

仮想化された機器監視アプリケーションの 計測時刻の誤差推定手法

金子 雄^{1,a)} 伊藤 俊夫^{1,b)} 原 隆浩^{2,c)}

概要: ビルや工場の設備機器を遠隔から管理する遠隔ビル管理システムには、ネットワークを介して機器の状態を計測するクローラと呼ばれるソフトウェアが存在する。計算機の仮想化 (VM; Virtual Machine) を利用し、ビルオーナーごとにクローラを実行することで、クローラの障害の影響範囲を限定しつつ、遠隔ビル管理システムに必要な物理マシン数を減らせる。しかし、複数のクローラを単一の物理マシン上で実行すると、クローラが機器の状態を計測する時刻が乱れ、正確な監視制御を行えなくなる可能性がある。そこで我々は、物理マシン、クローラ、VM の動作をモデル化し、モデルに基づくシミュレーションにより、クローラの計測時刻の誤差を推定する手法を提案する。本手法により、物理マシンで実行するクローラの数と、計測時刻の誤差の関係を推定できるため、適切なクローラの実行数を決定できる。シミュレーションによる推定結果を実機評価の結果と比較することで、提案手法の推定精度を評価する。

Estimation Method for Measurement Time Errors of Virtualized Device Status Measurement Applications

YU KANEKO^{1,a)} TOSHIO ITO^{1,b)} TAKAHIRO HARA^{2,c)}

Abstract: In a remote building management system (BMS) that manages facilities of buildings, there is an application called a *crawler* that measures statuses of facilities via a network. By running a crawler on a Virtual Machine (VM) for each building owner, the number of physical machines in the remote BMS can be reduced. However, running multiple crawlers on the same physical machine causes measurement time fluctuation, which may result in inaccurate facility monitoring and controlling. We propose an estimation method for measurement time errors of crawlers according to simulation using a model of crawlers, VMs, and physical machines. The maximum number of crawlers that can be run on the same physical machine can be estimated by the proposed method. In addition, we verify the estimation accuracy of the proposed method by comparing estimated measurement time errors to actual measurement time errors.

1. はじめに

ビルや工場の設備機器を遠隔から管理する遠隔ビル管理システム (BMS; Building Management System) がある [1]。遠隔 BMS は、インターネットなどの Wide Area Net-

work (WAN) を経由して、複数のビルに設置された設備機器を管理する。遠隔 BMS の管理対象は、空調の運転モードや目標温度、照明のオン・オフ状態、室内の温度や湿度や照度などである。以降、これらの管理対象のことを「監視点」、また監視点から得られるデータのことを「監視点データ」と呼ぶ。

遠隔 BMS の機能は通常の BMS と類似しており、例えば以下である。

- 可視化機能: 監視点データを、ビル管理者にわかりやすく表示する機能。最新値を表示したり、値の推移をグラフとして表示したりする。

¹ 株式会社東芝 研究開発センター
Research & Development Center, Toshiba Corporation,
Kanagawa, 212-8582, Japan

² 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University, Osaka, 565-0871, Japan

a) yu1.kaneko@toshiba.co.jp

b) toshio9.ito@toshiba.co.jp

c) hara@ist.osaka-u.ac.jp

- フィードバック制御機能: ある設備機器に対してフィードバック制御を行う機能。例えば、冷水の流量を調節するバルブの開閉度を、ある状態に維持する場合などがある。
- 課金機能: 監視点データから、設備機器の稼働時間を計算し、稼働時間に基づいてテナントに対する課金額を計算する機能。

これらの監視制御機能はいずれも監視点データを必要とする。そのため遠隔 BMS には、ビル群の監視点データを計測し、監視制御機能に提供するための機能が存在する [2][3]。本機能は Supervisory Control And Data Acquisition (SCADA) におけるデータ収集機能に相当する。以降、本機能のことを「クロウラ」と呼ぶ。上述の監視制御機能は、クロウラが計測するデータに基づいて動作するため、クロウラは遠隔 BMS の基礎をなす機能である。

遠隔 BMS が管理する各ビルにはオーナー (ビルオーナー) が存在する。ここで、ビルオーナーごとにクロウラを実行し、あるクロウラに障害が発生したとしても、他のビルオーナー用のクロウラに影響が及ばないようにできることが望ましい。例えば、あるビルのネットワークや機器の異常の影響で、クロウラの動作が遅れたり、停止したりする可能性がある。また、監視点の追加や削除などの設定変更の際に、設定内容や操作の誤りにより、クロウラが停止する可能性がある。このような障害による影響の範囲を制限できることが望ましい。

ビルオーナーごとにクロウラを稼働する単純な構成は、ビルオーナーごとに物理マシンを用意し、クロウラを実行する構成である。この構成は、ビルオーナーが保有するビル数が 1 棟でも物理マシンを用意することになるため、遠隔 BMS の運用者にとって物理マシン用のコストと設置スペースの増加が問題となる。単一の物理マシンかつ単一の Operating System (OS) 上で、複数のクロウラを実行する構成も考えられる。この構成は上記の問題を解決するが、各クロウラは OS が提供するファイルシステムなどの各種機能を共有することになるため、クロウラ間の干渉が生じやすい。例えば、複数のクロウラが同じポート番号を使用して通信を試み、通信に失敗する場合が考えられる。

