

M2M を用いた大規模データ収集システムの構築に関する研究

林 貴哉¹ 枝松 ちさと¹ 村林 竜司¹ 胥 徳文¹ 山下 浩一郎² 天笠 俊之³ 牧野 昭二³
高橋 伸³

概要: 近年、人を介すことのない M2M や大規模データに注目が集まっている。これらを組み合わせることでも有効なシステム活用ができると考えられる。このような M2M や大規模データを用いたシステムは全体的な負荷が大きくなることが多い。しかし、ストレスフリーな利用のためには高速に動作することが不可欠である。これに対し、部分的な高速化について検討されることは多い一方、システム全体としての最適化については、システムや環境に起因する複数の要因が関係するため、議論される機会は多くない。クライアント・サーバシステムでは、クライアントの最適化とサーバの最適化はトレードオフの関係にあることが多い。そこで本研究では、スマートフォンを利用した路面監視アプリケーションを取り上げ、そのシステムを構築し、システム全体としての最適化に関する考察を行った。具体的には、構成要素ごとに最適化を図るための実験を行うとともに、クライアント側の消費電力やサーバ側の負荷実験を通して、それぞれの最適化を図った。そこからシステム全体を最適化するための課題抽出・検討を行った。

1. はじめに

近年、人を介すことのない M2M に注目が集まっている。同様に大規模データも注目されている。しかし、M2M や大規模データを扱うシステムには問題や課題が残留していると考えられる。そこで、本研究では M2M の仕組みを用いて、大量のデータを収集するようなシステムを構築する。構築したシステムを通して、M2M や大量のデータを処理するうえでの課題を抽出し、それについて検討する。

1.1 問題背景

M2M とは人手を介すことのない仕組みである。活用事例として、電気自動車の充電状況の通知や無人駐車場の管理がある。これらは自動で利用者に通知を行うシステムである。一見人手を介していないように見えるが、実際には設定を行う必要があり、通知を受けた人はそれに対応したりと、厳密には人手がかかっているため M2M とは言えない。

大規模データ処理は、マーケティングや経営など様々な場面で利用されている。しかし、大規模データ処理は大きな負荷がかかるため、十分なリソースがないと活用が難しい。

1.2 検討する問題

ここで、本研究で構築するシステムで検討する問題について述べる。M2M の仕組みを用いたシステムを構築するので、解決を図る最適な事例は、現在人手がかかっていることが問題となっている事例である。そこでシステム構築の事例として注目するのが、路面監視である。路面監視は人手がかかっているため、十分に行われていない。

1.3 アプローチ

本研究では M2M の仕組みを用いて、一般の道路利用者から走行情報を集める。ここで、走行情報とは道路を車で走行した際の位置情報や加速度等の集合を指す。収集した走行情報を解析することで路面異常箇所の検出を行う。検出した路面異常箇所を提示することで、路面監視の効率化を図る。このようなシステムを構築し、それについて考察することで、M2M や大規模データを用いたシステムを構築する際の課題を抽出する。

¹ 筑波大学 システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science, Graduate School of Systems and Engineering, University of Tsukuba

² 株式会社富士通研究所
Fujitsu Laboratories Co. Ltd.

³ 筑波大学システム情報系情報工学域
Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba

1.4 得られた結果

本研究ではシステムの全体最適化を図るにあたり、まずクライアントやサーバに関する構成要素ごとの最適化を考察した。そこでシステムの構成要素ごとに負荷実験を行った。

クライアント側では、端末にスマートフォンを用いることから、消費電力や省パケット通信が必要と考察した。省電力に関して実験を行い、電力消費を占める原因が画面バックライトであり、センシングの周波数やGPSではないことを特定した。そのため、簡単な省電力化として画面使用を控えるような設計が必要である。

サーバ側では、サーバで行う処理ごとの時間を計測し、どのような設定が必要かを考察した。一般にシステムで扱うデータ数が多いと、データの検索に要する時間が伸びる。本研究で構築するシステムは扱うデータ数が多いため、実験結果もデータの検索を含む処理に多くの時間がかかっていた。そのため、データベースチューニングやインデックスの付与が必要である。

