

## サーバ仮想化技術を利用した アプリケーション障害再現システムの提案

樋口 毅<sup>†</sup> 國分 俊介<sup>†</sup> 近江 雅紀<sup>†</sup> 片山 吉章<sup>†</sup>  
<sup>†</sup>三菱電機株式会社 情報技術総合研究所

サーバ上で動作しているアプリケーションで障害が発生した場合、原因究明を行い、対策を実施する必要がある。原因究明を実施するためには、検証環境にて障害を再現させることが重要となるが、再現環境で障害を再現させることは困難である。本稿では、サーバ仮想化技術により取得できる仮想計算機のスナップショット情報とネットワークパケット情報を収集し、サーバアプリケーションの障害を検証環境で再現させる障害再現システムを提案する。また、ネットワークパケット情報収集によるサーバアプリケーションの応答時間への影響について評価した結果を報告する。

### A Proposal of an Application Failure Reproduction System Using the Server Virtualization Technology

Tsuyoshi HIGUCHI,<sup>†</sup> Shunsuke KOKUBU,<sup>†</sup> Masanori OHMI,<sup>†</sup> and Yoshiaki KATAYAMA<sup>†</sup>  
<sup>†</sup>Information Technology R&D Center, Mitsubishi Electric Corporation.

To investigate the cause of the failure of server applications, we need to reproduce the failure. However, it is difficult to reproduce the failure using the testing environment. This paper describes a proposal of an application failure reproduction system. This system collects snapshot and network packet information of the virtual machine, and reproduces the failure of server applications in the testing environment using the collected information. We describe the experimental result of the influence on response time of server applications by collecting network packet information.

#### 1. はじめに

サーバ上で動作しているアプリケーションやミドルウェアの障害が発生した場合、原因究明・対策を実施する必要がある。特に出荷後のシステムで障害が発生した場合には、原因究明のための情報(例えば、ダンプ情報やログ情報など)を採取し、検証センターなどの設備にて、採取した情報の解析、再現環境の構築、障害再現による原因究明、対策の実施を行うこととなる。しかし、障害の再現は時間を要する作業であり、検証センターの環境では再現が困難という課題がある。

本稿では、サーバ仮想化を実現するサーバ仮想化ソフトウェアが持つスナップショット機能を利用し、障害が発生したサーバへのネットワークパケット情報を用いて、検証センターの環境において障害再現を行うシステムを提案する。本稿の提案システムでは、障害再現に必要な情報を収集する専用の仮想計算機を動作させ、収集対象の仮想計算機のスナップショット情報と、収集対象の仮想計算機のネットワークパケット情報の収集を実施する。情報収集は一般に公開されている API を使用して実現しており、業務システムが動作している仮想計算機への情報収集のための特殊なモジュールの導入が不要であり、サーバ仮想化ソフ

トウェアの種別にも依存しない。ただし、ネットワークパケット情報の収集は、業務システム稼働中に実施されるものであるため、収集情報の保存による負荷が業務に影響を与える可能性がある。このため、収集したネットワークパケット情報のうち、再現確認に必要な応答パケットのデータ部分をハッシュ値に変換し、情報量を削減することで収集情報の保存による負荷の影響を抑える方式とした。

本方式の有効性を確認するために、プロトタイプシステムを構築し、障害再現情報収集時のリソース使用量、サーバアプリケーションの応答性能への影響について評価を実施した。評価の結果、本方式によりサーバアプリケーションの応答性能劣化は1%程度であり、ほぼ影響がないこと、特に Web サーバ等のように応答データが通信の大半を占めるサーバにおいて有効であることを確認した。

以下、2章では本システムを提案するに至った背景と課題について示す。3章では本システムの構成について示し、4章では障害再現システムの一構成要素である障害再現のための情報収集を実現する各機能の実現方法を示す。5章ではプロトタイプによる評価結果を示し、6章では実システムに適用するための課題について示す。

