

# スマートシティにおける 大規模住宅ログ活用プラットフォームのための API 実装

山本晋太郎<sup>†</sup> 松本 真佑<sup>†</sup> 中村 匡秀<sup>†</sup>

<sup>†</sup> 神戸大学 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

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

あらまし 我々は、スマートシティにおける住宅ログを蓄積・活用するためのデータプラットフォーム Scallop4SC の提案・開発を行なっている。これまでの先行研究で、Hadoop/HBase クラスタを用いた Scallop4SC のプロトタイプ の作成、省エネサービスや付加価値サービスの検討を行った結果に基づいた Scallop4SC 内のスマートシティ構成情報 のデータモデリング、汎用的な API の設計を行なってきた。本稿では、Scallop4SC 上で HBase を効率良く使用する住宅ログ向けデータモデリング、RESTful Java を利用した API の実装を行い、実環境で得られた住宅ログの蓄積 を行う。また、実装した Scallop4SCAPI を使用して、消費電力可視化アプリケーションの開発を行う。実アプリケーションの開発を通して、Scallop4SC の現状を評価し、今後必要となる機能について考察する。

キーワード スマートシティ, 住宅ログ, Scallop4SC, API, アプリケーション

## Implementing API of Large-scale House Log Data Platform in Smart City

Shintaro YAMAMOTO<sup>†</sup>, Shinsuke MATSUMOTO<sup>†</sup>, and Masahide NAKAMURA<sup>†</sup>

<sup>†</sup> Kobe University Rokkoudai-cho 1-1, Nada-ku, Kobe, Hyogo, 657-85301 Japan

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

**Abstract** Smart city is a next-generation city planning. In the smart city, some value-added services are provided using wide variety of logs collected from various appliances and sensors. In our previous work, we have been proposed and developed a smart city platform, called Scallop4SC that supports collecting and processing the extremely large-scale log data. This system stores variety of logs on key-value store, and supports the statistical processing of the logs on Hadoop. And we designed a meta-data scheme for smart city configuration information, designed Scallop4SCAPIs for accessing stored logs and meta-data. In this paper, we implement storing house logs and Scallop4SCAPIs based on previous work, and implement concrete application using Scallop4SC. And we discuss about the current state of Scallop4SC.

**Key words** Smart city, house log, Scallop4SC, API, Application

### 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、社会全体でのエネルギーの高効率化を

目的としたサービス、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情報を活用したスマートシティのサービス基盤として、我々は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) の提案を行っている [3]。Scallop4SC は以下に示す 4 つの特徴を持つ。

- P1: 都市内で発生する様々なログ情報の収集及び蓄積
- P2: 蓄積ログの効率的な処理

- P3: スマートシティに関する構成情報の一元管理
  - P4: P1 から P3 のデータへの汎用アクセス API の提供
- Scallop4SC の提供 API を利用することで、サービス開発者はログの収集や管理、その処理方法などを意識することなく、効率的にスマートシティサービスを開発することが可能となる。

先行研究 [3] で、上記の P1 (ログの管理) と P2 (ログの処理) の実現を目的として並列分散処理基盤 Hadoop [4] と HBase [5] を利用した Scallop4SC のプロトタイプシステムを開発した。このシステムは、都市内の様々なログ情報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 では全てのデータを単純な Key と Value の組として扱うため、多様性に富むログ情報を一元管理することができる。蓄積されたログは Hadoop を構成する複数ノードにより並列分散処理が施される。これらの並列分散処理基盤を利用することにより、テラバイトからペタバイトスケールのログが発生すると考えられるスマートシティ上での、効率的なデータ管理とデータ処理の実現が期待できる。

次に先行研究 [6] において、P3 (スマートシティ構成情報の管理)、及び P4 (汎用 API の提供) についての設計を行うために大規模な住宅ログを用いることで実現可能なサービスについて検討を行った。この検討結果に基づき、スマートシティ構成情報としてどのような情報を管理する必要があるか検討し、そのデータ設計を行った。同様にサービス実現のために、どのような API を提供する必要があるかを検討し、その API の設計を行った。

本研究では、先行研究 [6] での設計に基づき、スマートシティ構成情報及び Scallop4SCAPI の実装を行う。また、実際のスマートハウス環境における消費電力可視化アプリケーションを Scallop4SCAPI を用いて作成し、現段階での Scallop4SC の実用性評価を行う。

## 2. 先行研究

### 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]。

### 2.2 住宅ログ

スマートシティで収集・活用されるログ情報としては、交通や消費電力などの生活インフラに関するものや、住民自身の行動に関わるものなど多岐に渡る。本稿ではその中でも、住宅に密接に関わるもので取得可能なログ (住宅ログ) に焦点を絞

る。住宅ログはその性質からエネルギーログ (Energy)、機器ログ (Device)、環境ログ (Environment) という 3 つに分類できる。

### 2.3 Scallop4SC

我々は、スマートシティ内で取得される多種多様かつ巨大なログデータを蓄積・活用するための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) の開発を行なっている。Scallop4SC のアーキテクチャを図 1 に示す。各住宅から収集されたログ情報はネットワークを通じて、Scallop4SC の管理する HBase の分散 KVS データベース上に蓄積され、Hadoop の分散処理基盤により処理が施される。スマートシティ全体の構成情報は関係データベース上で管理される。各種スマートシティサービスは Scallop4SC の提供する API を通じてログ情報や構成情報にアクセスし、住民へのサービスとして提供される。

Scallop4SC は以下の 4 つの特性を持つ。

#### P1: 分散 KVS を用いた住宅ログ蓄積

ログ情報の肥大化と多様性に適応するために、HBase の分散 KVS データベースを用いてデータの蓄積と管理を行う。

#### P2: 蓄積した住宅ログの並列分散処理

効率的な蓄積ログの処理を実現するために、Hadoop を用いた並列分散処理をサポートしている。

#### P3: スマートシティ構成情報の管理

スマートシティの構成情報を RDB を用いて一元的に管理する。

#### P4: Scallop4SC にアクセスする汎用 API

P1 から P3 の機能へアクセスするための汎用 API を持つ。

### 2.4 HBase を用いた住宅ログの蓄積

住宅ログの蓄積には、分散 KVS である HBase を使用する。このとき分散 KVS の 2 つの特性を利用する。1 つ目は、大規模化する住宅ログを分散して蓄積することで高いスケーラビリティを得ること、2 つ目は、多様性に富みスキーマを決定しづらい住宅ログに対応するために住宅ログを Key と Value という単純な形式で蓄積することである。

例えば、以下のように消費電力や操作履歴、センサ値など、タイプや内容の異なる 5 つの住宅ログを Key と Value として柔軟に保存することができる。

Key	Value
2012-07-28T12:34:56.cs27.Energy.tv01	{value:600, unit:W}
2012-07-28T12:34:56.cs27.Device.tv01	{status:[power:off]}
2012-07-28T12:35:00.cs27.Device.tv01	{operation:on()}}
2012-07-28T12:34:56.cs27.Env.temp2	{value:24.0, unit:celsius}
2012-07-28T12:34:56.cs27.Env.pcount3	{value:3, unit:people}

この例では、Key を [date,time, house ID, log type, device ID] という、すべての住宅ログに対して固有の文字列としている。このように分散 KVS を用いることで高いスケーラビリティの確保とスキーマレスなデータ蓄積を可能としている。

### 2.5 スマートシティ構成情報

先行研究 [6] において、サービス実現を考慮したスマートシティ構成情報を設計した。図 2 にスマートシティ構成情報の

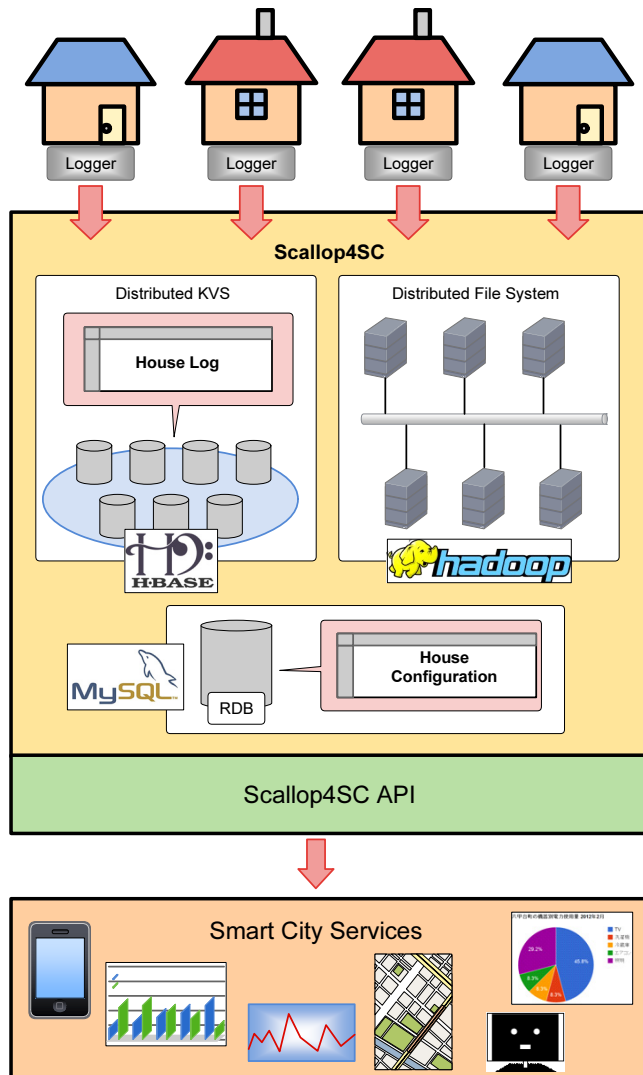


図 1 Scallop4SC のアーキテクチャ

ER 図を示す。この ER 図は文献 [7] で示された記法に基づいている。四角はエンティティを表し、データ項目を右に並べる。データ項目の下線は主キーを表し、[] は外部複合キーを表している。各エンティティの下部には、インスタンスを併記している (+ ) は親子関係 (+ ... ) は参照関係を表す。

提案するスマートシティ構成情報は、大きく住居情報 (House)、機器情報 (Device)、個人情報 (Person) という 3 つの情報に分類される。

これら 3 種類の情報を目的に応じて組み合わせることで、住宅ログの効率的な検索や統計が可能となる。

## 2.6 Scallop4SCAPI

??章と同様に、先行研究 [6] で、サービス実現のための Scallop4SCAPI を設計した。API がアクセス対象とするデータの種類やアクセス方法の違いから、以下の 5 種類のサービスに API を分類した。

- Log: ログデータの入出力 API
- Status: 現在状態の出力 API
- Configuration: スマートシティ構成情報の入出力 API
- Operation: スマートハウス上の機器操作 API

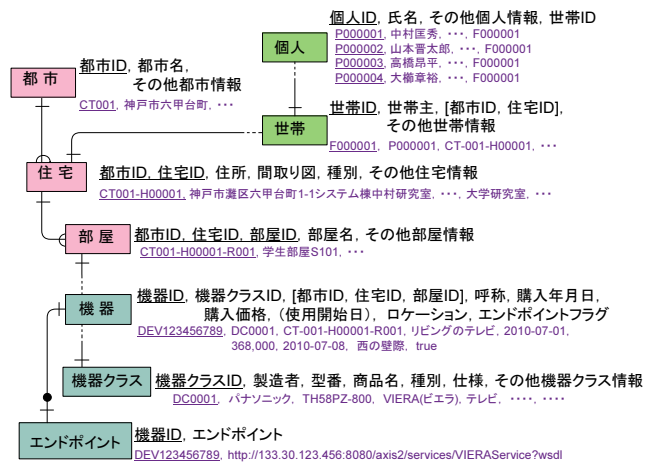


図 2 スマートシティ構成情報のデータスキーマ

また、4 つの API の具体的な中身について設計を行った。

図 3 に API の設計図を示す。左端が上記の 4 種類の API を示している。Type は操作対象のデータのタイプであり、3 種類のログデータと 3 種類の構成情報エンティティが含まれる。Access は操作内容の種類を示しており、データの入力 (put) とデータの取得 (get) の 2 つに大別されている。Query は API の呼び出しクエリであり、都市や家、機器、人などの、どのデータを取得 (あるいは入力) したいかを示すものである。

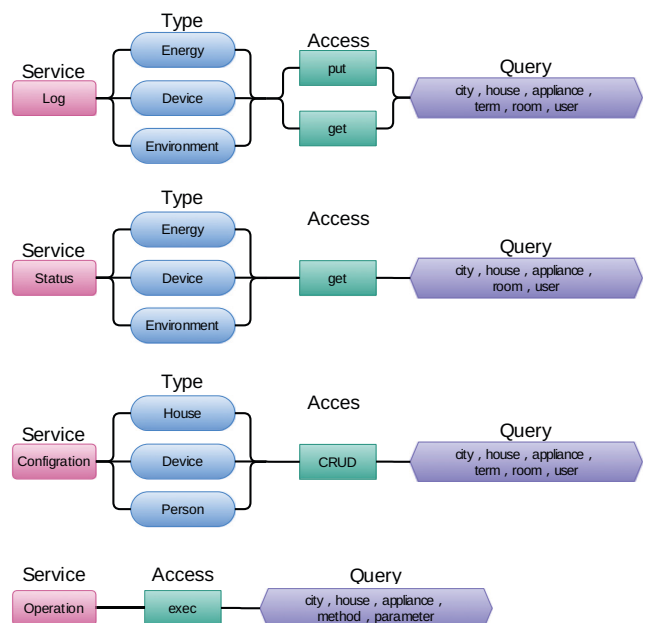


図 3 スマートシティ向け API 設計図

## 2.7 研究の目的とアプローチ

本研究では、先行研究 [6] での設計に基づき、住宅ログ蓄積および Scallop4SCAPI の実装を行う。また、実際のスマートハウス環境における消費電力可視化アプリケーションを Scallop4SCAPI を用いて作成し、現段階での Scallop4SC の実用性評価を行う。

表 1 House Log Table

Row Key	Column Families							
	Data:	Info:						
		date:	time:	device:	house:	unit:	location:	type:
hash()@<timeStamp>								
46018eab@2012-07-28T12:34:56	600	2012-07-28	12:34:56	tv01	cs27	W	living room	Energy
53cd5fad@2012-07-28T12:34:56	[power:off]	2012-07-28	12:34:56	tv01	cs27	status	living room	Device
8afd4abd@2012-07-28T12:35:00	on()	2012-07-28	12:34:56	tv01	cs27	operation	living room	Device
e85afaad@2012-07-28T12:34:56	24.0	2012-07-28	12:34:56	temp3	cs27	celsius	kitchen	Environment
13ffadbc@2012-07-28T12:34:56	3	2012-07-28	12:34:56	pcount3	cs27	people	living room	Environment
:	:	:	:	:	:	:	:	:

表 2 House Index Table

Row Key	Column Family
[timeStamp.type.house ID.device ID]!	hlkey:
	hash()@<timeStamp>
current.Energy.cs27.tv01	46018eab@2012-07-28T12:34:56
2012-07-28T12:34:56.Energy.cs27.tv01	46018eab@2012-07-28T12:34:56
2012-07-28T12:34:56.Energy.tv01.cs27	46018eab@2012-07-28T12:34:56
2012-07-28T12:34:56.cs27.Energy.tv01	46018eab@2012-07-28T12:34:56
2012-07-28T12:34:56.cs27.tv01.Energy	46018eab@2012-07-28T12:34:56
2012-07-28T12:34:56.tv01.Energy.cs27	46018eab@2012-07-28T12:34:56
2012-07-28T12:34:56.tv01.cs27.Energy	46018eab@2012-07-28T12:34:56
Energy.2012-07-28T12:34:56.cs27.tv01	46018eab@2012-07-28T12:34:56
Energy.tv01.2012-07-28T12:34:56.tv01.cs27	46018eab@2012-07-28T12:34:56
Energy.cs27.2012-07-28T12:34:56.tv01	46018eab@2012-07-28T12:34:56
Energy.cs27.tv01.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
Energy.tv01.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
Energy.tv01.cs27.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
cs27.2012-07-28T12:34:56.Energy.tv01	46018eab@2012-07-28T12:34:56
cs27.tv01.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
cs27.Energy.2012-07-28T12:34:56.tv01	46018eab@2012-07-28T12:34:56
cs27.Energy.tv01.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
cs27.tv01.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
cs27.tv01.Energy.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
tv01.2012-07-28T12:34:56.Energy.cs27	46018eab@2012-07-28T12:34:56
tv01.2012-07-28T12:34:56.cs27	46018eab@2012-07-28T12:34:56
tv01.Energy.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
tv01.Energy.cs27.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
tv01.cs27.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
tv01.cs27.Energy.2012-07-28T12:34:56	46018eab@2012-07-28T12:34:56
:	:

### 3. 住宅ログ蓄積

2.2章で述べた住宅ログは Scallop4SC の HBase 上で管理される。Scallop4SC では、分散 KVS として HBase を用いて高いスケーラビリティと大規模データの容易な分散を実現する。

しかし、KVS 特有の以下のような問題も幾つか存在する。

単純な HBase のデータモデルの一例を 2.4 章で示した。しかし、このようなデータモデルには以下のような問題がある。

#### (1) 柔軟な検索ができない

HBase では基本的に Row キーによる検索を行うという制約がある。この制約のために 2.4 章で示したモデルでは、日付による住宅ログの検索は可能だが、機器やログのタイプといった Row キーに含まれない情報による検索はできない。

#### (2) データが均一に分散されない

HBase 内部では Key によるソートを行いデータを蓄積するため、単調に増加するものを Key とした場合には、単一のサーバにデータレコードが集中する傾向がある。[8] によって、住宅ログが複数の HBase クラスタ内で均等に蓄積されず、パフォーマンスの低下を招く。

これらの問題を解決するため、我々は HBase 上に HouseLog と HouseIndex という 2 つの種類のテーブルの設計を行った。

HouseLog テーブルを表 1 に示す。HouseLog テーブルは住宅

ログを Data と Info という 2 つのカラムファミリーとして蓄積する。Data カラムは住宅ログの値を格納する。一方、Info カラムでは、住宅ログがもつ一般的なメタ情報を蓄積する。住宅ログの使用時に、Info カラムへのフィルタリングによって検索を行うためのものである。また、複数のサーバに均等にデータを蓄積するために、我々は Row Key として、ハッシュ関数によってエンコードされた文字列にタイムスタンプを付与して使用している。これにより Row Key を用いて直接 HouseLog テーブルを検索することができないが、この代用として HouseIndex テーブルを用いた検索を行う。

HouseIndex テーブルを表 2 に示す。HouseIndex テーブルは、HouseLog テーブルへのポインタを格納することでインデックスの役割を果たす。Row Key は、(1) タイムスタンプ、(2) ログタイプ、(3) 住宅 ID、(4) 機器 ID という要素の順列によって構成される。これにより、アプリケーションは任意の 4 種類の属性を指定することによって住宅ログを検索可能にしている。

また、最新の住宅ログへのポインタを格納する際には Row Key の先頭を 'current' にし、タイムスタンプを除去する。現在状態の取得の際にこの 'current' を検索することで検索を容易にするためである。hlkey というカラムファミリーには、HouseLog テーブルの Row Key が格納される。表 2 の全要素 (25 個 = 4!+1 個) は、全て同一の住宅ログ (46018eab) を指し示している。

アプリケーションは、まず HouseIndex テーブルに対して、タイムスタンプ、ログタイプ、住宅 ID、機器 ID のいずれかを指定する検索を行い、それに対応する HouseLog テーブルの Row Key を取得する。次に得られた Row Key の指定によって、HouseLog テーブルから住宅ログを得る。

このような提案データモデルにより (1) 柔軟な検索に限界がある (2) データが均一に分散されない、という 2 つの問題を解決する。

### 4. Scallop4SC の API 実装

2.6 章で示した設計に従って Scallop4SCAPI の実装を行った。実装には JAX-RS (Jersey v.1.21: JAX-RS (JSR 311) Reference Implementation for building RESTful Web services. [9]) を利用し、Tomcat 7.0 上で Web サービス化している。また、スマートシティ構成情報を保持する MySQL との通信に JDBC driver for MySQL (Connector/J v.5.1.21 [10])、HBase への通信に HBase の公式ライブラリを利用している。

具体的な API 実装の例を幾つか示す。

以下は指定した機器の指定時刻期間のエネルギーログを取得するための API の実装例である。

```
1  /**
2   * 指定時刻の範囲で与えられた機器の消費電力を返す。
3   */
4   エネルギーログ getEnergyLog(開始時刻 from,
5                               終了時刻 to, 機器 ID devId){
6
7       EnergyLog log[];
8       //検索開始キー
9       startRow = "Energy.deviceId.from";
10      //検索終了キー
11      stopRow = "Energy.deviceId.to";
12      //HouseIndex テーブルをスキャン
13      res1[] = indexTable.scan(startRow, stopRow);
14      for (r: res1) { //各行について
15          //キーのカラム値を取得
16          idx1 = r.getValue("hlkey", "");
17          //idx1 を行キーとして HouseLog テーブルから get;
18          res2 = logTable.get(index1);
19          //時刻とデータの組を log として挿入。
20          log.push({res2.getValue("info", "datetime"),
21                  res2.getValue("data", "")});
22      }
23      return log;
24 }
```

引数として開始時刻、終了時刻、機器 ID を渡し、複数の消費電力ログを取得する。6~10 行目でエネルギーログの Row Key (指定された時刻、機器 ID を含む) に対して検索を行うための検索キーを生成している。12 行目でその検索キーを用いて、HouseIndex テーブルへの検索 (スキャン) を行う。次に 15 行目~17 行目において、12 行目で取得したすべての結果に対して、hlkey の取得、hlkey を用いて HouseLog テーブルからエネルギーログを取得、という処理をする。最後に 19~20 行目のように、必要なデータを取得したエネルギーログから取り出し、返り値に追加し、エネルギーログを返す。

エネルギーログに限らず、機器ログや環境ログを取得するための API についても同様の処理によって実現することができる。

以下は指定した機器の現在の消費エネルギーを取得するための API の実装例である。

```
1  /**
2   * 与えられた機器の現在の消費エネルギーを返す。
3   */
4   消費エネルギー getEnergyStatus(機器 ID devId){
5
6       //検索キーをセット
7       rowKey = "current.Energy.devId";
8       //HouseIndex テーブルをスキャン
9       res1 = indexTable.scan(rowKey);
10      //キーのカラム値を取得
11      idx1 = res1.getValue("hlkey", "");
12      //idx1 を行キーとして HouseLog テーブルから get;
13      res2 = logTable.get(index1);
14      //データのカラム値を取得
15      return res2.getValue("data", "");
16 }
```

引数として機器 ID を渡し、消費エネルギーを取得する。6 行

目で指定された機器 ID を含むエネルギーログの現在状態を取得するための検索キーを生成している。この時、現在状態取得のために 3. 章で述べたように 'current' を検索キーの先頭とすることで容易に現在状態の検索ができる。8 行目ではその検索キーを用いて、HouseIndex テーブルへの検索 (スキャン) を行う。次に 10~12 行目において、8 行目で取得した結果に対して、hlkey の取得、取得した hlkey を用いて HouseLog テーブルからエネルギーログを取得する。最後に 14 行目のように、必要なデータを取得したエネルギーログから取り出し、返り値として返す。

先ほどと同じように、エネルギーログに限らず、機器ログや環境ログを取得するための API についても同様の処理によって実現することができる。

また、スマートシティ構成情報を操作するための API の実装には、JDBC を利用して SQL コマンドを検索キーとして生成するという処理を行なっている。スマートシティ構成情報を操作するための API は様々なものが考えられるが、スマートシティ構成情報を扱う API は SQL コマンドを柔軟に用いることで実装可能なので、有効な API を適宜追加することができる。

#### 4.1 Scallop4SCAPI の使用例

例えば、テレビ (tv01) の 10 月 26 日 18 時 00 分から 18 時 01 分間の消費エネルギーログを取得する場合を考える。このとき、getEnergyLog API を使用する。具体的には URL: <http://Scallop4SCAPI/Log/Energy.get?from=2012-10-26T18:00:00&to=2012-10-26T18:01:00&devId=tv01> にアクセスすると、以下のように XML 形式でテレビのエネルギーログが取得できる。

```
<houselog>
  <rowkey>
    1329fad6@2012-10-26T18:00:33
  </rowkey>
  <value>68.112</value>
  <info>
    <entry key="unit">W</entry>
    <entry key="home">cs27</entry>
    <entry key="time">18:00:33</entry>
    <entry key="location">s101</entry>
    <entry key="device">tv01</entry>
    <entry key="type">Energy</entry>
    <entry key="date">2012-10-26</entry>
  </info>
</houselog>
```

## 5. Scallop4SC を用いたアプリケーション

今回、我々は Scallop4SC を用いたアプリケーションとして消費電力可視化アプリケーションの作成を行った。

図 4 が今回作成したアプリケーションの動作画面である。

この消費電力可視化アプリケーションは、HTML および JavaScript によって実装された Web アプリケーションであり、JavaScript ライブラリとして JQuery(1.7.2) を用いた。また、インターフェースとして Google Chart API を利用している。

また、HTML ファイル、コメント、空行を除いたプログラム



図 4 消費電力可視化アプリケーション

としての有効行数は 148 行であり、開発時間はおよそ 5 時間ほどであった。

本アプリケーションは、2 種類の Scallop4SCAPI を使用して開発した。1 つ目は、指定した家の現在の消費エネルギーを返す Status API、2 つ目は、指定した家にある機器情報のリストを返す Configuration API である。まず、Status API を用いて消費エネルギーのリストを取得する。この時、消費エネルギーのリストだけでは、機器に関する情報が機器 ID のみでそれがどういった機器なのか解釈することができない。そこで次に Configuration API を用いて家中の機器情報を取得する。最後に機器 ID を用いて、機器情報と消費エネルギー情報をマッチングし、Google Chart API による可視化を行う。

## 6. 考察

### 6.1 Scallop4SC の現状と限界

5. 章で述べた消費電力可視化アプリケーション開発の結果、Web アプリケーション開発に関してある程度知識のある開発者ならば、Scallop4SCAPI を使用することで、このようなアプリケーションを少ない工数で作成することが可能だということがわかった。

また、今後、住宅ログが膨大なレコード数になったとしても、クラスタの数を流動的に増加させる、という対応がとれる。これにより、Scallop4SCAPI を通じて取得することが可能なデータは損失が生じる可能性が少ないといえる。

しかし、本稿で述べた Scallop4SCAPI では不足する部分もある。今回のように住宅ログを 1 つずつ参照し取得する API のみでは、膨大な数のレコードが関与する処理に対応できない。分散 KVS の特性から、膨大なレコード数の住宅ログを蓄積することは可能だと考えられる。しかし、膨大な数の住宅ログを API を用いて取得し、取得したすべての住宅ログを用いて合算などの処理をサービス実行毎に行うと、非常に効率が悪い。

### 6.2 目的に応じたビューの作成

6.1 章で述べたように API だけでは、膨大な数が生成され

る住宅ログに対して効率的な運用ができない。住宅ログの生データは、ライフログとしての欠損が少なく、広い範囲で正確な結果を算出可能である。しかし、膨大な計算量・通信量を必要とし、すべてのアプリケーションで用いることは非現実的である。そこで、HBase に蓄積された住宅ログの生データに対して、フィルタリングやマイニングを適用し、マイニングデータを適切な規模のデータベース（ビュー）に蓄積し直す、といったバッチ処理を Hadoop を用いて一定間隔で行うことが有効だと考えられる。ビューに蓄積されたデータは、汎用性は低いが、データサイズが抑えられアプリケーションとの連携に特化している。

## 7. おわりに

本稿では、スマートシティ上の住宅ログを収集・活用するために我々の研究グループが提案・構築している Scallop4SC 上で住宅ログを効率良く蓄積するための HBase 向けデータモデリング、JAX-RS を利用した汎用的な API の実装を行った。また、それらを用いて消費電力可視化アプリケーションを Web アプリケーションとして開発した。アプリケーション開発を評価し考察を行った結果、Scallop4SCAPI を利用したアプリケーション開発は可能だが、現状の Scallop4SCAPI のみでは、不足していることもあるとわかった。

今後は、目的別にビューを作成し、Hadoop でのバッチ処理を行うことで、多様なアプリケーションが効率良く扱える仕組みの実現を目指す。

謝辞 この研究の一部は、科学技術研究費（基盤研究 C 24500079, 基盤研究 B 23300009）、および、関西エネルギー・リサイクル科学研究振興財団の助成を受けて行われている。

## 文献

- [1] Robert 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] Arun Mahizhnan. Smart cities: The singapore case. *Cities*, Vol. 16, pp. 13–18, 1999.
- [3] 山本晋太郎, 瀬戸英晴, 榎本真佑, 中村匡秀. 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. 電子情報通信学会技術研究報告, 第 111 巻, pp. 207–212, March 2012.
- [4] D. Borthakur. The hadoop distributed file system: Architecture and design. [http://hadoop.apache.org/common/docs/r0.18.0/hdfs\\_design.pdf](http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf), 2007.
- [5] Ankur Khetrapal and Vinay Ganesh. Hbase and hypertable for large scale distributed storage systems. <http://www.uavindia.com/ankur/downloads/HypertableHBaseEval2.pdf>, 2006.
- [6] 山本晋太郎, 高橋昂平, 大柳章裕, 榎本真佑, 中村 匡秀. スマートシティにおける大規模住宅ログを活用したサービスの検討. 電子情報通信学会技術研究報告, 第 112 巻, pp. 019–024, June 2012.
- [7] 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.
- [8] Apache Software Foundation. Apache hbase reference guide. <http://hbase.apache.org/book/book.html>, 2012.
- [9] Jersey. <http://jersey.java.net/>.
- [10] MySQL Connector/J. <http://www.mysql.com/downloads/connector/j/>.