

実験環境でのプロファイル作成のための データ収集手法の基礎調査

神田 慎也^{†1} 榎本 真俊^{†1} 櫛山 寛章^{†1}
門林 雄基^{†1} 山口 英^{†1}

現在のインターネットは社会基盤として様々なサービスが展開されている。既存のサービスに影響を及ぼす恐れがあるため、インターネットは実験環境に適していない。そのため、インターネットに変わる大規模な実験環境が求められている。大規模な実験環境では、仮想化技術が実験に与える影響が大きく、実験で重要なログの収集に支障をきたす恐れがある。しかし、事前にそれらを知ることは難しく、実験後にログを見て判断するしか無い。そこで、実験プロファイルを作成し、過去の実験から新しい実験が失敗しないためのノウハウを得る。今回の論文では、プロファイルの内容について議論した。

Experiment Profiles to Understand Workload Characteristics

SHINYA KANDA,^{†1} MASATOSHI ENOMOTO,^{†1}
HIROAKI HAZEYAMA,^{†1} YOUKI KADOBASHI^{†1}
and SUGURU YAMAGUCHI^{†1}

Today's Internet provides a variety of services deployed as a social infrastructure. Because existing services could be affected, experiments are not carried out over the Internet. Therefore, there is a demand for a large-scale environment to perform Internet experiments. In a large-scale experiment environment, virtualization technologies have a great impact on the experiments, which may interfere with the important task of collecting experiment logs. However, it is difficult to anticipate such impact, and it can only be understood once the experiment is over. By creating an experiment profile, one can benefit from the know-how of past experiments in order not to fail new experiments. In this paper, we discuss the contents of the profile.

1. はじめに

現在のインターネット環境は、社会基盤として利用されており、様々なサービスが運用されている。新しい技術やサービスも次々と導入されており、インターネット上でサービスを導入するための実験を行いたいという要求がある。しかし、インターネットでは、既存のサービスに影響を及ぼす可能性があるため、実験には適していない。そのため、新しい技術やサービスのための実験環境として、インターネットに良く似た実験環境が必要とされている。

また、インターネットと共にシステムも肥大化しており、実験環境もそれに見合う大規模なものも求められている。現在の大規模実験施設として StarBED^{*1}や Emulab^{*2}が存在する。この実験施設は、数百台の実ノードで構成されており、仮想化技術を用いて模擬インターネット環境を構築することができる。仮想化環境では、1つの実ノードの上で複数のOSが動作するため、ゲストOS間で影響を及ぼし、実験で重要なログの収集に問題が発生する可能性がある。実験の規模に伴い、出力されるログの量も増大するため、ログ収集方式には注意を払わなければならない。ログは、実験において最も重要なもので、実験結果を知る上でログが記録できなかった場合には、実験は失敗となってしまふ。しかし、実験前にこれらの予測をすることは難しく、実験を行った後にログを確認する事でしかログの記録状況を把握する事が出来ない。そのため、大規模な実験のためにリソースの性能を最大限まで引き出すことと、ログを収集することがトレードオフとなってしまふ。

そこで今回は、過去の実験から実験プロファイルを作成し、新しい実験に役立てるための実験プロファイルを提案する。実験プロファイルには、過去の実験のデータを収集しており、新しい実験のための過去の実験ノウハウを取得することができる。また、実験の特徴と捕らえたデータを蓄積しているため、過去に行われた似ている実験を参照し、ログ収集をスムーズに行う事ができる。このように実験プロファイルを作成する事で、新たに行う実験のイメージをつかむ事ができ、実験の失敗を回避することができる。しかし、実験プロファイルを作成するためには、実験の情報を抽象化して取得しなければならないといった問題が残っている。本論文では、実験プロファイルのためのデータ収集手法の基礎調査を行った。