コストとスペースの問題を解決しつつ、各クロウラの実行環境の独立性を強く保証するための一手法として、ハードウェアの仮想化技術がある [4][5]。ハードウェア仮想化技術は、物理マシン上に、仮想的な CPU や NIC などを備えた Virtual Machine (VM) を作成する。各 VM は個別の OS を実行できる。すなわち、個別のファイルシステムや通信アドレス空間などをアプリケーションに提供できる。

遠隔 BMS の運用者としては、個々の物理マシン上に、できるだけ多くのクロウラ/VM を実行したい。そうすることで、遠隔 BMS に必要な物理マシン数を減らし、設備投資や、物理マシン用の電気料金を削減できる。しかし、

VM 上で動作するアプリケーションの性能は、複数 VM の並列実行により低下する [6][7]。そのため、多くのクロウラ/VM を単一の物理マシン上で実行すると、監視点データの計測時刻の誤差 (計測時刻誤差) が大きくなる。前述の監視制御機能の正常な動作のためには、計画された時刻に監視点データを計測する必要がある。

そこで我々は、VM 上で動作するクロウラの計測時刻誤差を推定する手法を提案する。本手法は、物理マシンや VM のモデルに基づくシミュレーションにより、複数のクロウラ/VM を単一の物理マシン上で実行した場合の計測時刻誤差を推定する。遠隔 BMS の運用者は、この推定結果と、計測時刻誤差の許容値にもとづき、単一の物理マシン上で実行できるクロウラ/VM の最大数を得ることができる。

以降、2 章で関連研究を、3 章で遠隔 BMS の概要を述べる。4 章で提案手法を説明する。5 章で提案手法による推定の結果を実機評価の結果と比較し、提案手法の推定精度を評価する。最後に 6 章でまとめる。

2. 関連研究

ハードウェアの仮想化、すなわち VM は、Web サーバや DB サーバなどのアプリケーションの実行環境として利用されている。VM を利用することで、アプリケーションの障害の波及範囲を制限しつつ、物理マシン数を減らし、電気代やスペースを削減できる。また、アプリケーションのバックアップやリストアが容易になる [8]。

VM の性能や安定性の向上に伴い、監視制御システムに適用するための研究も行なわれている [9]。監視制御システムのレガシーなハードウェアを仮想化する研究や [8]、テスト時に活用しようとする研究がある [10]。また、Programmable Logic Controller (PLC) や SCADA などの特定の監視制御アプリケーションを仮想化する研究もある [11][12][13]。

VM と類似する技術として、コンテナ技術がある [14]。コンテナは VM と比べて性能に対するオーバーヘッドが小さいことが知られている [15]。しかし、ライブマイグレーションやスナップショットなどの高可用性に関する機能が充実している VM のほうが、監視制御アプリケーションの仮想化には適していると考えられる [16]。

アプリケーションを VM 上で実行する場合の懸念の一つは、性能である。単一の物理マシン上で動作する複数の VM は、互いに干渉し、性能が低下する [17][18]。そのため、アプリケーション/VM を物理マシンに追加する際に、全てのアプリケーションの性能要件を満たせるかを判断する必要がある。

DeepDive [19] は、VM を配置する前に、VM の負荷を模擬する処理を物理マシン上で実行し、稼働中のアプリケーションの性能が低下するかを確認する手法である。稼働中

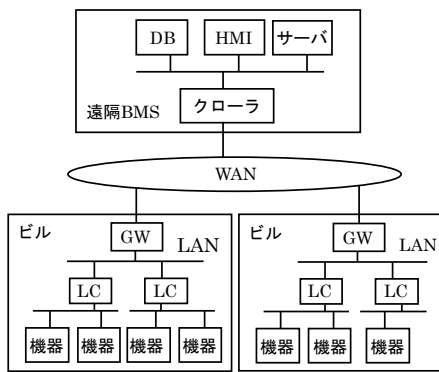


図 1: 遠隔 BMS のシステム構成

Fig. 1 System architecture of remote building system

の物理マシン上で負荷を発生させるため、性能低下の有無を正確に判定できるが、稼働中のアプリケーションの性能に影響を与える可能性が高い。また、物理マシンのリソース (CPU やメモリ) の残量が、追加予定の VM のリソース消費量よりも大きければ、VM を追加する手法も存在する [20][21][22]。これらのリソースに基づく手法は、Web アプリケーションを想定し、物理マシンのリソース利用率を高めることを目標としている。監視制御アプリケーションは性能要件が厳しいため、物理マシンのリソース利用率が低い状態でも、性能要件を満たせない場合がある [18]。したがって、リソースに基づく手法を適用すると、監視制御アプリケーションの性能要件を満たせなくなる可能性が高いと考える。

我々の提案手法は、物理マシンと VM、監視制御アプリケーションの動作をシミュレーションすることで、その性能を推定する。シミュレーションであるため、稼働中のアプリケーションの性能に影響を与えることはない。また、リソースの量ではなく、アプリケーションの性能に基づいて VM の追加可否を判断できる。