## 2. 背景と課題

### 2.1 非計画停止の要因

システムの停止には、バックアップやハードウェアメンテナンス等に伴う計画停止と障害等に伴う非計画停止がある。非計画停止の要因は、従業員エラー（オペレーション・エラー） 40%、アプリケーション・エラー 25%、作業遅延 15%、システム・エラー 10%、ハードウェア障害 5%、設備障害 2.5%、その他 2.5% というデータが公開されている[1]。

このうち、ハードウェア障害については、仮想障害試験システムにより、サーバ仮想化技術を用いて仮想計算機に対してハードウェア障害を模擬することにより、クラスタソフトウェア等の対策が正しく動作するかの検証を事前に実施することを可能としている[2]。従って、他の要因のうち、システム停止の大きな要因となっているオペレーション・エラーやアプリケーション・エラーが発生した場合の早期回復は重要となる。

### 2.2 障害解析のフロー

アプリケーションのエラー等が発生した場合、ログなどを利用し解析を行い、原因の究明を行い、対策を実施する。障害解析可能な情報が存在しない場合には、障害解析環境において障害を再現させる必要がある。しかし実際には障害の再現は難しく、長時間掛かる場合も多い。更に再現しない場合もあり、そのような場合には、稼働環境で追加情報収集の準備を行い、再現待ちをする[3]。

このことから、障害再現を容易に実現できることは、システムのダウンタイム削減につながり、信頼性向上につながる。また、障害再現は、障害原因の解析に役立つだけでなく、対策の妥当性の確認にも有効となる。

### 2.3 サーバ仮想化技術

プロセッサのマルチコア化や仮想化支援技術を搭載したプロセッサの登場により、1台の物理計算機上で複数の仮想計算機を動作させるサーバ仮想化技術が利用されてきている。サーバ仮想化技術を実現したソフトウェアは、仮想計算機のスナップショット情報として、ある時点のメモリ・ディスクのイメージを保存する機能を保持している。スナップショット機能は、復元ポイントを提供する機能であるため、パッチ適用やアプリケーションのアップグレードに失敗した場合に、適用前の状態に戻すことができる。

### 2.4 障害原因解析における課題

これまで述べてきたように、オペレーション・エラーやアプリケーション・エラーが発生した場合の解析において、障害を再現させることは重要であるが、実システムの環境を検証センター等の環境で構築するこ

とならびに障害を再現させることは困難である。

また、ある障害に伴って、アプリケーションが停止してしまった場合などは、情報収集も困難である場合がある。例えば、あるアプリケーションの障害により、別のアプリケーションプロセスに対してプロセス停止のシグナルを送信することでダウンさせてしまうというケースがある。このような障害が発生した場合、仮にダウンしたアプリケーションのダンプ情報が採取できたとしても、ダンプ情報からは、停止シグナルを受信したためダウンしたという情報がわかるだけであり、誰が何故停止のシグナルを送出したのかといった本来の障害原因を追究することが出来ない課題がある。

障害の再現が困難である、あるいはログやダンプの情報だけでは障害の原因追求が困難な場合があるといった課題を解決する手段として、スナップショット機能を利用することが考えられ、スナップショットを用いたデバッグの手法についても提案が行われている[4]。本デバッグ手法では、エラーが発生している状態のスナップショット情報を用いて、デバッグを実施するものであるため、エラーやオペレーションミスによりアプリケーションが停止してしまったケースでは、その障害が発生した状況を再現するための情報がなければ利用することが出来ない。

## 3. 障害再現システム

サーバアプリケーションの場合、オペレーション等の入力はネットワーク経由になることから、障害が発生した状況を再現するための情報として、対象となる仮想計算機へのネットワークパケット情報を利用することが可能である。ネットワークパケット情報とスナップショット情報を組み合わせることで、障害が発生する前のスナップショット情報から起動した仮想計算機に対して、ネットワークパケットを再度送出することで、障害を再現させることが可能となると考えた。従って、スナップショット情報とネットワークパケット情報の収集を行い、障害発生とは別環境で収集した情報を用いて、障害発生前の状態から障害発生までの過程を再現する障害再現システムを提案する。

