2008－OS－109 (16)
2008／8／7

レイヤ7プロトコルの状態を考慮した仮想マシンの移送

石川 豊　山田 浩史　浅原 理人　花岡 美幸　河野 健二

慶應義塾大学 理工学部 情報工学科
E-mail: {urube, yamada, asahara, hanayuki}@sslab.ics.keio.ac.jp, kono@ics.keio.ac.jp

仮想化された環境において，仮想マシンはメモリイメージと CPU の内容をコピーすることで他の物理マシンに移送することができる．仮想マシンの移送は，物理マシンのメンテナンスの時にサーバ機能を他の物理マシンに退避させる際などに有効である．しかし，移送中は仮想マシンを停止させる必要があるため，移送にかかる時間だけサービスの中断時間が生じる．本研究では仮想マシンとクライアントの間でやりとりされるレイヤ 7 プロトコルメッセージに注目し，クライアントに感知され難いタイミングで仮想マシンを移送することを提案する．サービスの中断をクライアントに感知され難い状態として，コネクションは確立しているがメッセージのやりとりをしていない状態や，長時間のデータ転送をしているためにサービスの中断時間を相対的に無視できる状態が挙げられる．これらの状態を，サーバ・クライアント間でやりとりされるレイヤ 7 プロトコルメッセージを解析して見つけだし，仮想マシンの移送を行う．提案手法を Xen 3.1.0 上に実装し，Web サーバを動作させた仮想マシンを提案手法と従来手法で移送し，Web サーバの応答時間を測定して比較する実験を行った．実験の結果，既存の手法ではクライアントにサービスの中断を感知されることがあったが，提案手法では全ての応答がクライアントに許容される時間内に行われ，サービスの中断を感知されることはなかった．

Live Migration of Virtual Machines by Exploiting Layer-7 Protocol Context

Yutaka Ishiakwa　Hiroshi Yamada　Masato Asahara　Miyuki Hanaoka　Kenji Kono

Department of Information and Computer Science, Keio University
E-mail: {urube, yamada, asahara, hanayuki}@sslab.ics.keio.ac.jp, kono@ics.keio.ac.jp

Virtual machine migration transfers a virtual machine (VM) on a physical host to another one by transfering its memory and CPU state. VM migration is useful, for example, for keeping a server running during the maintenance of the physical host. But VM migration cannot be operated without service downtime because VM must be stopped during the migration. In this paper, we propose VM migration without clients being aware of service downtime by exploiting layer-7 protocol context. There are the two states in which clients do not become aware of service downtime. One is the state in which a connection is established but a server and clients do not exchange messages. The other is the state in which data is being transferred for a long time and service downtime can be relatively negligible. To determine if a connection state is suitable for migration, we exploit layer-7 protocol of exchanged messages. We implemented our prototype on Xen 3.1.0, and conducted experiments that migrate VM on which a web server is running with the proposed and existing techniques. The experimental results show that clients notice service downtime with the existing technique. In contrast, with our technique, all responses are replied within the tolerable time and clients do not notice service downtime.

1 はじめに

近年，ハードウェアの仮想化を行う仮想化技術 [1] に注目が集まっている．仮想化技術は一つの物理マシンの上に複数の仮想マシンをつくり，それぞれの仮想マシン上で別々のオペレーティングシステム (OS) を動作させる技術である．物理マシンの高性能化が進み，パフォーマンスを落とさずに仮想マシン環境を実現できるようになったため，この技術に関心が集まるようになった．

仮想化技術を応用したもののひとつとして，仮想マシンの移送がある．仮想マシンの移送とは，メモリイメージと CPU の内容をコピーすることで仮想マシンを別の物理マシン上に移動させる操作のことである．仮想マシンの移送を用いることで，物理マシン上の資源が減少した際に資源を占有している仮想マシンを資源に余裕のある他の物理マシン上に移送することができ，資源が枯渇することを防ぐことができる．また，物理マシンのメンテナンスを行う前に物理マシン上の仮想マシンを他の物理マシンへ移送することで，仮想マシンの提供するサービスを中断させること無くメンテナンスを行うことができる．

