IMPLEMENTATION OF REAL-TIME TRANSFER SYSTEM OF SMART METERING WHICH SUPPORTS DEMAND-SUPPLY BALANCING

Naoki IWAMOTO
Chubu Electric Power Co., Inc. – Japan
Iwamoto.Naoki@chuden.co.jp

Katsuyuki KARIYA
Chubu Electric Power Co., Inc. – Japan
Kariya.Katsuyuki@chuden.co.jp

Takuya KAJIKAWA
Chubu Electric Power Co., Inc. - Japan
Kajikawa.Takuya@chuden.co.jp

Ryuji SHIBATA
IBM – Japan
sdragon@jp.ibm.com

Aki UCHIYAMA
IBM – Japan
auchiym@jp.ibm.com

Hideya SHIGEMATSU
Hitachi,Ltd – Japan
hideya.shigematsu.so@hitachi.com

ABSTRACT

Chubu Electric Power Company Inc. (CEPCO) implemented and operated a real-time, high-speed data transfer system of over 10 million smart meter's energy usage to contribute electricity supply-demand balancing. In this paper, we describe the system architecture, several technologies to improve system performance and its operational results.

INTRODUCTION

Since the nationwide tightness of electricity supply-demand balance was caused by the Great East Japan Earthquake on March 11, 2011, and the following Fukushima Daiichi nuclear disaster Japan government have promoted smart meter (SM) installations as a system infrastructure to control demand-side. In addition, in 2014 [1], all energy transmission and distribution companies got mandated to develop a system to provide 30 minutes usage values collected from all the SMs located in their business area to the Retailer and Transmission System Operator (TSO) within an hour (48 times a day), then to open the system to public in April 2016. Chubu electric power corporation (CEPCO), who has already installed over 5 million SMs and will achieve 10 million by 2022 as shown in Figure 1, has implemented and operated a real-time, high-speed data transfer system of SMs to contribute electricity supply-demand balancing with the large number of its own SMs in scope.

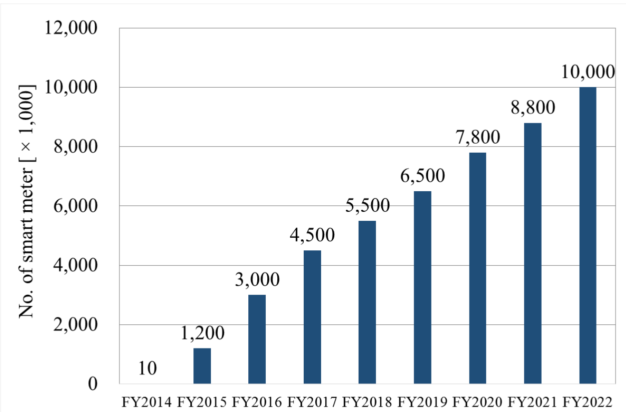

Figure 1: Number of SM installed in CEPCO area

Overview of CEPCO's SM system

SM

CEPCO's SM, which has common specification in all of Japanese Utilities, reads its meter by every 30 minutes and sends a reading not usage to Meter Data Management System (MDMS) via communication unit. Figure 2 shows a sample of CEPCO's SM.

Figure 2: A Sample Image of CEPCO's SM

Network

As shown in Figure 3, there are three communication media in CEPCO's SM network; Radio-Frequency (RF-) mesh, PLC and cellular network such an LTE. Due to reduction of network construction cost, almost all of SM uses wire-less network; RF-mesh is 94.3% and cellular is 5.5%. PLC, the communicates via power line, is only used for the area where wire-less network doesn't work well, for example inside a high tower building.

Using those three types of network, total 10 million SM's readings are sent to MDMS every 30 minutes. However, those networks are easily affected by network congestion and large physical object, for example large vehicles parked between each communication unit. Once network interruption happens, the SM reading couldn't be arrived to MDMS then it is recognized as a missing value. In order to complement the missing values, Head-End System (HE) in MDMS has a reacquiring SM reading process that communicates each SM with missing value. This reacquiring process is active within 48 hours after meter read.