### 3.1 システム構成

障害再現システムは、実システム環境で動作する障害再現のための情報収集する機能と、収集した情報を用いて障害発生とは別環境で障害を再現させる機能の二つから構成される。情報収集する機能は、収集対象の仮想計算機が動作している物理計算機上に情報収集のための専用の仮想計算機を配備し、スナップショット情報とネットワークパケット情報の収集を実施する。障害を再現させる機能は、実システム環境にて収集した情報を用いて、スナップショット情報から障害が発

生じた仮想計算機を起動し、ネットワークパケット情報を送出することで障害の再現を実施する。システム構成を図 1 に示す。本稿では、主に障害再現のための情報収集に関する機能について述べる。

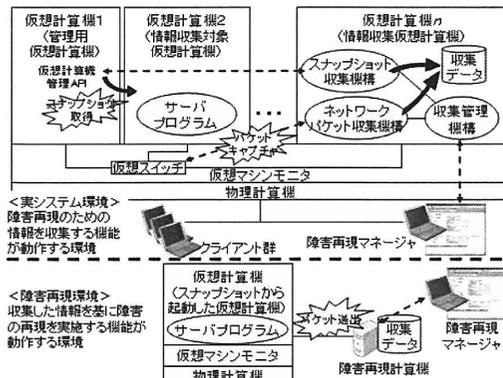


図 1：障害再現システム構成図

### 3.2 障害再現マネージャ

情報収集仮想計算機に対して、障害再現情報収集の開始・停止、収集ターゲットとなる仮想計算機の指定や収集周期等の指示を行う。また、障害再現時には障害再現計算機に対して、収集したスナップショット情報からの仮想計算機の起動ならびに収集したネットワークパケット情報を送出する指示を行う。

### 3.3 情報収集仮想計算機

収集対象の仮想計算機が動作している物理計算機上で動作する仮想計算機であり、表 1 に示した 3 つの機構を備えている。仮想計算機として実現することにより、サーバ仮想化ソフトウェアが持つ仮想スイッチを利用することが出来るため、障害再現情報の収集のために特別な機材の導入が不要であることや収集時のパケットロスがなくなるという効果を得ることが出来る。

表 1：情報収集仮想計算機が備えている機構

機構	概要
収集管理機構	障害再現マネージャからのリクエストに基づき、障害再現のための情報収集対象となる仮想計算機や収集周期を管理
スナップショット収集機構	収集対象となる仮想計算機のスナップショット情報を収集
ネットワークパケット収集機構	対象となる仮想計算機が実施しているネットワークパケット情報を収集

### 3.4 障害再現計算機

障害再現環境上で動作し、障害再現マネージャからの指示に基づき、情報収集仮想計算機が収集したスナップショット情報から仮想計算機を起動し、起動後ネットワークパケット情報を送出する。

## 4. 障害再現情報収集実現方式

障害再現システムの一構成要素である障害再現のための情報収集を実現する情報収集仮想計算機上の各機構の実現方式について示す。

### 4.1 収集管理機構

障害再現のための情報収集に関する制御は、障害再現マネージャからの要求に基づき行われる。収集管理機構で管理される情報は表 2 に示した 4 種類である。

表 2：管理情報

種別	概要
実行状態	障害再現情報収集の開始・停止
情報収集対象仮想計算機識別情報	情報収集対象の仮想計算機の識別情報（仮想計算機名、IP アドレス）
サーバ仮想化ソフトウェア種別	サーバ仮想化ソフトウェアの種類（VMware[5]、Xen[6]、Hyper-V[7]）
収集周期	収集周期（定時周期、時間周期）

#### 4.1.1 実行状態

障害再現情報収集の開始・停止要求を受け付け、現在障害再現情報の収集を実行中か停止中かを識別する情報。停止中に開始要求を受け付けると、スナップショット収集機構、ネットワークパケット収集機構に対して情報収集指示を行う等の制御に利用する。