3. 遠隔 BMS の概要

本章では、本稿が想定する遠隔 BMS の概要とクローラの性能要件について説明する。

3.1 遠隔 BMS による監視処理

図 1 は、遠隔 BMS によるビル群監視制御のシステム構成である。遠隔 BMS は、クローラの設定や計測した監視点データを保存するデータベース (DB)、監視制御機能が動作するサーバ、ビル管理者用の操作端末である Human Machine Interface (HMI) を備える。見やすさを考慮して、それぞれ 1 つしか描いていないが、実際には複数個が存在しうる。遠隔 BMS とビル群は WAN で接続される。

クローラは、監視点データを取得するため、ビルに設置されたゲートウェイ (GW) に、データ要求を送信する [2]。クローラと GW の間の通信プロトコルは、BACnet/WS

[23] や IEEE1888 [24] などの遠隔監視制御向けプロトコルが使用される [25]。GW は、データ要求を受けると、ローカルコントローラ (LC; Local Controller) にデータ要求を送信する。GW と LC の間の通信プロトコルは、ビル内監視制御向けのプロトコルである BACnet [26] などが使用される。同様に、LC は機器 (監視点) から監視点データを取得し、データ応答として GW に返す。GW はデータ応答をクローラに返し、クローラは監視点データを DB に蓄積する。もしくは、直接、監視制御機能に渡す。

上述のように、本稿ではリクエスト/レスポンス型の通信パターンを想定する。ビル側から遠隔 BMS に対して監視点データを能動的に送信するイベント型の通信パターンも考えられるが、両者は一長一短である。リクエスト/レスポンス型の通信パターンの長所は、いつ、どの監視点データを計測するかという設定情報を集中管理できる点である。これによりビル側の機能を簡素化でき、遠隔 BMS の導入が容易となる。

3.2 クローラの性能要件

クローラは、あらかじめ計画された時刻に監視点データを計測する必要がある。つまり、計画された時刻に、計画された監視点に対して、データ要求を送信する必要がある。なぜなら監視制御機能が、そのように計測された監視点データを必要とするからである。以下にその理由を述べる。

- 可視化機能は、監視点データの推移を折れ線グラフで表示する。監視点データが一定間隔で計測されていれば、ビル管理者にとって見やすいグラフ、すなわち、等間隔で値がプロットされたグラフを表示できる。
- 課金機能は監視点データから機器の稼働時間を計算し、テナントへの課金額を計算する。金銭取引の根拠となるデータであるから、一定間隔で稼働または非稼働を判定することが重要である。
- フィードバック制御機能は、監視点データが一定間隔で計測されないと、正しく機器を制御できない。もしくは、制御の品質が低下する。例えば、制御対象の機器の状態が不安定になる。

許容できる計測時刻の誤差 (計測時刻誤差) は、監視制御機能の種類や、目指す品質に依存する。例えば、課金機能や可視化機能は 1 秒間隔で監視点データを計測する場合があるため、1 秒の誤差は許容できない。またフィードバック制御の対象機器の特性によっては、100 ミリ秒の誤差を許容できない場合がある。

さらに、監視制御システムは 99.999% 以上の確率で正しく動作し続けることが要求される [27]。99.999% の稼働率は、IEC61508 で定義される Safety Integrity Level (SIL) の連続動作モードのレベル 1 に相当する稼働率である。したがって、99.999% 以上の確率で、計測時刻誤差が一定値

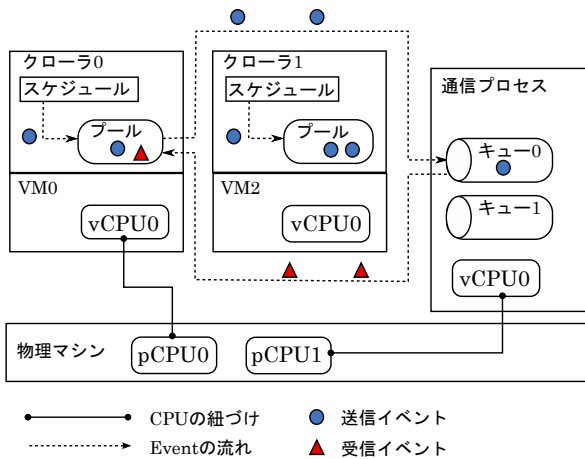


図 2: クローラ/VM と物理マシンのモデル
Fig. 2 Model of Crawler, VM, and Physical Machine

未満であることが、クローラの性能要件である。

3.3 クローラを追加するユースケース

遠隔 BMS では、新たなビルオーナーとビル管理の契約を締結した場合に、クローラを追加する。まず、ビルの GW の IP アドレスや、通信プロトコル、ビルが保有する監視点、監視点データの計測間隔などの情報を DB に登録する。これらの情報は、後でクローラが参照する。次に、クローラを追加する物理マシンを選択する。物理マシンを選択したら、VM を作成し、クローラを実行する。クローラを追加しても、その物理マシン上で動作している全てのクローラの性能要件を満たす必要がある。

契約の締結からクローラが実行されるまでの期間は、数時間～数日であればよい。したがって、仮に複数の契約が同時に発生しても、VM の作成やクローラの実行は逐次的に行える。また、一度実行されたクローラは年単位で稼働し続けることになる。頻繁に停止したり再起動したりすることはない。

