

追記・参照型データ管理プラットフォーム アーキテクチャの提案

赤間 浩樹 内山 寛之 三浦 史光 西岡 秀一
内藤 一兵衛 谷口 展郎 山室 雅司 櫻井 紀彦

日本電信電話株式会社 NTT サイバースペース研究所

センシング機器（センサ）や IC タグ等から発生する多量の情報、携帯機器の発達により収集される様々な個人の活動履歴情報、ブログなどコンシューマが生成する多量の情報など、ユビキタス社会の進展に伴って時々刻々と多種多様な情報が生まれ、ネットワークに流れ込んでいる。

これら多種多様で追記型の情報に対して、適切に蓄積・管理するとともに、利用者の目的に合った形で情報を統合し、検索・分析ができるためのデータ管理プラットフォームが必要になる。

本論文では、このプラットフォームのアーキテクチャとして、従来の蓄積型のデータを扱うデータベース管理システム(DBMS)とストリーム型のデータを扱うデータストリーム管理システム(DSMS)を統合した新たなアーキテクチャを提案する。また、そのプロトタイプシステムの概要といくつかの適用例のイメージを紹介する。

Data Management Platform Architecture that Specializes in Stream Addition and Reference

HIROKI AKAMA, HIROYUKI UCHIYAMA, FUMIAKI MIURA, SHUICHI NISHIOKA,
ICHIIBE NAITO, NOBUROU TANIGUHI, MASASHI YAMAMURO and NORIHIKO SAKURAI

NTT Cyber Space Laboratories, NTT Corporation

In the upcoming ubiquitous computing world, a lot of data such as sensing data, RFID logs, cellular phones logs and web logs are generated every moment, and they flow into the network, which is called 'information explosion.'

In this paper, we propose a new architecture of a data management platform that integrates database management system (DBMS) and data stream management system (DSMS). In the proposed architecture, we focus on flexibility and scalability of the system for real time aggregation and matching, as well as storage, retrieval and analysis of the enormous and various data for many different purposes. We describe the overview of a prototype platform system and applications implemented on the platform.

1. はじめに

情報の氾濫や爆発が言われて久しいが、情報増加の勢いは留まるどころを知らず、至る所で情報は増え続ける一方である。そして、環境にもセンサが埋め込まれ、個人携帯機器が情報を蓄積するようになる今後、その情報の増加は益々加速していくと予想される。

一方、ストレージの低価格化も驚くべきスピードで進んでおり、これまで蓄積できなかった生データを全て蓄積することも不可能ではなくなりつつある。

このような社会背景から、情報爆発時代に向けた新しい IT 基盤の研究[1]プロジェクトの開始や、各種センサ向けプラットフォーム (PF) の研究開発[2, 3]、ネットワーク・フォレンジックツールなどのログ記録システム開発などが活発化している。

我々の研究グループではネットワークに大量発生する時

系列ログ情報を、収集・構造化・半永久保存し、複数の情報を同期・相互連携し、企業や個人の各々の目的・条件に応じたフィルタリングや、時間・空間を跨った検索などを可能にする情報保管庫 (Lifebase) サービスを目指して、データ管理システム (以下 DMS: Data Management System) を研究開発している。

1.1 情報統合管理サービス PF

我々の目指すサービスを実現するための PF イメージを図 1 に示す。図の下側には情報源となる各種のセンサ群があり、図の上側には情報を活用する応用サービスが存在する。従来は、情報源と各種応用サービスは密に結びき、個々に構築・運用されていたが、社会的インフラとも言える情報源の増加に伴って、それら情報源を複数の応用に活用するというニーズが高まるのは必然である。そこで、我々の DMS は、それら情報源と応用の間に存在し、時間や空間を越えて情報を仲介接続し、新たな価値を創出することを