実験結果からそれぞれの最適化を試みているが、互いにトレードオフの関係にあることがわかっている。例えば、サーバ負荷を軽減するために路面異常箇所検出をスマートフォンで行うと消費電力が増加してしまう。そのため、システム全体を最適化しようとする場合、システム全体の負荷バランスを調整を行う必要がある。

1.5 本論文の構成

本論文の構成は、以下の通りである。2節で提案システムの概要について述べ、3節でシステムの構成について説明する。4節では、Android上のクライアントアプリケーションについて説明し、5節で本システムで採用した路面異常の検出手法について述べる。6節はシステム構成要素ごとの効率化、7節においてシステム全体の最適化について議論する。8節で本論文をまとめる。

2. 提案システムの概要

本研究では、M2Mの仕組みを用いた大規模データ収集システムを構築する。構築するシステムの解決を図る事例として、路面監視を取り上げる。

2.1 背景

社会インフラの一つである全国の道路網の多くが損傷などの問題を抱えている。また、国土交通省が新たに管理区間に指定した道路の総延長は年々増加する一方である。しかし、近年の財政難により道路の路面監視に必要な経費を捻出するのは困難である場合が多く、コスト削減のための路面監視基準の見直しが実施されている[1]。また、路面状態は自動車運転時の操縦性や乗り心地に影響を与えるだけでなく、落下物などの路面異常に起因した事故も発生して

いる[2]。

対策として現在、道路巡回パトロールが行われている。これは、現在人手で行っているため、非常にコストがかかっている。コストがかかっているため、巡回の網羅性・即時性は低くなってしまっている問題がある。それでは十分な対策とは言えないため、より低コストで路面監視を実現する仕組みが必要不可欠といえる。

そこで、本研究ではM2Mの仕組みを用いて大量の走行情報を収集する。それを解析することで路面異常検出を行い、検出結果を提示することで路面監視の効率化を図る。

2.2 システム概要

本研究で扱う事例に対するシステムの概要について述べる。事例として取り上げた路面監視に対して、M2Mの仕組みを用いた大規模データ収集システムを適用し、路面監視にかかるコストの削減を図る。開発システムの全体イメージを図1に示す。

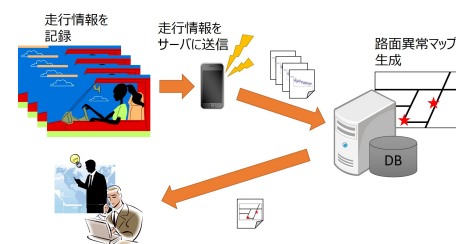


図1 システムイメージ

このシステムは、クライアントサーバモデルで構成する。主にクライアント側ではデータの収集を行い、サーバ側でデータの加工処理を行う。

大まかな流れは、まず一般の道路利用者が道路を走行した際のデータ（振動、速度、方位など）をスマートフォンに搭載されているセンサで計測し、“走行情報”として記録する。走行情報は一定のタイミングでサーバに送信される。サーバ側では、送られてきた走行情報を解析することで路面異常の検出を行い、異常箇所と判断された位置などの情報をサーバに保存する。そして、一般道路利用者や道路管理者の閲覧要求に応じて、路面異常箇所情報をクライアントにフィードバックとして返す。クライアント側では、地図に路面異常箇所のプロットを行った上で道路利用者や道路管理者に提示する。

なお、走行情報の収集から路面異常の検出まではシステムが自動的に行う。自動化により事例であげた路面監視にかかるコストの削減を図る。システムの効果として、路面監視の網羅性と即時性の向上が挙げられる。また、この自動化は“人手を介さない”ためM2Mであり、この部分を考察することでM2Mの一般的な最適化に通ずる知見を得る。