しかし，仮想マシンの移送では，CPU レジスタやメモリイメージの一貫性を保つために，移送する

際に仮想マシンを一度停止しなければならない．そのため，仮想マシンの移送は仮想マシンが提供するサービスの中断を伴う．移送による中断時間を減らす研究は数多く行われている．研究の例として，仮想マシンの移送を行う前に仮想マシンのメモリイメージの大半を転送しておき移送の時間を短縮させるというものがある [2]．この方法は非常に有効で，現在多くの仮想マシンのシステムに実装されている [3][4][5]．しかし，それらの方法を用いても中断時間を完全になくすことはできない．通信路が低速なワイドエリアネットワーク (WAN) 上で移送する場合を考えると，移送により発生するサービスの中断時間は無視できない．

そこで，本研究では仮想マシンで稼働しているサーバとクライアントの通信を考慮して，サービスの中断による影響が少ないタイミングで移送を行う手法を提案する．レイヤ 7 プロトコルメッセージを解釈して各コネクションの状態を入手することで，クライアントに感知され難いタイミングで移送を行う．サービスの中断をクライアントに感知され難いコネクションの状態には，以下の二種類の状態が考えられる．まずは，サーバがクライアントにサービスを提供していない状態である．サーバとクライアントの間でコネクションが確立していてもサービスに関するデータのやり取りをしていない場合は，コネクションが中断されてもクライアントは影響を受けない．そのため，移送を行ってサービスの中断時間が発生してもクライアントはそのことに気付き難い．もうひとつは，十分に大きいデータがコネクション上で転送されている状態である．転送データが大きければ転送完了までの総時間は長くなる．この時間が移送による中断時間よりも十分長ければ，相対的に移送による中断時間は無視できるほど小さくなり，クライアントはサービスの中断が発生したことを感知することが難しくなる．

提案手法は，サービスの中断をクライアントに感知され難い状態であることをレイヤ 7 プロトコルメッセージから判断する．まず，対象のレイヤ 7 プロトコルの状態の種類と状態遷移の条件を定義する．VM 上のサーバとそのクライアントの間で対象のレイヤ 7 プロトコルのメッセージのやりとりが行われたら，そのメッセージを取得し，定義にそってメッセージを解釈して状態遷移を行い，コネクションがクライアントにサービスを提供している最中かを判断する．また，メッセージの転送データサイズの記述部分から転送しているデータがどれだけの大きさを持っているのかを入手して，転送し終えるまでの時間が，移送による中断時間を無視できるほど大きいかを判断する．移送を行う時はこれを参照して，全てのコネクションが移送可能な状態であれば移送を実行し，そうでなければ移送を延期する．

提案手法を Xen [6] 上に実装し，その有効性を示すために実験を行った．実験では，Web サーバを動作させた仮想マシンに対してベンチマーク SPECweb2005 [7] を動作させ，仮想マシンの移送を 1 回行った．その結果，既存の手法では SPECweb に対する応答に失敗することがあったが，提案手法は常に SPECweb が許容できるとする時間内に応答を行い，サービスの中断を感知させなかった．

以下，2 章では仮想マシンの移送の説明および，仮想マシンの移送を行う利点を挙げる．3 章では提案手法の概要の説明を行い，レイヤ 7 プロトコルの利用方法について説明する．4 章では提案の実装について述べる．5 章では提案手法によってユーザーに気付かれ難いタイミングで移送をおこなえることを実験により示す．6 章では関連研究を挙げ，7 章では本論文をまとめる．

2 仮想マシンの移送

仮想マシンの移送とは，ある物理マシン上の仮想マシンを別の物理マシンに移送することである．仮想マシンの移送によって転送される内容は移送環境によって異なる．送り元の物理マシンと受け取り先の物理マシンが同じローカルエリアネットワーク (LAN) 上に存在する場合は，ディスクイメージはネットワーク接続ストレージ (Network Attached Storage) などで共有することができるので転送されず，CPU のレジスタ値とメモリイメージなどが転送される．一方，LAN 上に存在しない離れた物理マシンに対して WAN を通して移送を行うときは，ディスクイメージも転送する必要がある．ディスクイメージの転送を行うことに加え，WAN は LAN よりも一般に通信速度が遅いため，WAN 上の移送は LAN よりも長い時間がかかることが多い．

移送の具体例として Xen [6] における仮想マシンの移送の紹介をする．Xen の場合，LAN 内での仮想マシンの移送のみ実装されており，そのため転送されるのは CPU のレジスタ値とメモリイメージで，ディスクイメージは転送されない．Xen は移送方法に通常の移送とメモリ差分を考慮した移送 (Live