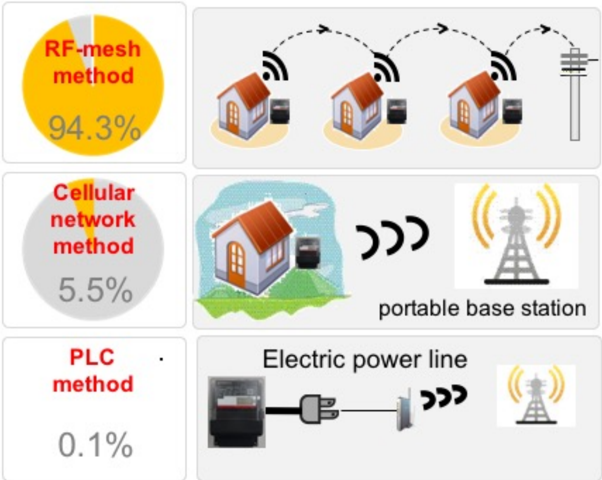

Figure 3: Overview of CEPCO's SM Network

MDMS
As mentioned in previous section, all SM readings are automatically stored in MDMS via SM network. If missing value is detected by monthly billing process, e.g. continuous network interruption occurred for more than 48 hours (Probability of this case is 0.1%), MDMS finally attempts to get a meter reading of the missing value by executing a get command to the SM. With this final completion process, almost all of SM readings are automatically stored in MDMS.

CEPCO's MDMS is composed of several sub systems as shown in Figure 4; Head-End System (HE), Meter Data Management (MDM) and Real-time Transfer System (RTS).

HE – communicates with each SM and collects SM readings by every 30 minutes. HE transfers each SM readings to RTS. To MDM, HE sends 6 hours packs, contains 12 readings, in batch process.
MDM – validates, estimates, finalizes and stores all of SM readings and energy usage.
RTS – receives SM readings, calculates and publishes energy usage. This is the system we mentioned in this paper.

Customer Information System (CIS) is also importantly related to CEPCO's MDMS.

CIS – manages and provides both customer and contract information of each SM. CIS calculates monthly energy usage of each customer from MDM data then send to billing system.

As following section, we describe the RTS system which receives a large amount of SM readings and publishes energy usages to Retailer and TSO in real-time.

BUSINESS REQUIREMENTS

The business requirements for RTS are following:

Functional Requirements

There are three main functional requirements.

Calculate energy usage for the unit of 30 minutes
Since all SMs in Japan send their meter readings by every 30 minutes not energy usages, the system has to convert meter readings to their energy usages.

Add Contract Information
Acquire contract information from CIS then add power supplier code to each energy usage.

Aggregate by Retailer
In order to disclose only Retailer contracting consumers' energy usage, aggregate the energy usage into a separate file by each Retailer. This separated file is also sent to TSO.

Non-Functional Requirements

Non-functional requirements for this system are following:

High-Speed Performance
In order to support demand-supply balancing for TSO, DSO has to provide every 30 minutes usage within 60 minutes. Considering transmission time on SM network and file uploading time to disclosed web server, actually only 18 minutes is allowed for this system to complete all processing.

High-Volume Data
Up to 10 million SM readings coming every 30 minutes.

Continuous Availability
In order to publish energy usage for the unit of 30 minutes within an hour, the system cannot be stopped.