目指す。

1.2 DMS への要求

情報統合管理サービス PF を目指す DMS に対して 4 つの要求がある。

(要求1) 多様な情報源への長期的対応能力

情報源は、図 1 に示すように物流情報、ヘルスケア情報、システム情報、道路交通情報、気象・環境情報といったセンシング情報など多岐にわたり、更に今後も新しい情報源が増加していく。それら情報源の多様性に対応可能であり、恒久的なインタフェースが求められる。

(要求2) 大量に発生する情報への長期的対応能力

全国で時々刻々と発生する情報は莫大なものとなる。その大量情報を受け付けるだけの能力、そして受け付けたものに対する各種処理能力が求められる。さらに、PF としての長期運用を前提とするため、負荷の増加に対してシステムが追従していく能力、機能の追加に対してシステムが追従していく能力、システムの故障の影響を局部的に留めてサービスを継続する能力が求められる。

(要求3) 統合データの活用・価値創出力

情報源から収集した情報に対して、様々な処理を行って新たな価値を創出する部分が必要になる。その価値の創出の本質的機能は、データマイニング機能である。そこで、PF には、その機能を組み込める能力が必要となる。また、マイニング対象は、現在流れているストリームデータに対して行うケースと、過去の蓄積データをも含めて行うケースがあり、それらの制御を統合して実行できる能力が求められる。

(要求4) 多様な応用サービスへの長期的対応能力

多様な情報源から集めた情報は、多様な応用先で利用されることで PF としての価値が生まれる。そこで、標準的な出力インタフェースが求められる。また、情報源のデータスキーマ変更との独立性を実現した恒久的なインタフェースが求められる。さらに、応用サービスの増加に対して、必要な情報を複製や分岐により配信したり配信先を変更する能力も求められる。

以下、本論文では、上記の要求解決に向けた DMS アーキテクチャを 2 つの分散軸によって示し、機能概要と特徴を紹介する。また、本アーキテクチャに基づいて実装されたプロトタイプを紹介し、応用例や評価実験に基づき DMS の今後の課題を考察する。

2. 追記・参照型 DMS アーキテクチャ

2.1 データの流れの方向に対しての分散 (追記部、フィルタ部、ビュー部)

図 1 に示したように、DMS アーキテクチャの特徴は、まず情報源から発生するストリームデータに対しては追記型であり、かつ、応用側に対しては参照型という点にある。各種情報源からの入力データは時々刻々と追記される。これはインクリメンタルな時系列情報であり、更新や削除よ

