

令和 6 年 6 月 11 日現在

機関番号：33907

研究種目：基盤研究(C)（一般）

研究期間：2021～2023

課題番号：21K11878

研究課題名（和文）仮想環境を使ったネットワークコンピューティングシステム性能評価法の研究

研究課題名（英文）Study of evaluation method for network computing performance on virtual environment

研究代表者

君山 博之（Kimiya, Hiroyuki）

大同大学・情報学部・教授

研究者番号：10757644

交付決定額（研究期間全体）：（直接経費） 3,200,000円

研究成果の概要（和文）：VMの性能記述モデルの確立を目的に、疑似的なクラウド環境で複数VMを同時に動作させて、その処理時間を計測する方法を検討した。評価のための静止画像の線形補間による拡大プログラムを開発し、複数のVMでそのプログラムを同時に実行させ、その処理時間を収集できる環境を構築した。その環境を使いプログラムを同時に実行させるVM数を1から30まで変えながら、VM毎の処理時間を計測し集計を行った。その結果、CPUコア数を使い切るまではメモリ競合により線形に平均処理時間が増加し、CPUコアを使い切るとCPUコアの空き待ちにより大きな傾きで増加することが確認でき、VM性能を線形モデルにより記述できることを確認した。

研究成果の学術的意義や社会的意義

本研究では、多くのICTシステムで利用されているVMの性能について、他のVMの処理がどのような影響を及ぼすのかを明らかにし、それを数式化することによって、システム構築前に予めVMの性能を予測するための技術の確立を目指している。本成果を使うことによって、複数のVMを使ったシステムを構築する際に、必要な数のVMだけを組み合わせることを可能にし、費用・エネルギー効率の良いシステムや、デッドラインが定まっているリアルタイム処理システム的设计・構築を可能にするものである。

研究成果の概要（英文）：To establish a model for describing VM performance, we researched a method for measuring the processing time of multiple VMs simultaneously running in a pseudo cloud environment. For the evaluation, we developed a program for enlarging still images using linear interpolation, and constructed an environment in which we could run the program simultaneously on multiple VMs and collect the processing time. Using this environment, we measured and compiled the processing time for each VM while varying the number of VMs that simultaneously executed the program from 1 to 30. As a result, we confirmed that the average processing time increases linearly due to memory contention until the number of CPU cores is exhausted, and once the CPU cores are exhausted, the processing time increases at a steeper rate due to waiting for free CPU cores, and we confirmed that VM performance can be described by a linear model.

研究分野：並列分散リアルタイム処理

キーワード：仮想環境 仮想マシン 並列分散処理 リアルタイム処理

1. 研究開始当初の背景

センサーや監視カメラなどの **Internet of Things (IoT)** の普及に伴い、爆発的に増加しつつあるデジタルデータを分析して業務や問題解決に生かすビッグデータ分析が盛んに行われている。このビッグデータ分析に利用されるのは、主にクラウド上の仮想計算機 (**Virtual Machine: VM**) である。VM は1台の計算機の CPU、メモリ、ストレージや計算機内のバス、計算機外部へつながるネットワークを含む全てのハードウェアリソースを複数の VM と共有して利用していることから、同じ物理計算機上で動作する他の VM の負荷状態によって VM の処理能力が変動することが、申請者や D. Novakovic らの研究によって確認されている。

複数の VM を使った分散型データ分析フレームワークとして、**Apache Spark**、**Apache Kafka**、**Google** の **Istio** などが利用されている。これらのフレームワークを利用する場合、前述した VM 間のリソース共有に起因する性能変動や転送速度変動により、システム全体性能を高い精度での事前評価が難しいことから、最適なリソース設計ができず、ベストエフォートでの処理とするか、必要数を超える大量の VM をデッドラインに間に合わせるために導入し、処理を行っているのが現状である。

一方で、第5世代 (5G) 携帯電話網の網内設備を **Network Function Virtualization (NFV)** のフレームワークを使って実装する試みが始まっている。これは汎用計算機上に構築した VM 上に、5G 携帯電話網の網内設備機能 (**Virtual Network Function: VNF**) を機能単位で実装し、それを VM 間の仮想ネットワークを使って結合することによって網内設備を実現するものであるが、前述したデータ分析と同様に事前の性能評価が難しいため最適なリソース設計ができない問題が発生する。そのため、使用するハードウェアを実績のある機種に限定するとともに、実装経験にもとづいて必要な VM 数や仮想ネットワークの帯域を決定しているのが現状である。