Migration) [2] のどちらかを選択できるようになっている．メモリ差分を考慮した移送とは，仮想マシンを動かした状態であらかじめメモリイメージだけを先に送っておき，ある程度のメモリイメージを送り終えたら仮想マシンを停止し，残りのメモリイメージと CPU レジスタ値の転送を行うというものである．これにより通常の移送より停止時間を短くすることができる．

仮想マシンの移送が有用となる状況はいくつかある．まず，物理マシン上の資源が底をついたときである．物理マシン上の資源は有限であるので，多くの仮想マシンが高負荷状態になると資源が不足することがある．仮想マシンの移送を使えば，資源不足を起こしている仮想マシンを資源に余裕のある他の物理マシンに移すことができる．すると，移送された仮想マシンは他の物理マシン上で豊富な資源を使うことができ，移送元の物理マシンには資源の余裕ができる．このように，仮想マシンの移送は複数の物理マシンの間で資源をマネジメントする際に有用である．

移送が有用な状況のもうひとつの例として，物理マシンをメンテナンスするときが挙げられる．メンテナンス中もサーバが提供するサービスを止めないようにするためには，代わりにサービスを行うサーバを立てなければならない．仮想マシンの移送を使えば，他の物理マシンに仮想マシン環境を整えるだけで速やかにサーバの機能を移すことができる．

以上のように仮想マシンの移送には大きなメリットがあるが，移送する際には必ず一度仮想マシンを停止しなければならないという問題がある．これは，仮想マシンを動作させたまま移送を行うと，転送済みの内容に更新が行われるなどしてデータの一貫性が保てなくなってしまうためである．メモリ差分を考慮した移送を用いても仮想マシンの停止を必要とし，サービスの中断時間をなくすことはできない．頻繁に仮想マシンの移送を行う場合や WAN 上で仮想マシンの移送を行う場合には，この停止時間が無視できないものになる．

3 提案

3.1 概要

本研究は，サービスを中断してもクライアントへの影響が少ないタイミングで仮想マシンを移送することを提案する．サーバとクライアント間でやり取りされているメッセージの内容を解析し，サービス

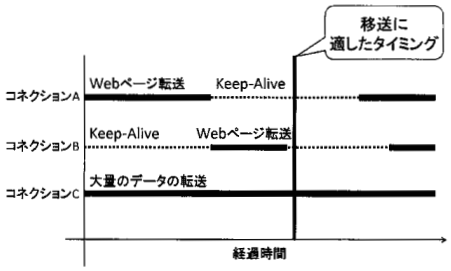

図 1: HTTP コネクションの状態遷移例

の中断時間をクライアントに感知され難いタイミングを見つける．このタイミングで仮想マシンの移送を行うことで，仮想マシンの停止による影響がクライアントの受けるサービスに及ばないようにする．

移送に適した状態として 2 種類の状態が考えられる．ひとつめはコネクションの状態が待機状態であるときである．待機状態とは TCP コネクションは確立しているが実際にメッセージのやりとりを行っていない状態のことである．待機状態中にサービスが中断されても，クライアントはサービスの提供を受けていないため，それに気付くことは難しい．待機状態の例として HTTP の Keep-Alive 状態が挙げられる．Keep-Alive 状態とは，HTTP においてリクエストに対応するレスポンスの処理が終わった後，次のリクエストを受信するまでの状態である．

移送に適したもうひとつの状態は，転送中のデータのサイズが十分大きい場合である．転送中のデータのサイズが大きいと，移送による中断時間がデータの総転送時間と比べて相対的に小さくなるため，クライアントはサービスの中断による影響を感知し難い．HTTP の GET 要求の応答で大きなファイルを転送している場合などがこれにあたる．

図 1 は HTTP サーバのコネクションの状態の遷移を表したものである．対象の仮想マシンとクライアントとの間に 3 つの HTTP コネクションが確立しているものとする．コネクション A とコネクション B は比較的小さなサイズのデータを HTTP でやり取りしており，データの転送を行っている状態と，TCP 接続は確立しているがデータの転送を行っていない Keep-Alive 状態の間の遷移を繰り返している．コネクション C は極めて大きいサイズのファイルの

