Research Journal of Applied Sciences 10 (1): 12-16, 2015
ISSN: 1815-932X
© Medwell Journals, 2015

Fast Access to Large Timeseries Datasets in SCADA Systems

Anton Tyukov, Olga Khrzhanovskaya, Alexander Sokolov,
Maxim Shcherbakov and Valerij Kamaev
Volgograd State Technical University, Lenina av. 28, 400005 Volgograd, Russia

Abstract: This study presents a new method of working with timeseries data gathered from sensors. Researchers use natural time cycles of timeseries to create data packages of different type. Each data package is a small piece of timeseries data of fixed size and granularity, supported with statistical information and data quality certificate. Data quality certificate provides information about following aspects of data: syntactic, semantic, pragmatic and punctuation. This study contains a concept of data infrastructure and experimental results which proves significant increase of data extraction time and perform data quality certification of each data package. All experiments were performed on large time series data collected from electricity, water and gas meters in public buildings.

Key words: SCADA, machine learning, data quality, timeseries data, building energy management system, BEMS, big data

INTRODUCTION

Industry dramatically increased numbers of requirements for sensor driven intelligent decision support systems in various domains. Being virtual, they need to bring serious value to real economy. Therefore, engineers and scientists develop new approaches to visualize more data at the same time in better way, sophisticated algorithms to improve operational system control (intellectual support systems, control, supervisory control). Unfortunately, majority of analytical systems were designed for offline data mining and were not supposed to be used in real-time because natural limitations of SCADA system are defined by speed of performance and maximum data capacity.

Usage of real-time applications increase requirements to data quality and overall system timing. Limitations are defined to the following features:

- Users annoyed to wait long each time when data extracted and processed
- There is completely no information provided on quality of stored data (Tyukov *et al*., 2013)
- There is nearly impossible to evaluate applicability of certain algorithm to existing data

Therefore, researchers developed new concept to manipulate with timeseries data to significantly improve access time and simplify access of data to real-time machine learning algorithms.

The scientific contribution of the study is creating a new method working with timeseries which is different from existing methods by usage of natural time circles of timeseries data and data quality certificates.

BACKGROUND

There are several factors influencing on usage of right infrastructure such as:

- Selection of right type of data storage: SQL or NoSQL, OLAP or OLTP, etc.
- Designing right architecture of the database
- Optimization of data export such as: software for parallel export like PDW software, Oracle Data Pump (Anonymous, 2011)
- Good application design
- Improvement of performance based on external factors such as server performance, quality of network connection, etc.

But, even these ways can not give acceptable processing speed to perform data manipulation (data fusion, change of granularity, data imputation, data export, calculation of data quality) with large and extremely large data sets in real time (Khrzhanovskaya *et al*., 2014; Vale, 2010; Shcherbakov and Tyukov, 2014).

The method is based on splitting data export into two stages: data preprocessing and export of processed data for further data manipulations.

Corresponding Author: Anton Tyukov, Volgograd State Technical University, Lenina av. 28, 400005 Volgograd, Russia

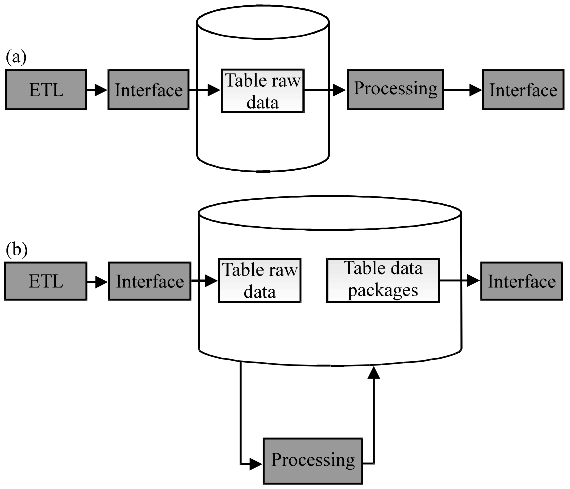


Fig. 1: a) Classical approach and b) Proposed approach

SOLUTION CONCEPT

In the most of the cases, data storage system consists on the following blocks:

- Data insertion module or complex event processing system
- Database
- Data processing module
- Data extraction application (Tyukov *et al*., 2012)

System with the following architecture stores much less data but data needs to be processed on demand which significantly increases latency time. Figure 1a represent the architecture of the system.

Researchers offers to preprocess data before usage by creating data of different resolution and creation of supportive data including data quality certificate. Figure 1b show the proposed architecture. The issue here is increasing time of preprocessing.