2.3 関連サービス

同様のサービスとして、A社製Xがリリースされている。これは、自治体などの道路管理者を対象としたサービスである。まず、巡回車にスマートフォンを搭載した状態で現行の業務と同じように道路巡回を行い、スマートフォンの各種センサから巡回時の走行情報を記録する。そして、記録したセンサの値を基に、舗装道路の凹凸情報を推定し、地図上に表示する。また巡回中に写真やコメントをアップロードし、後に作業報告として閲覧することも可能である。このサービスには、現行の業務負担を増加させることなく道路の路面監視を支援できるという特徴がある。しかしながら、このサービスは道路管理者向けのものであるため、取得できるデータ数・計測範囲は限られている。

B社より路面異常を検出を行うスマートフォン向けアプリケーションYが現在リリースされている。このアプリケーションでは、道路段差の分布状況を観測収集し、GoogleMap上で確認することが可能となっている。1回の解析に使うデータは入力された1件の走行情報のみとなっており、他の走行情報を参考にしていない。このため、網羅性や即時性は向上しない。また、1件の走行情報でより良い解析結果を取得するために、スマートフォンの設置位置に制約を設けたり、スマートフォン側で様々な処理が行われているため、端末の負荷が大きくなっている。

2.4 新規性

ここで、関連サービスと開発システムの差分を明示する。関連サービスでは、1件の走行情報に対し、解析を行っている。そのため、網羅性の向上につながらない。また、機器設置に制約を設けるなどの対策で、1件あたりのデータの精度を高めることで1回の解析精度を高めている。そのため、測定を目的とした走行が必要となる。

それに対し、本研究で構築するシステムは、一般の道路利用者から走行情報を収集しており、A社製Xに比べて広範囲に渡って道路の路面監視をサポートすることが出来る。道路管理者は計測のための走行を行わずに、路面異常箇所を事前に知ることができる。解析は大量の走行情報に対して行う。また、機器の設置について制約を設けないため、測定を意識せずに走行情報を収集できる。表1に、関連研究と開発システムの差分をまとめる。

表1 提案システムと関連研究の差分

| | 提案システム | A社製X | B社製Y |
|------------|--------|------|------|
| 測定を目的とした走行 | 不要 | 必要 | 必要 |
| 解析データ数 | 大量 | 1件 | 1件 |
| 機器設置 | 任意 | 固定 | 固定 |
| データの精度 | 低い | 高い | 高い |

3. システム構成

開発システムのハードウェア構成、ソフトウェア構成を図2、3に示す。

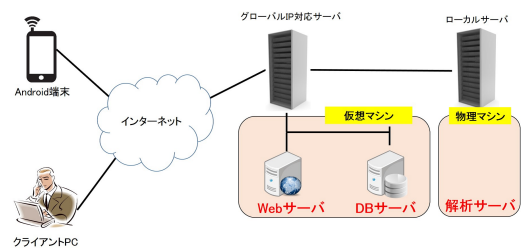


図2 ハードウェア構成図

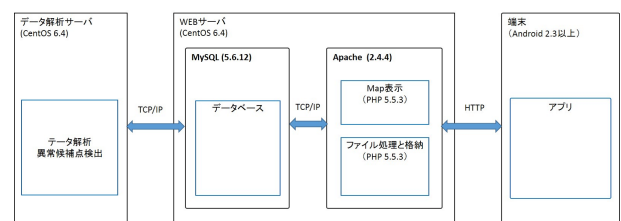


図3 ソフトウェア構成図

クライアント側では、Webブラウザから路面異常箇所を閲覧するために汎用計算機を、走行情報の収集などを行うためにAndroid端末を用いる。クライアントは、携帯回線を介してサーバとやりとりを行う。

サーバ側では、アクセス負荷を分散させるためにサーバを2台設置している。1台はクライアントからの要求を受け付けるWebサーバ、および走行情報や路面異常箇所情報などを保持しておくデータベースサーバの役割を果たし、もう1台は路面異常箇所を検出するための走行情報解析サーバの役割を持つ。

