様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２７ 年　 ６ 月 １７ 日現在

機関番号：１２１０２
研究種目：基盤研究(C)
研究期間：2012～2014
課題番号：２４５００１０６

研究課題名（和文）リアルタイム事象検知基盤に関する研究

研究課題名（英文）A Study on Real-Time Event Detection Infrastructure

研究代表者
川島　英之（Kawashima, Hideyuki）

筑波大学・システム情報系・講師

研究者番号：９０４０７１４８

交付決定額（研究期間全体）：（直接経費）　　4,000,000円

研究成果の概要（和文）：本研究の目的は膨大なセンサデータを手軽かつ効率的に処理するシステムを開発することであった．この目的を達成するため，本研究ではトランザクショナルストリーム処理機構，FPGAを用いた高性能ストリーム処理機構，効率的な複合イベント処理アルゴリズム，暗号化ストリームデータ処理，共有計算による効率的な複数ストリーム処理機構，そしてストリーム処理基盤システムの研究開発を行った．

研究成果の概要（英文）：The purpose of this research was to develop a system that efficiently and easily process enormous amount of sensor data. To achieve the purpose, we conducted a series of works: transactional stream data processing architecture, high performance stream processing architecture exploiting FPGA, efficient complex event processing algorithm, encrypted stream data processing, efficient multiple stream processing architecture by shared computation, and stream processing infrastructure system.

研究分野：DBMSカーネル技術

キーワード：ストリームデータ処理　センサー　データベース　DBMS

１．研究開始当初の背景

実世界で生じるイベントを検知すべく，センシングデバイスからデータを収集・解析する取り組みが行われていた．その例には，スマートルーム(東大 RCAST-森川研)，地震監視（猿渡先生（連携研究者)），気象監視（筑波大-日下研)，衛星監視（産総研 GEO-Grid）等が挙げられた．この状況に対して，センサデータ処理基盤に関する研究が様々に行われていた．海外研究には，科学データ処理を指向した ArrayStore(MIT 他)，ノイズ除去を行う MauveDB や FunctionDB(MIT)，複合イベント処理（CEP）を対象とする SASE（マサチューセッツ州立大)，関係ストリーム処理を支援する Borealis（MIT）や System S（IBM）が挙げられた．国内研究には，KRAFT（提案者）と StreamSpinner(筑波大-北川研)のみがあった．

２．研究の目的

上記の背景の元，膨大なセンサデータを手軽かつ効率的に処理するシステムを開発することが本研究の目的だった．

３．研究の方法

効率的なデータ処理システムを開発するために，複合イベント処理の高性能化技法，アクセラレータである FPGA を用いた高性能化技法に取り組んだ．また，リレーショナル演算子に加えて機械学習演算子を有するデータ処理システム自体の研究開発にも取り組んだ．

４．研究成果

4.1　複合イベント処理

RFIDやGPSなど多くのセンシングデバイスが当時も普及しつつあった．これらのセンシングデバイスからは継続的に終わりなくストリームデータが生成される．このストリームデータから，ユーザが指定したパタンを抽出する技術として，複合イベント処理がある．

センシングデバイスからは高頻度にストリームデータが配信されるので，複合イベント処理基盤はストリームデータを高速に処理しなければならない．複合イベント処理基盤の処理能力がストリームデータの到着レートよりも低い場合は，処理結果の鮮度や正確性が低下など，重大な問題を招く．そのため，複合イベント処理においてはスループットの改善が重要である．

複合イベント処理の既存手法として SASE が提案されているが，SASE では問合せの結果を生成する過程において，同じリンクを何度もたどるため，スループットが低下するという問題がある．本研究ではこの問題を解決するために，リンク先が同じイベントを１つに集約する“リンク集約”という手法を提案した．本研究では，BLA(Backward Link Aggregation) と FLA(Forward Link Aggregation)という２つの集約方法を提案した．合成データを用いた評価実験において，SASE よりも BLA と FLA のスループットが高いことを確認した．また，入力されるイベントの並びによっては，BLA や FLA がスループットを改善できないことも確認した．

次に本研究では，イベントの並びに依存せずにスループットを改善する手法として PRC(Partial Result Caching)を提案した．この手法は，1 度たどったイベントをキャッシュ構造内に格納し，問合せの結果を生成する際にはこのキャッシュ構造も利用することで，リンクをたどるよりも高速に問合せの結果を生成する．評価実験において，SASE と比較した場合，最大で 4.64 倍のスループット改善率を記録した．また，BLA や FLA と異なり，イベントの並びに依存せずにスループットが改善できることも確認した．

この結果を用いて，イベントに発生確率が付加されているような場合にも，PRC を拡張することで，スループットの改善を試みた．拡張した内容は先読み枝刈りというイベント枝刈り手法であり，この手法により確率計算の回数と問合せ結果生成時間を削減する．評価実験の結果，最大で 25.46 倍のスループット改善率を記録した．

4.2　暗号化ストリーム処理

継続的にストリームデータを生成し続けるストリーム情報源の数が増大しつつある．ストリームデータを処理する基盤システムとして，ストリーム処理エンジン（SPE）が開発されてきた．大量のストリーム情報源に対して処理を行うためには，SPE には非常に高い演算処理能力が要求される．このような処理の実行にはパブリッククラウドなどの分散並列処理基盤を用いることが有効であると考えられる．しかし，パブリッククラウドは一般に組織のファイアウォールの外側で第三者により管理されるため，情報の機密性を保持することができない．これに対し本研究では，安全性を考慮したストリームデータ処理の実現を目的として「暗号化ストリームデータ処理方式」の提案をした．

本研究で提案した暗号化ストリームデータ処理方式は，ストリーム情報源から発生するデータを様々な特徴を持つ 3 種類の暗号化アルゴリズムを用いて暗号化した上で SPE へ送信し，クエリに応じてそれぞれの暗号値を複合的に用いることで一切の復号を行う事なく基本的な関係演算を実現するものである．しかしながら本方式では，暗号化及び復号処理によるシステムの性能劣化や，データ量の増加に伴う通信帯域の圧迫や SPE のメモリ使用量の増大が課題となる．そこでこれらの課題に対してデータ量削減のための二つの効率化手法を示し，評価実験により有効性を示した．

次に，提案方式における効率的な暗号化鍵更新手法について三つの評価要素を挙げ，そのうちシステム停止時間を最適化した上で，

鍵の切り替えに要する時間と計算資源のオーバーヘッドを準最適化するような手法の提案を行った．

最後に，既存の SPE を用いた提案手法の実装を行った．既存の SPE として，我々の研究室で開発を行った SS*（エスエススター），及び商用の SPE である uCSDP（uCosminexus Stream Data Platform）を用い，それぞれにおいて実現可能な演算と生じる制約について検討した．また，SS*における暗号化ストリームデータ処理方式を実現するための追加機能の実装を行った．

4.3 トランザクショナルストリーム処理

スマートフォンより継続的に送信される位置情報や Web サーバにおける各種ログをはじめとするデータストリームの分析処理に注目が集まっていた．データストリームの分析処理では，データストリームの他にリレーショナルデータやクラス分類器のモデルデータなど様々な外部リソースを参照し，その結果を集計することが多い．しかし，こうした処理の最中に外部リソースが更新された場合，一つの集計結果を計算する過程で参照した外部リソースが一貫したものではなくなるおそれがあった．

本研究では，この問題を防ぐことのできる処理パラダイムとして，トランザクショナルデータストリーム処理を提案した．トランザクショナルデータストリーム処理は，継続的クエリにおいて連続的に実行されるリソースの参照処理群を継続的クエリ由来トランザクションと呼ばれるトランザクションへと割り当てる．そして，これらのトランザクションとリソースを更新するトランザクションの間で適切に同時実行制御をおこなうことにより，一貫性の問題を解決した．

本研究の前半では継続的クエリ処理の中で同時実行制御をおこなう方式を三つ提案し，実システムにおける実験によってそれらの有用性を示した．

本研究の後半では，トランザクショナルデータストリーム処理の効率化に取り組む．トランザクショナルデータストリーム処理では処理結果の不整合を防ぐためにオペレータの再実行処理がおこなわれ，性能低下の要因となる．これに対し，本研究では再実行されるオペレータ数がオペレータの実行順序に依存することを明らかにし，オペレータスケジューリングへ制約を付与することでオペレータの再実行回数を削減する方式を提案した．そして実験により，既存のスケジューリング方式に提案する制約をもうけた場合に再実行されるオペレータ数が削減され，制約をもうけなかった場合と比較しスループットが最大で 5.2 倍となることを示した．

4.4 機械学習とストリーム処理の結合

ストリームからの異常検出は重要である．例えば，パケットストリームから異常アクセスを検出することで侵入検知を行ったり，温度データストリームから異常温度を検出して観測対象の異常を発見したりする．

多数のマルウェア検知手法を効率的に運用することは容易ではない．運用作業は煩雑であるし，手法数の増加に伴い性能が劣化するからである．複数のマルウェア検知手法を運用する際，ナイーブな方式として，各手法を別のプログラムとして実装し，別々にコンパイルして，それらを別のプロセスとして動作させることが考えられる．この方式は単純であるため実現が容易であるが，次の 2 つの問題が存在する．

第一の問題は，管理に関する問題である．プログラムの整理・起動が煩雑であることに加えて，処理結果を受信するインターフェースが統一化されているとは考えられないため，処理結果を利用したプログラムの作成は複雑・困難であると考えられる．

第二の問題は，性能に関する問題である．システムの入力であるパケットストリームは別々の形式で処理されるため，いろいろなプロセスに同じパケットデータをフィードせざるを得ない．プロセス間通信にはシステムコールを要する．パケットストリームは非常に頻繁に到着するため，システムコールに伴う性能の劣化度合いを，小さいと看過することはできない．

これらの問題を解決するために，本研究ではストリーム処理システムをベースにしたマルウェア検知基盤システムを提案した．提案システムはパケットストリームをリレーショナルストリームとしてモデル化し，複数の分析手法を SQL ライクな問合せ言語により記述可能にした．これによりユーザは SQL を発行すれば結果をタプルストリームとして受信可能になる．

また，分析手法は連続的問合せとして一覧可能になり，分析手法の起動と停止は分析基盤を通して提供されるために実現され，ユーザプログラムから検知基盤へのインターフェースは統一化される．従って，提案基盤を用いることで，第一の問題は解決された．

複数のパラメータ設定に伴ってスケールに関する問題が発生する．マルウェアによるアクセスパターンをトラフィックにおける異常アクセスと見做すことがある．その異常アクセスを検知する手法として，時系列回帰分析は有力な手段の 1 つである．例えば NICTER においては Change Point Detection (CPD)と呼ばれる分析手法を用いて異常アクセスを検知し，異常アクセスをマルウェアによるアクセスと見做す研究が行われている．しかしながら，パラメータ設定が容易ではないという欠点がある．即ち，パラメータ設定を誤ると侵入検知が不能になる可能性がある．これを防ぐためには，異なるパラメータを持つ複数の CPD を並列に動作させることが求められる．

このような並列動作を高速化するために，

本研究では CPD の共有計算技法を提案した．SPS においては common sub expression を共有することで高性能を達成する手法が広く研究されてきた．しかしながらそれらの手法が対象とする演算はリレーショナル演算に限定されてきた．本研究ではリレーショナル演算ではない CPD について，その内部処理である SDAR アルゴリズムを分析し，共有可能な部分について考察した．

提案手法を用いるシミュレーション実験を行い，パラメータである忘却率と AR 次数が等しい複数の CPD を実行する場合，最大で 500%程度の性能向上率が観測された．しかしながら，忘却率が等しい複数の CPD を実行する場合，最大で 104%程度の性能向上率しか観測されなかったことから，パラメータである AR 次数と移動時間が異なる複数の CPD を実行する場合において，より効率的な手法を考える必要があることがわかった．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計　2　件）

1. Eric S. Fukuda, Hideyuki Kawashima, Taro Fujii, Koichiro Furuta, Tetsuya Asai, Masato Motomura, C-based Design of Window Join for Dynamically Reconfigurable Hardware, Journal of Computer Science and Engineering, Vol. 20, pp. 1-9, 2013.
2. Masafumi Oyamada, Hideyuki Kawashima, Hiroyuki Kitagawa, Data Stream Processing with Concurrency Control, ACM SIGAPP Applied Computing Review, Vol. 13, pp. 54-65, 2013. DOI:10.1145/2505420.2505425.

〔学会発表〕（計　10　件）

1. 島貫稔之，川島英之，シーケンス演算子の効率的実装と適応的処理機構，コンピュータセキュリティシンポジウム，2014 年 10 月 15 日．
2. 川島英之，建部修見，分析用データ処理系における効率的なデータ配送機構，情報処理学会研究報告,2014 年 7 月 21 日．
3. 西村直孝，川島英之，リンク集約とパタンキャッシュを用いた複合イベント処理の高性能化，データ工学と情報マネジメントに関するフォーラム，2014 年 3 月 3 日．
4. Naotaka Nishimura, Hideyuki Kawashima, Accelerating CEP queries over Uncertain Data with Event Pruning and Link Aggregation, WebDB Forum, Nov. 27, 2013.
5. Naotaka Nishimura, Hideyuki Kawashima, and Hiroyuki Kitagawa, A High Throughput Complex Event Detection Technique with Bulk Evaluation, 5th International Workshop on Streaming Media Delivery and Management Systems, Oct 28, 2013.
6. Oge Yasin, Takefumi Miyoshi, Hideyuki Kawashima, Tsutomu Terada, A Fast Handshake Join Implementation on FPGA with Adaptive Merging Network, International Conference on Scientific and Statistical Database Management, July 31, 2013.
7. 川島英之，In-DSMS 分析システムへ向けて，電子情報通信学会 ASN 研究会，2013 年 5 月 16 日．
8. 小山田昌文，川島英之，北川博之，トランザクショナルなストリームデータ処理の実現方式，電子情報通信学会データ工学研究会，2012 年 8 月 1 日．
9. Katsuhiro Tomiyama, Hideyuki Kawashima, Hiroyuki Kitagawa, A Security aware Stream Data Processing Scheme with Encryption, International Workshop with Mentors on Databases, Web and Information Management for Young Researchers, July 31, 2012.
10. 大桶真宏，川島英之，北川博之，ストリーム処理システムを用いたマルウェア検知基盤システム，マルウェア対策研究人材育成ワークショップ 2012，2012 年 10 月 30 日．

〔図書〕（計　0　件）

〔産業財産権〕
○出願状況（計　0　件）
○取得状況（計　0　件）

〔その他〕
ホームページ等

６．研究組織
(1)研究代表者

川島　英之（KAWASHIMA, Hideyuki）
筑波大学・システム情報系（計算科学研究センター）・講師
研究者番号：90407148

(2)研究分担者
（　　）
研究者番号：

(3)連携研究者

佐久間　淳（SAKUMA Jun）
筑波大学・システム情報系・准教授
研究者番号：90376963
猿渡　俊介（SARUWATARI Shunsuke）
静岡大学・情報学部・助教
研究者番号：50507811