4. 提案手法

提案手法は、クローラや VM、物理マシンのモデルに基づくシミュレーションにより、計測時刻誤差を推定する。

4.1 クローラ/VM および物理マシンのモデル化

提案手法における、物理マシンと VM、また、VM 上で動くクローラのモデルについて説明する。

4.1.1 モデルの対象

図 2 にモデルを示す。モデル化の対象は以下である。

- 物理マシン: 1 個以上の物理 CPU (pCPU) を持つ。クローラの処理は CPU バウンドであるため、メモリやストレージなどは考慮しない。
- VM: 1 個以上の仮想 CPU (vCPU) を持つ。vCPU の状態遷移は次項で説明する。

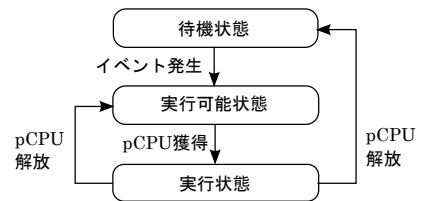


図 3: 仮想 CPU (vCPU) の状態遷移
Fig. 3 State machine of virtual CPU

- クローラ: 監視点データを計測するスケジュール (計測スケジュール) を持つ。計測処理は、「送信イベント」と「受信イベント」の 2 種類のイベントとしてモデル化する。クローラは、イベントを保持するイベントプールも持つ。
- 送信イベント: ある監視点に対して送信するデータ要求の作成や、そのデータ要求を送信する処理に相当する。処理を完了するために必要な時間 T_{send} をパラメータとして保持する。
- 受信イベント: 監視点データを含むデータ応答を受信する処理や、監視点データを解析する処理に相当する。処理を完了するために必要な時間 T_{recv} をパラメータとして保持する。
- 通信プロセス: 物理的な NIC を使用し、別の物理マシンとの通信を実際に行うソフトウェア。例えば、Xen であれば Dom0 に、KVM であればホスト OS に相当する [5][4]。通信プロセスは VM ごとにイベントキューを持つ。また、1 個以上の仮想 CPU (vCPU) を持つ。文献 [13] などで、通信プロセスに pCPU を占有させることで性能が向上することが示されているため、本モデルでは、通信プロセスに一部の pCPU を占有させることを前提とする。

4.1.2 仮想 CPU (vCPU) の状態遷移

図 3 は vCPU の状態遷移図である。vCPU の初期状態は「待機状態」である。

1 つ以上の送信イベントまたは受信イベントがイベントプールにあるとき、そのクローラを実行する VM の vCPU は、「実行可能状態」となる。pCPU は、実行可能状態である vCPU をランダムで 1 つ選ぶ。pCPU に選ばれた vCPU は、タイムスライスが割り当てられ、「実行状態」となる。タイムスライスとは、pCPU を解放することなく使用し続けられる時間である。通信プロセスの vCPU は、pCPU を占有できるため、常に実行可能状態か実行状態である。

実行状態となった VM の vCPU は、イベントプールにあるイベントをランダムに選択し、処理する。vCPU は、自身に割り当てられたタイムスライスを消費し、消費した分だけ、イベントの T_{send} や T_{recv} を減らす。pCPU に選ばれなかった vCPU は、イベントプールに蓄積されているイベントを処理できない。いずれかの pCPU が解放され、次に自身が選ばれることを待つ必要がある。

pCPU に選ばれた vCPU は、以下のいずれかの条件を満たした場合に、pCPU を解放する。

- 自身に割当てられたタイムスライスを使い切った場合
- イベントプールにイベントが無くなった場合

pCPU を解放した vCPU は、イベントプールにイベントがあるなら実行可能状態に、イベントがないなら待機状態に遷移する。解放された pCPU は、再び、実行可能状態の vCPU を選ぶ。

4.1.3 イベントの処理の流れ

クローラが監視点データを計測すべき時刻になると、送信イベントが作成され、イベントプールに挿入される。vCPU により処理され、 T_{send} が 0 以下になった送信イベントは、通信プロセスが持つイベントキューに挿入される。イベントキューにおける送信イベントは、処理を完了するまでの時間 T_{cp} が付与され、通信プロセスの vCPU により処理される。処理が完了した送信イベントは、イベントキューから削除される。その後、ある時間 T_{net} が経過した後に、受信イベントがクローラのイベントプールに追加される。 T_{net} は、監視点データを計測する際の応答遅延に相当する。イベントプールに追加された受信イベントは、VM の vCPU に処理される。 T_{recv} が 0 以下になった受信イベントは、イベントプールから削除される。

4.2 シミュレータによる計測時刻誤差の推定

4.1 節に記載したモデルにもとづいてシミュレータを実装する。シミュレーションにおける時刻は離散的に進む。シミュレータの入力パラメータを以下に記す。

- 物理マシンの物理 CPU コア数
- クローラ/VM の数
- 各 VM の vCPU 数
- 各クローラの計測スケジュール
- 各種イベントの処理に必要な時間
- タイムスライスの初期値
- シミュレーション期間
- シミュレーションの単位時間

