ファイルの更新期待値を用いたソフトウェア実行環境のマイグレーション方式

黒木 勇作[1]　畑 翔太[1]　横山 和俊[1]

概要：クラウドコンピューティング環境では，負荷分散や可用性を目的として，応用プログラムとその実行環境を移送することが頻繁に行われる．応用プログラムとその実行環境を移送するための代表的な手段として，仮想マシンを移送する方法がある．しかし，この移送方法は仮想マシン上の全てのファイル群を移送する必要がある．この問題点を解決するために，ファイルのアクセス情報を用いたソフトウェア実行環境の移送方式を提案している．提案方式では，アクセス情報を監視することで移送したい応用プログラムとファイル群を特定する．次に，移送対象であるファイル群をいくつかのグループに分けて分割して転送を行い，応用プログラムの停止時間を短縮する．この分割転送の問題点として，分割転送を行ったグループ内のファイルに更新があった場合，再送が発生する．頻繁にファイルが更新される場合や巨大なファイルを再送した場合，ネットワークに負荷がかかる問題がある．提案方式では，ファイルの更新頻度に着目し，移送する順番を考慮することでファイルの再送を抑制し，応用プログラムの停止時間を短縮する．その結果，従来方式と比較したとき，最終コピー転送量および，プリコピー転送量を削減できることを確認した．

1. はじめに

クラウドコンピューティング環境は，地理的に分散した広域分散システムとして実現されることが多い．これらのシステムの多くは，複数のデータセンタで構成され，計算機の効率的な利用を強く求められる．そのため，応用プログラム (以降，AP と略す) を実行する計算機を変更し再配置することが頻繁に行われる [1]．広域分散システムにおける AP の実行環境を移送させる代表的な技術として，仮想マシンを移送する方法がある．しかし，広域分散システム環境での仮想マシンの移送は，仮想マシン上にある全てのソフトウェア資源 (ファイル群) を移送先に転送する．つまり，移送したい AP が利用するファイル群だけでなく，他の AP のソフトウェア資源も転送されてしまう．この問題点を解消するために，我々は，ファイルへのアクセス情報を用いたソフトウェア実行環境の移送方式を提案している．提案方式では，AP が利用するファイル群へのアクセス情報を監視し，移送するファイル群を特定する．次に，移送対象のファイル群を，いくつかのグループに分けて移送する．本稿では，ファイルの更新頻度に着目し，移送するファイルの順番を制御することで，ファイルの再送を抑制し，AP の停止時間を短縮する移送方式について述べる．

[1] 高知工科大学
Tosayamada, Kami City, Kochi 782–8502, Japan

2. ソフトウェア実行環境の移送方式

我々が提案しているソフトウェア実行環境の移送方式について説明する．提案方式は，以下の 2 ステップから構成される．

(特定ステップ) ある AP を移送する場合，その AP が利用するファイルへのアクセス情報を用いて，移送対象とするソフトウェア資源 (ファイル群) を特定する．具体的には，AP が利用する open システムコールを監視し，利用するファイルを特定する．また，ファイルが複数の AP から共有されるため，その共有関係を追跡し，ある AP を移送することで影響を受ける他の AP とソフトウェア資源を特定する [2].

(転送ステップ) 特定したソフトウェア資源を一度に転送するのではなく，いくつかのグループに分割して移送する．分割転送は，AP の実行を停止せずに移送する「プリコピーフェーズ」と，AP の実行を停止して移送する「最終コピーフェーズ」に分かれる．例えば，ファイル群を 10 グループに分割した場合，最終グループ以外の 9 グループの転送は，AP を停止せずに移送する「プリコピーフェーズ」である。最終グループの転送は，AP を停止し移送する「最終コピーフェーズ」である．このように，分割転送では，最終グループでの転送のみ AP が停止するため，AP の停止時間を短

©2017 Information Processing Society of Japan

縮することができる．

上記の転送ステップでは，一度移送したファイルに対して更新が発生すると，該当ファイルを再送する必要がある．このため，ネットワークに負荷がかかる問題がある．この問題に対し，メモリを対象として再送を抑制する研究が行われている [3]．しかし，固定サイズのメモリブロックを対象としているため，可変サイズのファイルへ直接適用することは難しい．本稿ではファイルの再送を抑制し，AP の停止時間を短縮するプリコピーマイグレーション方式について述べる．

