2012 IEEE 4th International Conference on Cloud Computing Technology and Science

Design and Implementation of Service API for Large-Scale House Log in Smart City Cloud

Kohei TAKAHASHI, Shintaro YAMAMOTO, Akihiro OKUSHI, Shinsuke MATSUMOTO, Masahide NAKAMURA
Graduate School of System Informatics, Kobe University, JAPAN
1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo 657-8501, Japan
Email: {koupe, shintaro, okushi}@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 such as energy saving and optimization of traffic are provided using wide variety of logs collected from various appliances and sensors. We are currently developing a smart city platform, called Scallop4SC, which supports collecting and processing the extremely large-scale log data. This system stores variety of logs on the HBase key-value store, and supports powerful data processing by Hadoop MapReduce. The goal of this paper is to propose service API of Scallop4SC. The API should be implemented as a Web service, which allows heterogeneous clients to access large-scale log (of energy, device and environment), as well as house configuration data of the smart city. In order to cover queries from a variety of applications, we design possible combinations of service type, data type, access type and queries, based on practical service scenarios. As a case study, we present concrete smart city services for energy saving and quality of life improvement. Then, we evaluate how the proposed API can implement these practical services.**

I. Introduction

Smart City[1] is a next-generation city planning that aims to achieve **sustainable society** with ICT technologies. In the smart city, a variety of data are collected from sensors and peoples across the city. Typical collected data includes traffic information, energy consumption of houses and buildings, usage of home appliances etc. These data are stored as **log**, analyzed by advanced data mining techniques, and used for value-added services for life easier and more convenient to the citizen. The most important concept of the smart city service is to realize an environment friendly city by optimizing transport traffic[2], saving electric energy and so on. Furthermore, the smart city provide other value-added services in terms of safety, amenities and utilities based on the collected terabyte to petabyte-scaled data. In particular, the value added services include trend analysis of TV program in the particular area, health care for elderly citizen and anticrime activities with neighborhoods.

We have been proposed a platform for these data-based smart city services named **Scallop4SC** (SCALable LOgging Platform for Smart City)[3]. The Scallop4SC is a general-purpose service platform to support storing and processing such large-scale log data in the smart city. The Scallop4SC has the following four features.

- P1: Collecting and storing various logs of the smart city.
- P2: Efficient processing of stored logs.
- P3: Unify management of smart city house configuration.
- P4: Providing general access APIs to the above P1, P2, and P3.

Using the P4 APIs, a smart city service developer can easily develop the services without caring about collecting, storing and processing of smart city logs and smart city house configuration. The **house configuration** means static information of the smart city structure such as a list of houses in the city, household compositions, a list of installed devices in a home and so on.

In our previous work[3], we developed the prototype system of Scallop4SC which uses a parallel and distributed processing platform (Hadoop[4]) and a distributed key-value database (HBase[5]) to achieve P1 (log management) and P2 (log processing). In the Scallop4SC, smart city logs are collected via the internet, then they stored in the distributed database provided by HBase. The HBase simply manages various types of logs by a pair of key and value. The stored logs are processed by multi-node cluster of Hadoop.

In the smart city, it is considered that the terabyte to petabyte-scaled logs are generated. By using this parallel and distributed processing platform, efficient data management and processing can be provided. Now, we are operating a relational database (MySQL) to manage house configuration on the Scallop4SC (P3).

The goal of this paper is to achieve the feature P4. So this paper designs and implements the general-purpose APIs which are to access to the house configuration of the smart city and the log data collected from individual Smart Houses (**house log**). First, we sort out the API attributes in terms of service type, accessed data category and access method type. Then we discuss how combines the sorted out attributes and what APIs are provided to Scallop4SC users. Next, we consider some concrete smart city services from view points of **services for energy saving** and **services to improve Quality of Life** (**QoL**). Finally, as a case study, we evaluate our desined Scallop4SC APIs from "how the concrete services can be realized by combining the desgined

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

Scallop4SC APIs".

II. Preliminaries

A. Smart City

Smart city is a next-generation city planning that aims to achieve sustainable society with ICT technologies. In the smart city, a variety of data is collected from sensors and people across the city. Typical data include traffic information, energy consumption of houses and buildings, usage of home appliances, etc. These data are stored as log, analyzed by advanced data processing techniques, and used for value-added services for the sustainable society. As an example of concrete smart city services is an energy saving service based on power consumption in the area, optimization of traffic flow by assessment of the traffic situation, and so on. At present, smart city is not a theoretical speculation, experiment has been started in various cities around the world, such as Singapore and Amsterdam[6], [7].