#### 4.1.2 情報収集対象仮想計算機識別情報

収集対象の仮想計算機の識別情報を基に、スナップショット収集機構が取得する仮想計算機の指定やネットワークパケット収集機構が収集するネットワークパケット情報の識別に利用する。

#### 4.1.3 サーバ仮想化ソフトウェア種別

サーバ仮想化ソフトウェアの種類により、スナップショットの取得やスナップショットからの仮想計算機の起動の API が異なるため、スナップショット収集機構がサーバ仮想化ソフトウェアごとに取得コマンドを用意している。このコマンドの識別に利用する。

#### 4.1.4 収集周期

障害再現システムでは、収集したスナップショット情報から仮想計算機を起動し、収集したネットワークパケット情報を起動した仮想計算機に対して送出することで障害の再現を実現する。このため、スナップショット情報を取得してからのネットワークパケット情報をすべて保存しておくことが必要となる。仮想計算機の起動時にスナップショット情報を取得し、以降ネットワークパケットを収集し続けることでも実現可能であるが、保存するネットワークパケット情報が多量になるため、ストレージ容量の見積りが出来なくなる点、再現までに時間を要してしまう点が問題となる。本システムでは、定期的なスナップショット情報を取得することで、ネットワークパケット情報の情報量の削減、再現までの時間短縮を可能とする。

収集周期については、毎日負荷の低い決まった時間帯にスナップショットの取得を実施する定時周期と最初に起動した時刻からの時間周期（6時間周期等）の2種類の指定を可能としている。

#### 4.2 スナップショット収集機構

スナップショット情報の取得は、サーバ仮想化ソフトウェアが保持する機能を利用して実施する。取得するための API は各サーバ仮想化ソフトウェアによって異なるため、対応する API を用いたコマンドを用意しておき、利用されているサーバ仮想化ソフトウェアに対応したコマンドを実行することで、スナップショット情報の取得を行う。

スナップショット情報は、前回からの差分とした場合、取得した回数に応じてストレージ容量を必要とするため、常に前回取得分を削除した状態で取得を行い、必要となるストレージ容量の削減を実現する。

#### 4.3 ネットワークパケット収集機構

ネットワークパケットの収集はサーバ仮想化ソフトウェアが保持する仮想スイッチより `promiscuous` モードを利用し、パケット情報を取得し、情報収集対象の仮想計算機への要求・応答のネットワークパケットのみを収集する。スナップショット情報は、収集周期に指定された周期での収集であるが、ネットワークパケット情報の収集は、収集対象となる仮想計算機が稼働中、動作し続けるものであるため、収集の負荷を抑えることが重要となる。CPU、メモリについては、情報収集仮想計算機を定義する際に指定した量を超えることがないため、他の仮想計算機は最初からそのリソースはないと判断して構築することが可能であるが、収集したネットワークパケット情報をストレージに保存する I/O については、収集対象並びに他の仮想計算機と共存する可能性がある。このため、I/O 量を減らすことが必要となる。本システムでは、I/O 量を減ら

すための施策としてパケット情報の圧縮と応答パケット情報をハッシュ値に変換することで削減を実施した。

##### 4.3.1 パケット情報の保存

パケット情報の収集は、Wireshark[8]等のパケットキャプチャツールが使用している `pcap` 形式で保存することとした。これにより、収集したパケット情報を再現の目的のみならず、実際にどのような通信が行われていたのか等の解析を行う際に、パケットキャプチャツールの機能を利用することが可能となる。

##### 4.3.2 パケット情報の圧縮

I/O 量を減らすために、収集したパケット情報をストレージに保存する際、圧縮を行った。圧縮アルゴリズムについては、LZO[9]が高速に圧縮を行うことが出来るとの報告[10]がなされており、LZO を用いて圧縮を実施した。