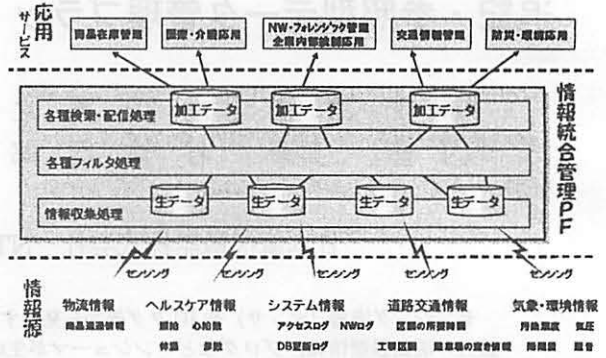


図 1 情報統合管理サービス PF

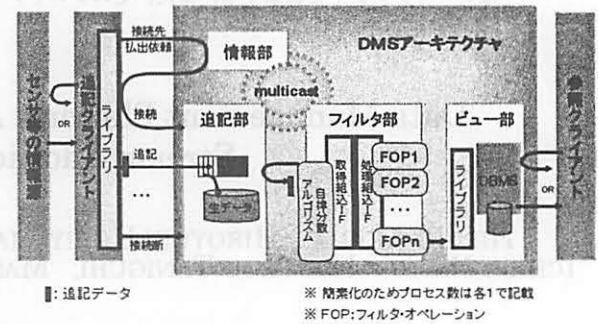


図 2 DMS アーキテクチャ

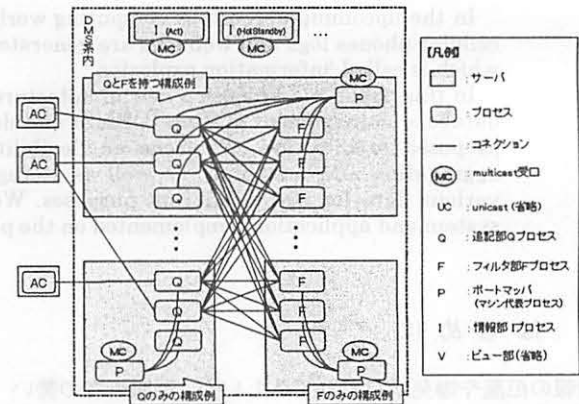


図 3 分散プロセス構成例

り追記が重視される。時系列情報は系内に将来の利用のために生データとしての保存が必要になる。一方、応用側には追記データを加工したデータを持たせ、参照を可能にしなければならない。そして、DMS アーキテクチャは、生データから価値を創出し加工データとするためのフィルタ処理が必要になる。そこでこれら図 1 に示した PF 内の 3 層にそのまま対応する各機能層を追記部、フィルタ部、ビュー部と呼ぶことにする。DMS は、この 3 つの基本機能を各々分散するプロセス構成で実現し、系内の任意のサーバに分散配置できる図 2 のようなアーキテクチャを採用。以下、各部の主な機能と、機能間の関係を説明する。

(1) 主要プロセスの構成

DMS を使ったサービス構築は、DMS 管理者、情報源側の追記クライアント (AC) 作成者、フィルタの処理実体であるフィルタ・オペレーション (FOP) 作成者、応用側のビュー利用者の 4 者の共同作業によって行われる。特に情報源は情報源 ID (SID) を持ち、その ID を使って DMS 内部の情報と対応付けが行われ、情報源のデータ型、情報源のデータに対する FOP の処理系列と各 FOP への引数などを DMS 側が管理する。系は主に 4 種類のプロセス群で構成される。

追記部(Q プロセス): AC から入力されたデータを受け取り、時刻印を付与し、生データとして DISK に書き込み、キューとして管理する。AC コネクションと Q プロセスは 1 対 1 で、1 つの Q プロセスに 1 種類のデータスキーマが対応する。なお、DISK 書き込みデータはローテーションされる。

フィルタ部(F プロセス): F は Q からデータを取得する前半部分と、受け取ったデータを処理する後半部分からなる。前半部分は自律分散アルゴリズムによって対象 Q を決定し、Q からデータを Pull する。後半部分は受け取ったデータに対応する SID に登録された FOP 系列を順に実行する。F はデータストリーム管理システム (DSMS [4-7]) として件数ベースの Sliding Window 処理を可能にしている。また系内の全ての F が同じ能力 (FOP 集合) を持つ。

ビュー部(V プロセス): データを応用に提供するものであり、既存の DBMS 等を組み込む。

以上が、主な 3 つの機能に直接対応するものであるが、DMS はさらに情報部と後述のポートマップを持つ。

情報部(I プロセス): 系内のプロセス構成をマルチキャスト (MC) によるハートビートでゆるやかに把握し、系内各マシンの起動コンフィグや SID 情報、更新用のフィルタ実行ファイル (F バイナリ) を管理し系内で http 公開する。AC からの接続要求に応じて Q プロセスの払い出しを行う。

(2) 分散プロセス間のデータ処理の流れ

AC からの追記データは図 2 に示したプロセス間の連携によって処理されていく。AC は I プロセスから接続先 Q プロセスの払い出しを受け、そこにデータを送信する。Q プロセスは、キューとしてデータを管理保持する。F プロセスは、組み込まれた自律分散アルゴリズムによって接続先 Q プロセスを決定する。Q プロセスに溜まったデータは、F プロセスによって Pull され、SID に指定されている FOP の系列に従ってフィルタ処理が順に実行される。その処理が終わると FOP はまた次のデータの取得を行うことを繰り返す。FOP では一般には各種フィルタ処理が実行されるが、汎用的な処理を記述定義できるため、V プロセスへのデータ送信も FOP として位置づけている。V プロセスでは FOP から受け取った加工データを蓄積し、参照クライアントの要求に応じて提供する。

2.2 データ増加の方向に対しての分散

DMS はもう一つの分散軸として、追記部およびフィルタ部について各々分散プロセス構成を採用する。図 3 はその構成例を示しており、Q プロセスや F プロセスは各マシンに自由に配置できる。その他、各マシンには Q や F の管理者としてポートマップ(P プロセス)が 1 プロセス存在し、系内のマルチキャスト(MC)を受信し情報部とマシン内プロセスの情報仲介、マシン内の F や Q プロセスの起動、次節で述べるような F バイナリ入替えの処理を行う。

追記部およびフィルタ部が複数のプロセスで構成されるため、それらを使って負荷の増加に対応することが可能になる。

(1) 追記データの増加への対応

AC から書き込むデータが追記部の能力を超える場合、AC は複数の Q プロセスとの接続を使って分散追記することができる。また、Q プロセス用のマシンを追加導入することもできる。

(2) フィルタ処理の増加への対応

F プロセスの能力が不足する場合には、系内に新たなマシンを導入し、系内の F プロセスを増加させることができる。

しかし、実際には、追記分散されたデータが最終的に同じビュー部へ辿り着くこと、負荷に偏りのあるキューに対して系内に追加された F プロセスがバランス良く処理を行えること、突発的に処理に時間がかかる F プロセスが発生しても他の F プロセスがキュー上に残るデータを処理できることなどが求められる。それらは、先に述べた 2 つの分散軸が揃って始めて実現できるようになる。

2.3 追記部・フィルタ部・ビュー部の独立性

2 つの分散軸を組み合わせ、系内のデータの流れに 2 回のネジリを発生させることで、追記部、フィルタ部、ビュー部の各機能層は、個々にサーバやプロセスを増加させることが可能になる。ここで 1 回目のネジリとは、追記部とフィルタ部の間の動的関係付けの仕組み (自律分散アルゴリズム) であり、これにより情報源とフィルタ間の独立性を実現する。2 回目のネジリとは、フィルタ部とビュー部の間の動的関係付けの仕組み (ビュー送信用フィルタ・オペレーションの動的組込み) であり、これによってフィルタと応用向けビュー間の独立性を実現する。これらの独立性は、下記のような負荷のロードバランスやシステムのスケールアウト、フィルタ処理や応用サービスの進化適応能力および持続性を実現し、PF への要求である各種長期的な対応を可能にする。

(効果 1) ロードバランス 1: AC が分散追記すること、およびフィルタが自律的にキューを選択肢処理することで、情報源の多様な負荷に対応できる。

(効果 2) ロードバランス 2: フィルタが自律的にキューを選択肢処理することで、フィルタ処理の多様な負荷に対応できる。

- (効果 3) **スケールアウト1:** 追記部の増設によって、情報源の負荷の進化 (増加) に対応できる。
- (効果 4) **スケールアウト2:** フィルタ部の増設によってフィルタ処理の負荷の進化 (増加) に対応できる。
- (効果 5) **進化適応1:** 新たな SID を持つ AC によって、情報源のスキーマの進化 (カラム追加) に対応できる。
- (効果 6) **進化適応2:** SID に対応する FOP 系列の追加や変更によって、フィルタ処理の進化 (フィルタ機能の追加) に対応できる。また、FOP 自体を新しく追加する場合には、FOP を追加した F バイナリの動的入れ替えによって対応できる。
- (効果 7) **進化適応3:** FOP 系列の追加や変更、FOP 自身の追加によって、応用サービスの進化 (応用先追加) に対応できる。
- (効果 8) **持続1:** 自律的なフィルタが処理を行うことによって、各種スケールアウトや各種進化適応の際に、既存の他の情報源を使った応用サービスに大きな影響を与えない。
- (効果 9) **持続2:** 追記やフィルタ処理を担当する一部のマシンに障害が発生しても系として停止せず、既存の他の情報源を使った応用サービスに大きな影響を与えない。

これらの効果を出すために自律的なフィルタ部は以下の処理を組み込む。

自律分散データ取得アルゴリズム: DMS は、フィルタ部が追記部に自律的にデータを取りに行くことによって、フィルタ処理のロードバランスやスケールアウトが可能なアーキテクチャになっている。自律分散によるフィルタ部からのデータ取得部分はアルゴリズム交換可能な作りになっており、様々なアルゴリズムの組み合わせが可能になっている。標準提供されるアルゴリズムとしてはラウンドロビンがあり、その効果は次章で述べるが、今後、様々なアルゴリズムの比較検証を行い別途報告する予定である。

FOP の動的追加と F バイナリの動的入れ替え処理: フィルタ部の処理組込インタフェース (IF) に従って新たな FOP を組み込んだフィルタの実行ファイル (F バイナリ) を作成する。その F バイナリを情報部に登録し、新たに追加した FOP を利用する SID 情報について FOP 系列の変更と F バイナリのバージョンの変更を行っておく。F プロセスは、Q プロセスからデータを取得する際にそのデータの処理に必要な能力 (バージョン) を自らが持っているかを判断し、バージョンが古い場合は、データの処理をあきらめ (他の F に譲る) て、マシン内代表プロセスであるポートマップを経由して情報部に登録されている最新の F バイナリを取得しプロセスの update を行う。

2.4 多彩な連携

DMS には PF として多様な情報源、フィルタ処理、応用処理への対応が必要になる。これらに対して DMS は図 2 に示すような各種連携のライブラリやインタフェースを

持つ。

多様な情報源への対応: 多様な情報源への対応は AC で吸収する。そのためのライブラリを提供している。

多様なフィルタ処理への対応: フィルタ部にユーザ定義処理 (フィルタ・オペレーション) の処理組込を可能にするインタフェースを提供する。

多様な応用処理への対応: フィルタから情報を受取りビュー側のプログラムに渡すライブラリを提供している。このライブラリを使って、既存の DBMS をビュー部として DMS の系内に取り込むことができ、さらに、ビュー部としてメールサーバなどとの連携をすることも可能になる。

3. プロトタイプの現状と考察

図 2 に示したアーキテクチャに添ってプロトタイプ DMS は設計され、動作している。開発言語は Java クライアント用ライブラリを除き C++ を使用し、IA マシン (EM64T) 上の Linux Fedora Core で動作している。分散構成を基本とする DMS は 20 台の環境での動作実績があるが、1 台構成でも動作可能でありシステムのスモールスタートが可能になっている。

3.1 標準で提供される組込対象

(1) **FOP:** 標準提供される FOP 用テンプレートとして、簡単な SQL 文でフィルタ条件を指定可能な SQLFilter と、処理結果を View ライブラリに送信する ViewFilter がある。SQLFilter は入力データに対して選択 (selection) と射影 (projection) からなる連続問合せ (Continuous Query) を可能にする。選択は引数の変更によって処理変更が可能であるが、射影はデータ列構成が変わるため FOP 自身の変更と F バイナリの変更によって処理変更を行う。

例) FOP

```
SQLFilter ("SELECT name, phone FROM myStrm WHERE age>20")
```

```
ViewFilter ("129.xx.xx.xx:12300:MyView")
```

(2) **ビュー部:** 標準提供されるビュー部としては PostgreSQL を内包するビュー部、および、リングバッファを用いた追記型の実体化ビュー部 (SimpleView) がある。

3.2 DMS の適用例

以下、現在構築中の 2 つの実験システムに関して、DMS のマッピング概要と、DMS に期待する効果について述べる。

(1) システム監査

背景: 2006 年 6 月成立の日本版 SOX 法によって産業界が内部統制・監査の対応を急いでいる。

期待する効果: 各種ビジネスシステムが出力するアクセスログ、DBMS に対する操作ログ、システム間の通信パケットなどを追記部の入力として、フィルタ部で不正利用パターンを検出し、ビュー部に警報を上げる。多数のシステムに対して適切な規模の監視システムを構築するためのスケールアウト能力や、監視ルールを FOP として動的に追加する構成の実現が期待される。

(2) 映像監視

背景: 最近の凶悪犯罪の増加に伴って社会インフラとして監視カメラの整備が期待されている。

期待する効果: 監視カメラの映像を追記部の入力とし、フィルタ部で非定常状態を検出し、ビュー部に警報を上げる。監視カメラの拠点の増加や解像度の向上による追記負荷の増加に上手くシステムの能力を追随させる能力や、映像内の顔検出などの画像解析フィルタ[10]の処理負荷増大に上手く対応できる構成の実現が期待される。

3.3 評価実験

映像監視試験モデルをベースに分散追記されたデータのロードバランス能力およびフィルタ処理のスケールアウト能力の確認を行った結果、Fプロセスを2台から10台まで増加させた状態で図4に示したようなスケラビリティが確認できた。

【測定条件】

- ・AC: JPEG ファイル (1000 個) 投入× (1 台 or 4 台)
- ・Q: 1 台、4 台 (各マシン 1 プロセス)
- ・F: 2 台→10 台 (各マシン 1 プロセス)
- ・V: なし
- ・Q での DISK 書込なし
- ・F の Sliding Window 幅は 2
- ・測定区間: データを Q に投入し全ての Q が空になった時点までの処理時間。

3.4 考察

以下、2章で示した DMS アーキテクチャの特徴に対する確認状況と残された課題を示す。

(1) 多様な情報源との連携 (システムログや Web カメラ映像 (JPEG))、多様なフィルタ処理の実現 (DB アクセスログからのシステムカタログ操作検出や変化量の大きな画像の検出)、ビュー組込ライブラリを使って PostgreSQL との連携に対応できることが確認できた。現在の制限としては並行に処理されたストリームデータについては、追記部で付与された時刻印または AC 側で付与したシリアル番号を使ってビュー部で sort する必要があり今後ビュー側ライブラリでの機能提供を図る予定である。

(2) 分散追記されたデータのロードバランス能力およびフィルタ処理のスケールアウト能力については、入力データが一樣である場合に、ラウンドロビンでも十分に効果がでていることは確認できた。ただし、情報源の多様な負荷やフィルタの多様な負荷への対応については今後の確認が必要である。また、実験の際、ビュー部がボトルネックとなる事象も確認できた。DMS はフィルタ部に処理をできるだけ寄せる利用法が並列度を上げるための理想である。ただし、フィルタ率が悪いなどでビュー部への書き込みがボトルネックになる場合には、ビュー部を水平分散させ、そのビューをアクセスする側 (または、マルチ DB 技術) で対応する必要がある。そのための DB 切り替え機構も課題として残る。

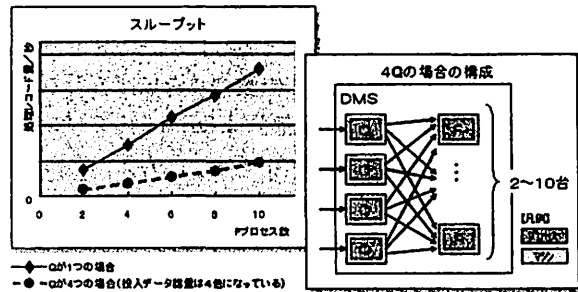


図4 Fプロセス数と処理性能

(3) フィルタの処理の進化 (フィルタ機能の追加) の対応についても、FOP の追加、F バイナリの自動更新で対応可能であることを確認した。ただ、現在はライブラリ等のインストールを含む FOP の配布や異種環境向けの対応などに課題を残しているため、今後改善を図る予定である。

(4) 情報源のスキーマの進化 (カラム追加) はビュー側対応、FOP 対応、SID の対応、追記クライアントの対応の順に手順を踏むことで対応できる。また、応用サービスの追加の場合には、ビュー追加、FOP 追加、SID 情報の FOP 系列修正、SID 情報更新通知という手順で対応できることを確認した。情報源の進化と応用サービスの追加の間の調停の仕組み等が今後の課題となる。

3.5 DMS のポジショニング

DMS は、従来の蓄積型のデータに対する検索を扱うデータベース管理システム (DBMS) と、ストリーム型のデータに対するフィルタリング処理を扱う分散データストリーム管理システム (DSMS) とを統合したアーキテクチャと考えられる。そこでまず、DMS を、既存の概念と比較することによって位置づけることを試みる。

(1) DMS の DSMS 部分と既存の DSMS の比較

DSMS[4-9]は、ストリームデータをタイムスタンプが付いた無限のタプル列とみなし、事前に指定した問合せを繰り返し適用し続ける連続問い合わせ (Continuous Query) という仕組みや、ストリームから有限のリレーションを抽出し処理するスライディング・ウィンドウという仕組みを持っている。DMS も SQL フィルタを持ち、条件として SQL の選択や射影を指定することができる。また、FOP として Tuple-based Sliding Window 処理も可能にしている点で、DSMS としての基本的能力を持っている。

(2) DMS の DSMS 部分と既存の DBMS の比較

DSMS としての DMS は、その応用領域の特性から DBMS の持っていた企業ビジネス (バンキング) 系トランザクション対応や、UPDATE、DELETE など多くの SQL 機能を落としている。また、過負荷に対しては即時応答を求めず、遅延を許容する仕組みとしている。代わりに、分散処理を取り入れて、データの追記などの各種処理のスケールアウトを可能にしている。

(3) DMS 全体と既存の DBMS との比較

DSMS と DBMS とを統合した DMS は、DBMS を置き換えるものではなく、DBMS の活躍の場を広げるものになっている。その場合、DSMS 部分は DBMS のフロントエンド部を拡張した構成と位置づけられ、入力データに対する能力を拡張したものとなる。

3.6 関連研究

以下ではより具体的に個々の研究との比較を行う。

DSMS の Aurora[5]や STREAM[4]は、ストリームデータの到着レートが過剰でストリームの処理が困難であると判定した場合に、到着データに対してユーザが定義する QoS に基づき自動的にフィルタリングを行い処理するデータ量を減らす load shedding 機能を提供している。これに対し、DMS は、追記クライアントが複数の Q に接続して分散書込することと、分散フィルタ処理することを可能にし、過剰なレートのデータ到着であってもストリーム処理を可能にする能力を提供している。ただし、現在の DMS はストリームの結合操作に対して Adaptive にアルゴリズムを切り替えるなどの仕組みはサポートしていない。Aurora の分散対応版である Aurora*[6]は広域ネットワークに対して通信の重みを考慮したストリームの流れ方向の分散を実現しているが、DMS のようなストリームの増加方向への分散は考慮していない。

KRAFT[3]は、データの鮮度と永続性の両立、類似シーケンス検索、周期的監視を可能にするセンサデータベースを実現している。多様なセンサデータを対象とし、センサの利用形態としてオンライン(周期監視)とオフライン(解析のための事後検索)の統合を狙う点は DMS の狙いとも類似する。しかし、KRAFT は永続性をリモートメモリを使うことによって実現している点、RDBMS の拡張として上記の機能を拡張している点、分散構成が DB サーバとログサーバに留まる点などが大きく異なる。

Harmonica[2]は、ストリーム処理エンジンと DBMS を統合し、ストリームデータに対する問合せやストリームデータを履歴データとして格納する機能を提供する。SQL をベースとしたストリームの連続問合せ処理やウィンドウ処理を提案している。DBMS として既存の DBMS と連携でき、DB ラッパーを経由することによりリモート DB との連携を考慮しているが、追記やフィルタ処理の過負荷に対して分散処理で解決を図るものではない。

DMS は Google File System [9]にいくつかの点で影響を受けている。それは、格安の PC サーバを多数並べて全体をスケールアウト構成にする点、多少のマシンの停止では系が停止しない可用性を実現する点、マシンの追加など運用コスト削減を狙う点、である。しかし DMS はセンサ・ビジネスの多様性を考え各種連携の仕組みを備え、DSMS として機能し、DBMS をも包含している点が大きく異なる。また、統合データの活用・価値創出力を狙っており、

マイニング対象が、現在流れているストリームデータだけではなく、過去に流れたストリームデータをも含み、それらの制御を統合して実行できる能力を目指す点が異なる。

広域で処理を分散し可用性も高める技術として P2P がある。たとえば PIAX[11]は、ユーザの位置や情報間の関係に基づくオブジェクトの発見と連携をスケーラブルに実現する PF であり、狙いとするユビキタスサービスのイメージは DMS のサービス適用イメージとも近い。現在の DMS はセンタ内に閉じている構成だが、今後は P2P 技術などとの連携した広域対応のあり方も考えていきたい。

4. まとめ

時々刻々と生成される各種センサデータに対して、情報を蓄積・統合し管理するサービス PF の構築に向けて、要求条件を洗い出し、ロードバランス、スケールアウト、進化適応、持続などの効果を持つ DMS アーキテクチャを提案し、プロトタイプでその効果を確認した。

今後、考察で述べた各種課題や、更なる応用の拡大に向けた機能拡充、運用コストを下げる観点、広域への対応などからアーキテクチャの精練と拡充を図っていきたい。

参考文献

- 1) 喜連川 優 他, “研究領域「情報爆発時代に向けた新しい IT 基盤技術の研究」”, <http://itkaken.ex.nii.ac.jp/iexplosion/>
- 2) 山田 真一, 渡辺 陽介, 北川 博之, “ストリーム処理とデータベースを統合した実世界情報管理基盤”, DEWS2006 5C-i7 (2006).
- 3) 川島 英之, 今井 倫太, 遠山 元道, 安西 裕一郎, “センサデータベースシステム KRAFT の設計と実装”, 情報処理学会論文誌, Vol.45, No.SIG14(TOD24), pp.39-53 (2004).
- 4) The STREAM Group, “STREAM: The Stanford Stream Data Manager,” IEEE Data Engineering Bulletin, 26-1 (2003).
- 5) D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul and S. Zdonik, “Aurora: A New Model and Architecture for Data Stream Management,” In VLDB Journal (12)2: 120-139 (2003).
- 6) M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, Y. Xing and S. Zdonik, “Scalable Distributed Stream Processing,” CIDR (2003).
- 7) 渡辺 陽介, “データストリーム処理システムに関する研究動向”, DEWS2004 ミニサーベイ (2004).
- 8) 白石 陽, “センサネットワークのためのデータベース技術”, 情報処理, Vol.47, No.4 (2006).
- 9) S. Ghemawat, H. Gobioff, S. Leung, “The Google File System,” ACM SOSP2003, pp29-43 (2003).
- 10) “遠隔映像モニタリング高付加価値化技術”, <http://www.ntt.co.jp/RD/OFIS/active/2006pdf/hot/ap/04.html>
- 11) 吉田 幹, 寺西 裕一, 春本 要, 下條 真司, “マルチオペレイと分散エージェントの機構を統合化した P2P プラットフォーム PIAX”, 2006-DPS-128, pp43-48 (2006).