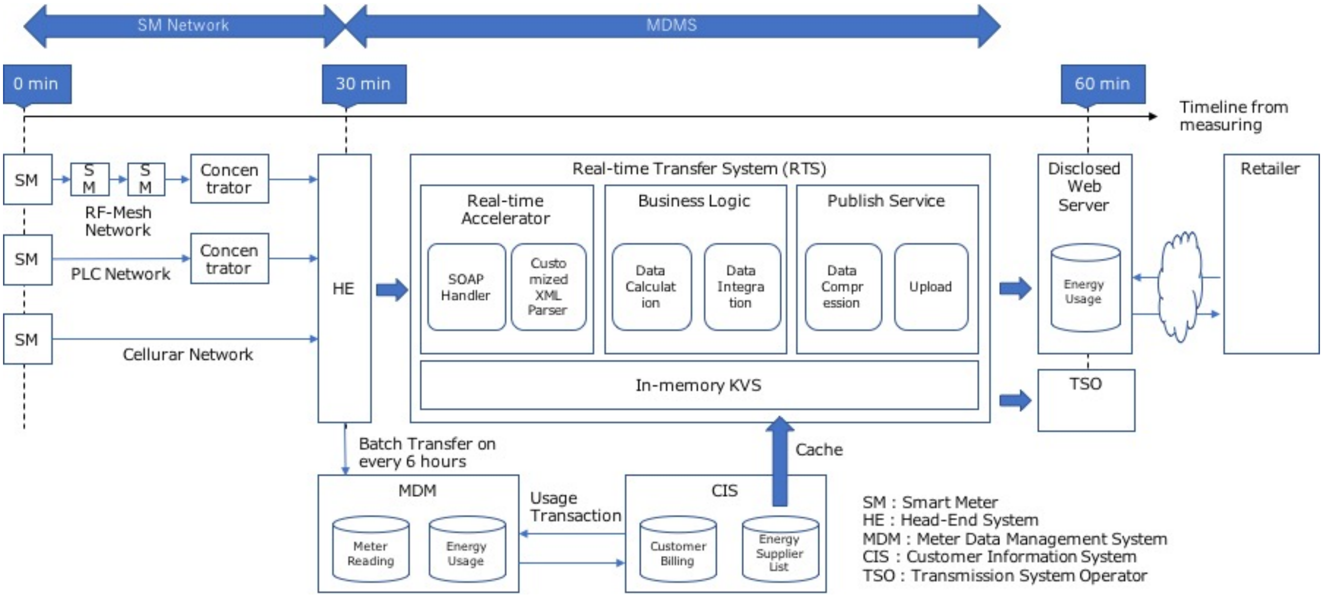

Figure 4: The System Architecture and SM Data Flow

SOLUTIONS

Without RTS, we assumed that it would be difficult to meet the requirement for the data amount and the time restriction (10 million SMs and within 60 minutes) with an approach using readings stored in MDM's database because the master data references needed on data calculation and integration might use much time. In order to satisfy the business requirements especially non-functional aspect, we implemented the RTS composed of an in-memory KVS, with about 1000 times more performance than on-disk DB, and of a stream computing architecture that pipelines a large and various data process while increasing the degree of parallelism. The RTS is powered by IBM Streams [2] composed of several component processes as shown in Figure 4.

Real-time Accelerator

In this component, the application receives SOAP messages from Head-End system via SOAP I/F, calculates each energy usage and stores with readings sent by smart meters (10 million) in 30 minutes intervals. Focusing on non-functional requirements, applied technologies are shown below.

SOAP Handler
Once the application receives a SOAP request, the SOAP message is temporally stored into RAM disk. This handler monitors RAM disk remaining amount and delays SOAP Responses in stages in case of unexpected vast SOAP requests. In other words, this handler performs flow controls of streaming processes.

Customized XML Parser
This parser uses Apache Xerces and skips readings with unnecessary XML tags based on pre-defined assumed XML structure which is customized for the schema conforming to Common Information Model (CIM) format defined in IEC61968-11 [3]. We also activate multiple parser processes in parallel to improve throughput.

Business Logic

Two of functional requirements, "Calculate energy usage for the unit of 30 minutes" and "Add Contract Information", are assigned to this component.

Data Calculation
One of functional requirements, "Calculate energy usage", is assigned here. Receiving tuple messages from Real-time Accelerator which contains current value of a SM reading, calculate energy usage for the unit of 30 minutes with following formula.

$$\text{energy usage} = (\text{current reading} - \text{previous reading}) \times \text{multiplier}$$

The key point is where to store and fetch previous reading. To achieve High-Speed Performance, we do not use disk but in-memory KVS provided by Real-time Accelerator. The key-value structure, corresponding to relational data-base schema, achieves fast access to previous reading of each SM.

Figure 5 describes the data store and fetch process on stream computing. The boxes named "Operator" are the application components of Business Logic and the other color objects are of Real-time Accelerator. The data store and fetch steps based on the diagram showing stream processing are following.

1. Once the SOAP Handler receives a SOAP Request from HE, the Customized XML Parser creates and sends a tuple message to the Operator #1.
2. The Operator #1 extracts both the meter ID and the time code of SM reading from the received tuple message to derive previous time code. Then, the Operator #1 also access the in-memory KVS to fetch the previous SM reading and the multiplier of the SM. Note that the multiplier of each SM has been loaded from Data-base (DB2) to in-memory KVS by a batch process. The details of the data load process is described in next "Data Integration" section.
3. The Operator #1 embeds the previous SM reading in the received tuple message and transfer it to next Operator #2.
4. The Operator #2 extracts the current and previous SM readings and the multiplier from the received tuple message, then calculates the energy usage by the formula denoted above.
5. The Operator #2 sends the energy usage with tuple message to next Operator #3.
6. The Operator #3 extracts both the energy usage and the current SM reading from the received tuple message, then access the in-memory KVS to store them onto memory. The stored data on memory is also stored into DB2 by background load process. Note that the current SM reading stored onto memory would be used as previous SM reading in the Operator #1 process when the next time code of reading is received.