これらの入力に対して、本シミュレータはクローラ/VM の動作をシミュレーションする。ある送信イベントがイベントプールに挿入された時刻と、その送信イベントの T_{send} が 0 以下になった時刻の差を、計測時刻誤差とする。

4.3 事前評価によるパラメータの決定

4.2 節で述べたシミュレータの入力パラメータのうち、「各 VM の vCPU 数」と、「各種イベントの処理に必要な時間」は、事前評価に基づいて値を決定する。

事前評価では、実機を用いて、任意の計測スケジュールに基づいてクローラを稼働させる。クローラを稼働させる時間を D とする。事前評価において、データ要求の送信先は、実際のビルの GW でなくてよい。例えば、評価用の模

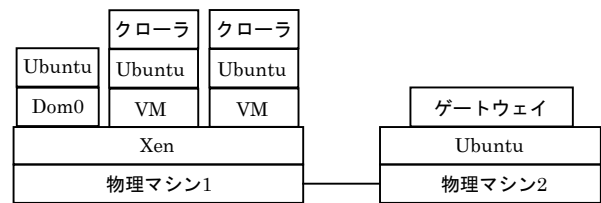


図 4: 実機評価の環境

Fig. 4 Experimental environment using real machines

擬 GW を用意してもよい。評価中、クローラを実行する VM の CPU 使用率を計測する。また、 D の間に、VM の CPU 使用時間 U_{vm} と通信プロセスの CPU 使用時間 U_{cp} を計測する。CPU 使用率や CPU 使用時間は、仮想化のソフトウェアが備えているコマンドを使用することで計測できる。例えば Xen [5] であれば `xentop` コマンドや `xl` コマンドを使用すればよい。

計測スケジュールにて定義されている「計測対象の監視点」と、「監視点ごとの計測時刻」から、データ要求の送信頻度 (rps; Request per Second) を計算できる。したがって、いくつかの計測スケジュールに対して上記の評価を実施することで、rps と平均 CPU 使用率の組み合わせのデータを複数得られる。これに回帰分析などを適用することで、rps と平均 CPU 使用率 E の関係を定式化する (式 1)。

$$E = f(rps) \quad (1)$$

式 1 を用いることで、rps, すなわち計測スケジュールから平均 CPU 使用率 E を計算できる。そして、以下の式 2 を用いることで、VM に割当てる vCPU の数 N_{vcpu} を決定する。

$$N_{vcpu} = \lceil \frac{E}{100} \rceil \quad (2)$$

例えば $E = 150$ であれば、 $N_{vcpu} = 2$ となる。

また、評価期間中に計測したデータ数 ($rps \times D$) と U_{vm} から、VM が一回の監視点データを計測する際に消費する CPU 時間 $T_{measure}$ を求める (式 3)。

$$T_{measure} = \frac{U_{vm}}{rps \times D} \quad (3)$$

提案手法のモデルでは、監視点データの計測処理は送信イベントと受信イベントに分割される。そこで、 $T_{send} = T_{recv} = \frac{T_{measure}}{2}$ として T_{send} と T_{recv} を決定する。

同様に、通信プロセスが送信イベントを処理する時間 T_{cp} は、以下の式 4 により決定する。

$$T_{cp} = \frac{U_{cp}}{rps \times D} \quad (4)$$

5. 評価

5.1 実機評価の環境

実機評価の環境を図 4 に示す。評価には 2 つの物理マシ

ンを用いる。1つはクローラを実行するために、もう1つはビルのGWと監視点を模擬するために用いる。2つの物理マシンは1000BASE-Tで接続する。2つの物理マシン間の時刻同期にはNTPを使用する。

クローラ用の物理マシンはIntel(R) Xeon(R) 1.80GHz CPU (8コア), 32GBメモリを備える。仮想メモリアドレスと物理メモリアドレスの変換処理を効率化するExtended Page Tables (EPT) 機能は有効化する。ビルGW用の物理マシンはIntel(R) Xeon(R) 2.60GHz CPU (32コア), 80GBメモリを備える。物理マシンとVMのOSとしてUbuntu 16.04 LTS (Linux 4.4.0)を使用する。

仮想化のソフトウェアとして、Xen [5]を採用する。Xenは、動作の安定性とメンテナンス性を重視した設計となっており、監視制御アプリケーションに適していると考えられる[28]。バージョン4.6.0のXenを使用し、デフォルトのスケジューラであるcreditスケジューラを使用する。creditスケジューラのタイムスライス、デフォルト値である30ミリ秒を使用する。また、CPU affinity機能を使用し、Dom0にpCPUを1つ占有させる。完全仮想化よりも準仮想化のほうが性能が高いため、仮想化の種類は準仮想化を使用する[18]。

クローラはC言語で実装する。クローラは、各監視点の計測周期を考慮して計測スケジュールを作成する[2]。そして、計測スケジュールに基づいてデータ要求を送る。データ要求を送信した時刻を計測時刻とする。各監視点の計測周期は全て1秒とする。データ要求にはBACnet/WS [23]のgetValuesリクエストを使う。BACnet/WSはHTTPをベースとする通信プロトコルである。本来であれば、クローラは計測した監視点データをDBや、監視制御機能に渡す。しかし、DBの性能に起因するクローラの計測時刻への影響を除外するため、本評価で用いるクローラは、計測したデータを破棄する。ビルのGWは、Apache HTTPサーバを用いて模擬する。Apache HTTPサーバはバージョン2.4.18を使用する。