4. Android アプリケーション開発

Androidクライアントアプリケーションの設計開発について述べる。本システムのAndroidアプリケーションは以下の機能を有する。

- センサデータを走行情報としてファイルに記録する
本アプリケーションでは、Android端末の各種センサで計測したそれぞれの値をまとめて走行情報としてファイルに記録する。記録するデータは、計測時刻、位置、加速度、地磁気、速度である。これらのデータはサーバ側で異常箇所検出の解析処理に利用される。なお、ファイル形式はCSVを用いている。採用した理由としては、CSVファイルの汎用性が高いこと、およびサーバ側でのデータベース操作が行いやすいことが挙げられる。
- 走行情報ファイルを圧縮する

1 件の走行情報ファイルのサイズは、現在のところ1時間あたりおよそ数十 MB と非常に大きい。このため、CSV ファイルのまま送信すると Android 端末・サーバ双方に負荷がかかってしまうことが予想される。そこで本アプリケーションではファイルの圧縮処理を実装し、データサイズの軽量化を図った。なお、圧縮のタイミングは走行情報ファイルの記録終了時となっている。圧縮形式には gzip を採用しており、先ほどのファイルはおよそ数 MB に圧縮される。

- 圧縮ファイルをサーバに送信する
gzip で圧縮した走行情報ファイルを Web サーバに送信する機能である。ファイル送信のタイミングは、走行情報ファイルの圧縮完了時となっており、1 回の計測で 1 件の圧縮ファイルを送信する。なお、プロトコルは HTTP を用いた。

5. 路面異常検出手法

本節では、サーバ側で行われている路面異常検出の手法について述べる。本手法では、同じ路面における複数の走行情報から解析を行うという点において特徴がある。しかし、一般の道路利用者が所有するスマートフォンから走行情報を取得する場合、計測時の車両や走行状況、端末の違いなどによる影響を考慮しなければならない。また、即時性の問題を解決するために、路面状態の経時変化に対応する必要がある。

本システムの路面異常検出アルゴリズムは大きく分けて次の 2 段階で行う。

STEP1) 連続した時刻で記録された走行情報一件から異常候補点の検出を行う。

STEP2) それ以前の同位置の走行情報を合わせて、その地点に路面異常が存在するか否かを 0~1 の数値の大ききで示す評価値を算出する。ここで計測条件の違いを平滑化するとともに、最近の走行情報ほど強く反映されるようにする。

これらの処理をスマートフォンから走行情報を受信するたびに実行する。以下、それぞれの段階における詳細な流れを説明する。

STEP1 : 一件の走行情報に対する処理

先行研究により、路面の損傷度合を表す国際ラフネス指数と車内で計測した重力方向加速度の 1 秒間の分散に相関があることが分かっている [3]。よって、これと同じ分散を用いることで異常候補点を検出することを考える。

ユーザビリティの観点から、計測時におけるスマートフォンの設置方法は指定しないことが望ましい。そこで任意の姿勢における重力方向の加速度を計測可能な情報から導出する。Android 端末では、図 4 の方位角、傾斜角、回転角、および端末に固定された座標系での 3 軸加速度を取

得することができる。

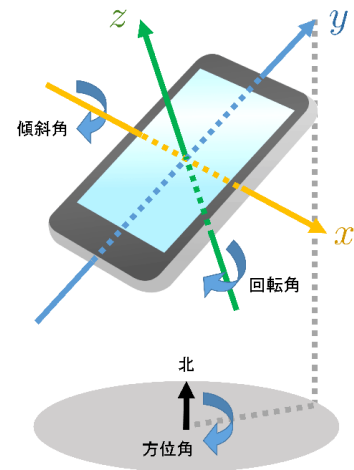


図 4 Android 端末の座標系

これらを用いて重力加速度を除いた重力方向の加速度を式 1 により求める。

$$z' = -x \sin(\gamma) \cos(\beta) + y \cos(\beta) \cos(\gamma) + z \sin(\beta) - g \quad (1)$$

