

分散ストリーム処理管理システム ORINOCO の評価

稲守 孝之[†] 渡辺 陽介^{††} 北川 博之^{†††} 天笠 俊之^{†††} 川島 英之^{†††}[†] 筑波大学 システム情報工学科 ^{††} 科学技術振興機構 戦略的創造研究推進事業 ^{†††} 筑波大学 計算科学研究センター

1 はじめに

近年、デバイスやネットワークの発達に伴い、時刻とともに変化する情報を次々と配信するストリームデータが増大している。ストリームデータの例は、web 上で配信される株価データやニュースコンテンツ、センサデバイスから配信される温度・光・音データ、カメラから配信される映像データなどである。web 上のデータや DB に格納されているデータは利用者が自ら取得するのに対し、ストリームデータは利用者に対して能動的かつ連続的に配信される。このような性質を持つストリームデータに対して、フィルタリングや他の情報源との統合のような問合せ処理要求が高まっている。

そこで、これらの要求を実現するストリーム処理エンジン [1] の研究・開発が行われている。我々の研究グループにおいても、StreamSpinner[2, 3] というストリーム処理エンジンを開発中である。

センサーのような実世界から得られるストリームデータは、その情報源が地理的に分散していることが多々ある。そこで、分散した情報源から配信されるストリームデータを効率的に処理するために、分散ストリーム処理の研究が行われている。我々の研究グループでも、StreamSpinner を構成単位とする分散ストリーム処理環境を構築した。本研究では、この処理環境を管理・運用するための ORINOCO システム [4] を開発した。ORINOCO システムは分散環境中で動作するアプリケーションを構築するためのフレームワークを提供しており、利用者はシステムの提供する標準演算と利用者自身が作成した処理を組み合わせることで、処理アプリケーションを容易に実現できる。本稿では、広域分散処理環境における実験を基に、ORINOCO システムを使用した分散ストリーム処理について評価する。

2 分散ストリーム処理

本節では、ORINOCO システムについて概説する。本研究では、分散ストリーム処理環境を構成するス

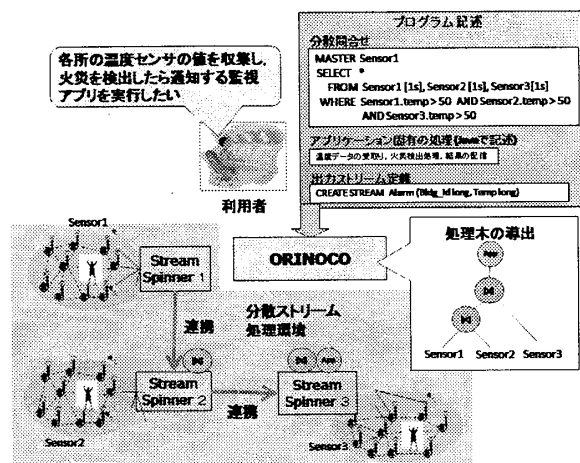


図 1: ORINOCO システムを用いた分散ストリーム処理の例

トリーム処理エンジンとして StreamSpinner を使用する。StreamSpinner は、情報源から配信されるストリームデータをリレーシオンのタプル形式に変換して扱うことで、ストリームデータに対し、リレーショナル代数における選択、射影、結合、直積演算処理を可能にする。また StreamSpinner は、他の StreamSpinner の処理結果を受け取る連携機能を有する。

StreamSpinner で構築された分散ストリーム処理環境を ORINOCO システムによって運用する例を図 1 に示す。図 1 では StreamSpinner が各ノードに分散配置され、近隣のセンサと接続し、温度データを逐次取得している。ここでは、利用者は「各所の温度センサの値を収集し、火災を検出したら通知する監視アプリを実行したい」という要求を持っているものとする。この要求を満たすために必要な処理は、(1) 温度データの収集処理、(2) 火災を検出するアプリケーション処理、(3) 検出結果の配信処理である。ORINOCO システムは、分散ストリームに対する問合せ（以下、分散問合せという）および分散問合せ結果に対しての任意のアプリケーション処理を定義できるフレームワークを有しており、利用者は上記 (1)~(3) の処理要求をプログラム記述として ORINOCO システムに与えることができる。プログラム記述および ORINOCO システムが有するフレームワークに関する詳細については [4] を参照されたい。ORINOCO システムはこのプログラム記述から、リレーショナル代数演算とアプリケーション処理で構成される処理木を導出する（図 1 右下）。そして各演算および利用者プログラムをそれ

An Evaluation of Distributed Stream Processing Management System ORINOCO

Takayuki INAMORI(tinamo@kde.cs.tsukuba.ac.jp)[†]

Yousuke WATANABE(watanabe@kde.cs.tsukuba.ac.jp)^{††}

Hiroyuki KITAGAWA(kitagawa@cs.tsukuba.ac.jp)^{†,†††}

Toshiyuki AMAGASA(amagasa@cs.tsukuba.ac.jp)^{†,†††}

Hideyuki KAWASHIMA(kawasima@cs.tsukuba.ac.jp)^{†,†††}

[†] Graduate School of Systems and Information Engineering, University of Tsukuba

^{††} Japan Science and Technology Agency, Core Research for Evolutional Science and Technology (JST/CREST)