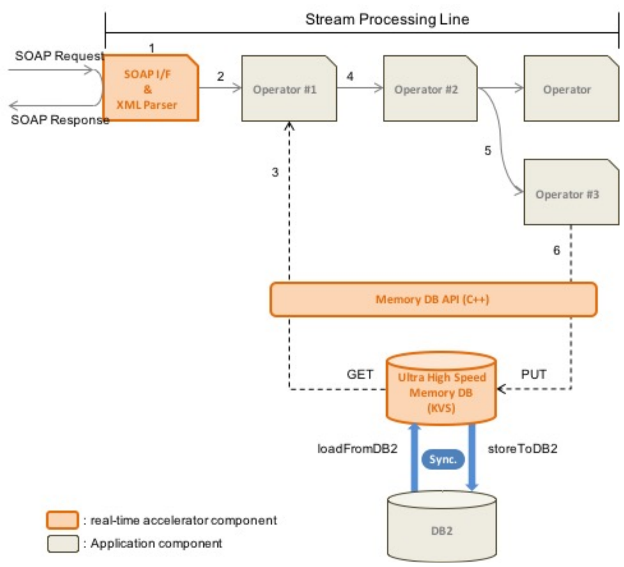

Figure 5: Overview of Real-time Accelerator and Stream Processing

Data Integration

"Add Contract Information" is assigned here. It is necessary to load contract information master data received from CIS, then extract and add operator code to each SM's energy usage. However, because of non-functional requirements especially Continuous Availability, we cannot stop streaming process even while master data loading. To meet this requirement, we prepared both active and inactive sides on the in-memory KVS. Master data is always loaded on the inactive side. After that, an application instantly swaps the active side with the inactive one, and stores data from the former active side (currently inactive) to DB2 as a background process.

Publish Service

The last of functional requirements, "Aggregate by Retailer", is assigned to this component. In this component, an application aggregates SM's energy usage into a file by each operator's code which derived in Business Logic component then compress it in Zip files and publish them to disclosed web server.

Data Compression

Considering upload of high-volume data covering 10 million energy usage in the next component, data compression is necessary here. In order to achieve High-Speed Performance requirement, instead of performing zip compression after all data outputted, the application uses incremental Zip that performs zip compression while outputting each record into a file.

CONSTRAINTS

The principal system constraints are following.

CIM Format

The XML in SOAP request from HE must be Common Information Model (CIM) format which defined by IEC61968-11 [3].

Deadline Time for Every 30-minutes Unit

SM readings arrived at the system after the predefined 'deadline time' for every 30-minutes unit would be considered as missing values. For example, in the case that the predefined deadline period is 20 minutes, a SM reading at 0:30 arrives after 0:50 would be considered as a missing in the real-time processing for energy usage of 0:30.

Published File is Not Sorted

To achieve the High-Speed Performance, the system doesn't sort the energy usages in the published file. Instead of sorting, the system separates the published file by each Retailer.

RESULTS

With this system, CEPCO achieves to provide all SM's energy usage for unit of 30 minutes within an hour. In actual, the results of FY2018 are following;

Data Volume – over 5 million records
Data Collection Rate – average 99.4 % within 30 minutes, average 99.9 % within 48 hours
Data Transfer Time – every energy usage, except missing value, published within 60 minutes after meter reading

The main purpose of this system is to publish energy usage calculated based on SM readings in real-time, however, the SM reading is sometimes missing due to several reasons of SM network condition such a radio wave congestion. To mitigate the situation, the system also receives energy usage of the lost period from MDM which has the reacquiring SM reading function. With daily batch process of data transfer from MDM, the Data Collection Rate of the system is finally improved to 99.9%.

Currently, the Retailer and the TSO utilize the energy usage of every 30 minutes which provided by this system for demand and supply balancing.

REFERENCES

[1] METI, 2014, "Results of examination concerning provision rules of information obtained from smart meter and future response", *Energy System Institutional Design working group*, vol.7, materials No.4.

[2] M. Ebbers, A.A. Gayed, V.B. Budhi, F. Dolot, V. Kamat, R. Picone, J. Trevelin, 2013, *Redbook: Addressing Data Volume, Velocity, and Variety with IBM InfoSphere Streams V3.0,* IBM, New York, United States, 15-56.

[3] IEC61968-11, *Application integration at electric utilities – System interfaces for distribution management – Part 11: Common information model (CIM) extensions for distribution,* IEC, 105-128.