

スマートシティにおける大規模住宅ログの 収集・活用プラットフォームの検討

山本晋太郎[†] 瀬戸 英晴[†] 裕本 真佑[†] 中村 匡秀[†]

[†] 神戸大学 〒 657-8531 兵庫県神戸市灘区六甲台町 1-1

E-mail: †{shintaro,seto}@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp

あらまし スマートシティとは、エネルギーをはじめとする生活インフラ全体の高度な効率化を目指した、次世代の都市のことである。スマートシティ環境では、都市中に設置されたセンサにより様々な都市データが収集され、社会サービスとして住民に還元される。本稿では、スマートシティ上での大規模ログデータの蓄積と活用を支援するプラットフォーム Scallop4SC を提案する。Scallop4SC は、分散処理システム Hadoop・HBase を組み合わせた並列処理分析インフラを構築し、巨大化するデータに対し効率的な収集・分析処理を可能とする。提案プラットフォームによるデータ処理効率を確かめるために、実際のスマートホーム環境で収集されたログデータを利用した評価実験を行う。さらに実験結果に基づき、実在の都市規模のログ処理に対する提案プラットフォームの適用可能性について検討する。キーワード スマートシティ、住宅ログ、Hadoop、HBase

A Study on Data Platform for Collecting and Using Large-scale House Log in Smart City

Shintaro YAMAMOTO[†], Hideharu SETO[†], Shinsuke MATSUMOTO[†], and Masahide
NAKAMURA[†]

[†] Kobe University Rokkoudai-cho 1-1, Nada-ku, Kobe, Hyogo, 657-8531 Japan

E-mail: †{shintaro,seto}@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp

Abstract Smart city is a next-generation societal concept which aims to provide low-carbon and sustainable social infrastructure. In the smart city environment, wide variety of city data are retrieved from sensors. Social services are provided from the collected data for smart city residents. In this paper, we propose a smart city platform, called Scallop4SC that supports collecting and using the extremely large-scale log data. The Scallop4SC constructs a distributed multi-node infrastructure using Hadoop and HBase, and provides high-scalability and high-performance data processing for petabyte scale data. We conduct an experimental performance evaluation of the proposed platform using actual house log obtained in our smart home environment. Based on the results, we discuss the applicability of Scallop4SC in city scale log data processing.

Key words Smart city, house log, Hadoop, HBase

1. はじめに

環境配慮型の都市実現を目的とした、スマートシティ [1], [2] と呼ばれる次世代の都市計画が着目を集めている。スマートシティでは都市中に設置された各種センサにより、交通網の状況や、各家庭のエネルギーの使用状況、家電機器の利用状態などの多種多様な都市ログが収集される。収集された都市ログは高度なデータ処理技術により解析され、社会サービスとして住民に還元される。具体的には、交通網最適化や地域単位での

省エネ計画の実現といった、エネルギーの高効率化を目的としたサービスに限らず、住居周辺地域でのテレビ番組などのトレンド把握といった付加価値創造のためのサービスなど多岐にわたる。

このような大規模ログに基づくスマートシティサービスの実現には、解決すべきいくつかの課題が存在する。まず、広域な範囲での膨大な数のログが定期的に収集されるため、ペタバイト級の大規模データを処理できる分析基盤が必須である。また、ログ管理のためのデータベースのスキーマを事前に決定するこ

とは難しい。スマートシティ内で取得される都市情報が多岐にわたること、及びどのようなログを取得するかは計画の発展に伴い拡張されることがその原因である。さらに数年という長期に渡るスマートシティ計画では、将来的な処理規模の拡大に応じて、そのデータの蓄積と分析を支える分析インフラを柔軟に拡張できる必要がある。

本研究の目的は、上記の問題を解決し、スマートシティ上で発生する大規模ログデータの蓄積と活用を支援するプラットフォームの実現である。その第一歩として、本稿では多種多様な都市ログのうち、個々の家庭内で発生するログ情報（住宅ログ）に焦点を絞る。

