ビッグデータをリアルタイムに処理するデータストリーム処理技術

喜田弘司　藤山健一郎　磯山和彦（NEC クラウドシステム研究所）

データストリーム分析とは？

ネットワーク技術，センシング技術などの発展により，新しいタイプのビッグデータ「データストリーム」が注目を集めている．ここで言うデータストリームとは，時間的に変化する大量のデータが日々刻々と生成・収集・活用される様子をデータの流れとしてとらえたものである．たとえば，株価などの金融データ，商品販売のような流通データ，Web サーバのアクセスログなどは典型例である．さらに，インターネット上のデータにとどまらず，実世界をセンシングしたデータや，人のライフログデータもデータストリームと見なせる．

現在，データストリームを活用したサービスの創出，価値創造への期待は大きく膨らんでいる．たとえば，クラウドに集められたスマートフォンやクルマの位置情報をリアルタイムに分析してユーザがおかれている状況を推測し，スマートフォンが秘書のように人の生活をナビゲートしたり，時々刻々と変化する交通状況を予測して最適な配達ルートを生成し業務効率化を図ることができる．

データストリームは，発生するデータ 1 つ 1 つは情報量が少なく活用が難しいが，それらを集めて時系列データとして分析することにより，さまざまな用途に活用できる．特に，時々刻々とデータが発生するデータストリームの特性を活かし，可能な限りリアルタイムに分析する技術が注目されている．本技術の特徴は，リアルタイム性を向上させるため，データへのアクセスをデータ発生時の逐次走査に限定する点にある．無限に発生し続けるデータストリーム全体を処理の対象とするのではなく，最近のデータだけを使って分析できるように工夫することにより，少ない計算機資源で，長期間安定して動作し続け，リアルタイムに分析結果をユーザに提供できる．

本稿では，データストリーム分析に関し，アルゴリズムの研究と，アルゴリズムを計算機上で実行する処理基盤の研究の 2 つの観点から解説する．

分析アルゴリズム

要件と戦略

データストリーム分析は，無限のデータストリームに対して，有限の計算機リソースだけを用いて，効率的な計算を行う点に本質的な難しさがある．たとえば，Web のアクセスログ解析において，「いつもと異なるアクセスパターンを検出する」という解析のケースを考える．いつもと異なるかどうかの判定は，過去のアクセスログと最近のアクセスログを比較する処理である．過去のアクセスログは増え続けるため，この比較処理の計算量は増え続けることになる．つまり無限のデータストリームに対しては破綻する．

こういったことがないように，データストリーム分析は以下の要件を満たす必要がある．

（要件 1）1 パス処理

データの走査をパスと呼び，1 つのデータに対して 1 度だけ走査を許す処理が 1 パス処理である．上記で述べたように，1 パス処理を適用しない場合には，無限のデータストリームを対象としているため，解析対象のデータ量は増え続けることになり，有限の計算機リソースを使い果たしてしまう．データへのアクセスは，データ発生時に逐次的に 1 回だけ走査することに限定する必要がある．

（要件 2）固定サイズ，小サイズメモリ

データストリームを解析対象にした場合，1つ1つのデータは小さいがその数が膨大であるために大きな計算機リソースが必要となることが多い．また，データストリームは，データ発生の流量が動的に変化するため，この解析には，最大流量に対応できる計算機リソースを予測し，準備しておく必要がある．また，長期間安定して動き続けるためには必要なリソースが一定である必要がある．

これらの要件を満たすために，ストリーム処理アルゴリズムの多くの研究では以下の戦略をとる．

「厳密解を計算することはあきらめて，近似解，確率的な解を求める」

一般に，解析精度と，解析に必要な計算機リソースはトレードオフ関係にあり，データストリーム分析アルゴリズムの研究では，解析精度より必要計算機リソースを優先する．これは，この技術の典型的な適用先では，正確な分析結果であるが分析時間がかかることよりも，分析結果の精度は落ちるが分析時間がかからないことを重要視しているためである．