3. 提案方式

3.1 プリコピーするファイルの順序

提案するプリコピーマイグレーション方式では，更新される可能性の低いファイルは，再送の可能性が低いことを利用する．つまり，更新される可能性の低いファイルは早い段階のグループでプリコピーし，更新される可能性の高いファイルについては，可能な限り後回しにして移送する．このことを実現するために，ファイルへのアクセス種別，更新頻度，ファイルサイズを用いて，プリコピーするファイルの順序を決定する．

(アクセス種別) ファイルが open される時のアクセス種別である，ReadOnly と ReadWrite を利用する．ReadOnly ファイルは更新されず再送の可能性がないため，最も早い段階でプリコピーする．

(更新頻度とファイルサイズ) ReadOnly ファイルの次は，ファイル単位で更新確率を利用する．基本的には，ファイルへの更新 (write システムコール) を一定回数監視し，ファイルの更新確率を求める．この更新確率が低いものからプリコピーする．この際に，更新確率だけでなくファイルサイズを考慮する．つまり，巨大なサイズのファイルについては，更新が発生した時，再送される転送量が巨大になる．このため，巨大なサイズのファイルについては，更新確率の低いものでも，後回しにすることも考慮する．提案方式では，以下に示す更新期待値を利用する．

更新期待値 = 更新確率 × ファイルサイズ

3.2 更新頻度に着目したプリコピー方式

提案するプリコピーマイグレーション方式の基本構成を図に示す．また，このプリコピーマイグレーションの基本動作を以下で説明する．

＜動作前＞ 特定ステップにより，移送対象のファイル群が特定される．

＜ReadOnly プリコピーフェーズ＞

(A) 特定ステップで得られた open 情報 (ReadOnly，または ReadWrite) を用いて，ReadOnly ファイルを抽出する．

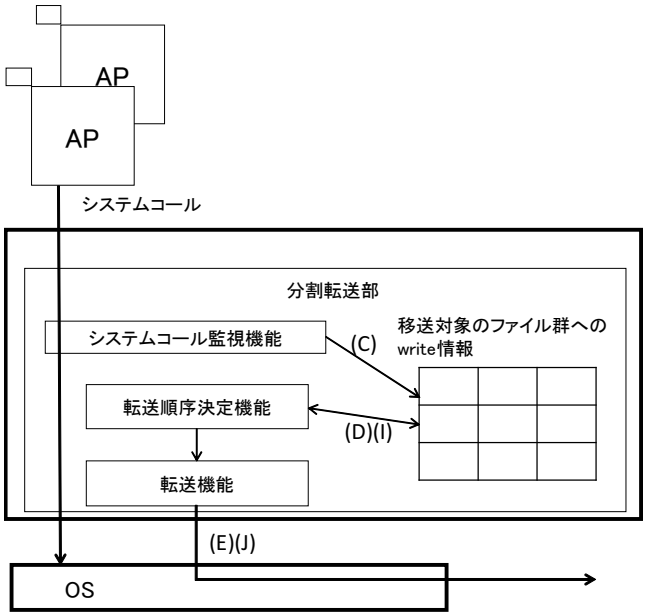

図 1 提案方式の概要

(B) ReadOnly ファイルを全てプリコピーする．

＜監視開始＞

(C) システムコール監視機能が，write システムコールを監視し，ファイル群への write 情報を記録する．

＜ReadWrite プリコピーフェーズ＞

(D) 直前の監視周期の write システムコール情報に基づき，全てのファイルの更新期待値を算出する．

(E) 更新期待値が低い順に，ReadWrite ファイルをプリコピーする．

(F) システムコール監視機能が，監視周期の終了を検出する．

(G) プリコピーフェーズ終了判定を行い，終了する場合は最終コピーフェーズへ，終了しない場合は（C）へ

＜最終コピーフェーズ＞

(H) AP を停止する．

(I) 未転送のファイル群，および，転送後に更新されたファイルを抽出する．

(J) 抽出したファイル群をすべて転送する．

3.3 プリコピーフェーズ終了判定

上記の (F) のプリコピーフェーズ終了判定は，最終コピーフェーズで転送されるデータ量に直接影響を与える重要な判定である．このため，更新される可能性の少ないファイルをできるだけ転送したことを表す判定条件が必要がある．提案方式では，未転送ファイル群の総量があまり大きく変化しなくなった場合，終了と判定している．具体的には，以下の終了判定を行っている．

(1) プリコピーフェーズ終了判定時の未転送ファイル群の総量の履歴を β 回分保持する．(古いものから 1 番とし，現在のものを β とする)