##### 4.3.3 ハッシュ値による応答パケット情報の削減

再現を行うことを目的とした場合には、収集対象の仮想計算機への要求パケットのみを利用することで実現可能である。このことから、I/O 量を減らすためには、収集対象の仮想計算機からの応答パケットについては収集対象外とすることが最も有効である。しかし、サーバ上で動作するアプリケーションの場合、コンソール画面への出力等がないため、再現時に同じ現象が発生しているかを確認することが困難である。また、圧縮の場合、パケットのデータの内容によって圧縮率が変化するため、バイナリデータ等のように圧縮の効果が期待できない可能性もある。

従って、応答パケットについては、データ部分をハッシュ値に変換して保存することとし、データ量の削減を行うこととした。これにより、再現時も同様にハッシュ値に変換し、変換された値の比較を行うことで、情報収集時と再現時に同じ現象が発生しているかを確認可能とした。更に、実システムでの動作時と再現環境での再現時で応答パケットのパケット長が変わってしまうことによりハッシュ値での比較が出来なくなることを避けるため、ハッシュ値変換する前のデータ長情報をデータとして保存することとした。

ネットワークパケット収集機構では、これらの方式を組合せ、図 2 に示した動作を行う。ネットワークパケット情報の収集はキャプチャプロセスで実施し、収集したネットワークパケット情報のディスクへの保存は、パケット保存プロセスが実施する。パケットキャプチャプロセスは、取得したネットワークパケット情報のうち、対象の仮想計算機への要求あるいは対象仮想計算機からの応答のパケットのみにフィルタリングした後、応答パケットについてはハッシュ値に変換し、共有メモリ上のパケット格納領域に格納する。パケッ

ト保存プロセスは、パケット格納領域一杯になると、その領域のデータを圧縮しディスクに保存する。この動作により、応答データのハッシュ値変換ならびに収集データの圧縮保存を実現する。

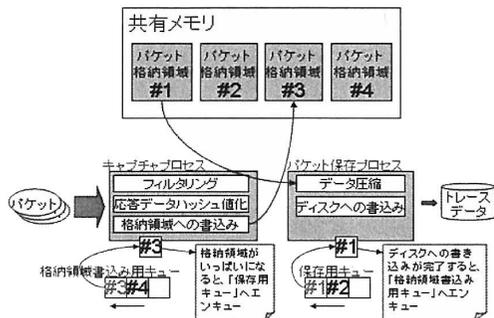


図 2：パケット情報収集方式

## 5. 評価

本システムのプロトタイプを構築し、情報収集におけるリソース使用量、情報収集実施による応答性能への影響について以下の評価を行った。

- (1) 障害再現の確認
- (2) 情報収集仮想計算機のリソース使用量の確認
- (3) 応答性能への影響の確認

### 5.1 評価環境

評価は、情報収集対象の物理計算機 1 台、情報収集対象のサーバアプリケーションに対してリクエストを行うクライアント 1 台、障害再現用の物理計算機 1 台の計 3 台の物理計算機を 100Mbps の LAN にて接続した環境にて実施した。

各物理計算機とソフトウェア等の構成を表 3 に示す。また、情報収集対象の物理計算機上で動作させた仮想計算機の構成を表 4 に示す。

表 3：評価環境構成

機種	情報収集対象／再現環境計算機
CPU	Intel Xeon X5160 3.0GHz (1ソケット2コア)
メモリ	8GB
ストレージ	SAS 72GB
サーバ仮想化ソフトウェア	VMware ESXi バージョン 3.5 Update 2

表 4：仮想計算機構成

	情報収集対象仮想計算機	情報収集仮想計算機
CPU	1CPU	1CPU
メモリ	512MB	1GB
OS	Red Hat Enterprise Linux 5	Red Hat Enterprise Linux 5

### 5.2 障害再現の確認

障害再現の動作を確認するため、クライアントから 5 秒ごと通信を行い、サーバ側は受け取ったデータに格納されている番号情報をサーバのコンソール上に表示し、特定の番号(今回は 10)を受信した際に、エラーとなり終了するという簡単なプログラムを作成し、再現の確認を行った。