B. House Log

In the smart city, there is various and wide-ranging kinds of log data, such as traffic state, energy consumption, action history of the citizens themselves, and so on. In this paper, we focus on the log data about closely related to housing (i.e., **house log**). The house log falls into three categories based on its nature.

1) **Energy Log**: means a history of energy consumption in the home. It corresponds to electric consumption, water consumption, gas consumption, and others.
2) **Device Log**: means a history of operation and state of home appliances. The examples are a TV was changed program 2 by user A, states of a floor light were power:ON, brightness:middle at 12:00:00 Jul. 27, and so on.
3) **Environment Log**: means a history of environmental changes measured by physical sensors. Specifically, it corresponds to temperature, humidity, light intensity, the number of people, and so on.

III. Scallop4SC

A. Architecture of Scallop4SC

We are developing a platform to support collection and using large-scale and wide variety log data occurred from the smart city, called **Scallop4SC** (SCALable LOgging Platform for smart city). Figure 1 shows the overall architecture of the Scallop4SC. Components of figure 1 are house loggers, Scallop4SC, smart city services.

The house logger preiodically measures house logs and stores to a database provided by Scallop4SC via the internet. The Scallop4SC is composed of distributed KVS database system, distributed data processing system and traditional relational database system. Each component is implemented by some existing systems: HBase, Hadoop and MySQL. House logs are stored to the KVS database and processed by the distributed processing system. In contrast to the house log, smart city house configuration has hierarchic and complex structure. So the house configuration is stored and managed by the relational database system. Smart city services are provided to citizens based on the house log and the house configuration via Scallop4SC APIs.

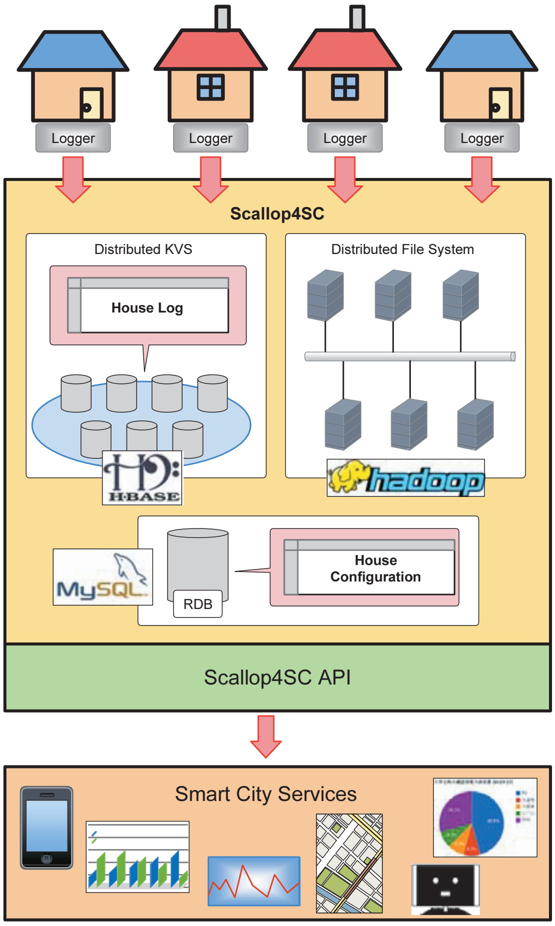

Figure 1. Architecture of Scallop4SC

B. Features of Scallop4SC

To cope with the growth of house log volume and its variety, the logs are stored and managed by the distributed KVS by HBase. Log data needs to be recorded periodically, and needs to be recorded every nook and cranny in the city. So, it is considered the amount of log volume will be explosively larger and larger. Additionally, the log data have a rich diversity (i.e., temperature log, appliance operation log, human activity log and so on). So the log cannot be presented in a uniformed format and we cannot decide its schema. By using a distributed KVS, it is not necessary to standardize data schema, and high scalability and high performance database can be achieved.

Moreover, in order to achieve efficient processing of stored log, Scallop4SC provides parallel distributed process-

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

ing system using Hadoop. Tallying or statistical processing of logs are required to utilize the terabyte-scaled data. Since Log data are highly independent of each other, these log data processing can be performed in parallel and distributed. Hadoop may provide high performance of house log processing. In addition, Hadoop can easily add its cluster nodes when declining its processing performance with expanding the smart city plan.