^{†††} Center of Computational Sciences, University of Tsukuba

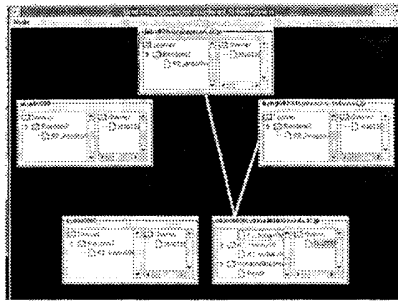


図 2: InTrigger 上の分散環境を管理する ORINOCO

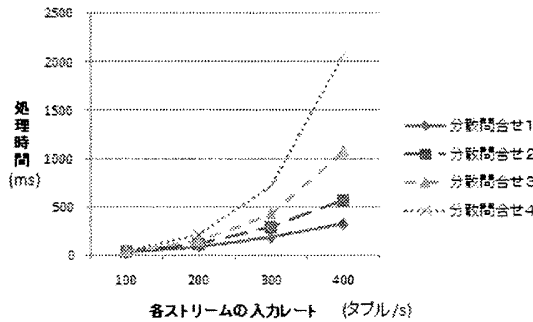


図 3: 各分散問合せにおけるストリームの入力レートと処理時間

ぞれノードに配置し、各 StreamSpinner を連携させることで分散ストリーム処理を実現する。

3 ORINOCO システムの評価

本節では、ORINOCO システムを使用した広域分散ストリーム処理の評価実験について述べ、その有効性を評価する。実験における分散処理環境を構築する基盤として、情報爆発プロジェクトの InTrigger[5] を使用した。InTrigger は、日本国内で 7 拠点にまたがり数百ノードを持つ、分散処理の研究などのために構築されたプラットフォームである。本実験では 5 拠点をし、各拠点内の 1 ノードに StreamSpinner を配置した。拠点ごとに 1 タプル 7 属性の人工データを配信するストリームを用意し、各 StreamSpinner はそれぞれからデータを取得する。本実験における ORINOCO システムの管理画面を図 2 に示す。

実験は以下に示す 4 種類の分散問合せを含むプログラムを ORINOCO システムに与えて分散環境中への配置を実行し、それぞれについて各ストリームの入力データレートを変動させた場合の分散問合せ処理の実行時間を計測する。

- 分散問合せ 1 : 2 ストリームの結合演算処理
- 分散問合せ 2 : 3 ストリームの結合演算処理
- 分散問合せ 3 : 4 ストリームの結合演算処理
- 分散問合せ 4 : 5 ストリームの結合演算処理

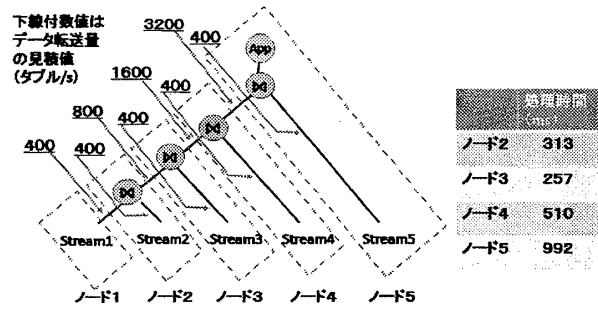


図 4: 各ノードの結合演算の処理時間 (各入力レート: 400, 分散問合せ 4)

ここでは、すべての結合演算の選択率を 0.05 とし、結合演算のアルゴリズムは入れ子ループ結合を用いた。各ストリームの入力データレートを毎秒 100 タプルから 400 タプルまで変動させた場合の全ノードの処理時間の合計値を図 3 に示す。この結果より、各ストリームの入力レートが毎秒 300 タプル以下ならば 5 ストリームの結合演算処理も 1 秒未満で完了することが判明した。図 3 では、入力レートが毎秒 400 タプルで分散問合せ 4 を実行した場合の処理時間の合計が 2 秒を超えているが、各ノードの処理時間は図 4 に示すとおり 1 秒未満であるので、この条件においても演算処理が滞りなく実行されることが証明された。

4 おわりに

本論文では、ORINOCO システムを用いた分散ストリーム処理の性能評価実験を行い、その有効性を立証した。今後の課題としては、ORINOCO システムが行う演算の配置最適化手法 [6] の実験を行い、その有効性を評価することが挙げられる。

謝辞 本研究の一部は、科学研究費補助金特定領域研究 (# 19024006)、科学研究費補助金基盤研究 (A) (# 18200005)、科学技術振興機構 CREST「自律連合型基盤システムの構築」による。

参考文献

- [1] Daniel J. Abadi, et al., "The Design of the Borealis Stream Processing Engine", Proc. CIDR, 2005.
- [2] 渡辺陽介, 北川博之. "連続的問合せに対する複数問合せ最適化手法", 電子情報通信学会論文誌, Vol.J87-D-I, No.10, pp.873-886, 2004 年 10 月.
- [3] StreamSpinner. <http://www.streamspinner.org>
- [4] 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.
- [5] InTrigger, <https://www.logos.ic.u-tokyo.ac.jp/intrigger/registration/>
- [6] 稲守孝之, 渡辺陽介, 北川博之, 天笠俊之, 川島英之. "分散ストリーム処理環境におけるアプリケーション配置最適化手法" 夏のデータベースワークショップ DBWS2007, 2007 年.