるので、約 50 周期で転送が停滞するが、その時点における最終コピーデータ量は

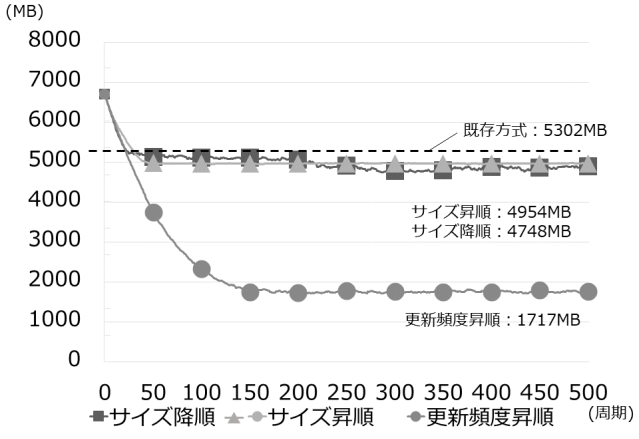

図 7 転送順序における最終コピーデータ量の推移

Vol.2018-DPS-175 No.16
Vol.2018-MBL-87 No.16
Vol.2018-ITS-73 No.16
2018/5/24

サイズ降順で転送を行った場合よりも小さくなっている。

5.3.3 更新頻度による転送に対する考察

更新頻度昇順による転送では、最も最終コピーデータ量を削減することができ、全体のデータ量の約 26%まで削減することができた。すなわち、約 74%プリコピーフェーズにおいて転送ができているということである。また、最終コピーデータ量が最も早く削減される。具体的には約 50 周期あたりで、削減が飽和している。用いたデータセットのファイル更新頻度に偏りがあるため、更新された回数が小さいファイルを優先的に転送することで無駄な再送を抑えつつ、プリコピーが行えたからであると考えられる。

6. おわりに

最終コピーデータ量を抑制し、AP の停止時間を短縮する転送順序を評価した。これにより、更新頻度を用いた転送順序が、最終コピーデータ量を大きく削減することを示した。また、プリコピーフェーズを十分に行なった場合、更新頻度を基に転送を行うと、最終コピーデータ量を最大、全体の約 26%まで削減できることを示した。

残された課題として、再送資源に対する制限を設けての転送がある。また転送順序の制御に加えて、実際の環境に合わせたシミュレーション評価、および、実アプリケーションによる転送評価の比較がある。

謝辞

本研究の一部は、科研費 (17K00107) の支援を受けて実施しています。

参考文献

[1] 畑翔太, 谷村直哉, 横山和俊, “ファイルの共有関係に着目した移送するプログラムと実行環境の特定方法”, 情報処理学会研究報告, 2015-DPS-164(11),pp.1-6 (2015).

[2] 大西史洋, 黒木勇作, 横山和俊, 谷口秀夫, “プログラム実行環境移送のための資源追跡機能のユーザレベルでの表現”, 情報処理学会第 80 回全国大会, 第 3 分冊,pp.319-320 (2018).

[3] 中井新太郎, 川島龍太, 斎藤彰一, 松尾啓志, “ファイルの共有関係に着目した移送するプログラムと実行環境の特定方法更新履歴に基づいたメモリページ転送順序スケジューリングによる仮想マシンライブマイグレーションの高速化”, 情報処理学会論文誌,Vol.56,No.2,pp.516-524 (2015).