本稿では、並列分散処理基盤である Hadoop [3] と HBase [4] を組み合わせたスマートシティ向けデータ処理プラットフォーム Scallop4SC (SCALable LOgging Platform for Smart City) を提案する。Scallop4SC では、まず各家庭の住宅ログをネットワークを通じて収集し、HBase 上の分散データベースに格納する。HBase は、データの正規化を意識することなく単純な Key と Value の組として扱うため、多様性に富む住宅ログを一元管理することができる。蓄積されたログは Hadoop の複数ノードによる並列分散処理が施される。各種スマートシティサービスは Scallop4SC の提供する API を通じて、住宅ログに基づいたサービスを実現する。

提案プラットフォームの有効性を確かめるために、実際のスマートホーム環境で計測された住宅ログを用いて、その処理効率に関する評価実験を行う。さらに得られた実験結果に基づき、実在の都市規模ログ処理に対する提案プラットフォームの適用可能性について検討する。

2. 準備

2.1 スマートシティ

スマートシティとは最新の IT 技術を駆使し、エネルギーをはじめとする生活インフラ全体の高度な効率化を目指した次世代の都市のことである。スマートシティでは都市中に設置されたセンサにより、エネルギーの使用状況や、交通量などの環境情報が計測される。計測されたデータは、広帯域のネットワークを通じて収集・統合され、リアルタイムでのデータ処理技術により社会サービスとして住民に還元される。具体的な社会サービスの例としては、電力消費把握による地域単位での省エネ計画や、交通状況把握による交通網最適化など多岐にわたる。現在、スマートシティは理論的な枠組みに留まらず、シンガポールやアムステルダムなどの世界中の様々な都市で実証実験が開始されている [1], [2]。

我々の研究グループでは、家庭内に設置された家電機器やセンサのネットワークに接続し、様々な付加価値サービスを実現するホームネットワークシステム（以下 HNS）の研究・開発を行っており [5]、実際の住宅機器を用いた HNS 環境 “CS27-HNS” を構築している。HNS では、機器の自立制御や複数家電の連携動作といった、快適で便利な宅内サービスを実現できる。また、家電機器の利用状況やセンサにより取得された環境値を、ネットワークを通じて容易に取得することが可能である。

本稿では、HNS のようなスマートホーム環境が設置された住宅の集合により構成される都市を、スマートシティの一つの実現形態と考える。このスマートシティ環境下では、各家庭の消費電力や家電機器の利用状況といった宅内のログ情報がリアルタイムに収集・分析され、様々なスマートシティサービスが実現される。

2.2 住宅ログに基づく社会サービス

本稿では、宅内における消費電力や温度などの環境値、及び家電機器の状態などの住宅に関するあらゆるログのことを住宅ログと呼ぶ。住宅ログの蓄積により、宅内での消費エネルギーの可視化や、機器の使用状況の振り返りといったサービスが実現できる。さらに単一家庭に留まらず、都市単位での横断的なログ蓄積により、より高い付加価値を持つサービスを実現できると考えられる。先に挙げた地域単位での省エネ計画はその一例である。また、他の家庭との比較による自己改善の推進や、住居周辺でのトレンド把握といった様々なサービスの提供も期待できると考える。

2.3 スマートシティにおけるデータ活用の課題

スマートシティでの住宅ログの収集・活用に対しては、以下に挙げる 3 つの課題が考えられる。

課題 P1: ログの肥大化

ログ情報は一定周期での計測が必須であるという特性から、その量が膨大になることが想定される。例えば、30 個の家電が設置された一家庭内での住宅ログを 1 分に 1 回取得することを考えた場合、一日で 43,200 件のログデータが発生する。市町村区といった地域単位では、さらに数千から数万の世帯数を乗じた値のログデータが発生する。また、高いリアルタイム性を持つサービスを実現するためには、そのデータ取得頻度を上げる必要があり、取得されるログデータ量も爆発的に増加する。

課題 P2: データベーススキーマを事前に決定できない

スマートシティでは、ログデータ格納のためのデータベースのスキーマを事前に決定することはできない。家庭によって設置される家電機器やセンサが異なることがその原因の一つである。また、家電機器の高性能化により取得すべき機器の状態が増加するといったケースも想定されるため、従来の関係 DB で用いられるような一貫性のあるデータスキーマのみでログを管理することは難しい。