ETL SYSTEM

ETL system is divided into two pieces: data crawlers which gets data from different sources (such as virtual com ports or html parsers) and prepares it in universal format and insertion method to the database. This part is out of scope of the research therefore, we are not presenting its details.

Database module: The database contains the following tables. Table T_Meters is a database mapping of real meters. Each meter can measure different parameters (T_Measurement). System stores data for each parameter in raw format (T_DataRaw) and preprocessed (T_DataPackage). Structure of the part of the database is presented on Fig. 2.

Table 1: Examples of data package

Data package	Level	Interval	Granularity	Where is used
Raw	0	1 value	1 value	Used only for preprocessing
d_15	1	Day	15 min	Daily graphs, meter readings, data export
w_h	2	Week	1 h	Weekly graphs
m_d	3	Month	1 day	Monthly graphs and reports
y_m	4	Year	1 month	Yearly reports and graphs
y_y	5	Year	1 year	Yearly reports

Data packages can be of different types, depending on granularity of data and size of the packages. Preselected data packages are presented in Table 1.

Each data package contains data and additional information with data quality. Supportive information contains additional information about the package such as amount of means, medians, averages, amount of gaps, etc. Data certificate contains information on data quality on the following criterias: syntactic, semantic, pragmatic and punctuation. Data supportive information and data quality certificate is out of scope of the research, therefore, the description is skipped.

Each table contains data in JSON format for high-speed export also it is native format for the most javascript frameworks of data visualization such as higharts.js or d3.js.

Processing module: The goal of the processing module is to process raw data into data packages of different type (Table 1). There are data packages of high granularity and low level level (For example: d_15, w_h, Table 1) are created or updated as soon as raw data arrive, data packages of low granularity and high level (m_d, y_m, Table 1) are processed only once per day. Processing module uses data package of previous level to generate data package of the next level. Schema of package creation algorithm is presented on Fig. 3.

Database interface: Interface module provides communication methods with the database the way that knowledge data structure is not needed for external applications and provides maximum abstraction to database tables. Researchers consider requests to the data as the most important for end-user application, therefore, they interfaces for import and export of data to the database. Examples of data queries are presented on Fig. 4 and 5.

Performance feedback: Researchers included possibility to track queries stored in the database which help to answer the following questions:

- Which users are using the system the most?
- Which queries are the most demanding?