The Hadoop cluster consists of 12 Linux machines. One of them is a master node for management distributed processing centrally. Another 3 nodes are ZooKeeper[1] nodes. The cluster has two spec types of nodes (Pentium series and Athlon series) for confirming Hadoop's advantage of allowing hetero configuration of hardware.

C. Data Structure of Scallop4SC

Scallop4SC manages two types of datas: **house log** which collected from devices and sensors and **house configuration**. As mentioned in Section III-B, house logs are stored into distributed KVS on HBase. Essentialy formats of the key and the value are represented as follows.

```
Key: LogType.houseID.deviceID.TimeStamp
Value: data, unit, location, etc.
```

Smart city house configuration is a static information of the smart city structure such as a list of houses in the smart city, household compositions, specification of installed appliances, and so on. The house configuration has highly hierarchic and complex structure. For example, the relationships between city and house, house and room, room and appliance, and so on. In contrast to house log, the house configuration has lower the frequency of update and has smaller data size. For these reasons, the house configuration should managed not by KVS but by RDB.

Figure 2 shows data schema on RDB for the smart city house configuration illustrated by Entity-relationship Diagram (ERD)[8]. Square represents an entity, "+ — ∈" represents a parent-child relationship and "+ — · · ·" represents a reference relationship. A text located on the right side of the square means attributes of the entity and a text undelying the attributes means examples of the attributes. An underlined attribute represents a primary key and an parenthetic attribute represents a composite foreign key.

The smart city house configuration is classified into following three information.

1) **House Information**: holds information about houses in the smart city and rooms in the house. This is classified into three entities: City entity, House entity and Room entity. They have parent-child relationship. Child entities are automatically deleted when its parent entity is deleted.

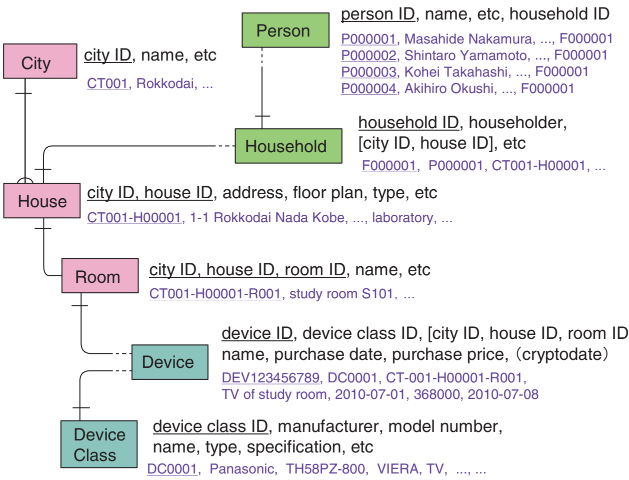
Person
person ID, name, etc, household ID
P000001, Masahide Nakamura, ..., F000001
P000002, Shintaro Yamamoto, ..., F000001
P000003, Kohei Takahashi, ..., F000001
P000004, Akihiro Okushi, ..., F000001
City
city ID, name, etc
CT001, Rokkodai, ...
Household
household ID, householder,
[city ID, house ID], etc
F000001, P000001, CT001-H00001, ...
House
city ID, house ID, address, floor plan, type, etc
CT001-H00001, 1-1 Rokkodai Nada Kobe, ..., laboratory, ...
Room
city ID, house ID, room ID, name, etc
CT001-H00001-R001, study room S101, ...
Device
device ID, device class ID, [city ID, house ID, room ID],
name, purchase date, purchase price, （cryptodate）
DEV123456789, DC0001, CT-001-H00001-R001,
TV of study room, 2010-07-01, 368000, 2010-07-08
Device Class
device class ID, manufacturer, model number,
name, type, specification, etc
DC0001, Panasonic, TH58PZ-800, VIERA, TV, ..., ...

Figure 2. Database schema of smart city house configuration

2) **Device Information**: has two entities: Device entity and Device Class entity. Device entity means concrete device which actually installed in a home. It holds information about device name, purchase date, and reference to its device class. Device Class entity includes abstract and common device information: model number, manufacturer and type.

3) **Personal Information**: has two entities: Household entity and Person entity. It holds information about citizens in the smart city. Person entity belongs to any one Household entity, and Household entity is linked to House entity.