この戦略をふまえ，データストリーム分析アルゴリズムの基本技法を述べる．**表 -1** に示した研究の多くのアルゴリズムでは，その工夫点に注目すると以下の 3 種類の技法に分類できる．

(1) **確率統計的計算**：データをサンプリングして確率統計をベースに解析結果を算出する．たとえば，整数カウンタを用いずに確率統計的にかぞえることで効率の良い計算が得られる [1]．

(2) **データ粗視化**：詳細な情報を得るためにすべてのデータをメモリに保持するのではなく，データをグループ分けしておおまかな情報，たとえば，グループの代表元や統計量などを保持することでデータを圧縮してメモリを節約する方式である．

(3) **適応的計算**：あらゆる解の可能性を最初から用意すると計算リソースが大きくなることが多い．そこで，最初は軽い計算により粗い解を求め，計算が進んで情報が増えてきたら，より詳細まで計算する戦略である．

処理内容	例	技法
異なり数 [1]	購買された商品の種類数	(1)
ホットリスト [2]	売り上げ最上位 K 個の商品	(1)
滑り総和 [3]	一定期間内の売り上げ総和	(3)
組合せ [4]	同時に買われる商品を検出	(3)
分類規則 [5]	購買傾向予測ルールを検出	(3)
クラスタリング [6]	売り上げ傾向が似た商品を検出	(2)

表 -1　ストリーム分析アルゴリズム

分析アルゴリズムの例

車両を動くセンサと見立て時々刻々と変化する車両情報（位置，速度，ワイパーの利用などであり，以下「プローブ情報」と呼ぶ）をサーバに集約し分析することにより渋滞情報などの交通情報をリアルタイムに提供する「プローブ情報システム」を例にデータストリーム分析アルゴリズムの実例を説明する．プローブ情報システムでは大量のクルマからのプローブ情報をいかに軽量に地図上の道路にマッチングさせるか（マップマッチング）が課題の 1 つである．これに対し，前記のデータストリーム分析の戦略を応用した（1）モザイクマッチング方式と（2）計算コスト可変近似方式の 2 つの方式で解決する．

（1）モザイクマッチング方式

本方式は，データ粗視化の技法を使った方式である．クルマと道路のマッチングは，交差点付近では細かくマッチングを行う必要があるが，それ以外ではおおざっぱなマッチングで十分であることが多い点を利用して粗視化する．

道路は交差点を意味するノードと，交差点から交差点までの道路セグメントを意味するリンクのネットワークで記述される．マップマッチングとはクルマの位置とリンクを対応させることである．

図 -1 の a から j の実線がリンクである．本方式のポイントはグリッドを数メートル程度の非常に細かなサイズにし，緯度方向と経度方向に平行にグリッド分けをすることである（図の格子）．このようにすることにより，ほとんどのグリッドは 1 つのリンクと重なり合っているために，プローブ情報の緯度，経度からグリッドを特定するだけでリンクが特定できる．たとえば，図 -1 の例では，クルマがいるグリッドはリンク b とのみ重なっているた

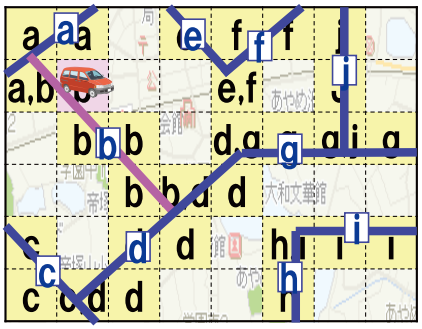

図 -1
グリッド分け
説明図

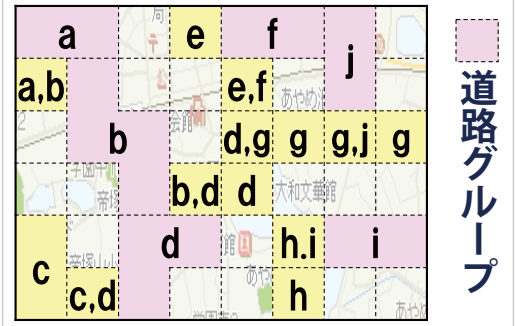

図 -2
グリッドの
結合説明図

め，リンク b と対応付ける．このように多くの場合，プローブ情報の位置と，リンクの距離計算など重い処理が不要となる．さらに，どのリンクがどのグリッドに属するかは事前に計算しておくことができるために，プローブ情報のマッチング時に計算する必要はない．一方，交差点付近では，1 つのグリッドに複数のリンクが重なっているため，プローブ情報の位置と，リンクの距離を計算してプローブ情報とリンクを対応付ける．