クライアントプログラム起動後、障害再現情報収集開始要求を障害再現マネージャから送出し、スナップショット情報とネットワークパケット情報の収集を開始(図 3)。エラーとなった時点で障害再現情報収集を停止し、再現環境に収集したスナップショット情報、ネットワークパケット情報を配備し、再現の確認を実施した。

結果を図 4 に示す。上の画面が通常動作時に障害が発生したことを示した画面で、下の画面が障害再現を実施した際の動作を示した画面である。上の画面で count=4 と count=5 の間が他の count の間の間隔と異なっていることがわかる。このデータの送受信間でスナップショットを取得している。下の画面では、count=4 まではスナップショットからの起動による画面状態であるため、実際の障害時と同じ時刻情報が表示されているが、count=5 以降は再現環境にて再現を実施した際の時刻情報となっていることがわかり、以降の通信が行われ、障害が再現されたことを確認した。

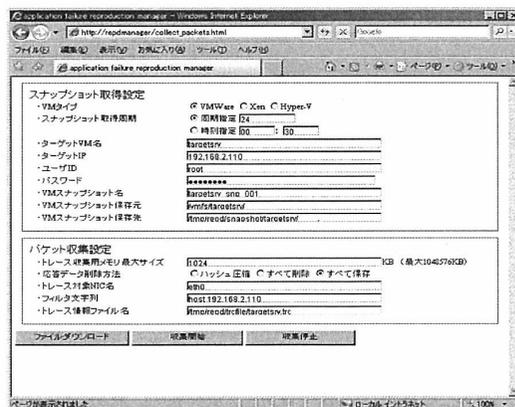


図 3：障害再現情報収集要求



図 4：障害再現時画面

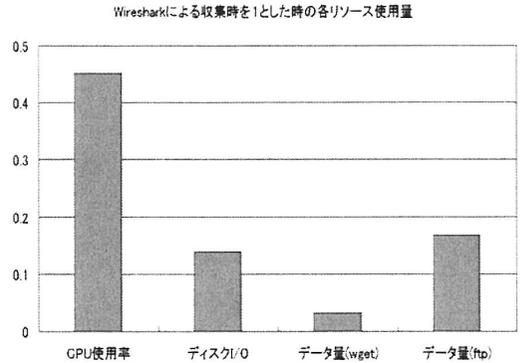


図 5：リソース使用量

### 5.3 情報収集仮想計算機のリソース使用量の確認

情報収集仮想計算機のリソース使用量を確認するため、表 5 に示したネットワーク通信を実施し、再現のための情報収集を実施する情報収集仮想計算機のリソース使用量を、Wireshark を利用して収集した場合と比較した。

それぞれ、データサイズとして 100MB、200MB、400MB のファイルを用意し、情報収集仮想計算機が使用した CPU 周波数、ストレージ I/O レートの測定、収集したデータ量の測定を実施した。

結果は、図 5 に示したように、wget、ftp のいずれの通信においても、Wireshark によるパケット収集に比べて、リソース使用量が 50% 以下であり、特に、当初課題として考えていたストレージに対する影響は、I/O レートが 14% 程度、データ量が wget 時に 3%、ftp 時に 17% 程度となり、本システムにおける提案方式の有効性を確認することが出来た。wget 時にデータ量が ftp 時より削減できている理由は、データ取得要求に対して応答データの比率が大きいため、ハッシュ値に変換したデータ削減の効果が出ていると考えられる。CPU に関しては、使用周波数が 1.3GHz 程度であったことから、現状のサーバが搭載している 1 コアが 3GHz 程度であることを考慮すると、情報収集仮想計算機に 1 コアを割り当てることで十分対応可能である。

表 5：評価用ネットワーク通信

通信内容	想定
wget によるデータダウンロード	WWW サーバの情報収集を想定。
ftp によるデータのアップロード	DB サーバへのデータのインポートを想定。