5.2 事前評価の結果

シミュレーションで使用するパラメータを決定するために、実機を用いた事前評価を実施した。事前評価では、1つのクローラに100rps, 200rps, 300rpsの計測スケジュールを与え、それぞれ、10分間動作させた。

この評価で得られたCPU使用率のデータから、rpsと平均CPU使用率の関係(式1)を、rpsを変数とする一次式として近似した。

$$E = 0.127 \times rps + 20.779 \quad (5)$$

この式を用いて、クローラのrpsから必要なvCPU数を決定する。

また、本評価の結果から、式3を計算したところ、

表 1: 実機評価のパラメータ

Table 1 Parameters of evaluation with real machines

パラメータ	値
クローラ/VMの数	1から20
クローラの追加間隔(分)	10
各クローラの監視頻度(rps)	50,100,150,200,250,300
各VMのvCPU数	1
クローラが使用できる物理CPUの数	7
監視頻度ごとの試行回数	5

表 2: シミュレーションのパラメータ

Table 2 Parameters of simulation

パラメータ	値
クローラ/VMの数	1から20
各VMのvCPU数	1
計測スケジュール	実機評価と同じもの
各種イベントの処理に必要な時間	事前評価から決定
クローラが使用できる物理CPUの数	7
タイムスライスの初期値(ミリ秒)	30
シミュレーション期間(秒)	5
シミュレーションの単位時間(マイクロ秒)	1
監視頻度ごとの試行回数	5

$T_{measure} = 686050$ ナノ秒となった。そのため、 $T_{send} = T_{recv} = 343025$ ナノ秒とする。また、式4を計算したところ、 $T_{comm} = 233333$ となった。

5.3 実機評価とシミュレーション評価の比較

本節では、実機評価で得られた計測間隔誤差と、シミュレータにより推定した計測時刻誤差を比較した結果について述べる。

実機評価のパラメータを表1に示す。クローラは10分間隔で追加され、追加されたクローラは、監視点データの計測スケジュールに基づいて、計測処理を開始する。クローラごとのrpsは50から300としたため、式5と式2から、各VMのvCPU数は1個となる。物理マシンのCPU数は8個で、そのうちの1つはDom0が占有する設定のため、クローラ群が使用できるpCPUの数は7個である。シミュレーションのパラメータは表2に示す。タイムスライスの初期値は、Xenのcreditスケジューラの設定値に合わせて30ミリ秒とする。

図5に、実測値と推定値の比較結果を示す。rpsが増加するに従い、計測時刻誤差の実測値も増加していることがわかる。50rpsの場合は、実測値と推定値の差は小さく、1ミリ秒程度である(図5a)。しかし、実測値の計測誤差は19VMまでほぼ一定だが、推定値の計測誤差は8VMから線形増加している。100rpsの場合も、実測値は16VMまでほぼ一定だが、推定値の計測誤差は8VMから増加し始