Because the smart city configuration includes citizen's private personal information, there is a need to manage under public institutions such as local public government. Also there is necessary to extend attributes in each entity depending on the expansion of smart city plan future growth of the smart city.

D. Challenges

In our present work, we have proposed Scallop4SC and implement a prototype system with focus on house log management and processing. Hence, the prototype system does not provide general APIs to access the features of Scallop4SC. In other words, feature P4 (general API) described in Section I are remaining challenges of Scallop4SC. In this paper, we aim to achieve design and implementation of the APIs which is provided by the Scallop4SC.

IV. Smart City Service API

A. Key Idea

In order to provide various smart city services, we must design and implement general-purpose APIs covering queries for various Scallop4SC data requests. At first, we sort out attributes of access method of Scallop4SC data in

[1] ZooKeeper is a service to support decentralized ensemble with other nodes on the distributed system.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

terms of service type, accessed data category and access method type. Next, we discuss possible combinations of above attributes Finally, we design Scallop4SC APIs which used from various smart city services.

B. *Sorting Out the Attributes of Scallop4SC API*

1) Service Type: First, we classify the service type into below 4 groups from the view point of the kind of a service.

- Log: API to access past house log.
- Status: API to access current house log.
- Configuration: API to access the house configuration.
- Operation: API to control appliances of a Smart House.

"Log" and "Status" also access same data source (house log), but "Log" is used for accessing past house log data and "Status" is used for accessing current and real-time house log. Smart city house configuration can be accessed via "Configuration". The above three service types are accessing stored data. When a service developer want to operate Smart House devices, he/she can use "Operation".

2) Data Category: Next, we classify accessed data categories for each above service type.

- Log: Energy / Environment / Device
- Status: Energy / Environment / Device
- Configuration: House / Device / Person
- Operation: none

Data categories accessed by "Log" and "Status" correspond to house log categories described in Section III-C. "Energy", "Environment" and "Device" correspond to energy consumption, environmental attributes measured by physical sensors and state of home devices. Data categories of "Configuration" corresponds to the three types of house configuration described in Section III-C which are "House", "Device" and "Person". Since the "Operation" is not a data access API, it has no data category.

3) Access Type: Finally, we classified access types of APIs.

- Log: get / put
- Status: get
- Configuration: CRUD
- Operation: exec

Because the house log is just a persistent information log, they do not need to update and delete operation. So both the "Log" and "Status" which access the house log have "get" access type. House log is created via "put" access operation provided by "Log". Since "Configuration" access house configuration stored in RDB, it has CRUD (create, read update and delete) operation. "Operation" provides just a "exec" access type.

C. *Designing Scallop4SC API*

From the results of sorting out of Section IV-B, we discuss the combinations of the three attributes: service type, data category and access type.

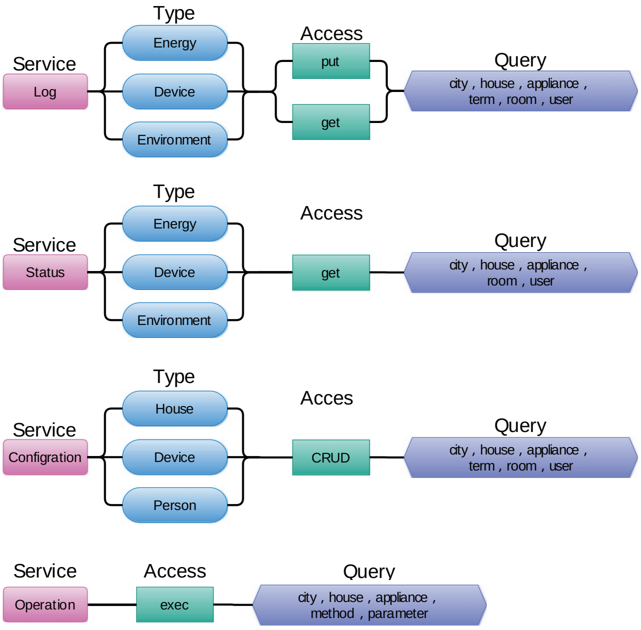

Figure 3. API Design of the Scallop4SC

Figure 3 shows four Scallop4SC APIs. The left hand boxes of the figure represent service types and correspond to API types. "Data"and "Access" represent accessed data category and data access type described in Section IV-B. "Query" means database access query that is used as API parameter. This query includes city, house, device, person, term and so on.