### 5.4 応答性能への影響の確認

表 5 に示した通信における転送速度を以下の 3 ケースで比較した。

- (1) ネットワークパケット情報未収集
- (2) Wireshark による収集・ファイル保存
- (3) 障害再現システムによるハッシュ値変換・圧縮による保存

結果は、図 6 に示したように、未収集時を 1 とした時の応答性能の劣化は、wget、ftp のいずれの通信においても Wireshark による収集時に比べ、小さいことがわかり、1% 以下であった。Wireshark を利用した場合でも本システムにて収集した場合であっても、ftp による通信の性能劣化が小さかった。これは、put 要求による書き込みであることから、ファイルシステムバッファへの書き込みにて処理が完了したこととなる点とサーバ仮想化ソフトウェアが、ftp による書き込み要求とネットワークパケット情報の保存のための書き込みを可能な限り、シーケンシャルへの書き込み順序制御等を実施していることの結果が出ていることが想定される。

一方、wget は読み込みのみで本来完了すべき動作に、収集したネットワークパケット情報の保存という書き込みの動作が処理の間に入ることになる。このため、ディスクへの読み込みと書き込みが混在したリクエストとなることから、読み込みへの影響が出ていることが性能劣化を引き起こしている要因と考えられる。このことから、収集したデータ量の削減を行うことは、業務への影響を抑える効果があると考えられ、本システムの有効性を確認することが出来た。

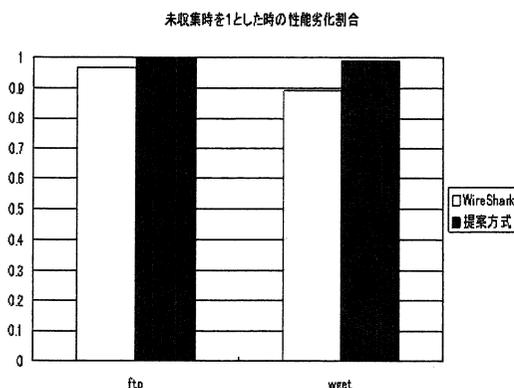


図 6：情報収集の影響

## 6. 今後の課題

本障害再現システムを用いることで、実システムへの影響を少なく、収集した情報を用いて、実システムで発生したサーバアプリケーションの障害を解析環境で再現させることが可能となる。これにより、再現環境で障害が発生したサーバアプリケーションプロセスに対してデバッガにてアタッチし、詳細な解析を実施することや、障害発生時のリソース使用状況の把握、実際の通信情報の解析等を実施することができるといった効果が期待できる。

また、定期的スナップショット情報やネットワークパケット情報を更新しながら、情報収集を行うことから、1年以上動作したことにより発生するメモリーリーク等の障害についても、収集周期の時間内で再現することが可能となる。

しかし、実システムで利用するためにはいくつかの課題が挙げられる。

### (1) ライブマイグレーション機能への対応

サーバ仮想化ソフトウェアには、オンラインで別の物理計算機に仮想計算機を移動させるライブマイグレーション機能が実装されている。収集対象の仮想計算機がライブマイグレーション機能により、他の物理計算機に移動した場合、元々動作していた物理計算機上の仮想スイッチからネットワークパケット情報の収集を実施しているため、移動した後のネットワークパケット情報の収集やスナップショット情報の収集を行うことが出来ない。将来、複数の物理計算機に跨る仮想スイッチの機能が実装されてくれば、ネットワークパケット情報の収集を実施できる可能性があるが、現状の仕組みでは、実現困難である。

### (2) 複数の仮想計算機の情報収集

通常、サーバ仮想化技術により、1台の物理計算機上には仮想計算機が4台以上動作することが想定される。これらの仮想計算機すべてに対して障害再現のための情報収集を実施する場合、その台数分のリソースが必要となり、実用上問題となる可能性が高い。特に、ネットワークパケットの取りこぼしが発生する可能性が高くなるため、応答データについては、収集しない等の対策を検討する必要がある。

