

# 高度なテレマティクスサービスを実現する データ活用プラットフォーム

塩田 尚基<sup>1,a)</sup> 菅 真樹<sup>1,b)</sup> 柴 純明<sup>1,c)</sup> 小林 大<sup>1,d)</sup> 一圓 真澄<sup>1,e)</sup> 伊賀 徳寿<sup>1,f)</sup>

## 概要 :

自動車の IT 化が進み無線ネットワークが発展することで、自動車向けのテレマティクスサービスが高機能化していくことが予想されている。多数の自動車から発生する大量のデータを収集・分析し、リアルタイムに利用するには、現在のデータ処理システムではさまざまな課題がある。本稿では、将来のテレマティクスサービスを実現するにあたって共通的に必要となる機能を考察し、それらの機能を備えたサービスプラットフォームを提案する。提案プラットフォームは分散ストリーム処理基盤、大規模データ分散処理基盤、および蓄積したデータを並列分散的に形式変換する処理基盤を備え、プログラム間の通信には Pub-Sub 形式の非同期通信を採用している。本稿ではさらに、提案したプラットフォームを利用してサービスアプリケーションを開発し、プラットフォームの有用性を評価する。

## 1. はじめに

近年の自動車の IT 化と無線ネットワークの発展により、様々なサービスが実現されてきている。例えば、自動車内に設置されたセンサ類から位置情報や車速、エンジン回転数などの情報（プローブ情報）を取得し、無線ネットワークを介してサーバに送ることが可能となったことで、自動車の状況にあった情報を配信することが可能となっている。実際の用途としては、自動車の周辺の天気予報や駐車場の満空情報などを配信したり、渋滞を検知してカーナビのルートを再設定するといったサービスが提供されている。

無線通信を用いた自動車向けのサービスはテレマティクスサービスと呼ばれ、自動車メーカーや IT サービスベンダが次期サービスの主力として開発に注力している。トヨタ自動車、日産自動車、本田技研工業、GM、ダイムラーなど、国内外を問わず様々な自動車メーカーが独自のテレマティクスサービスを提供している。

テレマティクスサービスにおいては、自動車から集めたプローブ情報を分析して新たな知見を得るといった側面も期

待されている。この有力な利用例として、自動車の情報を道路などインフラに設置されたセンサーの情報と組み合わせることで道路交通を効率化したり、都市計画に利用するといった用途が挙げられる。

テレマティクスサービスを実現するための情報処理プラットフォームについては、既に多くの研究開発が行われている。例えば、プローブ情報を集めて分析するプラットフォームである CarTel[4] や、Web インタフェースによる位置ベースサービスのプラットフォーム [5]、公衆無線 LAN のアクセスポイントを利用した情報配信サービスのプラットフォーム [1] などが提案されている。

その一方で、今後さらに自動車の IT 化と無線ネットワークの発展が進むことで、テレマティクスサービスが高度化することが予想されている。既に、自動車内に位置情報、走行距離、速度、加速度、時刻、エアバッグの動作状態、排気量、Wi-Fi 信号、Bluetooth 信号、OBD(On-Board Diagnostics) システム、カメラ、オーディオレコーダ、ドライバーの生体データ（心拍数など）などといった、多岐にわたるセンサ情報を利用したテレマティクスサービスに関して研究がなされている [3][8]。これらの先行研究においては、単にプローブ情報に含まれる情報の種類が増加するだけでなく、無線ネットワークの発展を背景として、画像や音声のようなリッチメディアデータがプローブ情報として利用可能となることが示唆されている。幅広いプローブ情報が利用可能になることで、多くの種類の、複雑な処理を伴うテレマティクスサービスが実現可能となる。

<sup>1</sup> NEC  
1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa  
211-8666, Japan

a) n-shiota@ay.jp.nec.com

b) kan@bq.jp.nec.com

c) sakae@bc.jp.nec.com

d) daik@ay.jp.nec.com

e) m-ichien@cd.jp.nec.com

f) n-iga@ah.jp.nec.com

しかしながら、このような高度なテレマティクスサービスを実現するにあたって、既存の情報処理プラットフォームで対応することは困難である。プローブ情報の量が増加し、また現在よりテレマティクスサービス対応の車両が増えることで、データ処理システムで扱うデータ量が爆発的に増加する。また、常に無線ネットワークが利用可能となることでリアルタイム性の高いサービスが求められ、システムの処理能力についての要求が高くなる。既存のプラットフォームはこのような環境下での利用を想定していない。

本稿では、多くの自動車で多種多様な情報がセンシング可能になり、随時ネットワークを介してサーバと通信が可能となった環境における、高度なテレマティクスサービスについて論じる。本稿は、様々な高度なテレマティクスサービスを構築するにあたり、共通的に利用できるプラットフォームを提案することで、高度なテレマティクスサービスの実現を加速させることを目的とする。

提案プラットフォームは、多数の自動車から集められる大量のプローブ情報を扱うため、分散ストリーム処理基盤、大規模データ分散処理基盤、および蓄積したデータを並列分散的に形式変換する処理基盤を備える。提案プラットフォームではさらに、新規サービスの追加を容易にするためプログラム間の通信に Pub-Sub 形式の非同期通信を採用し、プログラム同士の依存度を減らす構成としている。

本稿の構成を以下に示す。2節では想定される様々なテレマティクスサービスの例を挙げて、それらのサービスを実現するために必要となる機能について考察を行う。3節では2節の議論を踏まえて、テレマティクスサービスを構築するためのプラットフォームを提案する。4節では提案するプラットフォームを用いて実際に構築したサービス例について紹介する。5節ではサービスを構築する中で得られた知見をもとに、提案したプラットフォームについて評価・考察を行う。

## 2. 機能要求

本節では、将来的に実現が予想される高度なテレマティクスサービスの例を挙げ、それらのサービスを実現できる共通的なデータ処理モデルを考察する。

### 2.1 想定サービス

自動車の様々な情報がセンシング可能になり、常時通信が可能となることで具体的にどのようなサービスが実現可能となるか、いくつかの例とその実現方法を示す。

- コンテキストに応じた個別情報配信  
長時間運転して疲れているドライバーに対して周辺の休憩できる店舗や施設の情報を配信したり、燃料が少なくなった自動車に近隣の安いガソリンスタンドを紹介するといった、ドライバーのコンテキストに適した広告などの情報をタイムリーに配信するサービスが実

現可能になる。このとき、自動車からは現在の位置や連続運転時間、燃料残量といったセンサ情報をサーバに送信し、サーバでは登録されている広告情報とセンサ情報とを突き合わせて、配信に適した広告情報を自動車に送信するといった処理を行う。

- 緊急車両の運行ルート通知  
救急車や消防車などの緊急車両の通行ルートを、影響する自動車に対して予め通知できれば、緊急車両をスムーズに走行させることが可能になる。このとき、自動車からは現在の位置情報をサーバに送信し、サーバでは設定されている緊急車両の通行ルートと位置情報とを突き合わせて、影響のある車両を特定して通知を行うといった処理を行う。
- 運転の安全性向上  
事故を引き起こしそうな状態の自動車やドライバーに対して警告を出すことで、自動車の安全性を向上させることができる。このとき、自動車からは天候や路面状態、各種の自動車の内部センサの情報をサーバに送信し、サーバでは過去の事故事例データと送信されたデータを突き合わせて、危険度の高い車両に対して警告情報を送信するといった処理を行う。また、この処理とは別に、送信されたデータを蓄積しておき、事故が発生した場合に蓄積したデータを分析することで、事故事例のデータを新たに抽出するといった処理も考えられる。
- 渋滞回避ルート設定による渋滞の緩和  
自動車に対してリアルタイムな渋滞情報および渋滞を回避するルートを配信することで、早期の渋滞の緩和が可能になる。このとき、自動車からは現在の位置と進行方向といったセンサ情報をサーバに送信し、サーバでは渋滞の発生している道路の情報をセンサ情報と突き合わせて、渋滞の影響のある自動車に対して渋滞の情報と回避ルートの情報を送信するといった処理を行う。また、この処理とは別に、センサ情報を蓄積しておき、蓄積したデータを分析することで渋滞地点の抽出と渋滞回避ルートの計算を行うといった処理も行う。
- 道路開発計画のポイント抽出  
日時や天候による交通量の増加条件を予測して効果的な道路計画を行ったり、交通事故を起こしやすい道路の状態を計測して新しい道路の安全性を向上することが可能になる。このとき、自動車からは位置情報および天候や路面状態といったセンサ情報をサーバに送信し、サーバではセンサ情報を蓄積しておき、蓄積したデータを分析することで交通量が増加する条件や交通事故が発生する条件を抽出するといった処理を行う。

## 2.2 共通的な処理モデル

前述の各種サービスを実現する共通的な処理モデルを提案する。

データ処理を行うサーバの視点で見ると、自動車のセンサ情報がインプットの一つである点が共通している。センサ情報の内容は位置情報、燃料残量、路面状態などサービスによって様々であるが、自動車から送られる情報としてひとかたまりのデータと見ることができる。

このインプットに対して、処理の内容は大きく二通りに分かれる。一方は、インプットされたセンサデータに対して即座に適切な情報を返却する処理（リアルタイム処理）であり、もう一方はセンサデータを一旦蓄積して、蓄積したデータに対して分析を行う処理（蓄積データ処理）である。広告配信サービスや緊急車両の運行ルート通知サービスが前者の処理を、道路開発計画のポイント抽出サービスが後者の処理を利用して実現することができる。