(2) 計算コスト可変近似方式

本方式は，確率的計算，適応的計算の技法を使った方式である．同じ道路を走っている車両はデータが似ている点を利用して，確率統計的手法により，精度と速度のバランスをとったサンプリングレートを計算する方式である．また，計算機のリソースに余裕がある場合は詳しく計算し，余裕がない場合にはおおざっぱに計算するといった適応的な手法を適用することで，精度と速度のバランスがとれた分析を実現する．

具体的には，まず，グリッドを地理的な接続関係でグルーピングする（**図 -2**）．直感的には，各リンクは交差点付近とそれ以外にグルーピングされる．図 -1 の例では，リンク b はリンク a との交差点付近と，リンク d との交差点付近と，これら交差点をつなぐリンクでグルーピングされる．同じ道路を走っている車両はデータが似ているため，同じグループ内のデータは値が似ている．この性質を利用し，処理が間に合わない状況を検出すると，データ数の少ないグループを優先的に解析し，データ数が多いグループは処理を省略する．このように，全体の解析の進捗状況から，グループごとにサンプリングレートを制御することで精度と速度のバランスをとることができる．

サンプリングレートなどのパラメータは，道路の長さ，データの時間変化で算出する [7]．

データストリーム処理基盤

データフロー計算モデル

データストリーム処理基盤は，商用の基盤 [8]～[10]，オープンソースの基盤 [11] などさまざま公開されている．

これらは，文献 9）のように独自の言語でプログラミングする場合もあれば，文献 10）のように SQL をデータストリーム用に拡張した言語を用いる場合もある．また，文献 8）のように C 言語など汎用的な計算機言語で構築する場合もある．注意が必要なのは，データベースを中心とした蓄積データのプログラミングと計算モデルが異なる点である．プログラマはデータの流れを活かしたシステムを構築するために，データフロー計算モデルを利用する（**図 -3**）．

本計算モデルでは，処理ノードと呼ばれる解析処理の単位を複数用意し，これらを処理する順にリンクすることで処理ノードネットワークを構築する．各処理ノードは，解析待ちのデータを記憶する待ち行列キューと，待ち行列キューのデータから一度に解析するデータセットである解析ウィンドウと，待ち行列キューのデータを解析する解析関数とから構成される．

各処理ノードの待ち行列キューへは，前にリンクされている処理ノードから非同期にデータが追加される．解析ウィンドウは，待ち行列キューの状態を常に

監視し，一度に解析するデータが揃えば，解析関数に対象データを渡し解析を依頼する．解析関数は渡されたデータを解析し，後段の処理ノードの待ち行列キューへ解析結果を書き込む．これを繰り返し，全体として解析が行われる．

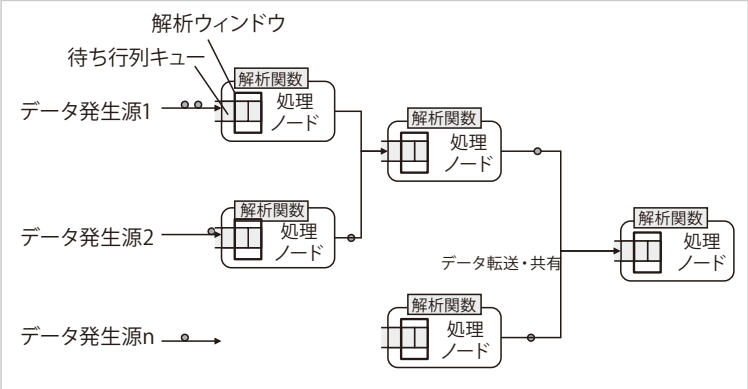

図 -3　データフロー計算モデル

データストリーム処理基盤 DSPP

データフロー計算モデルをコンピュータ上で実行させる DSPP（Data Stream Processing Platform）を紹介する[8]（**図 -4**）．DSPP は，アプリケーションに依存しない基本機能を提供する．この基本機能を使ってアプリケーションに依存する部分，すなわち，解析ウィンドウ，解析関数と，処理ノードネットワークを開発することでシステムを構築する．