For example, to obtain a energy log of a TV installed in Takahashi's home on Jul. 20, a user invokes a following API.

```
Log.Energy.get (city="", house="takahashi",
   appliance="tv", term=2012/07/20, user=*)
```

These return values are represented in the form of key-value pair described in Section III-C. Example of the return value of the above API is follows.

```
Key: Energy.takahashi.tv.2012/07/20T00:00:00
Value: {data=319, unit=W, location=living, ..}
Key: Energy.takahashi.tv.2012/07/20T00:01:00
Value: {data=323, unit=W, location=living, ..}
...
```

D. *Implementation*

We implemented the Scallop4SC API as a Web service with JAX-RS using Jersey v.1.21[2]. As a connection for MySQL, we used official JDBC driver for MySQL (Connector/J v.5.1.21[3]). Also for accessing to HBase, we used official Java library of HBase.

[2] http://jersey.java.net/

[3] http://www.mysql.com/downloads/connector/j/

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

V. Case Study

A. Overview

In order to evaluate the feasibility of our developed APIs, we conduct a case study with concrete smart city services. First, we discuss some concrete smart city services from two viewpoints: energy saving and quality of life improvement. Next, we discuss how the concrete services are realized by combining the Scallop4SC APIs.

B. Smart City Services

One of the most typical purposes of smart city is to realize an environment friendly and sustainable city by optimizing energy usage as typified by Smart Grid[9]. For the most of energy types such as electricity, water and gas, decreasing the total consumption of whole of the city is simple and effective approach. In addition, reduction or shift of peak electric power demand is also an effective approach because the electricity power is difficult to store at long-term. Other approaches to saving energy include to promoting awareness of energy usage by visualizing energy consumption and automatic operation of home appliances.

However, the energy saving activities require a certain amount of patience to citizens. To expand the future smart city, we need to not only forward the energy saving activities but also keep quality of life (QoL) for individual citizen.

In this section, we discuss some concrete smart city services from two viewpoints of **energy saving** and **improving individual QoL**. Below are examples of services for energy saving.

- **ENG1: Visualizing Power Consumption in Smart City** The service visualizes the total power consumption in the whole smart city to promoting awareness of energy usage for individual citizen. Through the visualization, the service urges to refrain from using at the demand peak time for some appliances which require large electric power and can be shifted to outside of demand peak such as wash machine and dishwasher. In addition, it may be able to promote conscious of energy saving.
- **ENG2: Predicting Peak Time of Electric Power Demand** To achieve an effective peak shift and peak cut of electric power consumption, accurate prediction of peak demand is required. By accumulating the electric power consumption log for long-term, this service can provides high accurate prediction in various situations.
- **ENG3: Optimal Operation of Home Appliances** This service monitors real-time state of every appliances, and then automatically optimizes the running of appliances. By combining the device logs with environmental logs, it can detects useless running such as leaving an appliance running without residents and leaving an air-conditioner running with a window open. User receives these information, and then he/she can save useless power consumption by remote operation via internet.

Below are examples of services to improve QoL.

- **QOL1: Detecting Intruder and Giving Notice** This service detects intruder and gives a notice to not only residents but also neighborhood. Some existing security service can detects intruders but it gives a notice to only house residents. This service can give enlighten in local community for anticrime activity. From the aspect of public government, they may be able to take more effective crime countermeasures by analyzing house logs on around a crime-infested area.
- **QOL2: Preventing Fire Caused by Leaving Appliance Running** This service prevents fire disaster caused by leaving appliance running. For example, when a heating appliance leaves running even if a house resident go out, this service automatically turns off the appliance. Or when a heating appliance, that exhausts carbon monoxide, turns on for a long time, this service automatically turns on a vent fan.
- **QOL3: Giving Notice the Degree of Road Congestion** Optimization of traffic flow is a typical purpose after the energy saving. The general way is to detect traffic jam from sensors placed in roads and cars. Additionaly, by using our proposed Scallop4SC, citizens can receive jum-up information of public transport infrastructure.
- **QOL4: Monitoring Life of Elderly Adults** This service provides remote monitoring of livelihood of elderly living alones. By periodically giving a notices of a usage of given appliance to their family, the family can keep check health of the elderly.

C. Implementing the Smart City Services with proposed Scallop4SC APIs

We evaluate whether such services discussed in Section V-B are be able to be achieved with the proposed Scallop4SC APIs designed in Section IV.Table I represents that each service is realized by using the Scallop4SC APIs.