一方、上記の二通りの処理を組み合わせたサービスも存在する。運転の安全性向上サービスや渋滞回避ルート設定による渋滞の緩和サービスは、リアルタイム処理と蓄積データ処理の両方を備え、連携させることで実現できる。すなわち、蓄積データ処理で得られた処理結果を用いてリアルタイム処理を行うという処理の組み合わせが必要である。

処理の結果得られるアウトプットについては、サービスによって性質が異なる。コンテキストに応じた広告配信サービスや緊急車両の運行ルート通知サービスではアウトプットのデータは自動車に送られるが、道路開発計画のポイント抽出サービスではアウトプットのデータは道路計画を行う自治体などに提供される。

以上の考察を踏まえて、サービスを実現するシステムの処理モデルとして図1のようなモデルを提案する。センサデータというひとつのデータのフローに対し、リアルタイム処理と、蓄積データ処理を並行して実行できるモデルとし、蓄積データ処理で得られた分析結果を随時リアルタイム処理に反映できるようなモデルとした。蓄積データ処理の結果を反映させる部分は、安全性向上サービスのようなデータの分析で得られた知見をリアルタイム処理にフィードバックするようなサービスを想定している。

## 3. 提案プラットフォーム

前節で述べた共通的なデータ処理モデルをもとに、高度なテレマティクスサービスを構築するプラットフォームを提案する。

### 3.1 プラットフォームの要件

様々な種類の高度なテレマティクスサービスを前節で述べた処理モデルで実現するにあたって、プラットフォームに必要とされる要件について考察する。

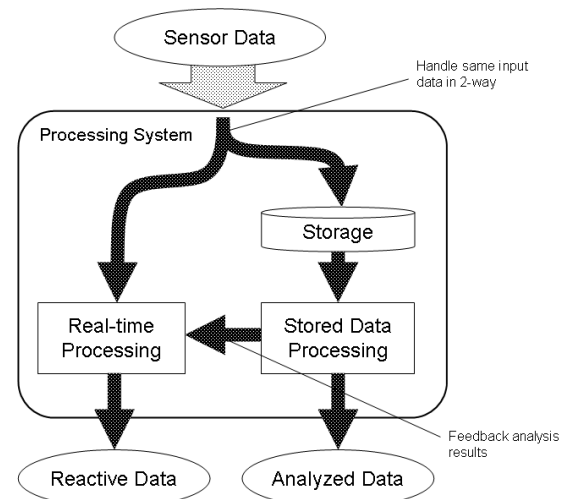


図1 データ処理モデル

- 低遅延かつスケーラブルなデータ処理  
多数のプローブ情報に対するリアルタイム処理を実現するため、低遅延かつスケーラブルにプローブ情報を処理できる必要がある。情報配信サービスにおいては、自動車およびドライバーの情報の変化に対して迅速に情報を配信できることが重要である。特に自動車の位置の情報は、歩行者などに比べると変動が早く、電車などに比べると予測しづらいという特徴がある。このような情報の迅速に変動に対応するため、短い時間で配信情報を判定処理できる必要となる。
- 大量のデータを蓄積して分析するデータ処理  
蓄積データ処理を実現するため、大量のプローブ情報を蓄積し、蓄積された大量のプローブ情報を並列分散的に処理できる必要である。長期にわたって蓄積されたプローブ情報はデータ量が非常に巨大になるため、多数のノードに分散させて保存した上で、現実的な時間で分析処理を完了させるために多数のノードで分散処理を行う必要がある。
- 蓄積されたデータに対する高速な形式変換  
蓄積された大量のプローブ情報に対して様々な分析を実行可能にするため、プローブ情報を高速に形式変換して分析処理に引き渡す必要がある。蓄積したプローブ情報に対しては目的に応じて様々な分析が行われるが、同じデータストレージに対して複数の分析処理が同時にアクセスすると、アクセスが集中することで性能の低下が発生する。また、収集したままのプローブ情報そのものを分析対象とすると不要な情報が多いため、分析処理の効率が低下してしまうという問題もある。そのため、個々の分析処理ごとに扱いやすい形式に変換し、分析用のストレージにインポートすることで効率化が必要がある。
- プログラムの独立性を確保する機能

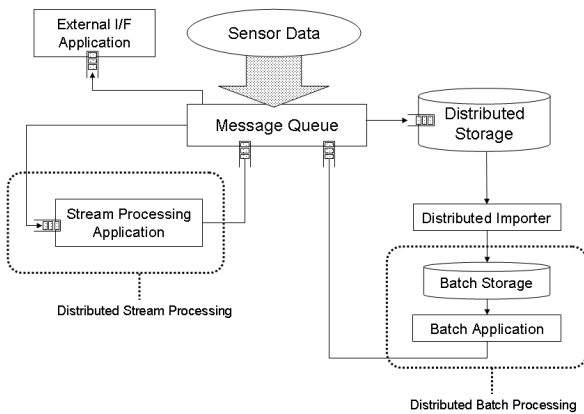


図 2 アーキテクチャ

サービスアプリケーションの保守性と拡張性を確保するため、サービスを構成する各処理を独立して実行できる必要がある。例えば、蓄積データ処理の結果を用いてリアルタイム処理を行うようなアプリケーションを考える。なんらかの理由でリアルタイム処理の動作が停止してしまった場合に蓄積データ処理も動作が停止してしまうと、後で両方の処理が復旧したとしても蓄積データ処理が完了するまでリアルタイム処理は動作することができず、サービスを再開するまで時間がかかってしまう。このように処理同士が密結合してしまうことでサービスの品質が下がることを防ぐため、各処理の独立性を高める機能が必要となる。

### 3.2 アーキテクチャ

前述のプラットフォーム要件を踏まえて、提案プラットフォームのアーキテクチャを図 2 の通りに定めた。

提案したデータ処理モデルに当てはめると、図の左側のモジュールはリアルタイム処理、右側のモジュールは蓄積データ処理に対応している。リアルタイム処理は分散ストリーム処理 (Distributed Stream Processing) によって実現し、蓄積データ処理は分散バッチ処理 (Distributed Batch Processing) と分散インポータ (Distributed Importer) によって実現する構成とした。また、蓄積データ処理の対象となるプローブ情報は分散ストレージ (Distributed Storage) に保存する構成とした。

分散ストリーム処理は、継続的に発生するデータをノードに分散させ、パイプライン的に逐次処理してレスポンスを返却する処理である。処理をノード分散させることでスケラビリティを確保でき、さらにデータをディスクに保存することなくオンザフライで処理できるため、データに対して低遅延で処理を行うことができる。

分散バッチ処理は、ストレージに保存されたデータに対するバッチジョブを、複数の処理ノードを用いて分散して処理する分散処理である。処理対象のデータ量に応じて十分な数の処理ノードを用意し処理を分散させることで、巨

表 1 プラットフォーム要件に対応する処理モジュール

プラットフォーム要件	処理モジュール	実装
低遅延かつスケラブルなデータ処理	分散ストリーム処理	Apache S4
大量のデータを蓄積して分析できるデータ処理	分散ストレージ	DKVS
	分散バッチ処理	Apache Hadoop
蓄積されたデータに対する高速な形式変換	分散インポータ	独自実装
プログラムの独立性を確保	メッセージキュー	Rabbit MQ

大なデータに対しても短時間で分析処理を完了させることができる。

分散インポータは、データストレージに保存されているデータを読み取り、複数の処理ノードに分散させて特定の形式に変換し、別のデータストレージに保存する処理である。ここでは、分散ストレージに保存されているプローブ情報を、分散バッチ処理が処理しやすい形式に変換することで分散バッチ処理の実行を効率化することができる。

分散ストレージは、複数のストレージノードを束ねて、大きい一つのストレージとして扱えるようにしたストレージである。十分なノード数を用意することで、長期にわたって発生する大量のプローブ情報を保存することができる。

リアルタイム処理の側と蓄積データ処理の側はメッセージキューによって互いに接続され、プローブ情報およびそれぞれの処理結果はメッセージキューを介して取得できる構成とした。処理結果を用いて自動車にデータを配信したり、分析グラフを表示させたりといったアクションを行いたい場合は、メッセージキューから該当のデータを取得するアプリケーション (External I/F Application) を用意する。

メッセージキュー (Message Queue) は、プログラム同士の通信を仲介し、Pub-Sub モデルの非同期通信を提供する通信ノードである。データの送信元のプログラムと送信先のプログラムを互いに隠蔽することができるため、プログラム間の独立性を確保することができる。

### 3.3 プロトタイプ実装

提案したアーキテクチャに基づき、研究プロトタイプを構築した。個々の機能については、アプリケーション開発のしやすさを考慮して、広く使われている OSS が存在する場合はできる限り利用した。該当するものが存在しない場合、もしくは機能・性能面に不足がある場合には独自に開発したモジュールを使う方針とした。

各プラットフォーム要件に対して用意した処理モジュールおよび実装の対応を表 1 に示す。

分散ストリーム処理を実現する基盤として、Apache S4[7] を利用した。Apache S4 はスケラブルな汎用分散ストリーム処理基盤の実装である。Yahoo! のシステムで実働実績があり、Java 中心で開発できる点を重要視して採用した。



図 3 サービス概要

分散バッチ処理を実現する基盤として、Apache Hadoop を利用した。Apache Hadoop は MapReduce[2] アルゴリズムを採用した、スケーラブルな分散処理基盤の実装である。既に広く利用されており、ドキュメントが充実している点を重要視して採用した。

分散ストレージとして、独自に開発している DKVS[6] を基にしたデータストアを利用した。プローブ情報の情報は複雑なデータ構造を持たず、アクセスは書き込みが中心であることから、生のプローブ情報を保存するストレージとしてはスケーラブルな分散 KVS を利用するのが適当と判断した。

分散インポータとして、DKVS から差分データを取得する処理と、定期的にデータをバルクロードして並列に HDFS に書き込む処理を実装した。

メッセージキューとして、Rabbit MQ を採用した。Rabbit MQ はオープンスタンダードなキューイングプロトコルである AMQP を採用した、メッセージキューの実装である。Java 用のクライアント API が用意されている点と、オープンスタンダードを採用しているため他システムや他実装との相互運用が容易である点を重要視して採用した。

## 4. サービスの構築

提案したプラットフォームの有用性を評価するため、テレマティクスサービスの一例を取り上げて、提案プラットフォームを用いてデータセンター上に構築した。本節では、構築対象としたサービスと開発したアプリケーションについて述べ、実際に動作させた環境について説明する。

### 4.1 サービス概要

構築するテレマティクスサービスとしては、自動車向けのターゲティング情報配信サービスを取り上げた。このサービスは、ドライバーの嗜好に応じて自動車の周辺にある施設の情報を配信するものである。具体的には、スーパーやレストランなどの商業施設が配信対象を絞った広告を配信するような用途を想定している。さらに、広告の登録者が自分の登録した広告の配信効果を測定できるように、広告効果を可視化するサービスも用意する。図 3 にサービス内容を図示する。

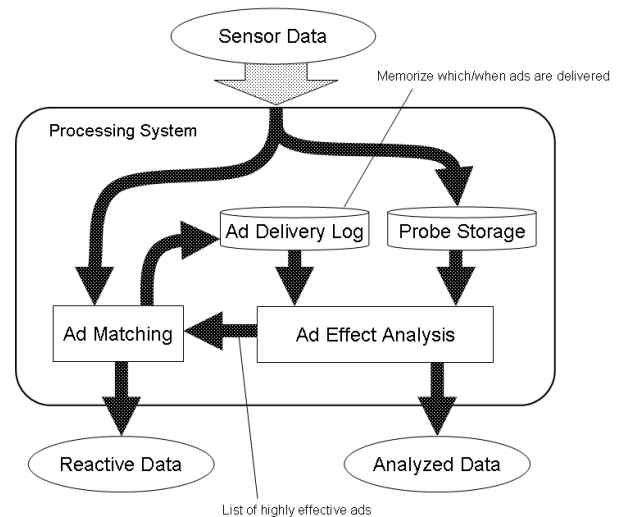


図 4 アプリケーションで行うデータ処理の概要

## 4.2 アプリケーションの構築

前述の情報配信サービスを実現するアプリケーションを、提案プラットフォームを用いて構築する。

### 4.2.1 データ処理の概要

アプリケーションで行うデータ処理の概要を図 4 に示す。自動車から収集したプローブ情報に対して、広告マッチングと広告効果分析処理を並行して行うモデルとした。

広告マッチング処理としては、自動車の位置情報とドライバーの属性（性別、年齢）をもとに適切な広告をリストアップするという処理を行うこととした。広告効果分析処理としては、自動車の位置情報から一定時間以上滞留した地点を抽出し、その自動車に過去に配信した広告情報と照らし合わせることで、「広告を受け取ってからその店舗に行った」ドライバーの数を数え上げる処理を行うこととした。

広告マッチング処理ではさらに、広告効果分析処理の結果を用いて、効果の高かった広告に対してフラグを設定する処理を行うこととした。これは、人気の高い広告をドライバーに提示して来店誘導を行うような用途に使うことを想定している。

### 4.2.2 データ処理プログラムの概要

前述のデータ処理を実現するため、提案プラットフォーム上に実装したプログラムについて概要を述べる。アプリケーション全体の構成図を図 5 に示す。

自動車のプローブ情報は自動車の車載機からフロントエンド Web アプリケーションを介してシステムにインプットされるものとした。実際はプローブ情報はプラットフォーム上のアプリケーションで共有されるデータのため、個々のアプリケーションの設計の範囲外である。ここでは実装の簡単化のため、自動車からのプローブ情報を HTTP リクエストで受け取り、広告情報を HTTP レスポンスとして返却する、情報配信サービス専用の Web アプリケーショ

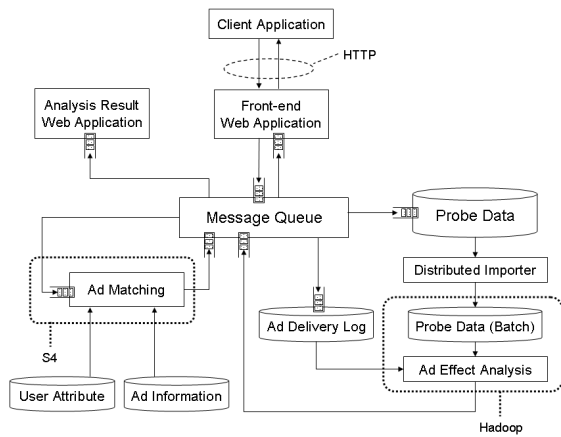


図 5 アプリケーション構成

ンを用意した。

S4 アプリケーションとして広告マッチング処理 (Ad Matching) を実装し, Hadoop アプリケーションとして広告効果分析処理 (Ad Effect Analysis) を実装した。広告マッチング処理ではプローブ情報に対して配信する広告を即座に判定し, 広告効果分析処理では蓄積したプローブ情報から 5 分ごとの広告の配信数および広告の利用者を抽出する。

さらに, 広告効果分析の結果をグラフとして表示するために分析結果表示 Web アプリケーション (Analysis Result Web Application) を用意した。

以下, 実装した各処理の内容について詳しく述べる。

- 広告マッチング処理

広告マッチング処理は, プローブ情報に対して適切な広告をリストアップする処理である。詳しい処理の流れを図 6 に示す。この処理は大きく分けて二段階の処理で構成されている。一方はプローブ情報から広告情報をリストアップする処理であり, もう一方はリストアップした広告情報に対し「ホット広告」フラグを設定する処理である。前者の処理では, プローブ情報内の位置情報およびドライバーの属性を, 広告情報内の配信範囲と配信対象属性 (性別, 年齢) とマッチングして, 該当する広告をリストアップする。後者の処理では, 広告効果情報 (後述) を用いて一定以上の利用率のあった広告に対してフラグを設定する。これらの処理は, 広告の配信範囲を 100m 四方のエリアに分割し, それぞれのエリアごとに PE (Processing Element) を割り当てることで分散処理を行った。

広告マッチング処理の結果としてリストアップされた広告は, フロントエンド Web アプリケーションを介してドライバーに配信されるほか, 広告配信履歴 (Ad Delivery Log) としてストレージに保存される。保存された履歴は, 広告効果分析処理の中で利用する。

- 広告効果分析処理

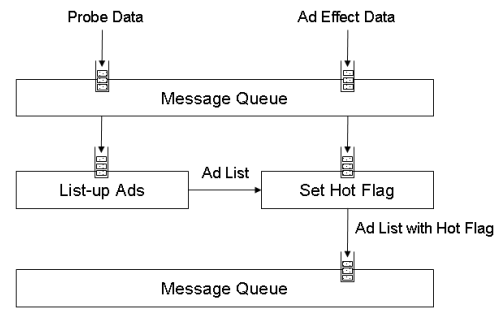


図 6 広告マッチング処理詳細

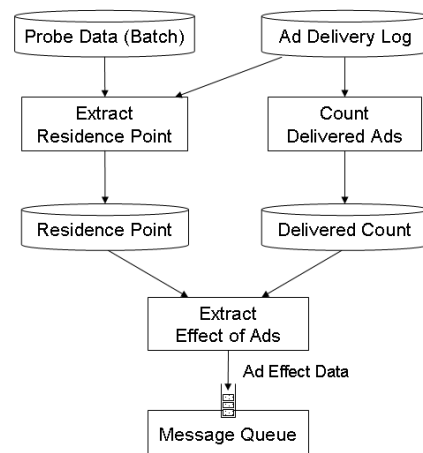


図 7 広告効果分析処理詳細

広告効果分析処理では, 自動車の位置情報および広告の配信履歴を用いて広告効果情報を抽出する。ここで広告効果情報とは, 具体的には各広告の 5 分間ごとの配信数および利用者数 (広告を受け取った後に店舗に行った自動車の数) である。この処理は 5 分間隔で実行され, 前回の処理からの差分のプローブ情報を対象として分析を行う。

広告効果分析処理は, ストレージに蓄積された自動車の位置情報を分析して, ドライバーが広告を受け取ってから店舗に行ったかどうかを判定し, そのユーザ数の割合を計測する Hadoop アプリケーションである。広告効果分析処理は 3 つの分析タスクから構成される (図 7)。一つ目はユーザが 3 分以上滞留した地点を抽出する処理, 二つ目は広告の配信数を数え上げる処理, 三つ目は広告の店舗に滞留したユーザ数を数え上げる処理である。これらの処理を経て, 最終的には広告ごとの配信数と利用者数を出力する。各分析処理は, 自動車や広告につけられた ID ごとに処理を束ねることで分散処理を行なった。

- フロントエンド Web アプリケーション

フロントエンド Web アプリケーションは, 前述のとおり自動車からのプローブ情報に対して広告情報を返却

表 2 自動車の台数による負荷の見積もり

自動車の台数 (千台)	見積り対象処理			
	広告マッチング (件/秒)	プローブ書込 (KB/秒)	広告効果分析 (件/回)	メッセージキュー 書込 (件/秒)
1	100	200	30,000	1,200
10	1,000	2,000	300,000	12,000
20	2,000	4,000	600,000	24,000
100	10,000	20,000	3,000,000	120,000
1,000	100,000	200,000	30,000,000	1,200,000

する Web アプリケーションである。このアプリケーションは、自動車の車載機にあたるクライアントアプリケーションからプローブ情報を含む HTTP リクエストを受け取ると、プローブ情報を抽出してメッセージキューに送信する。その後、メッセージキューから対応する広告リストを受信すると、HTTP レスポンスの形式に変換してクライアントアプリケーションに返却する。なお、実装には Servlet を利用した。

● 分析結果表示 Web アプリケーション

分析結果表示 Web アプリケーションは、広告効果分析処理の結果を Web ブラウザ上でグラフとして表示させる Web アプリケーションである。広告効果分析処理から 5 分ごとに出力される、広告の配信数と利用者数を変換し、HTML と JavaScript を用いて Web ブラウザ上に表示できる形式にする。なお、実装には Struts を利用した。

4.3 アプリケーションのデプロイ

データセンター上に提案プラットフォームおよびサンプルアプリケーションをデプロイし、実際に動作させた。

アプリケーションを動作させるにあたり、大量の自動車サービスを利用している状態をシミュレートするために、負荷ジェネレータを用意した。負荷ジェネレータは、大量の自動車が一定の地域内をランダムウォークする様子をシミュレートし、その結果の位置情報を、プローブ情報としてシステムに 10 秒に一度送信する。また、シミュレートする自動車の台数を変更することで負荷を調整できるようにした。

各処理に割り当てる物理ノード数を決めるため、シミュレートする自動車の台数ごとに発生するデータ量と各処理の負荷を見積もった。見積もった結果を表 2 に示す。各負荷の意味は以下の通りである。

- 広告マッチング処理の負荷は、毎秒何件の広告マッチング処理が実行されるかの件数である。
- プローブ書込の負荷は、プローブ情報を保存する DKVS に 1 秒当たり書き込まれるデータ量である。
- 広告効果分析の負荷は、5 分に 1 度実行される広告効果分析のバッチ処理で何件のプローブ情報を対象に分析が実行されるかである。
- メッセージキュー書込の負荷は、毎秒何件のデータがメッセージキューに書き込まれるかの件数である。

表 3 処理ごとの物理ノード台数

処理内容	台数
広告マッチング処理 (S4 アプリケーション)	2
広告効果分析処理 (Hadoop アプリケーション)	3
プローブ情報ストレージ (DKVS)	6
メッセージキュー (Rabbit MQ)	2

この見積もりをもとに、20000 台の負荷をかけた場合に各処理で必要となるノード数を算出した。各処理に割り当てた物理ノードの台数を表 3 に示す。

5. 考察

サービスを実現するサンプルアプリケーションの開発を通して得られた知見から、提案したプラットフォームを評価する。

5.1 機能要件に対する評価

3 節で挙げたプラットフォームの機能要件の観点から、提案プラットフォームを評価する。

● 低遅延かつスケーラブルなデータ処理

Apache S4 を用いて、プローブ情報をノード分散させて処理できることが確認できた。今回のアプリケーションではプローブ情報を位置情報によって処理ノードに振り分けたが、振り分ける条件はアプリケーションによって柔軟に設定できることは容易に推測できる。また、サンプルアプリケーションを実機環境で動作させ、車載機向けに実用的な処理時間で情報配信サービスを実現できることが確認できた。

● 大量のデータを蓄積して分析するデータ処理

DKVS を用いてプローブ情報を分散保存し、Apache Hadoop を用いて保存したプローブ情報を分散処理できることが確認できた。また、サンプルアプリケーションを実機環境で動作させ、20000 台の自動車の位置情報に対して 5 分間隔という短い間隔で分析結果を出力できることを確認できた。

● 蓄積されたデータに対する高速な形式変換

分散インポートを利用して DKVS に保存されているセンサデータを HDFS に並列分散的にインポートすることで、20000 台の自動車の位置情報に対して 5 分間隔という短い間隔でインポートを完了できることを確認できた。