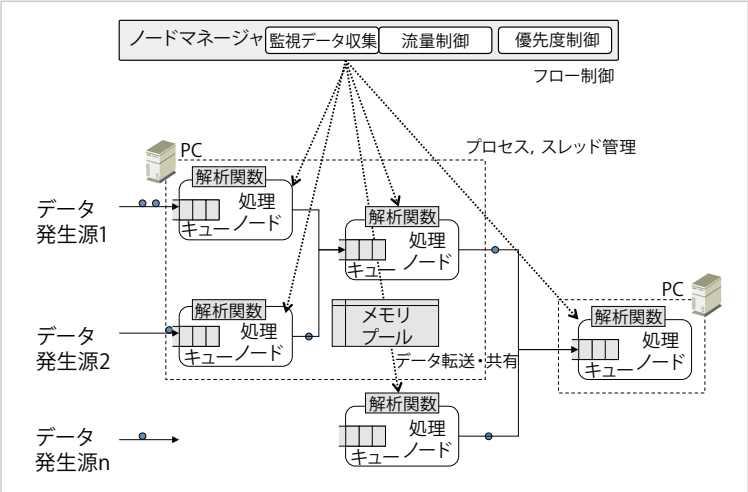

図 -4　DSPP アーキテクチャ

DSPP の基本機能は以下のとおりである．

- **処理ノードを実現する基本機能**：DSPP は処理ノードに共通の基本機能を提供し，アプリケーションを構築する際にはこの基本機能に独自の解析関数をプラグインすることにより各処理ノードを実現する．さらに，他の処理ノードへリンクする API（Application Program Interface）を利用して処理ノードネットワークを構築する．
- **処理ノードの実行プロセス，スレッド管理機能**：各処理ノードを非同期に独立に動作させる．
- **処理ノード間のデータ転送，データ共有機能**：小サイズのデータの逐次処理に特化した独自のメモリプールを利用して，高速にデータの受け渡しおよび，処理ノードのキューを管理する．
- **処理ノード間のフロー制御機能**：ノードマネージャと呼ばれるモジュールが，処理ノードネットワーク全体の処理の実行状態を管理する．ノードマネージャは，各処理ノードの待ち行列キューが溢れないように，あるいは系全体のパフォーマンスを向上させるために，各処理ノードへ流れるデータ量，処理ノードの処理優先度の制御を行う．

交通情報システムへの応用

実証実験概要

本章では，データストリーム分析を応用して実現した渋滞状況提供実証実験を紹介する．実証実験は，名古屋市にて実施されたプローブカーの実験であり，業務車両（タクシー）から，位置，速度，方向，ワイパーの動作などのデータを収集し，渋滞情報，旅行時間予測，降雨情報を提供するサービスを試行した．実験においては，プローブカーの台数は平均 1,700 台であり，ピーク時は最大 4,000 台が走行し，1 分間隔でサーバにアップロードし，5 分間隔で結果を更新した（**表 -2**）．

この実証実験では，データストリーム方式の有効性を検証するために，メモリデータベースを用いた従来方式と，データストリーム処理の方式の 2 つの方式で構築し性能比較を行った．

車両数	平均 1,700 台，最大 4,000 台
収集データ	位置，速度，方向
収集頻度	車両から 1 分間隔でアップロード
道路区間数	12 万区間
サービス	・5 分間隔で渋滞情報を更新・表示 ・渋滞を考慮して経路を探索
実行環境	CPU：Xeon2.4 × 2，メモリ：1.5GB