^{†1} 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

^{*1} StarBED Project, <http://www.starbed.org>

^{*2} Emulab, <http://www.emulab.net>

2. 大規模実験

大規模実験とは、実インターネット環境ではなく、それを模した環境をスケールアウトにより構築し、行う実験である。大規模な実験トポロジを構築するために、大規模な数のノードを用意することは、経済的、空間、保守などのコストがかかるため現実的ではない。実験施設では、1つのノードの上に複数のOSを展開し、スケールアウトを行う。大規模に行うことにより、小規模な実験ではできなかった挙動の観測ができ、実験の信頼性が上がる。他にもシミュレーションでの実験方法もあるが、大規模実験では既存のサービスの干渉など、より実インターネット環境に近い挙動を観測する事が出来る。

また、実インターネット環境で実験を行うことに比べ、他のサービスや影響を考慮する必要がなく、実験対象のパフォーマンスのみに集中して実験を行う事ができる。このように、実インターネット環境に1番近い実験環境として活躍しているのが大規模実験である。しかし、大規模実験では、1回の実験にかかるコストが大きく、トラブルや再実験などは回避しなければならない。

2.1 現在の実験施設

現在の実験施設では、StarBED¹⁾やEmulab²⁾といった例が挙げられる。これらの実験施設は、ネットワークテストベッドと呼ばれ、インターネット関連技術の研究開発の際に利用する大規模インターネット実証実験環境である。実ノードを用いて実験を行うため、実際にインターネットに投入される機器やソフトウェアを利用できるため、実インターネットの挙動に近い環境となる。

しかし、機器の調達や物理的な接続や設定、ノード制御のためのコストなどがソフトウェアシミュレータを利用した場合と比べて大きい。実験施設は、数百台の実ノードで構成されており、限られた資源の中でスケールアウトをしなければならない。そのため、仮想計算機環境を用いて模擬インターネット環境を構築する。小規模な実ノード実証実験での実験を行ったあと、実環境に問題があったアプリケーションなどは大規模実証実験で実験を行う事により、事前に発見できたり、フィードバックしたりなどのメリットもある。

大規模実験施設では、仮想化技術を用いて構築することが一般的であり、仮想化ソフトウェアの性能に左右される事がある。仮想化ソフトウェアを利用することで、大規模になる反面、トラブルの原因となる一面もある。

2.2 仮想計算機環境での特徴と問題

実験環境では、仮想計算機を利用することで、実ノードのリソースを最大限利用できる。

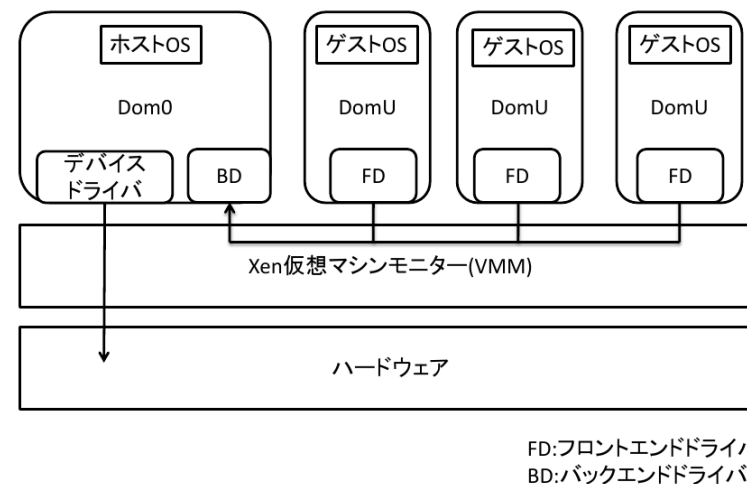


図1 Xenを用いた準仮想化環境の例
Fig. 1 Sample of Virtual Machine Environment

また、1つの実ノードの上に複数のOSを展開することで大規模な実験環境を構築できるといった可能性がある。仮想計算機環境で構築した例を図1に示す。この図では、1つのハードウェアの上に仮想マシンモニターとしてXen³⁾が展開している。物理的なハードウェアはデバイスドライバを介することでホストOSのみが操作することができる。そのため、ゲストOSはホストOSを介してログを出力するためボトルネックとなりやすい。大規模実験では、1つのハードウェアを複数のOSで共有しているため、1つのOSでトラブルがあった場合には実ノード上のOS全体に影響が波及してしまう可能性がある。