このように汎用のハードウェアで動作させることが可能であるにも関わらず、決まった機種のハードウェアしか利用することができないこと、また、VM の配置や動作させるソフトウェアも、動作実績のある組み合わせしか利用できないことが、システム構築の制約条件となっている。NFV のメリットをさらに享受するには、その規模に応じた汎用のハードウェアを使い、リソースを限界まで利用できるような VM の配置、動作させるアプリケーションや VNF の配置、仮想ネットワークと実ネットワークの接続方法や帯域などを最適に設計し、システムを構築する必要がある。そのためには、システムの構築前に利用可能な、複数 VM を使ったシステムに対する精度の高い性能評価法が不可欠であり、その評価方法の構築が望まれている。

2. 研究の目的

申請者は、複数クラウド上の VM を複数使い、様々な処理を実装可能な並列分散リアルタイム処理フレームワークを提案するとともに、このフレームワークを使った映像合成処理システムを実現し、その有効性を確認した。さらに、そのフレームワークの一部として、VM の性能変動をモデル化し、そのモデルを使った VM 同士のハードウェアリソース共有を考慮したシステム性能の事前評価法を提案し、前述の映像合成処理システムに対して、5%以内の誤差で全体性能を予測できることを実証した。一方で、この性能予測手法は、提案した分散処理方式に対して有効性を実証しただけで、任意のシステムに適用可能な汎用的な性能評価法に発展させることが課題となっている。

VM を使ったシステムの性能評価法に関する先行研究では、例えば R. C. Chiang らは運用中の VM 性能を観測し、**Support Vector** による機械学習を使い VM 追加時の性能予測する方法を提案している。これらの手法は因果関係に基づく予測ではなく、統計情報による推定であり、機械学習により生成した判定器を、素性の異なるアプリケーションが動作する VM に適用するのは困難である。本研究では、経験によらずに必要な VM 数などのシステム構成を事前に決定することを目的に、実測値にもとづいてリソース共有時の VM の性能変動モデルを構築し、そのモデルを使ったシミュレーションによって様々なシステム構成に対して性能評価が可能な評価方法を確立することが、本研究の目的である。

3. 研究の方法

本研究では、最初の2年間で上記の Step 1 と Step 2 を行い、Step 3 を最終年度に実施しシステム性能評価法を確立する。各 Step における具体的な進め方を以下に示す。

Step 1. 実測による汎用計算機上の VM におけるリソース競合時の性能評価

この Step では複数の性能の異なる汎用 PC サーバ機およびネットワークスイッチを導入し、疑似的なクラウド環境を構築し、KVM をはじめとする様々な Hypervisor を使って仮想環境を構築し、その上で VM を複数起動する。その VM 上でネットワークも含めたハードウェアリソースを使用するプログラムを動作させて、同時に動作する VM 数によってどのように性能が変動するかを実測する。さらに、**Apache Kafka** などの分散処理アプリケーションを実際に動かし、その性能が同時に動作する VM 数によってどのように変動するかを詳細に実測し、その結果について考察を行う。

Step 2. 汎用計算機上の VM におけるリソース競合時の処理性能モデルの検討

Step 1 で実測したデータを分析し、その傾向を確認するとともに、各ハードウェアリソースに対する競合処理がどのような処理になっているかを推定して仮説を立てる。その仮説と合致する処理性能記述モデルを、待ち行列理論等を用いて確立し、VM 数の関数として数式化する。この確立したモデルを用いて、性能の異なる汎用 PC サーバ機においてもリソース競合時の処理性能が再現できることを確認する。

Step 3. システム全体性能評価手法の確立

申請者が提案しているシミュレーション法をベースに、Step 2 で構築したモデルを組み込んでシステム全体性能を評価する手法を検討する。情報通信研究機構 (NICT) の管理運営する北陸 StarBED センター内の計算機環境を用いて、前述した Kafka または Istio を使った分散アプリケーションを実際に行い、性能評価を実施する。この Step で検討したシミュレーション法を用いて、その分散アプリケーションをシミュレートし、実測値との比較によりシミュレーション法の有効性を確認する。さらに、国立情報学研究所 (NII) が管理運営する SINET5 上の NFV 環境を用いて、パケットフィルタやパケットロギングなどの VNF を動作させ、処理性能を実測し、シミュレーションとの比較により、シミュレーション法の有効性を確認することによって、性能評価法の確立を目指す。

4. 研究成果

本研究では、画像を任意の大きさに拡大縮小するプログラムを独自に開発し、そのプログラムを複数の VM で同時に実行するための環境を自ら用意し、VM ごとの処理時間を計測し、集計する方法を検討した。検討の結果、Shell script および SSH を用いて、複数の VM に分散処理をさせて、その処理時間の集計をする方法が、様々な構成に簡単に対応出来ることが判ったことから、この手法を用いて評価実験を行った。

図 1 に、評価実験システムの構成を示す。図 1 中の左側の PC サーバにて複数の VM を動作させ、右側の PC サーバから SSH を用いて同時に画像拡大プログラムを実行し、そのときの各 VM での画像拡大処理時間を記録した。なお、この実験で用いた PC サーバのスペックを表 1 に示す。VM は、多くの環境で利用されている KVM を用いて作成した。

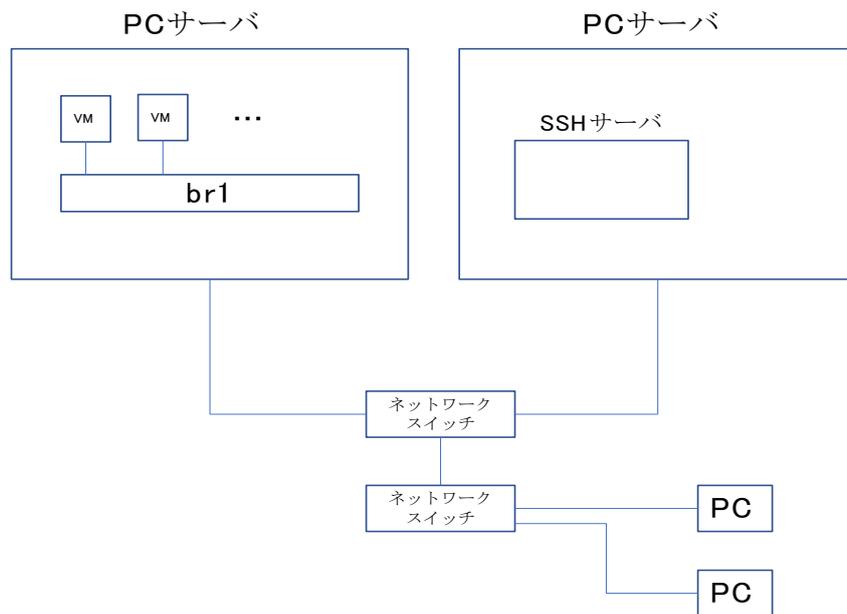


図 1 評価実験システム構成

表 1 PC サーバのスペック

ハードウェア	CPU	Intel Xeon Silver 4316 (2.30GHz) ×2 (20 コア)
	Memory	128 GByte (各 CPU に 64 GByte 割当)
	SSD	240 GByte+240 GByte +1 TByte (KVM 用 PC サーバのみ)
	Network Interface Card (100 Gbps Ethernet 用)	Mellanox ConnectX-5 (100 GbE)
	Network Interface Card (Internet 接続用)	Intel X710 10 Gigabit ethernet (10 GbE) オンボード
ソフトウェア	OS	Ubuntu 22.04.3 LTS server

図2に、1~30のVMを同時に動作させて画像拡大プログラムを動作させたときの平均処理時間を示す。この実験では、線形補間に16×16画素から16,384×16,384までの拡大処理を実行させている。この平均処理時間は、各VM数に対して5回の測定を行った平均処理時間であり、さらにVM間の平均処理時間でもある。図2から判るように、同時に動作させるVM数に対して比例して処理時間が増加することが判った。

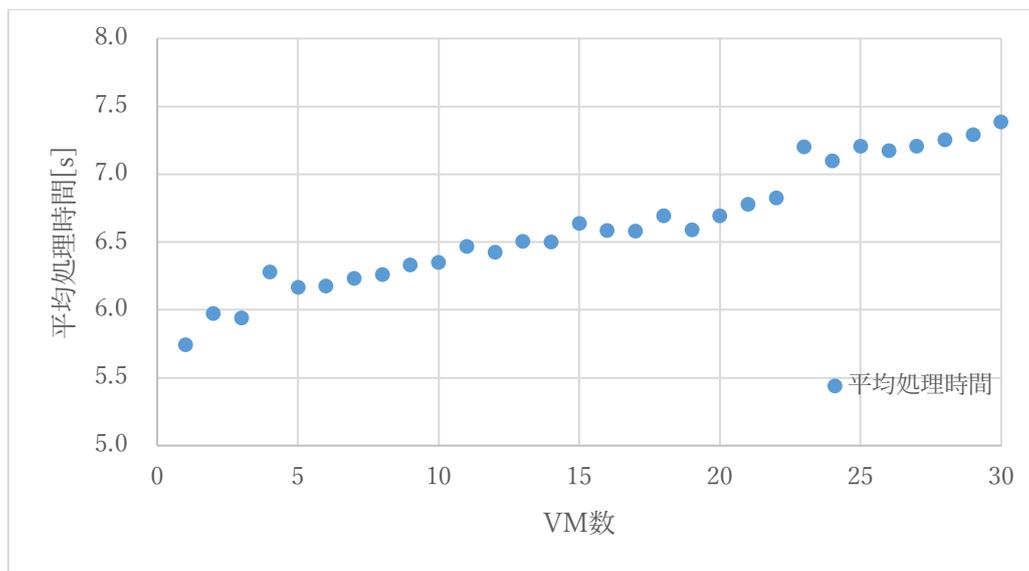


図2 VM数の増加における平均処理時間の変化

この平均処理時間のうちVM数が5~20までの値を最小二乗法で線形回帰させた。その結果、求めた直線は(1)式となり、その直線を図2にオーバーラップさせたものを図3に示す。

$$y = 0.0362x + 5.994 \quad (1)$$

この図から判るように、平均処理時間はVM数に対して、ほぼ直線上にのることが判った。その理由として、VM同士でのメモリアクセス待ちが考えられる。この実験で使用したPCサーバは表1に示すように、CPUコアが40コア存在する。各VMには2コアずつを割り当てたので、20のVMは同時に処理ができるが一方で、メモリは各CPUに対して1式ずつしかないため、メモリに対するアクセスは同時にほとんどが実行できないと考えられる。そのため、メモリ待ちが生じることから、VM数に対して増えていくと考えられる。今回の実験では、複数のVMで処理を同時に実行させたことから、ランダムに実行させたときに現れる指数関数的な増加とはならず、1つのVMの処理が終わったら、次のVMの処理と順番に処理されると考えられることから、このような直線的な傾向を示したものと考えられる。

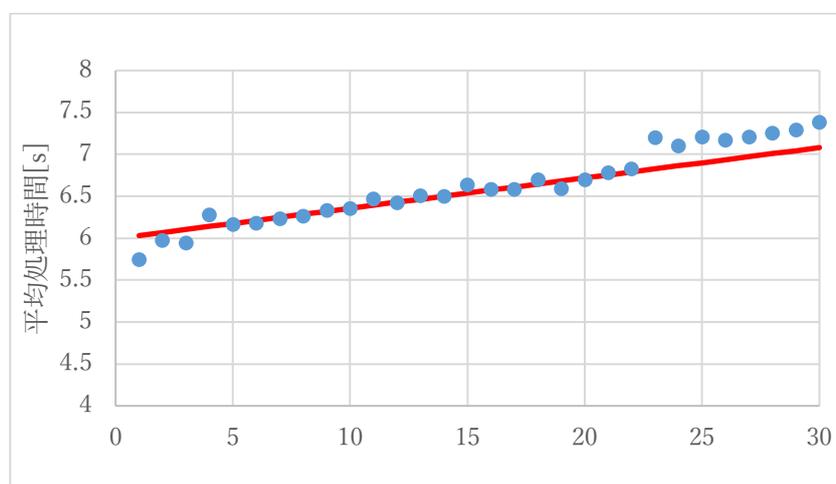


図3 VM数5~20の平均処理時間の回帰直線

一方で、VM 数が 20 を越えると、平均処理時間がこの直線から外れることが判った。そこで、VM 数 20~30 の平均処理時間の回帰直線を最小二乗法により求めた。その結果は、式(2)のようになり、その直線を図 3 にオーバーラップさせたものを図 4 に示す。

$$y = 0.0625x + 5.5376 \quad (2)$$

図 4 から判るように VM 数 23 のときの平均処理時間が直線から外れるものの、残りの値は、ほぼ直線的に増加することが判った。この理由としてメモリと CPU コアの待ちが入るためであると考えられる。前述したように全ての VM を同時に動作させているので直線的に処理時間が増加したものと考えられる。メモリと CPU コアを主に使用するプログラムを同時に動作させる場合は、VM の性能は VM 数に対して直線で記述できることが判った。

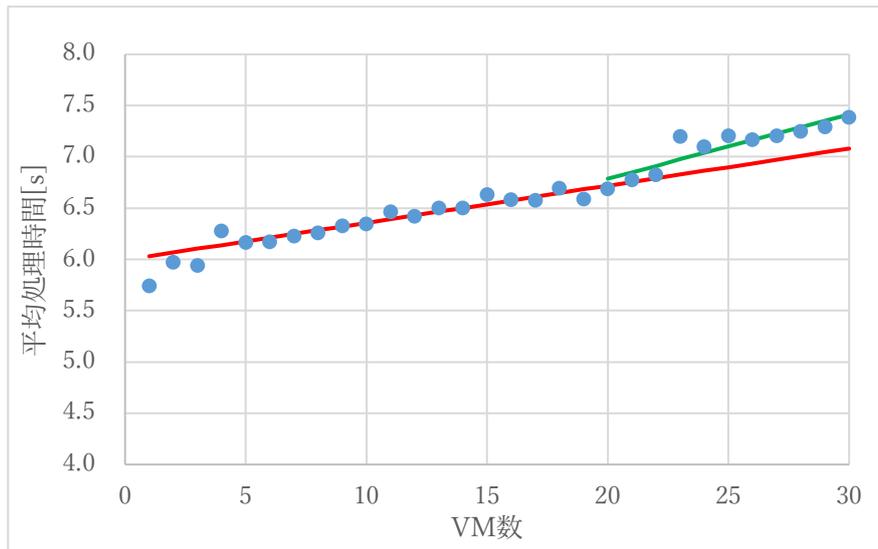


図 4 VM 数 5~20, 20~30 の平均処理時間の近似直線

この研究では、各 VM に同時に同じ負荷を与えて、同時に処理を実行させた場合の処理時間の変動について計測を行い、メモリや CPU へのアクセス競合により処理が遅延し、その遅延が VM 数に比例することが確認された。

測定環境が整ったことから、次のステップとして、待ち行列モデルでよく使われる CPU やメモリへのアクセスがランダムに行われる場合について計測を行い、待ち行列モデルによる記述が可能かどうかの検討を行っていく予定である。さらに、既存研究で作成したシミュレーションモデルを適用し、シミュレーションによる性能予測が可能であることを確認する予定である。

5. 主な発表論文等

〔雑誌論文〕 計0件

〔学会発表〕 計0件

〔図書〕 計0件

〔産業財産権〕

〔その他〕

2022年度の本学学生（須藤巧都，平野愛弓）の卒業研究論文「Apache Kafkaの分散処理機能に関する検討・評価」および，2023年度の本学学生（尾関日南子，竹内彩夏）の卒業研究論文「仮想計算機システムにおける他の仮想計算機の負荷が与える影響に関する評価」の中にまとめられ，これらの内容は，今年度の学会にて報告の予定である．

6. 研究組織

氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------