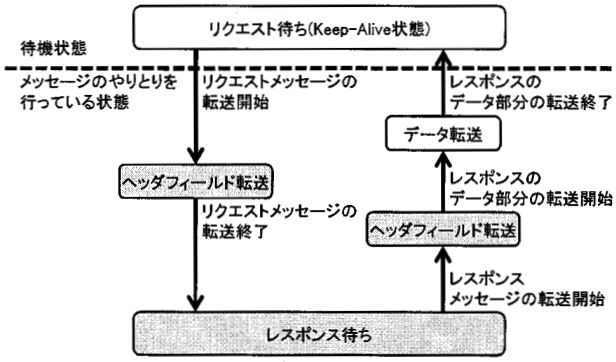

図 2: HTTP の状態遷移（簡単のため主要な状態のみを示す）

ダウンロードを行っており，この後も長い時間ダウンロードを続ける見込みである．図 1 の縦線のタイミングでは，コネクション A とコネクション B は Keep-Alive 状態であり，コネクション C は長時間のデータ転送を行っている状態である．提案手法は，この線のタイミングのようにすべてのコネクションが移送に適した状態になっているタイミングで仮想マシンの移送を行う．移送の際に仮想マシンは一度停止するが，図 1 のタイミングならば仮想マシンが停止してもクライアントに提供されるサービスに大きな影響はないと考えられる．

3.2 レイヤ 7 プロトコル

提案手法では，コネクションが仮想マシンの移送に適した状態であるかどうかを，サーバ・クライアント間でやりとりされるメッセージをレイヤ 7 プロトコルに基づいて解析することで判断する．

レイヤ 7 プロトコルは複数の状態を持ち，レイヤ 7 プロトコルメッセージをやりとりすることで状態遷移していくものと考えることができる．そのため，コネクション上でやり取りされるレイヤ 7 プロトコルメッセージを解釈して状態を追えば，コネクションの現在の状態を入手することができる．提案手法は，この特徴を利用して仮想マシンの移送タイミングを判断する．まず，レイヤ 7 プロトコルの状態のうち，待機状態に分類される状態を仮想マシンの移送に適した状態として定義しておく．通信が行われた際には，プロトコルの定義に従ってメッセージを解釈して状態遷移を行い，コネクションが移送に適した状態かを判断する．

また，レイヤ 7 プロトコルからは，転送対象のデータ以外に複数の情報を入手することができる．たとえば，一部のレイヤ 7 プロトコルはデータを転送する前にデータのサイズを相手に通知する．提案手法は，この特徴を利用してレイヤ 7 プロトコルメッセージを解析し，どのようなサイズのデータの転送を行っているかを知り，コネクションが移送に適した状態か否かを判断する．

HTTP を例にして，レイヤ 7 プロトコルから移送に適した状態をどのように検出するかを説明する．HTTP の状態遷移は図 2 のように定義できる．HTTP はコネクションが確立すると，まずリクエスト待ち状態になり，リクエストメッセージを受け取るとリクエストのヘッダフィールド転送状態に遷移する．ヘッダフィールドとは HTTP メッセージの先頭部分に書かれている内容のことで，通信方法などの情報を持つ部分である．今回は簡単のためリクエストメッセージはヘッダフィールドしか持たないものとする．リクエストメッセージの受け取りを終えると，コネクションの状態はレスポンス待ちに遷移し，サーバからのレスポンスメッセージを待つ．レスポンスメッセージがサーバから返されると，レスポンスのヘッダフィールド転送状態に遷移する．レスポンス側ではヘッダの転送が終わると，それに続くデータの転送を行う状態に遷移する．データの転送が終了すると，再びリクエスト待ちの状態に遷移する．

リクエスト待ちの状態は，メッセージのやりとりを行っていない待機状態であるので，仮想マシンの移送に適した状態とする．レスポンス待ちの状態で

は，仮想マシンの移送を行うと移送による中断時間分サーバのレスポンスが遅れるので，クライアントにサービスの中断を感知されやすい．よって，レスポンス待ちの状態は移送に適さない状態とする．ヘッダフィールド転送状態やデータ転送状態はメッセージの転送の途中なので，仮想マシンの移送を行うとメッセージの転送が中断されることになり，クライアントにサービスの中断を感知されやすい．よって，ヘッダフィールド転送状態とデータ転送状態も仮想マシンの移送に適さない状態とする．ただし，レスポンスのヘッダフィールドの Content-Length の値が十分に大きければ，転送されるデータサイズが大きく，データ転送が長時間に渡るものと考えられる．この場合，移送によるサービスの中断時間は相対的に小さくなるので，データ転送状態も移送に適した状態とする．