仮想計算機環境での資源管理では、OSの資源管理機構が入れ子構造となり完璧なリソースアロケーションができず、ゲストOS同士でのリソースの競合が発生する。そのため、ログに割り当てられるリソースが不明瞭となり、リソース競合によるログ収集の支障が発生する。実験では、ログを見る事により、実験対象のパフォーマンスやプロセス、エラーや障害などを発見する。ログの収集が最も重要であり、ログを収集できなかった場合には再実験を行わなければならないコストがかかる。

2.3 大規模実験による問題

大規模な実験環境では、ログの収集が大きな課題である。仮想化環境で構築された実験環

境は、実ノードを最大限利用しているため、資源管理が完璧ではない。そのため、実験で重要なログを落としたり、収集できなかつたりといった問題が発生する。

しかし、現状では事前にこれらの問題の存在を知ることは難しく、実験後にログを見て判断するしかない。大規模実験環境では、1回の実験にかかるコストが大きいため、再実験は避けなければならない。そういった観点から実験前にログ収集の問題を知ることが望ましい。本研究では、過去の実験データから新しく行う実験のログ収集に関する問題を取り出すことでこれらの問題を解決したいと考えている。そのためには、負荷特性の把握やプロファイルリングを行い、過去の実験から抽象化されたデータを取得することが必要である。これにより、類似した実験のプロファイルデータを用いて実験の計画立てを行う事が有効となる。

3. プロファイル

プロファイルとは、過去の実験からリソースやネットワークの情報を取り出して抽象化されたデータのことである。図2にプロファイルのイメージを示す。まず最初に、実験者はマイクロベンチマーク及びハードウェア情報などの初期値を登録しなければならない。プロファイルでは、初期値を Ver1.0 位置づける。その後、他の実験者がプロファイルを参照、編集及び情報の追加を行う。プロファイル情報を編集、追加した場合には、プロファイルのバージョン番号を増やしていく。

プロファイルの要件には、追実験できること、トラブル地点が予測できること、実験の特徴を表せることが挙げられる。その情報とは、主にハードウェア構成、使用するソフトウェア、設定ファイルである。トラブル地点を予測するには、予めスケールアウトの実験が行われており、その際何かしらトラブルがあった場所の記録をとっておかなければならない。また、実験の特徴を表すには、ノードのマイクロベンチマークが必要である。その他にも実験を行っている際の CPU、メモリ、ディスク、トラフィック使用量を計測しておかなければならない。

これらのことを考えると、依存するものによって、プロファイルデータとして取得するデータが変化する。依存するものとして、OS、ハードウェア、アプリケーション、トポロジが考えられる。OS 依存の場合、取得すべきデータはカーネル情報である。ハードウェア依存の場合、取得すべきデータは I/O 情報である。アプリケーション依存の場合、取得すべきデータはネットワークに流れるトラフィックの量やパケット情報である。