表 -2　実験環境

メモリデータベースの方式は，まずプローブデータをデータベースに蓄積し，これに対して 5 分間隔にバッチ処理により分析を繰り返す（図 -5）．一方，データストリーム方式では，各クルマからプローブデータがアップロードされてくるごとにデータドリブンで順に以下の処理を実行する．

- 地図との対応をとるマップマッチング
- データの欠損を補うための経路探索処理
- 統計処理による旅行時間計算

これらは，DSPP 上の処理ノードとして実装することにより，パイプライン処理される．また，特に処理が重いマップマッチングに関しては，すでに紹介したデータストリーム分析アルゴリズムであるモザイクマッチング方式と計算コスト可変近似方式により軽量に実現した．

性能評価

実験の結果を図 -6 に示す．横軸は，5 分間あたりの処理データ件数（処理データ投入数），縦軸は処理時間（秒）である．つまり，このグラフでは，縦軸の処理時間が 300 秒（＝ 5 分）を超えてしまった場合は，処理できないデータが発生していることになり，データベース方式の場合のその件数は，約 8,000 件である．一方，データストリーム方式では，余裕を持って処理できており，8,000 件のデータでも約 50 秒で処理できている．全体平均として，データベース方式より約 6 倍高速であった．

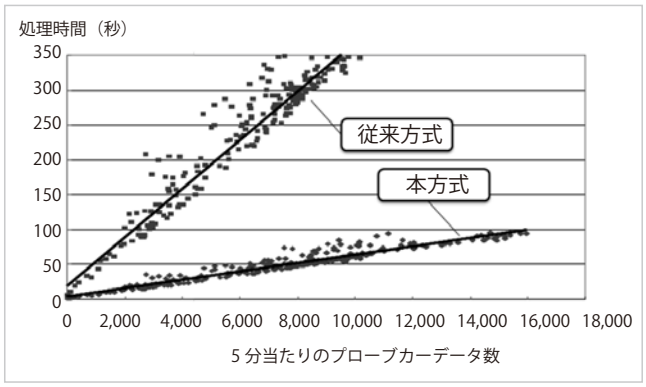

図 -6　タクシーによる渋滞情報システムの従来方式と本方式の処理時間比較

サービスイメージ

以上の実験結果をふまえ，データストリーム分析を応用した場合のサービスの変化を述べる．特に，渋滞情報は見える化するだけではなく，解消することが期待されており，この観点から考察する．

（1）きめ細かく渋滞情報を提供

図 -7 にデータベース方式と，データストリーム方式の渋滞情報表示の例を示す．データベース方式は，その性能値から算出すると，対象は主要幹線道路に限られ，範囲も大まかになる．実際に，渋滞を解消するためには，細街路も使い混雑している道路を回避するように経路案内する必要があり，主要幹線道路だけでは不十分である．これに対し，データストリーム方式では，細街路も含めた道路を対象に渋滞情報を生成することができる．