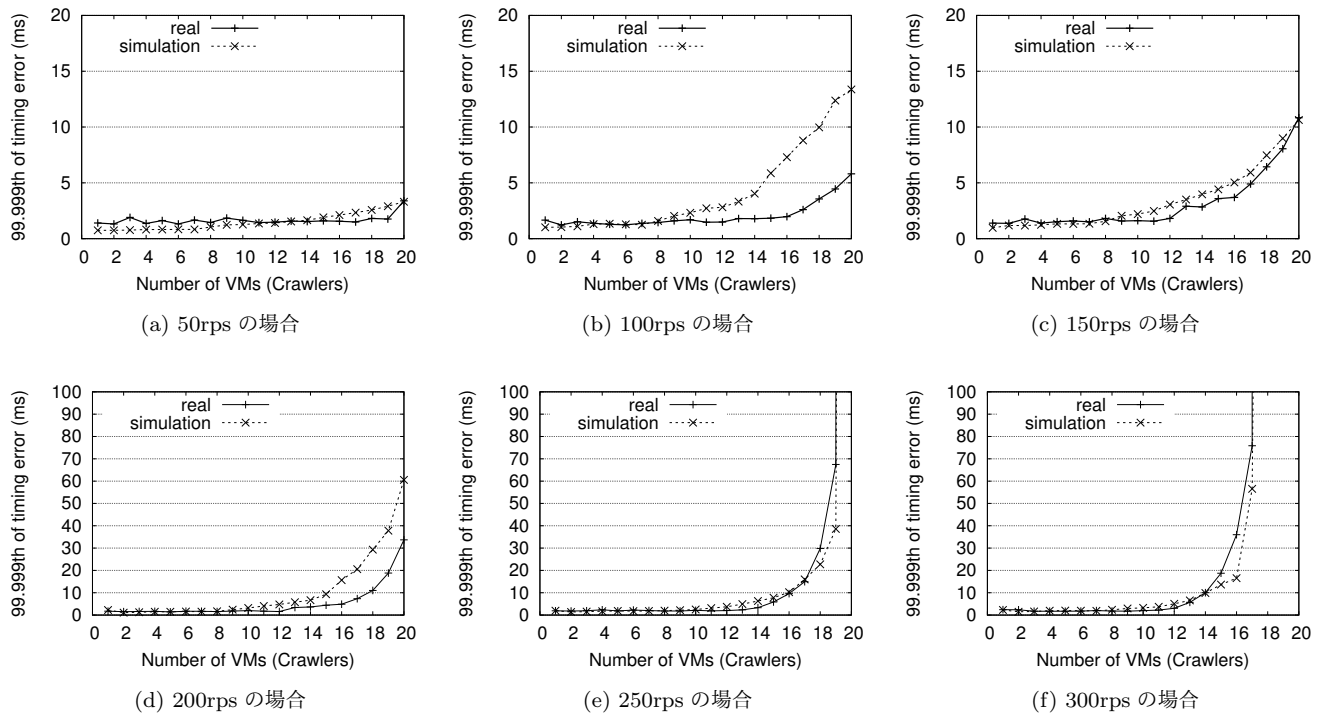


図 5: 計測時刻誤差の 99.999th 値の実測値と推定値の比較

Fig. 5 Comparison of 99.999th of measurement time error between actual values and estimated values

めている。つまり、提案手法のモデルは、実機よりも CPU 競合が発生しやすいモデルになっていると言える。4.1.2 項で述べたように、提案手法のモデルにおける vCPU は、タイムスライスを使い切るか、処理すべきイベントがなくなるまで pCPU を解放しない。しかし実機の場合、送信処理や受信処理の途中に、一時的に pCPU を解放するが発生している可能性がある。この点は実機とモデルの擦り合わせが必要である。

150rps の場合も同様に、推定値の計測誤差は 8VM から増加しているが、実測値と推定値の差は小さく、2 ミリ秒程度である (図 5c)。15VM 以降において、100rps の場合よりも推定値が小さい原因については、実装したシミュレータにバグがある可能性もあり、調査が必要である。

200rps の場合は、15VM から実測値と推定値の差が広がり、約 30 ミリ秒となった (図 5d)。15VM から、推定値の増加率が増したためである。一方、250rps と 300rps では、途中から実測値のほうが推定値よりも大きくなった (図 5e, 図 5f)。実機の場合、rps が大きくなると、VM 数が増えた場合の誤差の増加率が大きくなるが、この傾向をシミュレーションで再現できていないと考えられ、改善が必要である。

250rps では 20VM で、300rps では 18VM で、実測値と推定値の計測時刻誤差はどちらも 1 秒を超えた。これはクローラ群の総処理量が、物理マシンの処理性能を超えたためであり、遠隔 BMS の監視制御機能として許容できない

誤差である。提案手法のモデルとシミュレータは、その状況を適切に再現できたとと言える。

本評価の結果を以下にまとめる。

- (1) 推定値の計測誤差は 8VM から増加するが、実測値の計測誤差はさらに VM 数が増えてから増加する。
- (2) 15VM 以降の実測値の増加率は、推定値の増加率よりも大きい。
- (3) 実測値と推定値の誤差は、最大で約 30 ミリ秒である。
- (4) クローラ群の総処理負荷が物理マシンの処理性能を超える状況は、推定できている。

上記の 1, 2, 3 は改善すべき点であり、今後の課題とする。

6. まとめと今後の課題

監視制御アプリケーションの 1 つであるクローラを想定し、VM 上で動作するクローラの計測時刻の誤差を推定する手法を提案した。実機を用いた評価結果と、提案手法による推定結果を比較した結果、VM 数の増加に対する計測時刻誤差の増加率を、正確に再現できていないことがわかった。実測値と推定値の誤差は最大で約 30 ミリ秒であった。一方、クローラ群の総処理負荷が物理マシンの処理性能を超える状況は、推定できた。

今後は、CPU 競合のモデルの見直しや、VM 数が増加した場合のオーバーヘッドを考慮することで、計測時刻誤差の推定精度の向上を図る。また、推定結果に基づいてクローラ/VM を物理マシンに配置する VM 管理ソフトウェ