表 2 変数表

| | |
|-------------------------------|----------|
| 重力方向加速度 [m/s ²] | z' |
| 端末 x 軸加速度 [m/s ²] | x |
| 端末 y 軸加速度 [m/s ²] | y |
| 端末 z 軸加速度 [m/s ²] | z |
| 方位角 [deg] | α |
| 傾斜角 [deg] | β |
| 回転角 [deg] | γ |
| 重力加速度 [m/s ²] | g |

次に、時刻 i における z' の分散 $v(i)$ を求める。サンプリング周波数が f のとき、一秒間の加速度の分散を、

$$v(i) = \frac{1}{f} \sum_{j=0}^{f-1} (z'(i+j) - \overline{z'(i)})^2 \quad (2)$$

とする。ここで、 $\overline{z'(i)}$ は $z'(i), z'(i+1), \dots, z'(i+f-1)$ の相加平均である。

以上により求めた時系列の分散が、予め設定した閾値 T を超えた点を異常候補点とする。

STEP2 : 複数走行情報を反映した評価値処理

STEP2 では図 5 に示すように、緯度経度を一定幅 d で格子状にする。STEP 1 の走行情報および異常候補点から、経路上の格子に対して次のように評価を行う。

$$p(i, j) = \begin{cases} 1 & (\text{異常候補点あり}) \\ 0 & (\text{異常候補点なし}) \end{cases} \quad (3)$$

次に、それぞれの格子に対して新たな評価値 J_{new} を、計算した p 、重み係数 w_1, w_2 ($0 < w_1, w_2$)、既存の評価値 J_{old} を用いて式 4 により更新する。

$$J_{new}(i, j) = \frac{w_1 J_{old}(i, j) + w_2 p(i, j)}{w_1 + w_2} \quad (4)$$

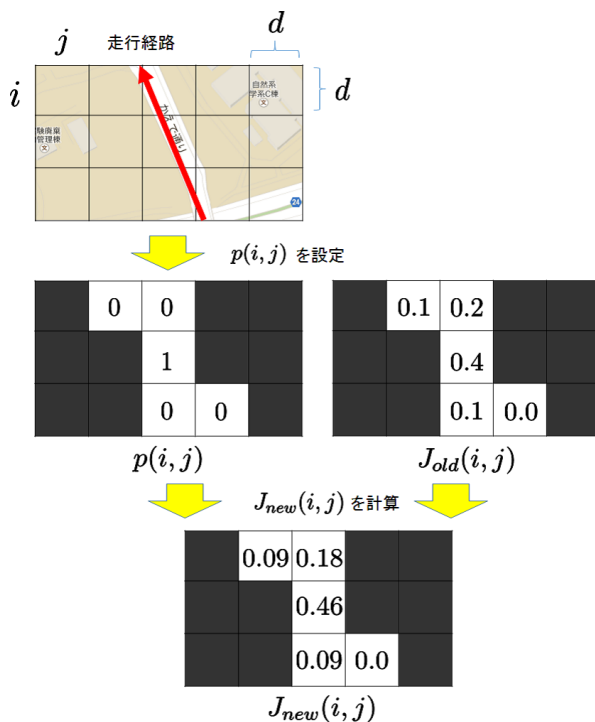


図 5 $(w_1, w_2) = (0.1, 0.9)$ とした評価値更新処理の例

同位置に新たな走行情報を受け取るたびに式 4 が繰り返されるため、過去の影響は指数関数的に減衰する。すなわち、最近のデータほど大きく反映することになる。新しいデータの影響を小さくしたい場合は、 w_1 に対して w_2 を大きく設定すればよい。また、初めて路面候補点を検出した格子は J_{old} が存在しないため、適当な値で初期化する。