### (3) 動作時に通信するデータが変化するアプリケーションの再現

動作時に乱数等を使用して、クライアント・サーバ間でネゴシエーションを行うようなアプリケーションの場合、再現時は、実行時にネゴシエーションした情報に基づいた通信を実行しようとするが、再現時にサーバ上のアプリケーションが別の値を作り、それを基にした通信を実施しようとした場合には、再現時には通信失敗となり、再現することが出来ない。

## 7. おわりに

実システムにおいてサーバアプリケーションの障害が発生した際の原因追及のための解析には、検証センター等の環境で障害再現を実施することが重要である。しかし、障害を再現させることは困難であり、原因追及に時間を要するという課題があった。

本課題を解決するため、障害再現システムを提案した。障害再現システムは、再現に必要な情報を収集する機能と収集した情報を基に障害を再現する機能から構成され、本稿では特に情報を収集する部分の実現方式について述べた。障害を再現させるために必要な情報は、仮想計算機のスナップショット情報とネットワークパケット情報である。特にネットワークパケット情報は、業務稼働中に情報収集処理が動作するため、収集情報の保存の負荷が業務に影響を及ぼす可能性がある。従って、本障害再現システムでは、応答データをハッシュ値に変換し、収集したデータを圧縮して保存することでデータ量の削減を行う方式を導入した。

検討した方式の有効性を確認するため、通常のネットワークパケット情報を収集するツールである Wireshark を用いた方式と収集に要するリソース使用量について比較した。評価の結果、1.3GHz 程度の CPU リソースを収集専用の仮想計算機に割り当てることで、障害再現のための情報の収集が可能であることを示した。特に、Web サーバ等のように応答データが通信の大半を占めるサーバにおいては、データ量の削減効果が顕著となり、I/O 量の削減が行われ、業務を行っている仮想計算機の I/O 処理に対する影響の削減に有効であることを確認した。

一方で、ライブマイグレーション機能への対応や複数の仮想計算機の再現情報の取得等に課題があることを示した。

今後は、常に動作させているわけではなく、障害が発生した場合の網張りモジュールの位置づけで、オンラインで本機能を有した仮想計算機をアドオンし、障害再現情報の収集を可能とするなど、実システムへの適用に向けた技術開発を検討していく予定である。

## 参 考 文 献

- [1] 庄内 享：“IT システム高信頼化技術の動向と課題－メインフレームからオープン時代へ－”，電子情報通信学会 信学技報 Vol.107, No.17(20070413) pp. 25-30, 2007.
- [2] 國分 俊介, 片山 吉章, 樋口 毅, 松本 利夫, 相浦 利治：“仮想化環境におけるハードウェア障害模擬と HA クラスタシステム試験への適用”，電子情報通信学会 信学技報 Vol.108, No.181(20080805) pp. 1-7, 2008.
- [3] OSS 技術開発・評価コンソーシアム：“2005 年度上期オープンソースソフトウェア活用基盤整備事業「OSS 性能・信頼性評価／障害解析ツール開発」OS 層～障害解析の手順・ツール評価編～”，2005.
- [4] 菊地 誠, 阿部 洋丈, 梅村 恭司：“スナップショットを用いたデバッグ環境の構築”，情報処理学会研究報告(2008-OS-108) pp.187-194, 2008.
- [5] VMware Inc.: <http://www.vmware.com/>
- [6] Xen.org: <http://www.xen.org/>
- [7] Microsoft Corp. : <http://www.microsoft.com/windowsserver2008/en/us/hyperv.aspx>
- [8] Wireshark.org: <http://www.wireshark.org/>
- [9] oberhumer.com : LZO real-time data compression library : <http://www.oberhumer.com/opensource/lzo/>
- [10] 清水 奨, 風間 一洋, 廣津 登志夫, 後藤 滋樹：“リアルタイム圧縮によるパケットキャプチャの高速化”，情報処理学会論文誌：コンピューティングシステム, Vol.47, No. SIG 7(ACS 14) pp.183-193, 2006.