図 2 の状態遷移の定義を用いると，たとえばレスポンス待ちの状態のときにサーバがレスポンスメッセージを送信した場合，コネクションの状態はヘッダフィールドの転送状態，データの転送状態と遷移して，データの転送が完了したときにリクエスト待ちの状態になり，移送に適した状態になる．この様にして，状態遷移から仮想マシンの移送タイミングを判断できる．

4 実装

提案手法を Xen 3.1.0 の Paravirtualization 環境に実装した．OS は Xen 用に修正を加えられた Linux を使用した．

4.1 全体像

本研究で提案する手法を実装したシステムは役割ごとに大きく通信パケット取得部，レイヤ 7 プロトコルメッセージ解釈部，仮想マシン移送実行部にわけることができる．各実装部分の関係性を図 3 に示す．通信パケット取得部は Domain-0 (Dom-0) 上を通過する通信パケットを取得して，レイヤ 7 プロトコルメッセージ解釈部に渡す．レイヤ 7 プロトコルメッセージ解釈部は，まず，通信パケットをコネクションごとに分類して，レイヤ 7 プロトコルメッセージを復元する．次に，事前に定義しておいたプロトコル解析ルールで復元したメッセージを解析して，コネクションのレイヤ 7 プロトコルの状態と転送データの総量を入手する．そして，これらから各コネクションが移送可能な状態かを判断して，移送不可能なコネクションの総数を求める．仮想マシン

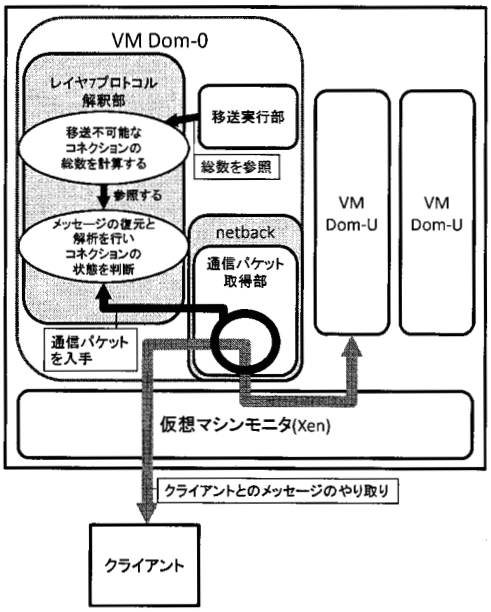

図 3: 提案手法の実装の概略図

移送実行部は移送不可能なコネクションの総数を参照して，仮想マシンの移送を行うか延期するかを決定する．

4.2 通信パケット取得部

Xen 上の全ての仮想マシンの通信パケットは特権仮想マシンである Dom-0 を通過する (図 3)．通信パケット取得部は通過するパケットの情報を取得する役割を持ち，Dom-0 の netback というインタフェースに実装される．この netback はゲスト仮想マシン Domain-U (Dom-U) と通信パケットのやりとりを行うためのインタフェースで，Dom-0 の Linux カーネルの内部に存在する．通信パケット取得部は取得したパケットをレイヤ 7 プロトコルメッセージ解釈部に渡す．

4.3 レイヤ 7 プロトコルメッセージ解釈部

このレイヤ 7 プロトコルメッセージ解釈部は TCP ストリームフィルタ [8] と TCP リアセンブラ [9] を使用して構成されている．TCP リアセンブラは，取得した通信パケットをコネクションごとに分類してレイヤ 7 プロトコルメッセージに復元し，TCP ストリームフィルタエンジン (TSF エンジン) に渡す．TSF エンジンは事前に定義されたレイヤ 7 プロトコ