課題 P3: インフラ需要の拡大に対する不透明性

スマートシティのデータ活用を支えるインフラに対して、将来必要とされるデータベースの規模や計算コストを見積もることは困難である。数年や数十年といった長期に渡るスマートシティ計画を支えるためには、世帯数の変化や家電機器の高性能化といった様々な環境の変化に追従できる必要がある。また、スマートシティ計画自体も実証実験が始まったばかりであり、将来どのように発展し拡大化されていくかも不透明である。

2.4 Hadoop と HBase

本節では本稿での提案フレームワークに組み込む 2 つの技術、Hadoop と HBase について説明を行う。

Hadoop は大規模なデータを効率的に分散処理するためのソフトウェア基盤である。複数ノードでの並列化を行うため大規

模データの処理に適しており、またサーバを構成するノードを柔軟に追加、拡張可能である（スケールアウト）といった利点を持つ。Hadoop 上でのデータ処理は Map フェーズと Reduce フェーズの 2 つのフェーズで行われる。Map フェーズでは、入力データを更に細かい単位に分割し、複数のノードに分配する。Reduce フェーズでは、Map フェーズでの処理結果を集約し、最終的なデータ処理リクエストに対する結果を得る。

Hadoop におけるサーバクラスターは一つの Master ノードと複数の Slave ノードから構成される。Master ノードは、クラスター全体のファイルの管理と処理の管理を担当する。Slave ノードは分散されたデータの格納と、Master ノードからの通信に従い Map フェーズ、Reduce フェーズに応じた処理を行う。

HBase は Hadoop 上で構築される並列分散データベースシステムであり、大規模かつ大量のデータを複数のノードマシンで管理する。HBase の分散ファイルシステム上では、KVS (Key Value Store) [6] と呼ばれるデータの管理方法が用いられる。KVS で管理される各データは、データを一意に識別するための値 (Key) と、保存対象となるデータの内容 (Value) の単純な組で表される。DB からのデータ取得・検索は Key のみが用いられるため、膨大な量のデータであっても高速なデータの取得が可能である。KVS は関係データベースのようにデータの正規化を行わないため、複雑なデータ構造の管理には不向きであり、またデータの更新処理ができない一方で、ログデータのような構造がシンプルなデータの処理に適する。

3. スマートシティにおける大規模住宅ログの収集・活用プラットフォーム

3.1 目的とアプローチ

本研究の目的は、スマートシティにおける大規模住宅ログデータの収集と活用を支援するための枠組みの提供である。そのための手段として、大規模並列処理のためのフレームワーク Hadoop と、並列分散データベース HBase を利用した住宅ログの収集・活用基盤、Scallop4SC を提案する。

スマートシティを対象とした大規模データの収集・活用基盤に並列分散データベースを組み込むことで、膨大な量のデータ蓄積と処理に対して高いスケーラビリティを確保することが可能である。これにより課題 P1 の解決を試みる。HBase では各データ要素を Key と Value の組で表すため、厳密なデータベーススキーマを決定する必要がない。この特徴は課題 P2 の解決策となる。さらに Hadoop の持つスケールアウトという特性により、課題 P3 に挙げた将来的な計算コスト需要の拡大に対して、柔軟に対応することが可能となる。

3.2 全体アーキテクチャ

提案プラットフォーム Scallop4SC の全体アーキテクチャを図 1 に示す。構成要素は住宅ロガー、スマートシティ向け住宅ログ収集基盤、スマートシティサービス用 API、スマートシティサービスである。

まず、各家庭で発生する住宅ログは住宅ロガーにより収集され、ネットワークを通じて基盤に蓄積される。基盤は、HBase の提供する分散 KVS と、Hadoop による分散処理システムが