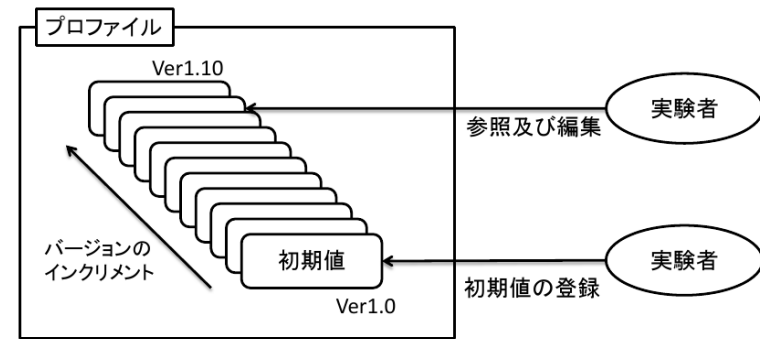


図2 プロファイルのイメージ図
Fig.2 Image of Profile

3.1 取得するデータ

プロファイルを作成するにあたり、取得しなければならないデータとして、特徴がある。ノードの特徴に関しては、マイクロベンチマークによって取得する。ベンチマークツールは現在数種類存在しており、bonnie++⁴⁾ や IOzone⁵⁾, Netper⁶⁾ などがある。実験の特徴に関しては、CPU 情報、メモリ、I/O、トラフィックなどの情報を取得する。これらの情報は top, iostat, vmstat などのコマンドで /proc の中を見て取得することができる。

実験中はそれ以外にも、ログを収集する必要がある。ログが出力される量や出力先を監視しなければならない。もし、ログが出力されているにもかかわらずディスクが飽和状態になり書き込めていなかった場合には、実験は失敗となってしまう。

また、ネットワークのトポロジも取得しておかなければならない。ネットワークテストベットでは、ネットワークの構成も重要になる。実験環境の中でどのようにネットワークが構成され、実験の特徴として取得しておかなければならない。実験施設でのバックボーンネットワークは広帯域のため、実験自体に影響を及ぼすことは少ないが、汎用的なデータとして取得しておく。

3.2 プロファイルの登録

プロファイルの登録には、生データの取得及びデータの抽象化が必要である。生データとは、実験を行う際に出力される実際のログであったり、実験対象のコンフィグであったり、抽象化されていないそのままのデータの事である。データを取得する場合、最初はこのような生データで取得することとなる。しかし、プロファイルでは、生データは汎用性に欠け

るためプロフィールとして登録しても利用することができない。プロフィールは環境が異なる場合でも利用することが出来るように、汎用性を持つデータで登録しておく必要がある。さらに過去の実験者のリコメンデーションとしてコメントと共に登録しておけば有効に利用することができる。

抽象化されたデータは、汎用性を持つデータであり、ボトルネックの発見に一役を担う。実験で利用したリソース配分や OS のパラメータは汎用性を持つデータであり、プロフィールにそのまま利用することができる。ボトルネックが発覚した場合は、プロフィールの中から過去のデータを抽出し、トラブルシューティングを行う。もし、プロフィールの中に解決策が無い場合は、プロフィールのためのデータを取得し直さなければならない。プロフィールのためのデータ取得は、ベンチマークツールを使用する場合と予め決められた手順通りにデータを取得できるよう決めておく必要がある。

3.3 データの使用方法

データの使用方法は、ベンチマークを取得した環境かどうかで対応が変わってくる。ベンチマークを取得した環境では、そのデータを信用して再実験であったり、スケールアウトであったり、抽象化したデータとして利用する事が出来る。ベンチマークを取得していない環境では、パラメータのみ抽出し、マイクロベンチマークを行い、検証を行う必要がある。このようにデータを取得した環境かどうか、既にあるデータを信用するかどうか、プロフィールとして登録してあるか、といった項目が必要である。

3.3.1 プロフィールを取得した環境で利用する場合

プロフィールデータを取得した環境で利用する場合、再実験やスケールアウト、データを抽象化することが挙げられる。すでに同じ環境でプロフィールが作成されている場合は、そのデータを信用するならば、そのままデータを転用し、再実験を行うことができる。また、そのままスケールアウトの実験を行う事もでき、同じミス回避したり、設定を変えた場合のプロフィールなど元々のプロフィールに更にデータを付与することができる。そのため、データを抽象化していくだけで、プロフィールを作成することが可能になる。