©2017 Information Processing Society of Japan

(2) (β/2+1 から β の平均) ÷ (1 から β/2 の平均) を求め，その値が 99 %から 101 %の場合，プリコピーフェーズを終了する．

4. 評価

4.1 評価項目

提案するプリコピーマイグレーション方式の有効性を，シュミレーションにより評価した．既存方式として，文献 [3] の方式をファイルに適用したものを用いる．なお，既存方式では，反復回数が一定回数となっているため，ここでは 10 回に固定して比較を行う．

評価に使用した環境を表 1 に示す．

表 1 実装環境

ホスト OS	Windows10
ゲスト OS	CentOS 7.2.1511(Core)
CPU	Intel Core i5-6500 CPU @ 3.20GHz
メモリ	2GB
アプリケーション	VMware Workstation 12 Player

評価に用いた可変長のファイルのデータセットは，以下の通りである．

(1) 固定長データセット
200KB のファイルを 11,000 個用意し，1,000 個のファイルには ReadOnly，10,000 個のファイルには Read-Write を行う．ReadWrite を行われたファイル 10,000 個を 5,000 個ずつに分割し，5000 個のファイルには 20 %の確率でアクセス，別の 5000 個には 80 %の確率でアクセスされるとしている．

(2) 可変長データセット
可変長のファイルを用いるデータセットは，100KB，200KB，300KB，…，1,000KB までの 10 種類を用意し，ReadOnly のファイルを 100 個× 10 種類，ReadWrite のファイルを 1000 個× 10 種類用意した．ReadWrite として用意したファイルのうち，500 個× 10 種類で 20 %の確率でアクセス，別の 500 個× 10 種類には，80 %のアクセスされるとしている．

また，ネットワークの帯域を変化させ，評価を行った．つまり，帯域が狭く低速なネットワークと，帯域が広く高速なネットワークでの有効性を確認する．

4.2 評価結果

4.2.1 固定長データセット

固定長データセットによるプリコピーデータ量の評価結果を図 2 に，最終コピーデータ量の評価結果を図 3 に示す．図 2 に示すように，5Mbps から 30Mbps までの帯域において，最大 94 %，最小 43 %のプリコピー転送量を削減を達成している．帯域が狭い低速なネットワークでは，ファイルを転送する時間が長くなる．このため，転送が長時間化するため，プリコピーされたファイルへの更新が多くなり再転送が増加する．既存方式では，転送順序が固定化されているため，直近の更新頻度が高いファイルを再転送してしまい，プリコピーフェーズが繰り返される．一方，提案方式では，周期的にファイルの更新期待値を算出している，つまり，直近のファイル更新の影響を反映し，更新される可能性の少ないファイルをプリコピーする．つまり，再転送が抑止できプリコピーフェーズは早い段階で終了するため，プリコピーデータ量が削減できる．一方，帯域が 40Mbps 以上の評価結果では，既存方式と提案方式を比較すると，ほぼ同等である．この場合は，帯域が広いため，1 周期で転送できるファイルが多くなる．このため，提案方式で，本来プリコピーすべきではない更新期待値の高いファイルまでプリコピーされてしまっているため，再送が頻繁化する．結果として，プリコピーフェーズが長期化し，プリコピーデータ量が増加する．

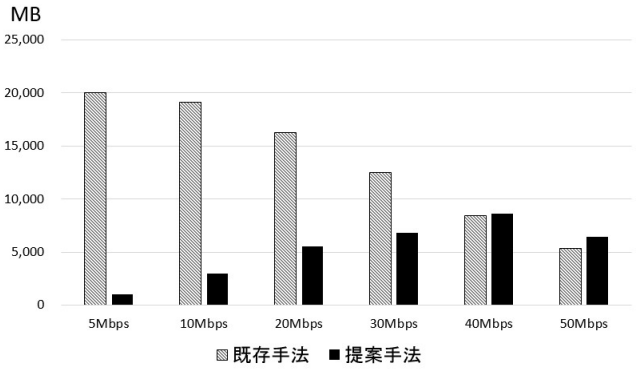

図 2 固定長データセットによるプリコピーデータ量

図 3 に示すように，帯域が広くなると，最終コピーデータ量は双方とも減少する．これは，より広く高速になるため，数多くのファイルをプリコピーできるからである．また，既存方式と提案方式を比較すると，提案方式の方が最大で 65 %，最小で 6 %の削減を実現している．提案方式の最終コピーデータ量が少ない理由は，更新期待値により，プリコピーされたファイルへの更新を抑制できているからである．

