

ハイブリッドクラウド環境における性能と実行コストに基づいた最適負荷分散ミドルウェアの提案と実装

笠江 優美子[†]

小口 正人[†]

[†]お茶の水女子大学

1. はじめに

近年、コンピュータシステムにおける情報量が爆発的に増加している。その処理プラットフォームとして、ハイブリッドクラウドが注目される。ハイブリッドクラウドには、プライベートクラウドとパブリッククラウドを併用することで、自社の持つリソースを無駄なく使いながら、スケーラブルな処理が実現できるというメリットがある。しかし、実社会において、データ配置の難しさや、特有のコストバランスが生じてしまう点などから、ハイブリッドクラウドの積極的な導入が進められていない。

そこで本研究では、ハイブリッドクラウド環境において、ローカルにあるストレージプールに対し、パブリッククラウドから遠隔ストレージアクセスを行い、データ自体を外部サイトに保存しない前提において、生じるコストという評価軸の元、最適なジョブ配置を行う手法を提案し、ミドルウェアとして実装した。本ミドルウェアを適用することで、想定環境において、アプリケーションの特性に依らず、最適なジョブ配置が行えることを示す。

2. ハイブリッドクラウド導入への懸念

ハイブリッドクラウドとは、主に IaaS において、プライベートクラウドとパブリッククラウドを併用する利用形態のことである。しかし、実社会において、ハイブリッドクラウドの積極的な導入は行われていない。これは、以下の様な懸念があるためだと考えられる。

2.1 クラウドにおけるデータ配置

ハイブリッドクラウドの積極的な導入が進まない理由として、データ配置の難しさが挙げられる。特に、パブリッククラウドでは、企業の持つセキュリティポリシーなどにより、データを外部サイトに保存できないという制約が存在することが多い。総務省によって行われた平成 23 年通信利用動向調査 [1] の結果では、企業がクラウドサービスを導入しない理由の第 2 位にセキュリティへの不安を挙げている。

本研究では、企業の持つセキュリティポリシーに従って、データはローカルにあるストレージプールに保持しておき、ハイブリッドクラウドでデータ処理を行う場合には、パブリッククラウドから遠隔アクセスを行う。これは、パブリッククラウドのインスタンス（仮想マシン）から、処理能力のみを借りる場合を想定したものである。

2.2 コストのトレードオフ関係

ハイブリッドクラウドでの処理には、トレードオフ関係となるコストが生じる。つまり、パブリッククラウドを多

く使って処理を行えば、処理が速く行える分、パブリッククラウドの従量制料金が嵩む。逆に、プライベートクラウドのみで処理を行うと、時間がかかり過ぎてしまう。

そこで、本研究では、性能と金銭的成本という、2つのコスト評価軸において、最適なジョブ配置を行う手法を提案し、ミドルウェアとして実装した。金銭的成本としては、パブリッククラウドの従量制料金とプライベートクラウドの消費電力料金を考える。

3. 想定するハイブリッドクラウド環境の構築

本研究で想定するクラウド環境を、IaaS のクラウド環境構築ソフトウェア Eucalyptus[2] を使って構築した。構築した 2 つのクラウドの間を人工的な遅延を発生させる Dummynet で接続することで、ハイブリッドクラウド環境を実現した。さらに、ローカル環境側には、データを保持する 3 台のストレージから成るストレージプールを設けた。構築したハイブリッドクラウド環境を、図 1 に示す。

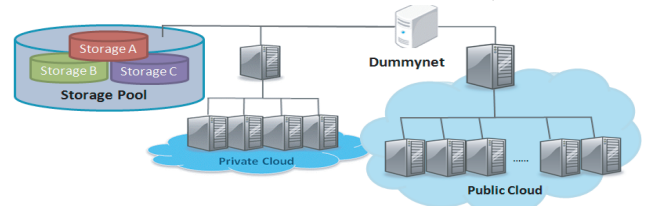


図 1: 構築したハイブリッドクラウド環境

この環境において、それぞれのクラウドからストレージプールに対し、iSCSI アクセスを行う。iSCSI とは、データ転送を行う SCSI プロトコルを TCP/IP ネットワーク上で使用する規格のことである。専用のケーブルを用いて構築される FC-SAN に比べ、安価に SAN 環境を構築できる。

4. 最適配置ミドルウェアの提案方式

4.1 学習フェーズ

本研究では、ハイブリッドクラウドのリソースに対し、無駄のない利用を目指す。そのため、「リソースを使い切った状態」を定義する必要がある。CPU 処理においては、CPU 使用率 100 % を CPU リソースを使い切った状態と定義できる。しかし、ディスク処理においては、環境依存であるため、学習フェーズとしてディスク処理の性能測定を行っていく。本手法では、自作ベンチマーク Disk Bench を用いたジョブを複数同時処理していき、その処理時間と、処理の混雑度を示すディスクに溜まるキューの長さを測定する。そして、測定した処理時間と、その分のジョブを逐次処理した時の処理時間と比較する。ディスク処理に余裕があれば、同時処理した方が処理が速いはずであるが、飽和状態となると処理が遅くなる。本手法では、その点を見つけ、その時のキューの長さを飽和の条件とする。また、