3.3.2 プロフィールを取得した環境ではない環境で利用する場合

異なる環境でプロフィールデータを取得した場合は、パラメータのみ抽出、マイクロベンチマーク、検証等を行う。異なる環境でプロフィールが作成されていた場合には、そのデータを全て信用することができない。似ている実験であれば、パラメータのみ抽出し、それ以外は自分自身でもう一度プロフィールを作成する必要がある。プロフィールを作成するには、マイクロベンチマークを行う必要がある、この手順とツールを予め決めておかなければ

ならない。また、それらのデータを比較することで異なる環境間でのプロフィールを検証することができ、実験として似ているかどうかの指標の1つとなる可能性がある。

3.4 プロファイリング方法

実験プロフィールを作成した後は、プロファイリングを行う。プロファイリングとは、過去の実験プロフィールから新しく行う実験のログ収集のための問題を知る事である。プロファイリング方法には、3つの情報が必要である。

1つ目は過去の実験である。まずは、実験プロフィールを作成しなければならない。過去の実験から実験プロフィールで利用するデータを収集しなければならない。しかし、自動的に収集するだけではなく、実験者からどのような設定で行ったかなど詳細を記入してもらう必要がある。

2つ目は、新しく行いたい実験の情報である。プロフィールを参照するための項目を取得する必要があり、実験のための実験が必要である。この実験は、小規模に行う必要があり、この結果からプロフィールに参照を行う。このときの情報はマイクロベンチマークによりデータを収集する必要がある。

3つ目は、検索方法である。過去の実験から新しく行いたい実験で利用できる情報を検索するための検索方法が必要である。検索には、類似度検索を行って過去の実験とやりたい実験を結びつける必要がある。このときの類似度検索には、2種類の検索方法により、過去の実験と新しく行いたい実験を結びつけて行く。

3.5 類似度検索

類似度検索には、ベクトル型検索と木構造による検索の2種類で行う。ベクトル型検索では、プロフィール情報にベクトル情報を付加し、距離を出す事による類似度検索を行う。ベクトル情報をパラメータとして計算式を算出することができる。

また、重要な情報には重みをつけることで計算での重要度を上げる。木構造による検索では、ログの出力レベルや仮想マシンモニタの種類、ハードウェアの情報によって検索を行う。選択肢が少ない情報は予め選択式にしておき、木構造で検索することで検索スピードを上げる。木構造にできるデータとして、物理ノードの台数、仮想化環境の有無、ネットワーク構成、アプリケーションが挙げられる。

3.6 プロフィールの利用手順

一般的な大規模実験施設での実験手順を下記に示す。まず、実験の計画をたて、その実験内容に沿ってレイヤ3の実験ネットワークポロジを決定する。また、実験ネットワークポロジ上のどのノードでプログラムを動かすかを決定する。

次に、実験の規模に合わせてノード、スイッチを借りる。実験計画に合わせて実験ネットワークトポロジのノードに対し、大規模実験施設の物理的なノードを割り当てる。割り当てたノードに IP アドレスを割り当て、それに合わせて、実験ネットワークトポロジ内のサブネットワークに VLAN とサブネットワークアドレスを割り当てる。このとき、ネットワークインタフェース数や帯域、CPU といったノードの性能を考慮しながら的確な割り当てを行う必要がある。実際のノード、レイヤ 2 のスイッチ設定を行う。各ノードに対してマスターノードにスレブノード用の OS イメージやアプリケーション、各スレブノードごとの設定ファイル、実験シナリオファイルを設置する。借用したスレブノードを管理用ネットワークから起動し、起動した OS イメージにネットワークインタフェース、経路制御、名前解決の設定を行う。また、レイヤ 2 スイッチに実験ネットワークトポロジに合わせた VLAN の設定を行う。

設定完了後、実験を実施する。実験計画時にたてたシナリオ終了後実験データの回収を行う。再実験が必要であれば、実験環境を初期化後実験を行う。全ての実験終了後、利用したノード、スイッチの設定を全て初期化し返却する。