4.3 可変長データセット

可変長データセットによるプリコピーデータ量の評価結果を図に，最終コピーデータ量の評価結果を図に示す．図に示すように，プリコピーデータ量については，最大で 93 %，最小で 71 %の削減を達成しており，固定長のデータセットの時より，削減効果が大きい．帯域が狭いネットワークでは固定長データセットと同様であるが，可変長データセットの場合は帯域が 40Mbps 以上でも，提案方式の方がプリコピーデータ量が少ない．この場合は，帯域が広いため，1 周期で転送できるファイルが多くなる．既存

©2017 Information Processing Society of Japan

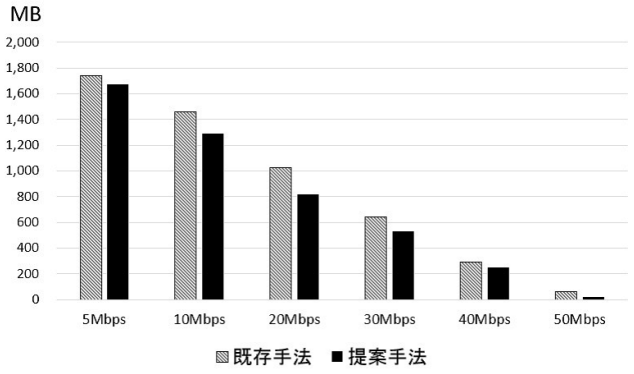

図 3 固定長データセットによる最終コピーデータ量

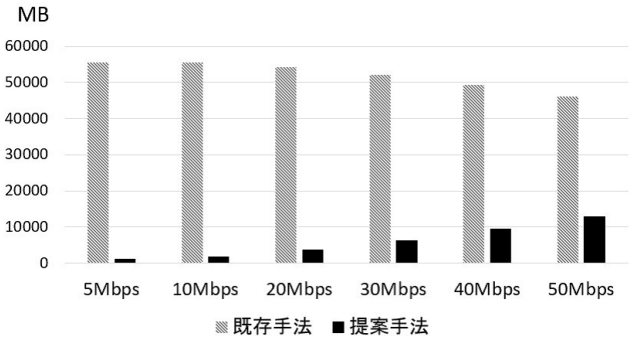

図 4 可変長データセットによるプリコピーデータ量

方式では，転送順序が固定であり，かつ，ファイルサイズも考慮せずにプリコピーする．このため，サイズが大きく更新頻度の高いファイルをプリコピーしてしまい，再転送量が増大している．一方，提案方式は，ファイルサイズを意識しているため，巨大なファイルの再転送を抑止できている．

図に示すように，帯域が広くなると，最終コピーデータ量は，双方とも減少する．これは，固定長データセットの時と同様に，ネットワークが高速になるため，数多くのファイルをプリコピーできるからである．また，既存方式と提案方式を比較すると，提案方式の方が 3 %から 10 %の削減を実現している．提案方式の最終コピーデータ量が少ない理由は，更新期待値により，プリコピーされたファイルへの更新を抑制できているからである．

5. おわりに

本研究では，ファイルのサイズとアクセス頻度に着目し，転送順序を制御することによって，再送を抑制し，AP の停止時間を削減する提案を行った．その結果，狭い帯域においてプリコピー量が大幅に削減し，最終コピー量も削減できることを確認した．今後の課題として，様々なデータセットを用いた評価や，実アプリケーションのファイルアクセスでの評価がある．

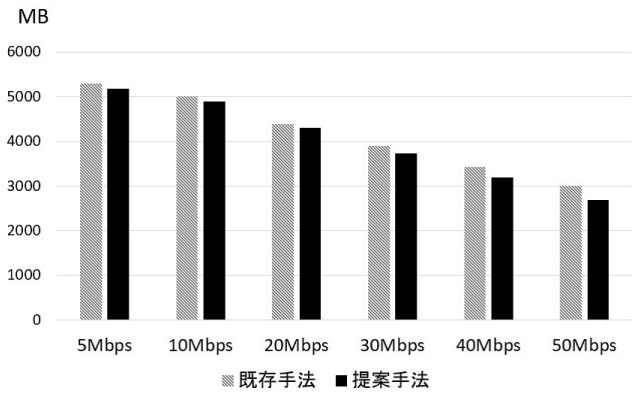

図 5 可変長データセットによる最終コピーデータ量

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] 広渕 崇宏，マウリシオ ツガワ，中田 秀基，伊藤 智，関口 智嗣，“仮想マシンの超広域ライブマイグレーションにむけたベストエフォート型状態同期機構の試作，”情報処理学会研究報告，2012-OS-121(11)，(2012)．

[2] 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)，(2015)．

[3] 中井 新太郎，川島 龍太，斎藤 彰一，松尾 啓志，“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)．

©2017 Information Processing Society of Japan