StreamSpinnerのEC2における評価

João Felipe Santiago dos Santos Orui[†]　川島英之[‡,‡‡]　北川博之[‡,‡‡]
筑波大学第三学群情報学類[†]　筑波大学大学院システム情報工学研究科[‡]
筑波大学計算科学研究センター[‡‡]

1　はじめに

カメラ，センサなどのストリームデータを処理するためのデータストリーム技術が注目されている．今までのデータストリーム技術に関する研究は，関係データモデルを拡張してストリームを扱うためにデータモデルの提案，そのモデルに基づく処理系である Stream Processing Engine（以降，SPE）において性能向上とメモリ使用量の制御に関する研究をはじめ，分散処理および耐故障性向上というものがあった．最近の SPE に関する研究では，明確なアプリケーションを想定した，専門用途システムの構築が研究されている．用途例には，RFID，ネットワークトラフィック，監視など，幅広いアプリケーションが含まれる．

このような SPE の技術が提供されているものの、評価は現段階ではまだ行われていない．本研究では分散ストリーム処理機構である StreamSpinner[1]のクラウドにおける利用方法の提案を行うとともに StreamSpinner の分散環境でのスループットの評価を実施する．

本稿は以降の構成は以下の通りである．2 節では Amazon Elastic Compute Cloud について記述する．3 節では分散ストリーム処理環境について記述する．4 節ではクラウド環境における利用の提案について記述する．5 節では本稿のまとめと今後の課題について述べる．

2　Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2) [2]は米アマゾン社が計算資源を提供し，利用者がアプリケーションをクラウド環境で実行することができる AmazonWeb Services(AWS) の１つである．EC2 のウェブサービスインターフェイスを利用して仮想マシン(インスタンス) を起動することより，任意のアプリケーションが実行できる環境を短時間に用意することが出来る．必要に応じたインスタンスの作成・起動・停止が可能であり，柔軟に構成台数を変えることができる．利用者はネットワーク使用，インスタンスの起動時間，データ保存使用に応じた利用料課金が行われる．1 インスタンスを 500 時間実行しても，500 インスタンスを 1 時間実行しても起動時間に対する利用料は同じである．インスタンスの柔軟性は高く，さまざまなオペレーティングシステム(OS) を利用することが可能であり，選択した OS に適用するメモリ・CPU・インスタンスストレージ設定を多数のインスタンスタイプから選択できる．インスタンスが実行している OS に対する root 権限を持っていることから，通常のサーバマシンと同様に扱うこともできる．EC2 を利用することで，必要に応じたサーバ台数をオンデマンドで短時間に調達しやすく，また必要がなくなった場合はインスタンスを停止することができる．

Evaluation of StreamSpinner on EC2
João Felipe Santiago dos Santos Orui[†](joao@kde.cs.tsukuba.ac.jp)
Hideyuki Kawashima[‡,‡‡] (kawasima@cs.tsukuba.ac.jp)
Hiroyuki Kitagawa[‡,‡‡] (kitagawa@cs.tsukuba.ac.jp)
[†]College of Information Sciences, University of Tsukuba
[‡]Graduate School of Systems and Information Engineering, University of Tsukuba
[‡‡]Center for Computational Sciences, University of Tsukuba

3　分散ストリーム処理環境

分散ストリーム処理環境の構成はストリーム処理エンジンと分散環境を管理・運用するシステムである．本研究ではストリーム処理エンジンの StreamSpinner と StreamSpinner 運用管理システムとして提案された ORINOCO[3]を利用する．StreamSpinner については 2.1 節で述べる，ORINOCO システムについては 2.2 節で述べる．

2.1　StreamSpinner

StreamSpinner はストリームデータ管理のための情報基盤システムである．Java 言語により開発されており，多数の機能を有する．それらの機能には，複数のストリームデータや既存の DBMS 中のデータを扱うための統合環境，SQL ライクな問合せ要求記述言語，新規データの到着や時間の経過に連動したイベント駆動型の連続的問合せ実行機構，大量の問合せの同時実行を