Disk Bench 内には、読み込み量を指定するパラメータが設けられており、学習フェーズでは、これらを変化させることで、様々なアクセスパターンを再現する。

4.2 提案するミドルウェアの動作

本ミドルウェアは、まずプライベートクラウドのリソースに対してジョブを配置する。同時に、それぞれのリソースに対して、定期的に処理状況を問い合わせ、リソースが飽和しているか判断する。つまり、学習フェーズで定義した値を閾値とし、閾値以下であるとジョブ処理を実行し、閾値以上であるとリソースの飽和と判断する。リソースが飽和すると、プライベートクラウド内の別の空いてるリソースに負荷分散を行う。そして、プライベートクラウド内のすべてのリソースを使い果たすと、パブリッククラウドへスケールアウトする。また、ジョブがディスク処理を行う場合、ストレージプール内へデータアクセスする。

5. ミドルウェアの評価実験

5.1 実験概要

想定環境での本ミドルウェアの有用性を示すため、データインテンシブアプリケーションの、ハイブリッドクラウド環境での処理を、本ミドルウェアを介して行っていく。その際に、参考値として、ディスク処理における負荷分散の閾値を、学習フェーズで得られる値より、かなり小さな値やかなり大きな値に設定した場合など、様々な閾値での実験も行う。また、実験時は、生じる処理コストとして、処理したジョブの処理時間と、パブリッククラウドの従量制料金、プライベートクラウドの消費電力料金を測定しておく。この実験により、負荷分散の閾値として、特に学習フェーズで得られた値を設定することで、ミドルウェアの評価にどのような違いが現れるを考察する。

本稿の実験では、適用したデータインテンシブアプリケーションとして、TPC-Hの簡易版であるDBT-3の検索クエリを用いた。また、設定した往復遅延時間は、4msecとした。これは、東京都内にある企業が、自社内にデータを保存するストレージプールとプライベートクラウドを設け、さらにパブリッククラウドプロバイダなどが都内近郊で管理するデータセンタから、必要なCPUリソースを借用する場合を想定する。

5.2 学習フェーズの実行

まず学習フェーズによって、ディスク処理の飽和の条件を定義していく。学習フェーズでは、Disk Bench を使って、様々なアクセスパターンで測定を繰り返すが、本稿では、その結果の一例を示す。

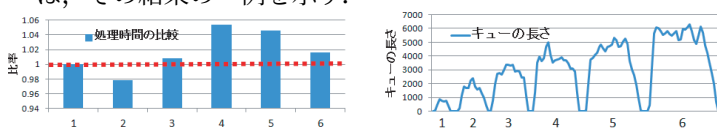


図 2: 学習フェーズ: 処理時間の比較 (一例)

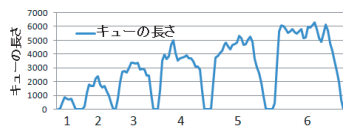


図 3: 学習フェーズ: キューの長さ (一例)

図 2 が逐次処理と同時処理の処理時間の比較結果で、縦軸が比率となっており、1 以上が同時処理の方が遅い、つまり、ディスクの処理的な飽和状態を意味する。図 2 から、

同時処理数 3 で、ディスクの処理的な飽和が生じていることがわかる。また、この時のキューの長さの推移の結果が、図 3 である。同時処理数とディスクに溜まるキューの長さが比例関係にあることがわかる。

この処理時間による飽和の判断と、キューの長さの増加傾向を、様々なアクセスパターンで分析した結果、キューの長さが 2200~3200 で、本実験環境では、ディスクの飽和状態となっていることがわかった。このことから、ディスク処理の飽和の条件を、上記値と定義する。

5.3 コスト評価

図 4 に、本実験での、様々な閾値を設定した時の、生じたコストの評価結果を示す。横軸が、処理時間という時間的コスト、縦軸が金銭的成本である。金銭的成本は、実行時に測定した、パブリッククラウドの従量制料金とプライベートクラウドの消費電力料金の和である。

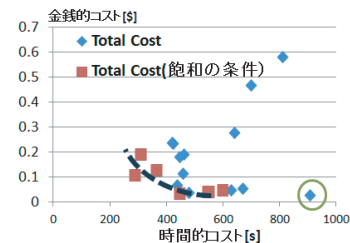


図 4: コスト評価 (DBT-3 検索クエリ)

本ミドルウェアによって、学習フェーズにより決定したディスクリソースを使い切った状態の条件となるキューの長さ、つまり 2200~3200 の値付近に設定することで、リソースを効率的に使いながら、コスト評価においてパレート最適の点となっていることがわかる。

また、緑の丸で囲まれた点は、プライベートクラウドのリソースのみでジョブ処理を行った場合のコスト評価結果である。遅延発生時でも、プライベートクラウドのリソースのみで処理する場合と比較して、本提案手法の方が、効率の良い処理が提供されることが確認できる。

6. まとめと今後の課題

ハイブリッドクラウド環境での効率的な処理に関して、実環境に沿った環境で、どちらのクラウドをどれくらい使えば良いのかを、性能と必要なコストに基づき判断して、最適なジョブ配置を行える手法を提案し、ミドルウェアとして実装した。

今後の課題としては、本研究で想定する環境において、様々なアプリケーションでの考察を行なっていきたい。また、現在の実装では、学習フェーズで取得した閾値を、ミドルウェアに適用してから実行する形となっているが、これをミドルウェアの一部に取り込み、自動化を実現したい。

参考文献

- [1] 平成 24 年度情報通信白書:
<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/pdf/24honpen.pdf>
- [2] Eucalyptus:<http://www.eucalyptus.com/>