アを試作し、関連研究と比較する。

参考文献

- [1] Chaiboonruang, P., Pora, W., Ochiai, H. and Esaki, H.: Small buildings energy management system based on IEEE1888 standard with data compression, *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference on*, pp. 1–6 (2014).
- [2] 伊藤俊夫, 米良恵介, 金子 雄, 松澤茂雄: 通信エンドの負荷ピークを低減するためのビル設備情報収集スケジューリング作成方法, 電子情報通信学会技術研究報告. IN, 情報ネットワーク, Vol. 111, No. 197, pp. 77–82 (2011).
- [3] Suresh, K., Kirubashankar, R. and Krishnamurthy, K.: Research of Internet based supervisory control and information system, *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pp. 1180–1185 (2011).
- [4] Kivity, A., Kamay, Y., Laor, D., Lublin, U. and Liguori, A.: kvm: the Linux Virtual Machine Monitor, *Proceedings of the Linux Symposium*, pp. 225–230 (2007).
- [5] 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, *SIGOPS Oper. Syst. Rev.*, Vol. 37, No. 5, pp. 164–177 (2003).
- [6] Apparao, P., Makineni, S. and Newell, D.: Characterization of network processing overheads in Xen, *Virtualization Technology in Distributed Computing, 2006. VTDC 2006. First International Workshop on*, pp. 2–2 (2006).
- [7] Koh, Y., Knauerhase, R., Brett, P., Bowman, M., Wen, Z. and Pu, C.: An Analysis of Performance Interference Effects in Virtual Environments, *Performance Analysis of Systems Software, 2007. ISPASS 2007. IEEE International Symposium on*, pp. 200–209 (2007).
- [8] Breivold, H., Jansen, A., Sandstrom, K. and Crnkovic, I.: Virtualize for Architecture Sustainability in Industrial Automation, *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, pp. 409–415 (2013).
- [9] Givehchi, O., Trsek, H. and Jasperneite, J.: Cloud computing for industrial automation systems - A comprehensive overview, *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pp. 1–4 (2013).
- [10] Breivold, H. and Sandstrom, K.: Virtualize for test environment in industrial automation, *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pp. 1–8 (2014).
- [11] Givehchi, O., Imtiaz, J., Trsek, H. and Jasperneite, J.: Control-as-a-service from the cloud: A case study for using virtualized PLCs, *Factory Communication Systems (WFCS), 2014 10th IEEE Workshop on*, pp. 1–4 (2014).
- [12] Shahzad, A., Musa, S., Aborujilah, A. and Irfan, M.: A Performance Approach: SCADA System Implementation within Cloud Computing Environment, *Advanced Computer Science Applications and Technologies (AC-SAT), 2013 International Conference on*, pp. 274–277 (2013).
- [13] Mahmud, N., Sandstrom, K. and Vulgarakis, A.: Evaluating industrial applicability of virtualization on a distributed multicore platform, *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pp. 1–8 (2014).
- [14] Dua, R., Raja, A. and Kakadia, D.: Virtualization vs Containerization to Support PaaS, *Cloud Engineering (IC2E), 2014 IEEE International Conference on*, pp. 610–614 (2014).
- [15] Felter, W., Ferreira, A., Rajamony, R. and Rubio, J.: An updated performance comparison of virtual machines and Linux containers, *Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium on*, pp. 171–172 (2015).
- [16] Li, W. and Kanso, A.: Comparing Containers versus Virtual Machines for Achieving High Availability, *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pp. 353–358 (2015).
- [17] Pu, X., Liu, L., Mei, Y., Sivathanu, S., Koh, Y., Pu, C. and Cao, Y.: Who Is Your Neighbor: Net I/O Performance Interference in Virtualized Clouds, *Services Computing, IEEE Transactions on*, Vol. 6, No. 3, pp. 314–329 (2013).
- [18] Kaneko, Y., Ito, T. and Hara, T.: A measurement study on virtualization overhead for applications of industrial automation systems, *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8 (2016).
- [19] Novaković, D., Vasić, N., Novaković, S., Kostić, D. and Bianchini, R.: DeepDive: Transparently Identifying and Managing Performance Interference in Virtualized Environments, *Proceedings of the 2013 USENIX Conference on Annual Technical Conference*, USENIX ATC'13, Berkeley, CA, USA, USENIX Association, pp. 219–230 (2013).
- [20] Zhou, W., Yang, S., Fang, J., Niu, X. and Song, H.: VM-CTune: A Load Balancing Scheme for Virtual Machine Cluster Using Dynamic Resource Allocation, *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*, pp. 81–86 (2010).
- [21] Tomás, L. and Tordsson, J.: Improving Cloud Infrastructure Utilization Through Overbooking, *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*, CAC '13, New York, NY, USA, ACM, pp. 5:1–5:10 (2013).
- [22] Meng, X., Isci, C., Kephart, J., Zhang, L., Bouillet, E. and Pendarakis, D.: Efficient Resource Provisioning in Compute Clouds via VM Multiplexing, *Proceedings of the 7th International Conference on Autonomic Computing*, ICAC '10, New York, NY, USA, ACM, pp. 11–20 (2010).
- [23] ASHRAE: Addendum c to ANSI/ASHRAE Standard 135-2004 - BACnet/WS (2006).
- [24] IEEE: IEEE Standard for Ubiquitous Green Community Control Network Protocol, IEEE1888-2011 (2011).
- [25] Schachinger, D., Stampfel, C. and Kastner, W.: Interoperable integration of building automation systems using RESTful BACnet Web services, *Industrial Electronics Society, IECON 2015 - 41st Annual Conference of the IEEE*, pp. 003899–003904 (2015).
- [26] ASHRAE: Annex J to ANSI/ASHRAE 135-1995 - BACnet/IP (1999).
- [27] Grossman, E., Gunther, C., Thubert, P., Wetterwald, P., Raymond, J., Korhonen, J., Kaneko, Y., Das, S. and Zha, Y.: Deterministic Networking Use Cases, Internet Draft (2015).
- [28] Fraser, K., Hand, S., Neugebauer, R., Pratt, I., Warfield, A., Warfield, A., Williamson, M. and Williamson, M.: Reconstructing I/O, Technical Report UCAM-CL-TR-596 (2004).