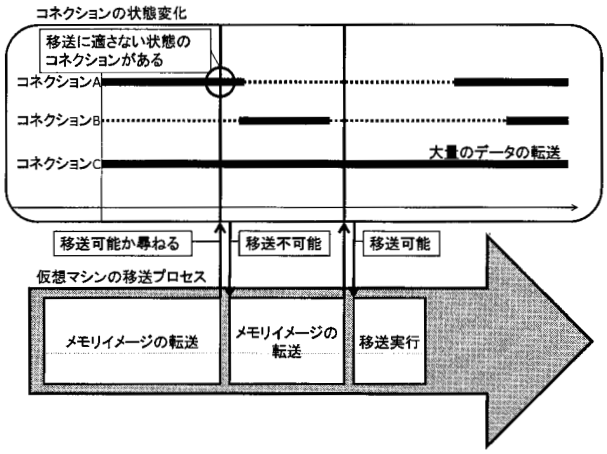

図 4: 提案手法の仮想マシンの移送プロセス

ルのルール (TCP ストリームフィルタルール (TSF ルール)) から生成される構文解析エンジンであり，レイヤ 7 プロトコルメッセージの解析を行う．

TSF エンジンはコネクションごとに存在し，TCP リアセンブラから渡されたメッセージを 1byte ずつマッチングしてコネクションの状態遷移を行い，コネクションが待機状態か判断する．また，TSF エンジンはマッチング時にレイヤ 7 プロトコルメッセージ内の転送データの大きさの情報を取得し，データの転送に長い時間がかかるか判断する．コネクションが待機状態もしくは長時間にわたるデータの転送中であった場合，TSF エンジンはコネクションが仮想マシンの移送に適した状態であると判断する．コネクションごとの TSF エンジンの判断結果はコネクション監視エントリによって参照され，仮想マシンごとに集計される．

TSF エンジンの元となる TSF ルールはプロトコルごとに定義する必要がある．今回は HTTP のみ実装した．これは Web サーバがネットワークで使われるサーバの中でも一般的であるという点と，HTTP の状態遷移が比較的単純であるという点から提案手法を使用するのに適当だと判断したためである．状態を追う記述と転送データのサイズを取得する記述を書けば他のレイヤ 7 プロトコルの TSF ルールも記述することができる．

4.4 仮想マシンの移送実行部

提案手法は，メモリ差分を考慮した移送 (Live Migration) [2] を使用する．メモリ差分を考慮した移送は繰り返しメモリイメージの転送を行い，メモリ転送の終了後に仮想マシンの移送を実行する．メモリ転送を終了する条件は，メモリの転送回数があらかじめ指定されている最大回数を越えた場合や，転送レートの上限に達し，かつ，今回のメモリ転送量が前回の転送量よりも大きい場合などである．

本実装では，メモリ転送の終了条件を満たしたときに，さらにレイヤ 7 プロトコルメッセージ解釈部のコネクション監視エントリの持つ集計結果を参照するようにした．全てのコネクションが移送に適した状態であると判断されていれば，クライアントに感知され難い移送が可能であると判断して仮想マシンを停止させ，残りのメモリイメージと CPU レジスタ内容の転送を開始する．そうでなければ再びメモリイメージの転送を行い，すべてのコネクションが移送に適した状態になるのを待つ．

図 4 はこの仮想マシンの移送のプロセスを表したものである．図 4 の上側の 3 本の横線はレイヤ 7 プロトコルのコネクションの状態遷移を表す．太線の部分はメッセージのやりとりをしている状態を表し，点線の部分は待機状態であることを表す．図 4 の下側の矢印は仮想マシンの移送プロセスを表す．仮想マシンの移送プロセスは，まず従来のメモリ差分を考慮する移送と同じようにメモリイメージの転送を行い，メモリイメージの転送が終了条件を満たしたところでコネクションが移送に適した状態であるかを尋ねる．コネクションのいずれかが移送に適さない状態であれば，メモリイメージの転送を再び行う．すべてのコネクションが移送に適した状態であれば，仮想マシン移送を実行する．

表 1: 実験結果 (既存の手法)

	Good (3 秒以内)	Tolerable (5 秒以内)	Fail (5 秒以上)
1	89.1%	99.2%	0.8%
2	95.7%	99.6%	0.4%
3	95.1%	100.0%	0.0%

表 2: 実験結果 (提案手法)

	Good (3 秒以内)	Tolerable (5 秒以内)	Fail (5 秒以上)
1	87.2%	100.0%	0.0%
2	95.7%	100.0%	0.0%
3	95.1%	100.0%	0.0%

