クラウドを利用したElasticなデータストリーム処理の検討

石井 惇志† 鈴村 豊太郎†‡

†東京工業大学 ‡IBM 東京基礎研究所

1. 研究の背景

データストリーム処理を用いたWebアプリケーションを運用する場合、データレートが増大した場合においても、レイテンシなどで定義されるSLA（Service Level Agreement）を守れることが望ましい。SLAを守ることが出来ない場合、多くのユーザーに不満を与えてしまい、アプリケーションを使用するユーザー数の減少にも繋がる。

SLAを守る為の手段として、計算機の能力を増強する、すなわち計算資源を増やすことが考えられる。しかし、経済的な問題、及び計算機を配置するスペースの問題により、実際に有することが出来る計算資源の量には限度がある。更に、データレートの増大に対応するために計算資源を増やすことを考えると、手続き的な問題により、即時の計算資源の追加というのは困難である。

このように、経済的、手続き的な問題などの複数の要因によって、データレートの増大に対し、即時に計算資源を追加することは困難である。

そこで、本研究では、追加する計算資源としてクラウド環境を考える。クラウド上のVM（Virtual Machine：仮想マシン）ならば、数分程度の所要時間で使用することが可能になり、また、物理的なスペースを必要としないので、上限を気にすることなく計算資源を追加することができる。したがって、自身の保有する計算資源では処理が追いつかなくなった、あるいは追いつかなくなることが予想される場合に、クラウド上のVMを立ち上げ、処理を委譲することで、SLAを守るための即時の対応が可能になると考えられる。本研究ではこの機能を可能にする手法、及びアーキテクチャについて考察する。

また、クラウド環境を利用すると、利用状況に応じた経済的コストが発生する。SLAを守るためとはいえ、不必要に多くのVMを起動すれば、経済的損失が大きくなる。この、SLAの保証と経済的コストのトレードオフを、本研究では最適化問題として捉え、SLAを守りつつ、必要な経済的コストを最小化するスケジューリング手法についても考察する。

本研究における技術的な挑戦は、以下の4点が挙げられる。

- どのようなアルゴリズムで、クラウド上に処理を委譲するか
- どのように起動するクラウド上のVMの数を決定するか
- どのタイミングで、クラウドのVMを起動・停止させるか
- 現行のデータストリーム処理系には、実行時にノードを動的に追加する機構がないので、それをどのように解決するか

これらの4点に対して、本研究では、クラウド上に処理を委譲するための手法を提案し、更に、SLAを堅守した上で、クラウドを利用するための費用を最低限に抑えるための、必要最低限のVM数を求める最適化問題を提案する。また、現行のデータストリーム処理系にはクラウド上のノードを動的に追加する機構がないので、System Sの機能を用いて外部的にその機能を実装するアーキテクチャを提案する。

本研究では、クラウド環境としてEucalyptus[1]を、データストリーム処理系としてSystem S[3]を使用する方針である。

また、アプリケーションは応答時間の緩やかなものを想定し、SLAはレイテンシで定義されるものとする。

2. データストリーム処理

データストリーム処理とはデータを蓄積せずに逐次処理するという新しい計算パラダイムで、リアルタイムの応答が要求される場合や、前後の僅かなデータのみを参照すればよい計算、データの蓄積が困難な処理に適している。

処理系にはIBM ResearchのSystem Sが存在するが、System Sはデータフロー図から直感的に処理を記述できるSPADE[3]という宣言的言語を持ち、自動性能最適化を行うSPADEコンパイラとSPC[4]という実行基盤を用いてデータストリーム処理を実現する。SPADEでは組み込みオペレータで処理を記述するが、C++やJavaといった汎用言語を用いた独自のオペレータや関数により高度に柔軟な処理も実装できる。このように、データストリーム処理やミドルウェアの研究に適するため、本研究ではSystem Sを利用する。

3. StreamCloud

本研究で提案する、高データレートな状況で処理をクラウド上に委譲するアーキテクチャを、StreamCloudと呼ぶことにする。本章では、StreamCloudを構築する上で必要となる、SLAを順守した上で、クラウド利用のコストの最小化する最適化問題の構成について論じる。

3.1 クラウド環境を利用するための費用の最小化

第1章で述べたように、クラウド環境を利用するには、それに応じた経済的コストを支払う必要がある。例えば、Amazon EC2[2]の場合、クラウド環境を利用した場合の課金方法は、

- 1VM毎の稼働時間
- 保管するデータ量
- データ転送量

にそれぞれ依存する。実際のWebアプリケーションの運用において、保管するデータ量は、余分なデータを排除するなどして最適化を施せば削減することはできるが、アプリケーションを運用する以上、一定の量は必ず使用する。そこで、本研究では保管するデータ量による課金は固定費用とみなし、最小化には考慮しない。すると、クラウド利用コストを最小化するためには、VMの起動台数、及びその稼働時間、そしてデータ転送量を最小化しなければならない事になる。

$$Cost = \sum^{App}\left(\sum^{Cloud} \sum^{I} m \cdot t \cdot p_{time} \right)$$
$$+ \sum^{App}\left(\sum^{Cloud} \sum^{I} m\left(n_{in}\, p_{In_c} + n_{out}\, p_{Out_c}\right) \right)$$
$$+ \sum^{App} \sum^{Cloud} \sum^{I} X \cdot m \cdot l \qquad \ldots(1)$$
$$\forall l \le SLA + error \qquad \ldots(2)$$