以上の STEP1, STEP2 が路面異常検出の一連の流れである。実装時には格子内での進行方向から対向車線と区別してそれぞれ評価値を計算している。

6. システム構成要素ごとの効率化

本システムでは、多くの走行情報を扱うため、主にスマートフォンとサーバの観点から最適化を図る必要がある。スマートフォン側では、消費電力や通信について、サーバ側では、データベース処理や解析の負荷についてを考慮しなければならない。これについてのいくつかの実験を行う。

6.1 スマートフォンの消費電力実験

本システムでは、走行情報収集にスマートフォンを用いる。そのため、バッテリー稼働のスマートフォンにおいて消費電力や端末の動作速度について考慮が必要である。実際

にスマートフォンユーザの懸念事項にもなっている [4]。そのため、アプリケーションによる消費電力を極力抑える。そこでスマートフォンの電力消費の要因を調査した。

6.1.1 仮説

スマートフォンの電力を消費する要因として次の 3 つがあると仮説を立てた。

- 画面バックライト
一般に、消費電力が大きいと考えられている要因である。
- GPS の測位
一般に、地図アプリケーションなどで GPS を利用していると電力の消費が激しくなると考えられる。
- センサデータのサンプリング周波数
サンプリング周波数が高いと、それだけセンサデータを読み出す頻度が高くなるということになり、消費電力が増大すると考えられる。

6.1.2 検証

それぞれの要因が電力消費の原因となっているかを検証するために、アプリケーション利用時の消費電力を計測し、比較することにした。以下に実験内容を示す。

事前条件

実験のためのアプリケーションは、次の機能を有する。

- CSV ファイルを生成する機能
- 一定頻度で取得したセンサデータなどを CSV ファイルに書き込む機能
- CSV ファイルを GZip 形式で圧縮する機能

なお、記録するデータは、ファイル名、データ取得時刻 (日付・時刻)、位置 (緯度・経度)、3 軸加速度 (x, y, z)、速度 (方角・速さ)、傾き (方位角・傾斜角・回転角)、3 軸地磁気 (x, y, z)、照度の計 8 種類 17 項目である。また、WiFi 機能は無効にし、3G もしくは LTE 回線下で本実験を行うこととした。

実験手順

- (1) アプリケーション開始前のバッテリー残量 [%] を記録する。
- (2) 各実験条件でアプリケーションを実行してセンサデータの記録を開始し、端末を固定して 1 時間置く。
- (3) 1 時間経過後、アプリケーションを終了する。このときのバッテリー残量 [%] を記録する。

実験の各要素をまとめたものを表 3 に示す。実験 1, 実験 2 で GPS の、実験 2, 実験 3 で画面バックライトの消費電力の計測を行う。また、実験 3, 実験 4 でセンサデータのサンプリング周波数の違いにおける消費電力の差を観察する。

実験端末

実験端末には表 4 に示した 3 台を利用した。

6.1.3 実験結果

6.1.2 項の実験結果を図 6, 図 7 に示す。縦軸は消費電力

表 3 各実験のパラメータ設定の一覧

| | 画面バックライト | GPS 機能 | サンプリング周波数 [Hz] |
|------|----------|--------|----------------|
| 実験 1 | ○ | × | 100 |
| 実験 2 | ○ | ○ | 100 |
| 実験 3 | × | ○ | 100 |
| 実験 4 | × | ○ | 40 |

表 4 実験端末一覧

| 端末機種 | OS バージョン | バッテリー容量 | バッテリー電圧 |
|------------|---------------|---------|---------|
| A 社スマートフォン | Android 2.3.6 | 1650mAh | 3.7V |
| B 社スマートフォン | Android 4.2.2 | 2300mAh | 3.6V |
| C 社スマートフォン | Android 4.0.3 | 1800mAh | 3.7V |

量 [W] を表しており、この値 P は式 5 より算出している。

$$P = \frac{(B_{before} - B_{after})}{100} * C * E * 1000 \quad (5)$$

ここで B_{before} は実験前のバッテリー残量 [%], B_{after} は実験後のバッテリー残量 [%], C は実験端末のバッテリー容量 [mAh], E は実験端末のバッテリー電圧 [V] を表す。

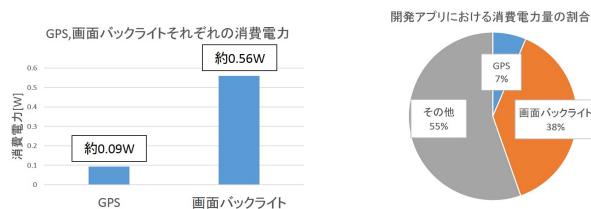


図 6 実験 1, 2, 3 における結果

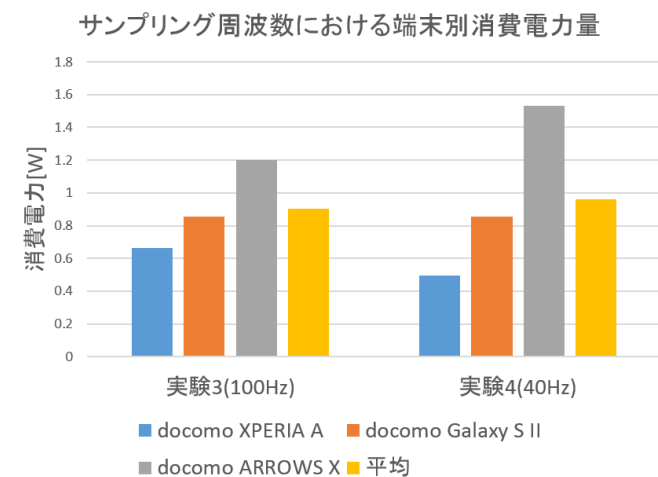


図 7 実験 3, 4 における結果

6.1.4 考察

6.1.3 節での結果をふまえ、電力を消費する要因について考察する。

- 実験 1, 2 より、本アプリケーションに占める GPS の消費電力量は 1 時間で平均 0.09W であった。これは、実験 2 におけるアプリケーション全体の消費電力のうち、およそ 6% を占めていることになる (図 6)。
- 実験 2, 3 より、本アプリケーションに占める画面バツ

ックライトの消費電力量は 1 時間で平均 0.56W であった。これは、実験 2 におけるアプリケーション全体の消費電力のうち、およそ 38% を占めていることになる (図 6)。

- 実験 3, 4 より、センサデータのサンプリング周波数の違いによって消費電力量に大きく差が出るわけではない (図 7)。

GPS の消費電力については、全体の消費量に対して 1 割を切っており、バッテリー消耗の大きな要因とはなっていないことが分かった。したがって、早急に対応を講じる必要はないと考えられる。

画面バックライトについては、全体の消費電力量に対する割合が 4 割近くまで占めており、実装する際には必要に応じて消灯するなどの対応策を講じなければならないことが分かった。

センサデータのサンプリング周波数における消費電力については、それぞれに大きな違いは見られなかったことより、サンプリング周波数を高く設定しても電力消費には大きく影響しないものと考えられる。これより、解析の精度を向上させるためにサンプリング周波数を高くし、より詳細なデータを記録できるよう実装できる可能性があることが分かった。

6.1.5 対策

実験結果をふまえ、Android アプリケーションを開発するにあたって、以下の点に留意することとした。

- 画面バックライトの点灯は極力控える
- センサデータのサンプリング周波数は 40Hz とする

6.1.4 項で、センサデータのサンプリング周波数を高くできる可能性について述べたが、サンプリング周波数とデータサイズはトレードオフの関係にある。開発システムではサーバで大量のデータを取り扱うことを想定しており、サーバ側の負荷も考慮する必要がある。本システムでは解析精度を損なわない程度のサンプリング周波数に設定するのが妥当だと判断した。

6.1.6 課題

実験 4 での平均消費電力量は 0.96W であった。しかしながら、実験 4 での機能はシステムが動作するのに必要最低限のみに限定しており、解析結果を地図に表示するなどの拡張機能を追加すると、消費電力量はさらに増加するものと予想され、懸念事項となっている。

また、まだ検討を行っていない省パケット処理の問題についても考察が必要である。

6.2 データベースサーバの負荷実験

データベースサーバの負荷実験を行う。本システムのデータベースに対する要求は作成、読み出し、更新、削除である。まず、本システムで稼働しているデータベースに対して、負荷の大きな動作を調査した。実験条件を以下に

示す.