実験計画の段階で過去の実験プロファイルが存在するか検索を行う。もし、存在した場合はそのプロファイルを参照し、実験計画に反映する。このとき、実験プロファイルを作成した環境と同じならば実験プロファイルを変更なしに反映することが可能となる。しかし、実験プロファイルを作成した環境と異なる環境で実験を行う場合、マイクロベンチマークによる再検証が必要である。このとき、プロファイル作成に十分ではない場合、必要な情報を追加する必要がある。シナリオが決まった段階で、データの収集方法を決定する。

4. 考 察

今回の論文では、実験プロファイルのあり方について述べた。実際に取得するデータは、top, iostat などのコマンドから得られるが、これらの出力は、単位時間ごとにファイルに出力し結果を得る。出力されたデータは、単位時間ごとの値をその間の値として取得し、グラフ化することとする。コマンドから出力された値は様々な要素を含んでいるため、パースすることにより必要なデータのみを抽出する。利用できるコマンドには、top, iostat, vmstat がある。

一方ログは、汎用的なデータでは無いため、そのままプロファイルデータとして利用する事ができない。ログの出力は、主にメッセージが書かれているため、そのメッセージを記録しておく事にする。また、ログが出力されるスピードは計測しておき、プロファイルにデータとして登録する。メッセージは、プロファイルに登録し、エラーメッセージ検索により、

エラーの回避方法を検索する。そのため、ログは主力されるスピードとメッセージ内容の 2 つをプロファイルとして保持しておく。類似度検索には、良く出るログメッセージと、ログの出力スピードを用いる。

ネットワークトポロジでは、実験で構成したトポロジを把握しなければならない。しかし、基本的にネットワークはスター型リング型バス型の 3 つのタイプに分類される。そこで今回は、これらの 3 タイプ、及びそれらを併用した 7 種類に分けることができると推測する。類似度検索には、木構造での検索の際に重要度を低くして検索を行う。

また、大規模実験施設では、ネットワークの実験を行うため、トラフィックも計測しなければならない。トラフィックを計測するには、スイッチから取得する場合とゲスト OS で取得する方法がある。スイッチから取得した場合には、実験全体のトラフィック量としてプロファイルに登録する。ゲスト OS で取得する場合には、snmp を導入し、ゲスト OS での入出力のトラフィック量を計測する。

仮想化ソフトウェアである Xen では、xentop 等を用いて情報を取得する。このコマンドにより、ゲスト OS の情報を取得することができ、実験中にどのような負荷がかかっているかがわかる。コマンドから出力された情報は抽象化し、プロファイリングに登録する。ゲスト OS のパフォーマンス情報が、実験の特徴となるため、値の変移を中心に値を取得する。

5. 終わりに

本稿では、実験環境でのプロファイル作成のためのデータ収集手法の基礎調査を行った。実験プロファイルに求められるものを明らかにすることでどのようなデータを収集すべきか明確になる。

今後は、実際にデータを取得しながら、プロファイルを作成していきたいと考えている。しかし、大規模実験はたくさんの種類がある訳ではなく、個人でできる量も少ないため、実験を行う人にプロファイルを作成してもらうことが望ましい。また、様々な実験のプロファイルを作成することで、新しく行う実験のためのデータが増えるため、スムーズに実験を行うことが期待できる。

参 考 文 献

- 1) StarBED Project,
available from (<http://starbed.nict.go.jp/>, <http://starbed.org/>)
- 2) Emulab, available from (<http://www.emulab.net/>)

- 3) Xen, available from <http://xen.org/>
- 4) bonnie++, available from <http://www.coker.com.au/bonnie++/>
- 5) IOzone, available from <http://www.iozone.org/>
- 6) Netperf, available from <http://www.netperf.org/netperf/>
- 7) Modeling Virtual Machine Performance: Challenges and Approaches, Omesh Tchoo, et al.
- 8) Diagnosing Performance Overheads in the Xen Virtual Machine Environment, Aravind Menon, et al.
- 9) VCONF: A Reinforcement Learning Approach to Virtual Machines Auto-configuration, Jia Rao, et al.