可能にする複数問合せ最適化技術，ストリームを用いたアプリケーション開発のための Java API，複数ノードによる分散ストリーム処理機構，そして仮想マシン技術と連携した問合せ処理の高信頼化があげられる．

2.2 ORINOCO システム

StreamSpinner の複数ノードによる分散ストリーム処理環境における運用管理システムとして「ORINOCO」が提案されている．ORINOCO システムを利用することで、分散ストリーム処理環境上で利用者が作成したアプリケーションを実行できる．分散ストリーム処理中のどのノードに処理の割当てを行うかの決定の分散問合せ最適化機能を持つ．

4 クラウド環境における利用方法の提案

StreamSpinner を利用した分散ストリーム処理環境と ORINOCO の運用管理システムを利用して EC2 上で分散ストリーム処理環境をクラウドサービスとして提供する方法とストリームデータの入力コネクタの提案について述べる．

4.1 EC2 上の分散ストリーム処理環境

ORINOCO システムの拡張として，EC2 インスタンスの管理機能を提供する．あらかじめ事前に用意した StreamSpinner のストリーム処理環境がインストールされている OS イメージ（Amazon Machine Image，以降 AMI）のインスタンスを起動する機能を提供する．AWS が提供している EC2 API である，RunInstances クエリパラメータの AMI ID と起動するインスタンス数を指定して起動要求をする．自動的に分散ストリーム処理環境を構築するため、実行中のインスタンス一覧を取得できる DescribeInstances クエリを利用している．定期的に一覧を取得して、現在管理していないノードがあれば、ORINOCO システムの管理下に設定する機能を実装している．また，分散ストリーム処理環境を縮小または終了するためにすべてのノードのインスタンスを TerminateInstances クエリの利用より実現している．

4.2 ストリームデータの入力コネクタ

外部からクラウド環境上にある ORINOCO システムと StreamSpinner ノードの要素で構成されている分散ストリーム処理環境に対して情報源のストリームデータを送信する仕組みである．ストリームデータ入力コネクタは JAVA で実装されており，ORINOCO システムと StreamSpinner ストリーム処理環境に接続する機能を持っている．

また，ORINOCO システムと StreamSpinner ストリーム処理環境はコネクタからの接続を受け付ける機能を持っている．２つのシステム・環境と通信を行う必要があるため，コネクタを 2 つ用意する．１つ目は ORINOCO システムと通信をする CloudStreamConnector を提供し，二つ目は StreamSpinner と通信をする CloudStreamNodeConnector を提供する．ストリーム情報を送信するための流れは次のようになる．CloudStreamConnector を利用して ORINOCO システムに対して提供したいストリームデータのストリーム名・フィールド名・フィールド型を登録する．この情報を受け取った ORINOCO システムはストリーム数の少ないノードを選び，そのノードに対してストリームデータを受け付けるためのラッパーを設定する．その直後に，ラッパー設定が完了したノードに対してストリームデータを送信するための接続先アドレスとポート番号を返す．接続先情報を CloudStreamConnector が受け取ると、その情報を利用して CloudStreamNodeConnector を生成してクライアントに返す．今度は，CloudStreamNodeConnector が StreamSpinner のノードに対して接続をし、ストリームデータを送信できる状態になる．最後に，クライアントがデータ送信のためのメソッドを呼び出すことより，登録したストリームのノードを物理的に意識しないでデータストリームを送信できる．

5 まとめと今後の課題

本論文では Amazon EC2 における StreamSpinner の分散配置に関する我々の検討を述べた．今後は大規模実験を行う予定である．

謝辞

本研究の一部は科学研究費補助金基盤研究(A)(#21240005）による．

参考文献

[1] StreamSpinner http://www.streamspinner.org/
[2] Amazon Elastic Compute Cloud (Amazon EC2) http://aws.amazon.com/ec2/
[3] 稲守孝之, 渡辺陽介, 北川博之, 天笠俊之．“分散ストリーム処理環境のための運用管理システムの提案” DEWS2007，2007 年