As an example, we pick up a service named “ENG1: Visualizing Power Consumption in Smart City”. At first, “Configuration.Device.get(houseId)” returns a list of installed appliances in the specified house from the house configuration database. Then, “Log.Energy.get(deviceId)” returns all log of energy consumption of each listed appliance. Finally, by processing and visualizing the obtained data, the ENG1 service is achieved. In a similar way, all the discussed smart city services can be achieved by proposed Scallop4SC APIs.

D. Advantages and Limitations

All the concrete smart city services can be achieved by Scallop4SC APIs. Since the APIs are designed in general access way to both house log and house configuration, we

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

Table I
SMART CITY SERVICES AND ITS IMPLEMENTATION WAY OF PROPOSED SCALLOP4SC APIS

Smart city service		Scallop4SC API			
Service ID		**Service**	**Type**	**Access**	**Query**
ENG1	Visualizing Power Consumption in Smart City	Configuration	Device	get	city
		Log	Energy	get	device, term
ENG2	Predicting Peak Time of Electric Power Demand	Log	Energy	get	city, term
ENG3	Optimal Operation of Home Appliances	Status	Device	get	house
		Status	Environment	get	house
		Operation	—	exec	device, method, param.
QOL1	Detecting Intruder and Giving Notice	Status	Environment	get	device
		Configuration	House	get	city
		Configuration	Device	get	house, device
		Operation	—	exec	device, method, param.
QOL2	Preventing Fire Caused by Leaving Appliance Running	Log	Environment	get	house, term
		Log	Device	get	house, term
		Operation	—	exec	device, method, param.
QOL3	Giving Notice the Degree of Road Congestion	Operation	—	exec	device, method, param.
QOL4	Monitoring Life of Elderly Adults	Log	Device	get	house, term
		Log	Environment	get	house, term

consider that the API can be respond to requests for implementation of future smart city service. However, because we only focus on the house log, Scallop4SC has no public area logs such as traffic. Therefore, QOL3 (traffic optimization) provides passive activity compared with traditional traffic optimization approach. In the future, we need to extend our Scallop4SC to handle public area logs.

Most Scallop4SC APIs designed in this paper have just for a data accessing method. To improve more feasibility and provide a general-purpose APIs, the Scallop4SC APIs are required to extend to provide data processing method such as statistic processing.

Finally, the performance evaluation of Scallop4SC APIs for handling terabyte and petabyte scale log data is also our future works.

VI. CONCLUSION

In this paper, we have designed and implemented the general-purpose API to access large-scale house log and house configuration managed on the Scallop4SC. As a case study, we have presented concrete smart city services and have evaluated how the proposed API can implement these practical services.

In our future work, we will conduct an experimental evaluation of Scallop4SC APIs. Also the personal privacy problem is one of the important future challenges.

ACKNOWLEDGMENTS

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No.24500079), Scientific Research (B) (No.23300009)], and Kansai Research Foundation for technology promotion.

REFERENCES

[1] A. Boulton, S. D. Brunn, and L. Devriendt, “18 cyberinfrastructures and ‘smart’ world cities: physical, human and soft infrastructures,” *International Handbook of Globalization and World Cities*, p. 198, 2011.

[2] E. Brockfeld, R. Barlovic, A. Schadschneider, and M. Schreckenberg, “Optimizing traffic lights in a cellular automaton model for city traffic,” *Phys. Rev. E*, vol. 64, p. 056132, Oct 2001.

[3] S. Yamamoto, H. Seto, S. Matsumoto, and M. Nakamura, “A study of services using large-scale house log in smart city,” in *Technical Report of IEICE*, vol. 111, no. 470, March 2012 (in Japanese), pp. 207–212.

[4] D. Borthakur, “The hadoop distributed file system: Architecture and design,” 2007, http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf.

[5] A. Khetrapal and V. Ganesh, “Hbase and hypertable for large scale distributed storage systems,” 2006, http://www.uavindia.com/ankur/downloads/HypertableHBaseEval2.pdf.

[6] 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.

[7] A. Mahizhnan, “Smart cities: The singapore case,” *Cities*, vol. 16, pp. 13–18, 1999.

[8] K. Watanabe, *Hanbai Kanri System de Manabu Modeling Koza*. Shoeisha, 2008 (in Japanese).

[9] S. Massoud Amin and B. Wollenberg, “Toward a smart grid: power delivery for the 21st century,” *Power and Energy Magazine, IEEE*, vol. 3, no. 5, pp. 34–41, 2005.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE