

# 大規模仮想化環境におけるオンラインランザクション処理の詳細解析

石田 渉† 横山 大作‡ 中野 美由紀‡ 豊田 正史‡ 喜連川 優‡

† 東京大学大学院 情報理工学系研究科

‡ 東京大学 生産技術研究所

## 1 はじめに

近年 Amazon EC2[2], Google Compute Engine[3] といったクラウドサービスが広く利用されている。クラウドコンピューティングは計算資源を動的に増減できるエラスティシティ(伸縮性)が大きな特色となっている。eコマースやオンラインゲームなどのウェブサービスアプリケーションは利用者数の変動、アクセスされるデータ量の変動が大きく、処理負荷が容易に予測できない。一方、これらのアプリケーションではOLTP(オンラインランザクション処理)が必要不可欠であり、クライアントアプリケーションに加えてデータベース等のミドルウェアと連携したエラスティックな計算環境の提供が強く望まれている。

本論文では大規模なオンラインランザクション処理を行うアプリケーションのエラスティシティを効率良く実現する手法を検討する。クラウドコンピューティングではKVM[4]などの仮想化技術を利用して、エラスティシティを実現している[1]。しかしながらクライアントアプリケーションの負荷のみを考慮してVMライブマイグレーションを行なっても、データベースのキャッシュミスにより、かえって性能が低下することもある。そこで、オンラインランザクション処理を行うアプリケーションとデータベースのVMライブマイグレーション時における挙動を実機により計測、その結果を解析し、データベースキャッシュの重要性を明らかにする。さらに大規模仮想化環境においてクライアントアプリケーションのエラスティシティを実現する手法について検討する。

## 2 クライアントアプリケーションのエラスティシティを実現する際の問題

本節ではアプリケーションサーバであるVM(ヴァーチャルマシン)、それをホストするHV(ハイパーバイザー)、そしてDB(データベース)から構成される仮想化環境(1)でクライアントアプリケーションのエラスティシティを実現するための課題を考察する。議論を簡単にするためVM群はVM1~VM4, HVをHV1, HV2, DBをDB1, DB2とする。DB1とDB2はレプリケーション

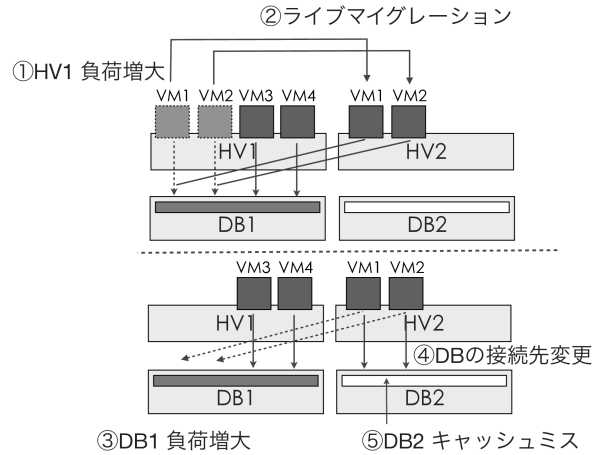


図 1: 想定環境

により、どちらを参照しても同様の結果が得られるとする。はじめすべてのVM群はHV1上で稼働し、DB1を参照している。ここでVM群の負荷が上昇し、HV1が十分な計算資源を提供できなくなった場合を考える(図1①部)。VM群のうちVM1とVM2をHV2にライブマイグレートすることでHV1の負荷分散をすることができる(図1②部)。一方負荷分散後、VM群はより計算資源を得る、すなわちDB1への負荷が増加する(図1③部)。ここでDB1の計算資源が足りないと、アプリケーション全体としてはスループットが上がらない。そこでVM1, VM2の接続するデータベースサーバをDB1からDB2に切り替えると、データベースサーバの負荷も分散する(図1④部)。しかし、VM1, VM2の利用するデータがDB2のキャッシュにない場合、キャッシュミスが発生し、DB1へリクエストし続ける場合と比較し、クライアントアプリケーションの性能が劣化する可能性が生じる(図1⑤部)。つまり、仮想化環境においてはクライアントアプリケーションのVMライブマイグレーション、DBの接続先変更だけでは十分にアプリケーションの性能を担保できない。

## 3 仮想化環境におけるオンラインランザクション処理の詳細解析

前節で述べた性能劣化が実際に生じることを確かめるため、VM内で実行するアプリケーションとしてOLTPベンチマークであるTPC-C[5]を利用し、実機上にて計測を行った。

TPC-Cのwarehouseは16で固定し、connection数を

†Wataru ISHIDA ‡Daisaku YOKOYAMA ‡Miyuki NAKANO  
‡Masashi TOYODA ‡Masaru KITSUREGAWA  
†Graduate School of Science and Technology, The University of Tokyo  
‡Institute of Industrial Science, The University of Tokyo

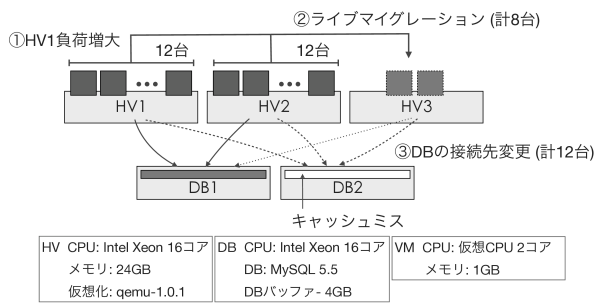


図 2: 計測環境

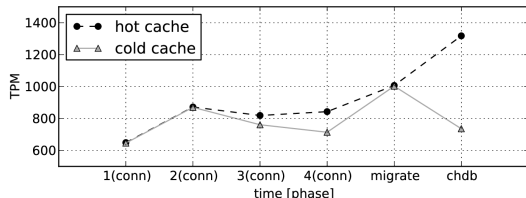


図 3: 各フェーズにおける TPM

変化させることで負荷の変動を実現する。またアプリケーションの性能は TPM(transaction per minitue) で評価する。計測環境を図 2 に示す。計測の流れは次のようになっている。まず HV1 と HV2 の上でそれぞれ 12 台の VM を稼働させ connection 数を 1 としアプリケーションを実行する。すべての VM は実行開始時は DB1 に接続する。なお、DB1 のキャッシュは予め温めておく。段階的にすべての VM の connection 数を 4 まで増加させる (図 2 ①部)。connection 数の増大によって HV がボトルネックとなるため、ここで HV3 に HV1 と HV2 上で稼働する VM を計 8 台 (4 台ずつ) ライブマイグレートし HV の負荷分散を図る (図 2 ②部)。しかし HV のボトルネックが解消されることで今度は DB がボトルネックとなる。そこで DB1 に接続する計 24 台の VM のうち半数を DB2 に接続変更する (図 2 ③部)。ここで DB2 のキャッシュが温まっていない場合、前節で述べた通りキャッシュミスによるパフォーマンス劣化が予想できる。実験では DB2 のキャッシュを温めた場合と温めていない場合の 2 通りで計測を行った。

図 3 に計測の各フェーズにおける VM の平均 TPM を示す。横軸 1~4 は段階的に connection 数を増加させたときの TPM, migrate がマイグレーション後の TPM, chdb が接続するデータベースを変更した後の TPM である。グラフから分かる通り、DB2 のキャッシュを温めていない場合は温めた場合に比べて約 35% の性能劣化が見られ、connection 数 2 のときと同程度の性能しか出せていない。これより前節で指摘した問題が実際に生じていることがわかる。また connection 数 3, 4 では接続数は増えているにもかかわらず TPM が connection 数 2 に比べて低い。これは connection 数 3, 4 のとき HV の CPU 資源に余裕がないため仮想化のオーバーヘッドがクエリの発行処理を阻害するためと考えられる。

## 4 OLTP 仮想化環境におけるエラスティシティ手法の検討

ライブマイグレーション時の性能低下の原因はデータベースのキャッシュミスによるものである。そこで、アプリケーションのデータ参照局所性に着目し、データベース自体はデータアクセスのログがあることを利用し、ライブマイグレーションの対象となる VM, 接続変更先となる DB を決定する。

局所性はアプリケーションによって異なるため、各 DB ノードのキャッシュの状態は接続している VM 群により異なる。そこで DB 間で各 VM が参照するデータ情報を共有し、ライブマイグレーション後に接続先を変更する際、複数ある DB ノードから似た参照範囲の VM が既に接続している DB を選択する手法を検討している [7]。DB のエラスティシティを実現する手法には Sudipto らによる Albatross[6] がある。Albatross は DB のみのエラスティシティを考慮しているのに対し、検討手法では HV と DB のエラスティシティを同時に考慮ことでアプリケーション全体の性能を担保しつつ、処理変動への対処が可能となる。

## 5 まとめ

本論文ではクラウドコンピューティングにおいてオンライントランザクション処理を対象とするクライアントアプリケーションのエラスティシティを実現する上での課題を明らかにし、実機により確認した。さらに DB へのデータ参照局所性を利用したアプリケーション、DB のエラスティシティ実現の手法を検討した。今後は実機計測で得られた知見を元に、検討手法の詳細と実装方式を考えていきたい。

## 参考文献

- [1] C. Clark, K. Fraser, Steven Hand, et al. *Live Migration of Virtual Machines* NSDI, 2005.
- [2] Amazon Elastic Compute Cloud(Amazon EC2). <http://aws.amazon.com/ec2/>.
- [3] Google Compute Engine. <https://cloud.google.com/products/compute-engine>
- [4] KVM [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page)
- [5] TPC-C <http://www.tpc.org/tpcc/>
- [6] Sudipto Das, Shoji Nishimura, Divyakant Agrawal, Amr El Abbadi *Albatross: Lightweight Elasticity in Shared Storage Databases for the Cloud using Live Data Migration* VLDB, 2011.
- [7] 石田 渉, 横山 大作, 豊田 正史, 中野 美由紀, 喜連川 優 *オンライントランザクション処理における VM 挙動の詳細解析* 電子情報通信学会 DE 研究会, 2012.