5 実験

5.1 概要

提案手法が実際にクライアントに感知されないタイミングで仮想マシンの移送が行えることを確認するため実験を行った．本実験では Web サーバの標準的なベンチマークである SPECweb2005 [7] を使用し，動作中に仮想マシンの移送を 1 回行った．また，比較のために既存手法でも同様の実験を行った．

5.2 実験環境

実験には Pentium4 2.80GHz CPU，メモリ 512MB，32GB のハードディスクで構成された物理マシン 4 台を使用した．これらの計算機から，仮想マシンを動作させるマシンに 2 台，SPECweb のバックエンドシミュレータに 1 台，SPECweb のクライアントに 1 台を割り当てた．すべての物理マシンは単一のスイッチにギガビットイーサで接続されている．仮想マシンを動作させる 2 台のマシンには Xen 3.1.0 をインストールした．Dom-0 の OS には Fedora7 (Linux-2.6.18-xen) を使用し，割り当てメモリは 192MB とした．この Dom-0 のカーネルに提案手法が実装されている．移送の対象となる Dom-U の OS には Gentoo 2.86 (Linux-2.6.18-xen) を使用し，割り当てメモリは 256MB とした．SPECweb の評価の対象となる Dom-U の Web サーバには Apache 2.2.6 と php 5.2.5 を使用した．Dom-U のディスクイメージは 1 台の物理マシン上に用意し，もう 1 台の物理マシンからはこれを Network File System を使用してマウントすることで同一のディスクイメージを参照するようにした．バックエンドシミュレータとクライアントの OS には Fedora7 (Linux-2.6.23.8-34.fc7) を使用した．バックエンドシミュレータの Web サーバには Apache 2.2.6 と FastCGI 2.4.0 を使用した．SPECweb2005 は 3 種類のワークロードを提供しているが，本実験ではそのひとつである Ecommerce ワークロードを使用した．仮想マシンの移送がサービスに与える影響を明確にとらえるために，SPECweb は同時接続数を 100 とし，動作時間を 30 秒と短い時間に設定した．

5.3 実験結果

提案手法と既存手法，それぞれの場合で 3 回評価を行った．実験結果を表 1 と表 2 に示す．本実験で用いた Ecommerce ワークロードでは応答時間が 3 秒以内だった場合は十分に早く応答が返ってきているものとみなされ，5 秒以内ならば許容できる応答速度であるとみなされる．5 秒よりも長い時間がかかった場合は応答に失敗したものと判断される．

実験結果を見ると，既存手法では 1 回目と 2 回目のテストで応答に失敗する通信が存在する．これに対して，提案手法を使用した場合は失敗とみなされる通信はなく，常に許容できる時間内に応答が返される．このことから，提案手法は既存手法に比べて仮想マシンの移送によるサービスへの影響を緩和できていることが示された．

6 関連研究

メモリ差分を考慮した移送 (Live Migration) [2] や VMotion [3] は，移送を行う前にあらかじめメモリイメージを転送しておくことで，仮想マシンの移送を行う際に転送するデータの総量を小さくし，仮想マシンの停止時間を短縮する移送方法である．初めに仮想マシンのすべてのメモリイメージを転送し，次に，前回の転送中に更新があったページを再送する．この再送を繰り返し更新ページが少なくなったら仮想マシンを停止させ，移送を開始する．これらの移送方法は中断時間の短縮に有効であるが，コネクションについては考慮していない．そのため，たとえば多くのクライアントがデータをダウンロードしているときでも，クライアントのコネクションを全て一時中断して移送を行う．本研究では，クライアントに中断を感知されるような状態のコネクションが存在するときには仮想マシンの移送を延期し，クライアントに感知され難いタイミングを待って移送を行う．

メモリ差分を考慮した仮想マシンの移送はメモリとCPUの転送をおこなうが，ディスクイメージは転送対象とされておらず，使用がLAN上の移送に限定されている．Live Wide-Area Migration [5] はこのディスクイメージの転送も行い，WAN上での仮想マシンの移送を実行するものである．この手法によるサービスの中断時間は1分程度になる．サービスが1分中断されるならばそのタイミングは重要であり，なるべく感知するクライアントの少ないタイミングで移送を行った方がよい．本研究ではWAN上の移送にも使うことのできる，コネクション状態を考慮した仮想マシンの移送を行うことが期待できる．

Sandpiper [4] は物理マシンの資源の使用状況を考慮し，仮想マシンに資源が行き渡るよう適切に仮想マシンを移送するシステムである．Sandpiperは各仮想マシンのCPU，ネットワーク，メモリの使用率を考慮し，複数の物理マシンで構成されたクラスタ上で自動的に仮想マシンの再配置を行う．SandpiperはDom-0上のデーモンから各仮想マシンの資源の使用状況を入手し，これらの情報を元に現在どの仮想マシンが高負荷状態にあるか，将来どの仮想マシンが高負荷状態に陥るかを判断して仮想マシンの移送を行う．Sandpiperは資源の使用状況の変化から仮想マシンの移送タイミングを決定しており，コネクションの状態を考慮せず移送を行うのでサービスが中断されたことをクライアントに感知されやすい．本研究では，コネクションの状態を考慮して，クライアントに感知され難いタイミングで仮想マシンを移送する．

7 まとめと今後の課題

本研究は，クライアントとの通信におけるレイヤ7プロトコルメッセージに注目し，これを考慮して移送タイミングを決定する方法を提案した．提案手法ではレイヤ7プロトコルメッセージを分析して，コネクションが現在どのような状態であるかを判断し，コネクションを中断してクライアントに感知されるかどうかを判断する．そして，クライアントにいずれのコネクションの中断も感知され難いタイミングで仮想マシンの移送を行う．

提案手法を用いてWebサーバの動作しているVMを移送する実験を行った．その結果，既存の手法ではベンチマークソフトSPECwebにサービスの提供に失敗していると認識されるのに対し，提案手法を用いた場合は常にサービスの提供に成功したものと認識された．よって，提案手法は既存の手法と比べて移送がサービスに与える影響が少ないことが示された．

今回は，提案手法の実装をHTTPで行ったが，レイヤ7プロトコルメッセージのやり取りを行うサーバはWebサーバだけではない．提案手法がレイヤ7プロトコルを扱う仮想マシンの移送に常に有効であることを示すために，今後はFTPなどの他のプロトコルの実装も行いたい．

文献 [5] ではLive Wide-Area MigrationはWAN上で仮想マシンの移送を行えると報告されているが，現在公開されているXenにはまだ実装されていないため，実験することができなかった．しかし，クライアントとの通信を考慮した移送を行う提案手法はこのような中断時間の長い仮想マシンの移送にこそ有効なものである．よって，今後はWAN上で仮想マシンの移送を行えるようにXenに変更を加え，WAN上の仮想マシンの移送実験を提案手法を用いて行いたい．

参考文献

[1] Goldberg, R. P.: Survey of Virtual Machine Research, *IEEE Computer Magazine*, Vol. 7, pp. 34–45 (1974).

[2] Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I. and Warfield, A.: Live Migration of Virtual Machines, *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI '05)*, pp. 273–286 (2005).

[3] Nelson, M., Lim, B.-H. and Hutchins, G.: Fast Transparent Migration for Virtual Machines, *Proceedings of the 2005 USENIX Annual Technical Conference (USENIX '05)*, pp. 391–394 (2005).

[4] Wood, T., Shenoy, P., Venkataramani, A. and Yousif, M.: Black-box and Gray-box Strategies for Virtual Machine Migration, *Proceedings of the 4th Symposium on Networked Systems Design and Implementation (NSDI '07)*, pp. 229–242 (2007).

[5] Bradford, R., Kotsovinos, E., Feldmann, A. and Schiöberg, H.: Live Wide-area Migration of Virtual Machines Including Local Persistent State, *Proceedings of the 3rd International Conference on Virtual Execution Environments (VEE '07)*, pp. 169–179 (2007).

[6] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A.: Xen and The Art of Virtualization, *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, pp. 164–177 (2003).

[7] Standard Performance Evaluation Corporation: The SPECweb2005 benchmark, `http://www.spec.org/web2005/`.

[8] 河野健二, 品川高廣, ラハトカビル: TCPストリームに対するフィルタリングによるインターネット・サーバの安全性向上, 情報処理学会論文誌. コンピューティングシステム, Vol. 46, No. SIG4(ACS9), pp. 33–44 (2005).

[9] Hanaoka, M., Shimamura, M. and Kono, K.: TCP Reassembler for Layer7-Aware Network Intrusion Detection/Prevention Systems, *IEICE Transactions on Information and Systems*, Vol. E90-D, No. 12, pp. 2019–2032 (2007).