（2）鮮度の高い渋滞情報を提供

従来の多くの渋滞情報は，20 分から 30 分程度の間隔で更新されている．つまり，ドライバーへ提示されている情報は 20 分か

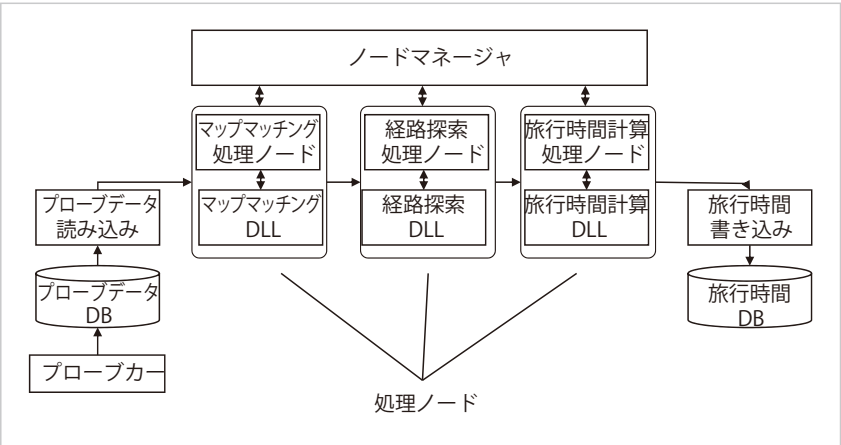

図 -5　DSPP によるシステム構成

図 -7　従来方式と本方式で提供する渋滞情報サービスの比較
（地図データはインクリメント P（株）の製品 MapDK を用いて作成されたものです）

ら 30 分前の情報である．渋滞を解消するためには，渋滞の起き始めの対処が重要であることが知られており，可能な限り渋滞情報の遅延を少なくすることが重要となる．これに対し，データストリーム方式の更新頻度は 5 分間隔である．このように鮮度の高い渋滞情報は，信号の制御や，カーナビの経路案内などにより交通をコントロールし渋滞を解消することができる．

展望

データストリーム処理技術に関して，分析アルゴリズムと実行基盤に関して解説した．また，交通情報システムへの応用例を紹介した．

今後の ICT（Information and Communication Technology）産業は PC や携帯電話といった ICT そのものの普及を目的としたものではなく，交通やエネルギー等の社会基盤産業と同期することでさまざまな社会問題の解決に貢献することが期待されている．この実現のためには，実世界の「今」何が起きているのかをリアルタイムに ICT で理解する必要があり，この実現としてデータストリーム分析技術は非常に重要な技術と考えられる．データストリーム分析技術が，人と地球にやさしい社会の実現に貢献できることを期待している．

参考文献

1）Flajolet, P. and Martin, G. N. : Probabilistic Counting, in Proc. FOCS'83, pp.76-82（1983）.
2）Gibbons, P. B. and Matias, Y. : Synopsis Data Structures for Massive Data Sets, in J. Abello, editor, External Memory Algorithms, Vol.50 of DIMACS Series in Discrete Mathematics, pp.39-70, DIMACS（1999）.
3）Babcock, B., Babu, S., Datar, M., Motwani, R. and Widom, J. : Models and Issues in Data Stream Systems, in Proc ACM PODS'02, pp.1-16, ACM（2002）.
4）Manku, G. S. and Motwani, R. : Approximate Frequency Counts over Data Streams, in Proc.VLDB'02, pp.346-357（2002）.
5）Domingos, P. and Hulten, G. : Mining High-speed Data Streams, in Proc. ACM SIGKDD'00, pp.71-80（2000）.
6）Zhang, T., Ramakrishnan, R. and Livny, M. : Birch : A New Data Clustering Algorithm and its Applications, Data Min. Knowl. Discov., pp.141-182（1997）.
7）喜田弘司，藤山健一郎，三津橋晃丈，中村暢達：次世代プローブ情報システム（2）～大規模高速マップマッチングアルゴリズムの提案～，情報処理学会，マルチメディア分散協調とモバイルシンポジウム論文集（2007）.
8）喜田弘司，藤山健一郎，今井照之，中村暢達：データストリーム処理による大規模プローブカーシステムの開発と評価，情報処理学会，高度交通システム研究会，Vol.2008, No.83（2008）.
9）IBM InfoSphere Streams, http://www-06.ibm.com/software/jp/data/infosphere/streams/
10）日立製作所： uCosminexus Stream Data Platform, http://www.hitachi.co.jp/Prod/comp/soft1/cosminexus/sdp/
11）Yahoo S4, http://incubator.apache.org/s4/

（2012 年 6 月 6 日受付）

謝辞　本研究開発成果の一部は，2008 ～ 2010 年度総務省委託研究「ユビキタスサービスプラットフォーム技術の研究開発」による．

喜田弘司（正会員）kida@da.jp.nec.com
日本電気（株）主任研究員．エージェント，検索エンジン，システムセキュリティの研究に従事．現在，M2M の研究に従事．博士（工学）．

藤山健一郎（正会員）k-fujiyama@cw.jp.nec.com
日本電気（株）主任．データストリーム処理の研究に従事．現在，M2M の研究に従事．

磯山和彦　k-iso@bc.jp.nec.com
日本電気（株）主任．現在，複合イベント処理（CEP）の研究に従事．