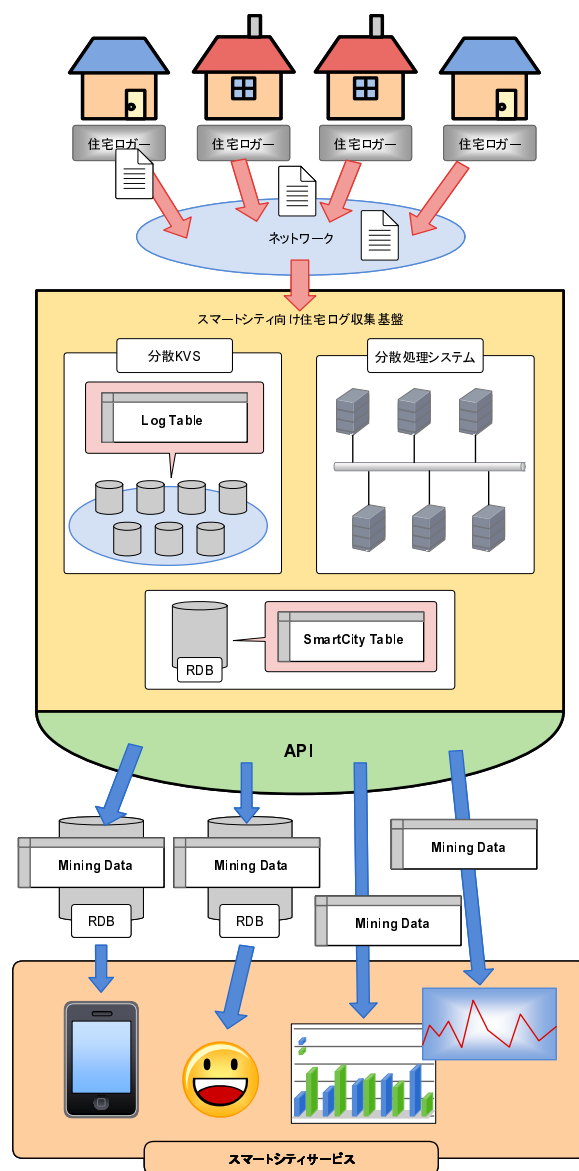


図 1 提案プラットフォームの全体アーキテクチャ

ら構成されている。取得された住宅ログは、HBase 上の分散 KVS と関係データベースを組み合わせたデータ蓄積基盤に蓄積・管理され、Hadoop の提供する分散システムにより解析処理が施される。基盤を利用するサービス開発者は、基盤の提供する API を通じてログの解析結果を取得することが可能である。

3.3 住宅ロガー

住宅ログの収集方法としては、現在様々な技術が利用可能である。家電状態を取得する場合、ネットワーク家電のようなりモート制御可能な機器を用いるなどの手段が考えられる。また消費電力は、分電盤経由で宅内の全機器の消費電力を計測する方法や、個々のコンセントにネットワーク機能を持った専門の端末を接続し、LAN 経由で計測するといった様々な方法がある。

CS27-HNS では消費電力の計測方法として、ECO マネシステム [7] を用いて個々の家電機器の消費電力を取得している。また、全ての家電とセンサが Web サービスとして公開されており、ネットワーク経由により機器状態と環境値を容易に取得で

きる．具体的には，`http://cs27-hns/TVService/getStatus` といった REST プロトコルによる呼び出しを行うことで，テレビの機器状態を `[power:on, channel:6, volume:15]` のようなデータ形式で取得できる．

住宅ロガーは全ての機器とセンサ，及び消費電力の計測端末に定期的に関わり合わせ，住宅ログの収集を行う．収集ログに対して，その取得日時や家庭固有の機器 ID，家庭 ID の付与といった DB 投入のためのデータ整形・変換を行う．最終的にスマートシティの持つ分散データベースへログを投入する．

ログの計測頻度と DB への投入頻度が高いほど，リアルタイム性の高い社会サービスが実現可能である一方で，データの規模が爆発的に増加する傾向がある．そのため，ログの計測・投入頻度はどのようなサービスを提供するかによって決定されるべきである．

3.4 大規模住宅ログ向けデータ設計

スマートシティ向けのデータ蓄積基盤として，住宅ログを HBase の分散 KVS に蓄積し，住所や設置機器といった住宅情報を関係データベース (RDB) に蓄積するという方針を取る．前者の住宅ログは更新不要かつ大量に発生する傾向があり，また分析処理の対象となりやすいため，分散 KVS への設置に適する．後者の住宅情報は更新処理が必須で，かつデータ構造が複雑であるため RDB での管理が適する．これら 2 つのデータベースシステムを用いることで，高いスケーラビリティを確保しつつ効率的なデータ処理が実現できる．

住宅ログ記録のための KVS データベースを表 1 に，住宅情報記録のための RDB スキーマを図 2 に示す．KVS の Key の値として，家 ID と機器 ID を用いており，これら 2 つの ID は図 2 に示す住宅情報データベースと対応する．スマートシティサービスでは，家庭単位でのデータ取得や，機器単位でのデータ取得が頻りに利用されると考えられるため，これら 2 つの ID を Key として組み込んだ．Key の最後のタイムスタンプは特定の期間のデータ取得・検索を可能とするために付与した．Column Families にはログの内容 (Value) が記録されており，“state” の部分が具体的なログデータの内容となる．1 列目の場合，山本家 (家 ID: 00036) のテレビ (機器 ID: 001) が 2012 年 1 月 30 日 1 時 45 分に電源がオンでチャンネル 2，音量 14 であったことを示す．また 2 列目はその時のテレビの消費電力が 300W であったことを意味する．

図 2 に文献 [8] の表記法に従った ER 図を示す．四角はテーブルを表しており，テーブル名，主キー，属性のスキーマが並ぶ．テーブル下部にはインスタンスを併記している「+←€」はテーブル間の親子関係を表す．図 2 では，山本家 (家 ID: 00036)，本田家 (家 ID: 00037)，鈴木家 (家 ID: 00038) の 3 つの家庭がスマートシティ上に存在しており，山本家には機器として TV (機器 ID: 001) と天井照明 (機器 ID: 002) が設置されていることを表している．

3.5 スマートシティサービス用 API

基盤が提供する API としては，以下に示す 3 つの種類が考えられる．1．単純なデータの取得操作，2．汎用的なデータの処理・分析操作，3．MapReduce による複雑なデータ処理操

表 1 分散 KVS 上の住宅ログスキーマ

Row Key	Column Families		
<家 ID>-<機器 ID>-<LongMax>	date: 日付	time: 時刻	state: データ型:値
00036-001-1327...5500	20120130	1:45:27	power:on, ch:2, vol:14
00036-001-1327...5500	20120130	1:45:27	300Wh
00036-002-1327...5500	20120130	1:45:27	power:on, brightness:10
00036-002-1327...5500	20120130	1:45:27	40Wh

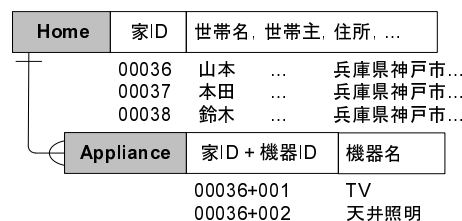


図 2 RDB 上の住宅情報スキーマ

表 2 使用ソフトウェアのバージョン

Software	Version
Hadoop	hadoop-0.20.2+737
HBase	hbase-0.90.3
Pig	pig-0.9.1

作．1 つ目は DB 上の生のデータを取りだす API であり，単純で最も汎用的であるが，具体的なデータ処理解析は個々のサービスが独自に行う必要がある．2 つ目は蓄積データに対する汎用的なデータ処理操作を提供するものである．具体的には数値の集計や平均，最大，最小値の取得などである．これら 2 つの API の実現方法として，Hadoop 上のデータ取得や MapReduce 処理を対話型で実現する Pig [9] を利用する．Pig を基盤に組み込むことで，上記 2 つの API を対話型 API として実現することができる．3 つ目の API は個々のサービスに特化した MapReduce 処理を支援するためのものである．Map 処理と Reduce 処理は個々の処理内容に依存するところが大きく，汎用的な API 操作のみではその実現可能性に限界がある．また処理効率の観点からも，汎用操作の組み合わせではなく処理内容に特化したプログラムを作成することで，ノード間の通信オーバーヘッドを削減し処理の大幅な向上が見込める．この API はサービス開発者が独自に作成した MapReduce プログラムを入力として，その処理結果をサービス開発者に提供するという方法を現在検討している．