- データベースに 700 件のデータを保存
- データベース内のデータ数毎に処理時間を計測
- 処理はファイル取得, ファイル移動, 路面異常解析, 評価値更新, コミット

負荷テストの結果から, ファイル取得や移動などの検索を用いる処理に時間がかかっていることがわかった. このことから, 検索を高速化するためにインデックスの付与やバッファサイズの変更等のデータベースチューニングを行う必要がある. しかし, 挿入や更新などの他の処理に影響があるため, 最適な設定を考える必要がある.

また, スマートフォン側で解析やデータ補正等の処理を行うことで, サーバ側の負荷を軽減できる. しかし, サーバよりも性能で劣るスマートフォン側で処理を行うと, スマートフォンへ大きな負荷を与えるため, 消費電力等に大きな影響を及ぼす. そのため, 処理の分散は双方のバランスを考えて行う必要がある.

7. システムの全体最適化

これまでにシステムの構成要素ごとの最適化について実験を行った. しかし, それぞれの最適化を行うと他に影響を及ぼすものが多い. そのため, その他の構成要素とのバランスを考える必要がある. スマートフォン側から考えれば, データの計測と送信以外はサーバ側で行い, データの精度向上のためにサンプリング周波数を高く設定することが好ましい. サーバ側から考えれば, 処理量が多いため, ある程度の処理をスマートフォン側で行うことで, 高速に動作することが考えられる.

この状況に, 新たな機能が加わるとこのようなトレードオフの関係が増加すると考えられる. そのため, システム全体としてバランスを考えていくことが, このようなシステムを開発する際の重要な要因となりえる.

8. まとめ

本研究では M2M を用いた大規模データ収集システムを構築し, そこから課題抽出・検討を行った. 構築したシステムに対し, クライアントとサーバで消費電力実験や負荷実験を行い, システムの構成要素ごとに最適化を図った. 今後はその結果から, システム全体の最適化を行う. その際, システムが自動で負荷状況を判断するような仕組みを考える必要があると考えている. 現在, 以下のことを検討している.

- サーバの負荷に応じて, スマートフォンでの処理を行うか否かを動的に判断する
- サーバの負荷に応じて, 解析等のタイミングを自動的にスケジューリングする

謝辞 本研究をに関して, 活発な議論にお付き合い頂いた株式会社富士通研究所の山下浩一郎様に感謝いたします.

す. 本研究を進めるにあたり, ご指導を頂いた筑波大学システム情報系情報工学域の天笠俊之准教授に感謝いたします. また, 日常の議論を通じて多くの知識や示唆を頂いた筑波大学システム情報系情報工学域の牧野昭二教授, 高橋伸准教授に感謝いたします.

参考文献

- [1] 小澤隆 (2007). “道路維持管理の現状と課題”. http://www.ndl.go.jp/jp/data/publication/refer/200704_675/067503.pdf
- [2] 国土交通省 (2012). “国道 (国管理) の維持管理等の現状と課題について”. <http://www.mlit.go.jp/road/ir/ir-council/roadmaintenance/pdf/4.pdf>
- [3] 八木浩一. “スマートフォンの加速度センサを用いた路面段差検出手法”. 第 9 回 ITS シンポジウム 2010 災害時交通流監視システム研究会
- [4] 総務省 (2012). “スマートフォンアプリケーションに係る利用者の動向”. 利用者視点を踏まえた ICT サービスに係る諸問題に関する研究会. スマートフォンを経由した利用者情報の取扱いに関する WG http://www.soumu.go.jp/main_content/000150948.pdf