表 1： 目的関数と束縛条件

3.2 レイテンシの順守及び最小化

また、クラウド利用コストを最小化すると言っても、SLA を守れる範囲でなければならない。更に、クラウドのデータセンターの物理的所在地によって、通信に要するレイテンシは変わってくる。委譲先のクラウド環境を複数箇所想定する場合には、課金額だけではなく、その環境を利用する場合のレイテンシについても考慮する必要がある。

本研究では最適化問題を金額単位で構成することを考えているので、時間から金額に換算するパラメータを定義し、その最適な値についても考慮する必要があると考えている。

3.3 目的関数と束縛条件

現在検討を行っている、最適化問題の目的関数と束縛条件を表 1 に示す。[6]では二値線形計画法を用いて最適化問題を構成しているが、本研究では複数の VM に同じ処理を委譲するケースが考えられるので、整数計画法を用いて、最適化問題の解として、必要な VM の数を取得出来るようにすることを考えている。(1)式が最小化すべきコストを表す目的関数で、(2)式が SLA を守るための束縛条件である。

(1)式は、第一項が稼働時間に依る課金の項、第二項がデータ転送量による課金の項、そして第三項がレイテンシの項である。App はアプリケーション総数、Cloud はクラウドプロバイダの総数、I は起動する VM の種別の総数を表す。m は各クラウドプロバイダの VM の種別ごとの、起動する VM の数である。また、p_{time} が稼働時間によるの課金額を表し、t は稼働時間を表す。p_{in}、p_{out} はそれぞれデータ送受信に関する課金額で、n_{in}、n_{out} がデータの送受信の量を表す。l はレイテンシで、X は時間から金額への換算パラメータである。

また、(2)式の束縛条件は、全ての VM で行われる処理のレイテンシが、SLA の基準と許容誤差の範囲内に収まっていなければならないことを示す。

4. StreamCloud システムのアーキテクチャの検討

本章では、現在想定している、StreamCloud に必要となるコンポーネント及び全体アーキテクチャの概要を検討する。

必要となるコンポーネントは、

- VM の起動・停止の管理
- 今後のデータレートの予測
- データストリームの VM への振り分け
- 最適化問題のソルバ
- データレートと VM のタイプからのレイテンシ予測

が考えられる。

アーキテクチャの概要としては、各アプリケーションを、データストリームを受け取る部分と実際の処理を行う部分を分割し、処理を行う部分を VM 上で実行させ、その数を動的に変更することで、並列化による処理能力の向上を図ることを検討している。

5. 実験・評価に向けて

本章では、実験と評価手法のの構想について述べる。

実験の対象とするアプリケーションは、StreamCloud が応答時間の緩やかなアプリケーションを想定しているので、ネットワーク負荷が高く、CPU 負荷が軽いアプリケーションを想定している。

現在検討しているアプリケーションは次のとおりである。

- VWAP

  株式市場における「出来高加重平均価格(Volume Weighted Average Price)」の略称であり、成立した売買について、価格毎の出来高を加味して加重平均を求めた値である。このアプリケーションでは加重平均を求める処理を各 VM 上に分散させることで処理委譲を行う。
- Twitter の投稿情報のデータマイニング

  マイクロブログサービスである、Twitter で公開されている StreamingAPI を利用することで、Twitter への投稿情報の一部をデータストリームとして取得できる。このアプリケーションでは、Twitter の投稿情報について、データマイニングを行うことを考える。

  具体的な処理の例としては、特定の単語を含む投稿や、日本人ユーザーなど、特定カテゴリのユーザーの投稿のフィルタリングが考えられる。

6. まとめと今後の展望

本研究では、データストリーム処理を用いた Web アプリケーションの運用において、データレートの増大に伴うレイテンシの増加に対して、クラウド上のデータストリーム処理系に処理を委譲することでその解決を図る手法を提案し、その実装である StreamCloud のアーキテクチャ、クラウド利用に伴う経済的コストの最小化問題について検討した。

最適化問題の更なる検討、及びアーキテクチャの実装、アプリケーションを用いた実験と評価が今後の課題である。

参 考 文 献

[1] Daniel Nurmi et.al., The Eucalyptus Open-source Cloud-computing System, Proceedings of the 2009 9th IEEE/ASCM International Symposium on Cluster Computing and the Grid

[2] Amazon Elastic Compute Cloud (Amazon EC2),http://aws.amazon.com/ec2/

[3] Bugra Gedik, et.al., SPADE: The System S Declarative Stream Processing Engine" SIGMOD 2008

[4] Lisa Amini, et.al., SPC: A Distributed, Scalable Platform for Data Mining DM-SSP 2006

[5] Barzan Mozafari et.al., Optimal Load Shedding with Aggregates and Mining, ICDE 2010

[6] Ruben Van Bossche et.al., Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads, IEEE CLOUD 2010