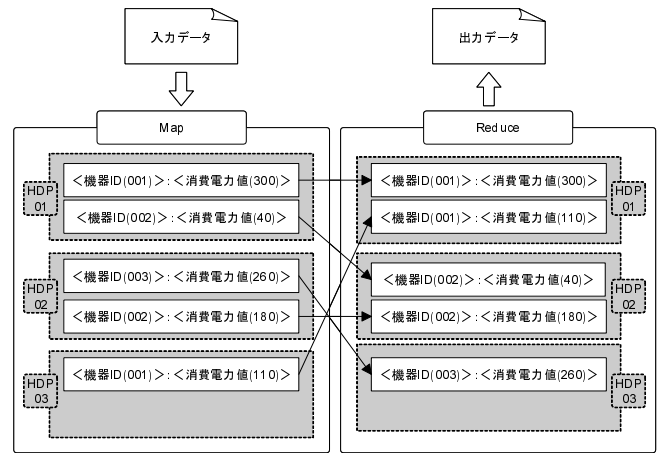
3.6 システム実装

これまでの議論に基づき，実際に Scallop4SC の実装を行った．利用ソフトウェアの概要を表 2 に，構築されたクラスタ構成を表 3 に示す．クラスタは全 12 台のマシンから構成されており，HDP00 が分散処理を一元管理するの Master ノードである．HDP01 から HDP03 は ZooKeeper^(注1) ノードとした．Hadoop の持つハードウェアのヘテロ構成が可能であるという利点を実際に確かめるために，2 種類のスペック (Pentium 系と Athlon 系) のノードからクラスタを構成している．

(注1): 分散システム上でのノード間の分散協調を支援するサービスである [10]．

表 3 クラスタ構成

Name	Type		Spec	
	Hadoop	HBase	CPU	Mem.
HDP00	Master	HRegionMaster	Pentium4 3.00GHz	400MB
HDP01	Slave	Zookeeper	Pentium4 3.00GHz	1GB
HDP02	Slave	Zookeeper	Pentium4 3.00GHz	1GB
HDP03	Slave	Zookeeper	Pentium4 3.00GHz	1GB
HDP04	Slave	HRegionServer	Pentium4 3.00GHz	1GB
HDP05	Slave	HRegionServer	Pentium4 3.00GHz	1GB
HDP06	Slave	HRegionServer	Pentium4 3.00GHz	1GB
HDP07	Slave	HRegionServer	Pentium4 3.00GHz	1GB
HDP08	Slave	HRegionServer	Pentium4 3.00GHz	1GB
HDP09	Slave	HRegionServer	Athlon 64 3800+	1GB
HDP10	Slave	HRegionServer	Athlon 64 3800+	1GB
HDP11	Slave	HRegionServer	Athlon 64 3500+	1GB



4. 評価実験

4.1 実験概要

提案プラットフォーム Scallop4SC を利用した際のデータ処理効率を確かめるために、実際の HNS で蓄積された住宅ログを用いた評価実験を行う。実験では、Scallop4SC 環境を利用した汎用 API によるデータ処理方法と MapReduce プログラムによるデータ処理方法、及び関係データベースを用いた従来手法の 3 つの比較を行う。題材として、電力消費の振り返りサービスを取り上げる。具体的な実験のタスク内容は数ヶ月から数年規模の住宅ログの中から、機器ごとの消費電力を集計するというものである。

4.2 処理対象のデータ

CS27-HNS ではこれまでに約 2 年間に渡り、32 個の家電機器に関する消費電力ログと家電状態ログを蓄積している。消費電力ログは 6 秒に 1 回、家電状態ログは 1 分に 1 回の頻度で蓄積しており、全体で約 4 億 6 千万件のログデータとなっている。実験では、この住宅ログのうちの消費電力ログデータの 1 年分（約 1 億 8 千万件）を、スマートシティ内の単一家庭の 1 年分のログデータに相当すると仮定する。このログデータを数回複製することで、単一家庭で発生する数ヶ月～数年規模のログデータを擬似的に生成し、その処理効率を計測する。

4.3 データの処理方法

4.3.1 提案手法 1 : Pig 法

Scallop4SC 環境上での Pig による汎用的なデータ処理操作を組み合わせ、消費電力の集計を行う手法である。MapReduce プログラムを作成せず、システムとの対話を通じてデータ処理を行えるという利点を持つ。Pig による消費電力の集計処理コマンドを以下に示す。

```

1 records = LOAD 'ConsumptionLog.csv'
2 USING PigStorage(',')
3 AS (<機器 ID>:int, <消費電力>:long);
4 grouped_records = GROUP records BY <機器 ID>;
5 sum = FOREACH grouped_records
6 GENERATE group, SUM(records, <消費電力>);
7 STORE sum INTO 'Result.txt';

```

1～3 行目は Hadoop に保存されているデータの解釈部分で

ある。4 行目は同じ機器 ID の列を連想配列の形でまとめる Map 処理であり、5～6 行目で同機器 ID の消費電力を集約する Record 処理を行っている。

4.3.2 提案手法 2 : MapReduce 法

Scallop4SC を利用し、消費電力集計のために作成した MapReduce プログラムにより、データ処理を行う手法である。本手法の処理の流れを図 3 に示す。まず Map 処理により、複数ノードに分散配置されたログデータに対し、機器 ID をキーとした連想配列形式に変換する。Shuffle 処理によって複数ノードへのデータの再配置が行われ、Reduce 処理により機器 ID ごとの消費電力が集計される。実際に実験に用いた MapReduce 用の Java プログラムは、空行とコメント行を除き 57 行であった。

4.3.3 従来手法 : SQL 法

関係データベースに格納された住宅ログから、SQL コマンドによりデータの集計を行う従来手法である。具体的には SQL の “GROUP BY” コマンドにより、家電機器ごとの消費電力を集計し出力する。実際に用いた処理コマンドを以下に示す。

```

1 SELECT ApplianceId, Sum(powerConsumption)
2 FROM ConsumptionTable
3 GROUP BY ApplianceId

```

4.4 結果

実験の結果を図 4 に示す。横軸がデータ件数であり、左のプロット点から順に 1ヶ月、2ヶ月、3ヶ月、6ヶ月、1年、2年、3年、4年の 1 世帯のデータに該当する。縦軸は処理時間を表す。

どの手法についても、その処理時間はログデータの規模とほぼ正比例する結果となった。1ヶ月のデータのみ SQL が最速であったが、おおむね MapReduce, SQL, Pig の順で処理速度が速かった。最も巨大な 4 年分のログを処理する場合、SQL では約 26 分であったが、Scallop4SC を利用した MapReduce では約 14 分と半分程度の時間で処理を終えていた。一方で同じ Scallop4SC を利用する Pig はどのケースにおいても最も処理速度が遅かった。

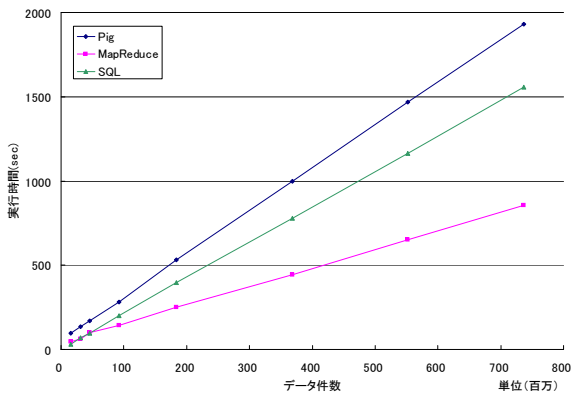


図 4 実行時間の計測結果

表 4 3つのデータ処理方法の特徴

	スケーラビリティ	処理速度	実装工数
Pig	high	low	middle
MapReduce	high	high	large
SQL	low	middle	middle

表 5 実験結果より推測されるスマートシティ規模ログの処理時間(分)

ログ取得頻度	神戸市	灘区	名護市
1分に1回	730	65	27
6秒に1回	7,298	651	267

5. 考 察

5.1 実験結果に対する考察

データの処理速度という観点では、MapReduce が最も高速であった。1ヶ月程度の小規模データでは複数ノード間のネットワーク通信がボトルネックになっていた一方で、大規模データでは、複数ノードによる並列処理の効果が顕著に表れたと考えられる。一方 Pig は MapReduce と同じ並列処理を行っているにも関わらず、最も速度が遅かった。

実験結果と上記の考察から得られる各手法の特徴を表 4 に示す。MapReduce と Pig は Hadoop 環境上で動作するため高いスケーラビリティを持つ。処理速度の面では MapReduce が優秀であるが、分散処理を意識したプログラミングが必要であり、3つの手法のうち最も実装工数が大きい。Pig は対話操作可能であるため少ない工数で実現可能であるが、処理性能が低い。Scallop4SC 上では、一時的な結果の確認やテストのためには Pig を利用し、実際のデータ処理は MapReduce を用いるといった使い分けが有効であると考えられる。

5.2 スマートシティ規模データへの擬似適用

実験では単一世帯における数ヶ月から数年規模のデータ処理速度を計測した。この実験結果に基づき、複数世帯から構成されるスマートシティ規模データへの適用可能性について考察する。題材都市は政令指定都市である神戸市(約 70 万世帯)と、神戸市の一区である灘区(約 6 万 2 千世帯)、及び沖縄県の名護市(約 2 万 5 千世帯)の 3つである。各都市の世帯数分のデータ複製を行った場合の処理時間を、実験結果から擬似的に算出した。推測される各都市の 1 日分の処理時間を表 5 に示す。

まず 1 分に 1 度ログを取得する場合について議論する。神戸市規模に対するデータ処理時間は約 12 時間であった。消費電力集計のような比較的単純なデータ処理でも、約半日の処理時間が必要という点で非実用的な結果であるといえる。一方、灘区や名護市のような数万世帯規模であれば、約 1 時間と実用的な時間で処理を終えることが可能であった。

よりきめ細やかで正確な消費電力の把握を行う場合を想定し、ログの取得頻度を 10 倍(6 秒に 1 度)に変更した場合について考察する。最も規模の小さい名護市であっても処理時間は 4 時間以上であり、このログ取得頻度に対しては大幅な性能改善が必要であった。Scallop4SC はスケールアウトを容易に実現できるため、現在 12 台のノードで構築されたクラス構成を拡張することで、大都市、及び高頻度のログデータ処理に適応できると考えられる。

6. おわりに

本稿では、スマートシティにおける大規模住宅ログの収集と活用の支援を目的とし、Hadoop と HBase を組み合わせたデータ処理プラットフォーム Scallop4SC を提案した。実際の HNS で収集された住宅ログを用いた評価実験により、Scallop4SC を用いることで、従来の SQL コマンドに比べて約半分程度の時間で大規模データを処理できることが明らかとなった。

現在 Scallop4SC の提供する API は設計段階であり、その実装に関しては今後の必須課題である。また、Scallop4SC を利用した住宅ログを活用する具体的なスマートシティサービスを考察し、実装、評価を行いたい。

謝辞 この研究の一部は、科学技術研究費(基盤研究 B 23300009, 若手研究 B 21700077, 研究活動スタート支援 22800042)、及び、ひょうご科学技術協会の助成を受けて行われている。

文 献

- [1] R.G. Hollands, "Will the real smart city please stand up?," City: analysis of urban trends, culture, theory, policy, action, vol.12, no.3, pp.303-320, 2008.
- [2] A. Mahizhnan, "Smart cities: The singapore case," Cities, vol.16, pp.13-18, 1999.
- [3] D. Borthakur, "The hadoop distributed file system: Architecture and design," 2007. http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf
- [4] A. Khetrapal and V. Ganesh, "Hbase and hypertable for large scale distributed storage systems," 2006.
- [5] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K. Matsumoto, "Constructing home network systems and integrated services using legacy home appliances and web services," International Journal of Web Services Research, vol.5, no.1, pp.82-98, 2008.
- [6] M. Seeger, "Key-value stores: A practical overview," Computer Science and Media. Ultra-Large-Sites, vol.SS09, pp.1-21, 2009.
- [7] パナソニック 電工株式会社, "Eco マネシステム". <http://denko.panasonic.biz/Ebox/kahs-eco/>
- [8] 渡辺幸三, 販売管理システムで学ぶモデリング講座, 翔泳社, 2008.
- [9] Apache Pig. <http://pig.apache.org/>
- [10] T. White, Hadoop, オライリー・ジャパン, 2010.