● プログラムの独立性を確保する機能

メッセージキューである Rabbit MQ を用いてプログラム間の通信を仲介することで、各データ処理ごとに独立して開発を進められることを確認できた。今回のサンプルアプリケーションでは、S4 アプリケーションおよび Hadoop アプリケーションの開発にあたって、処理間の通信に用いるオブジェクトのフォーマットの

み決めるだけで、それぞれの開発を独立に行うことができた。また、それぞれの処理に割り当てる物理ノード数を決める際に、処理ごとに個別に負荷を見積もってノード数を決めることができた。

## 5.2 課題

プラットフォームの検討とアプリケーションの開発を通して明らかになった課題について述べる。

提案プラットフォームは個々のサービスの開発に関しては有用であることがわかったが、複数のサービスが同時に動作するケースを想定すると機能が不十分である。具体的には、以下のような課題が明らかになった。

- サービスごとに、データに対するアクセス権が設定できる必要がある。提案プラットフォームでは多くのプログラムが同じメッセージキューを介して通信するため、複数のサービスが同時に動作していると、あるサービスのプログラムが別のサービスのプログラムのデータにアクセスすることができてしまう。これは、サービスの提供者が異なる場合にはセキュリティ的に問題がある。
- サービスごとに必要な処理リソースを保証できるようにする必要がある。サービスの応答性やバッチの完了時刻などの品質を保証するために、サービスごとの負荷の変動に追従して必要な処理リソースを算出し、割り当てを行う仕組みが必要である。
- 複数のサービスを横断して処理の負荷をバランスさせる必要がある。複数のサービスが動作する場合、S4やHadoopの処理ノードを共有して分析処理を行うため、サービスの間で処理の量をバランスさせる必要がある。Hadoopの処理に関しては、対象とするプローブ情報の量によって必要となるバッチ用ストレージのサイズも異なるため、ストレージの使用量に関してもバランスさせる仕組みが必要である。

## 6. まとめ

本稿では、高度なテレマティクスサービスを実現するために必要となる共通機能を提供するプラットフォームを提案し、実際にサービスを構築することでプラットフォームの有効性と課題を明らかにした。分散ストリーム処理、分散ストレージ、分散バッチ処理、および分散インポータをメッセージキューで接続した提案プラットフォームにより、サービスを容易に構築できることを確認した。一方で、プラットフォーム上での複数のサービスの同時実行に関しては更なる検討が必要である。

自動車は、新興国を中心に今後さらなる普及が見込まれており、人とICTの接点として非常に期待されている分野である。日本でも東日本大震災以降、各地の状況を正しく認識し、人々に正しい情報を速やかに届けることは大きな

課題として認識されており、老若男女・地域を問わずに利用されている自動車を利用するサービスは平時・災害時両用のインフラとして有望視されている。このような社会情勢を鑑み、我々は高度テレマティクスの実現を加速することで、ICT技術による人と地球にやさしい情報社会の実現に大きく貢献できるものと確信している。

## 謝辞

提案プラットフォームの評価にあたり、サンプルアプリケーションの開発に助力いただいた方々にこの場を借りて深謝する。

## 参考文献

- [1] C. Bisdikian, I. Boamah, P. Castro, A. Misra, J. Rubas, N. Villoutreix, D. Yeh, V. Rasin, H. Huang, and C. Simonds. Intelligent pervasive middleware for context-based and localized telematics services. In *Proceedings of the 2nd international workshop on Mobile commerce (WMC '02)*, pp. 15–24, 2002.
- [2] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM - 50th anniversary issue: 1958 - 2008*, Vol. 51(1), pp. 107–113, 2008.
- [3] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang. Framework for security and privacy in automotive telematics. In *Proceedings of the 2nd international workshop on Mobile commerce (WMC '02)*, pp. 25–32, 2002.
- [4] CB. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: A distributed mobile sensor computing system. In *SenSys '06 Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 125–138, 2006.
- [5] D. Kim and B. Jang. The development of telematics service platform in web environment. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS '05)*, pp. 1698–1700, 2005.
- [6] 小林大, 菅真樹, 大野善之, 鳥居隆史. 構成ノード電源停止によるシステム省電力化のためのインメモリ分散データストア設計. 第3回データ工学と情報マネジメントに関するフォーラム (DEIM 2010), pp. C10–3, 2011.
- [7] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari. S4: Distributed stream computing platform. In *IEEE International Conference on Data Mining Workshops (ICDMW 2010)*, pp. 170–177, 2010.
- [8] J. Rodrigues, A. Aguiary, F. Vieiray, J. Barroso, and J. Cunhaz. A mobile sensing architecture for massive urban scanning. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC 2011)*, pp. 1132–1137, 2011.