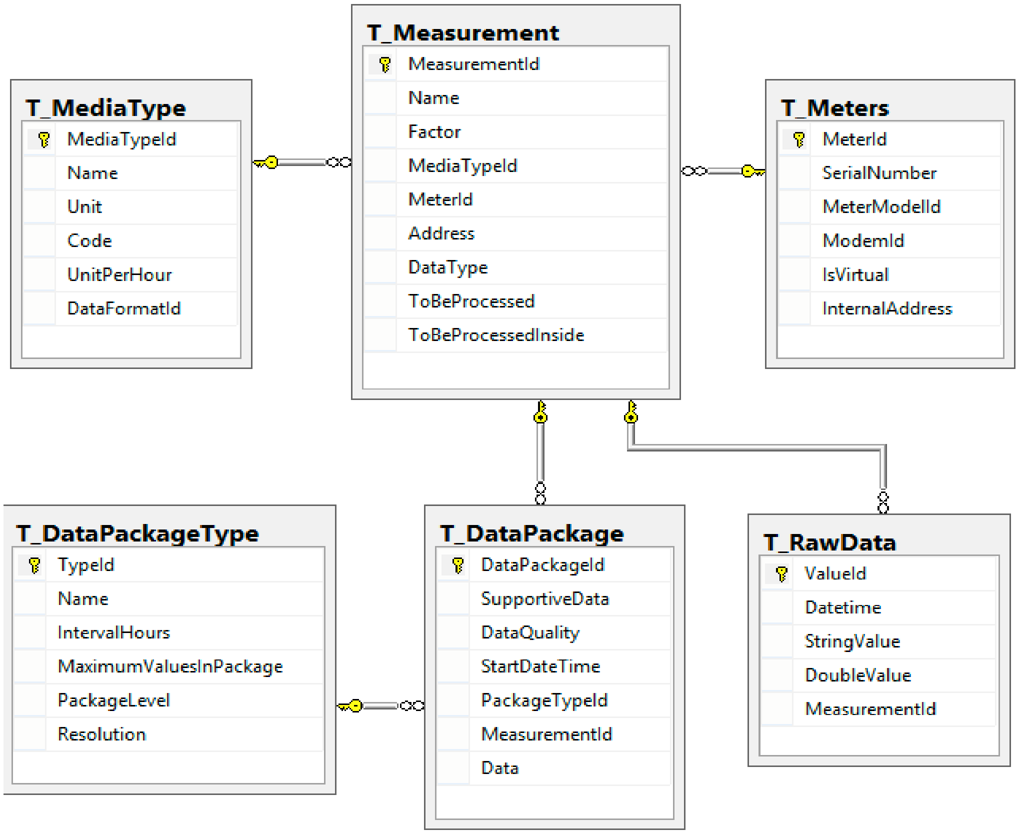


Fig. 2: Fragment of database architecture

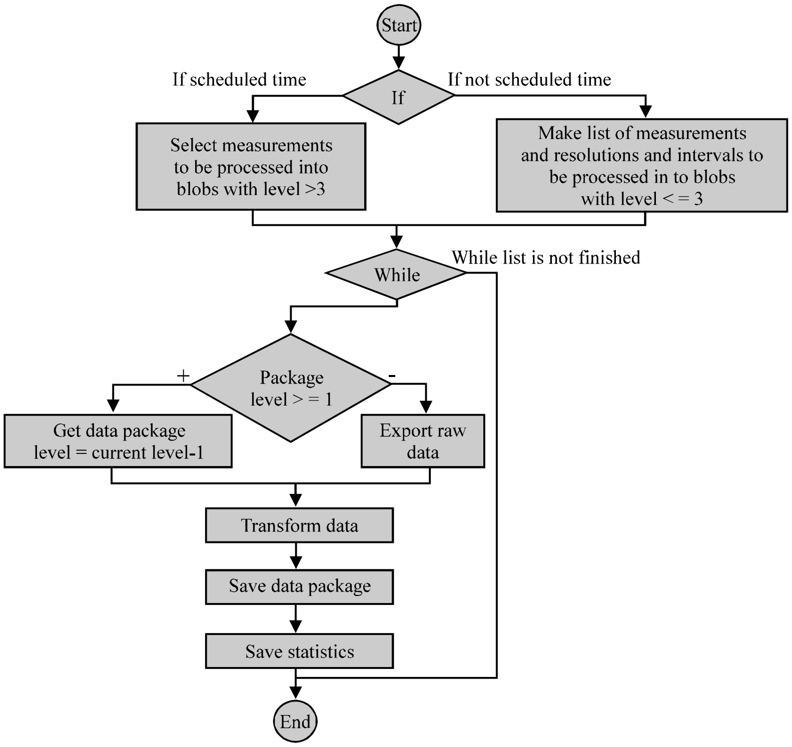
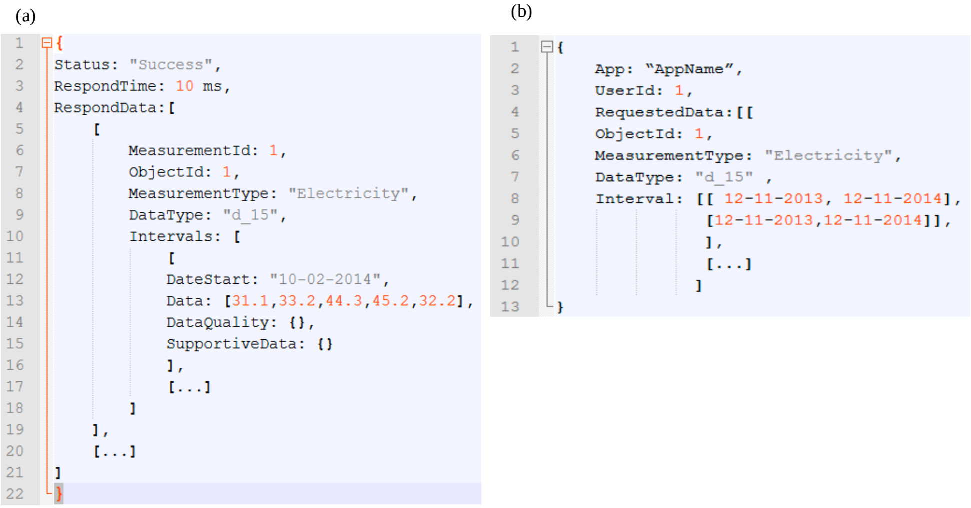

Start
If scheduled time
If
If not scheduled time
Select measurements to be processed into blobs with level >3
Make list of measurements and resolutions and intervals to be processed in to blobs with level < = 3
While
While list is not finished
+
Package level > = 1
-
Get data package level = current level-1
Export raw data
Transform data
Save data package
Save statistics
End

Fig. 3: Performance of processing algorithm



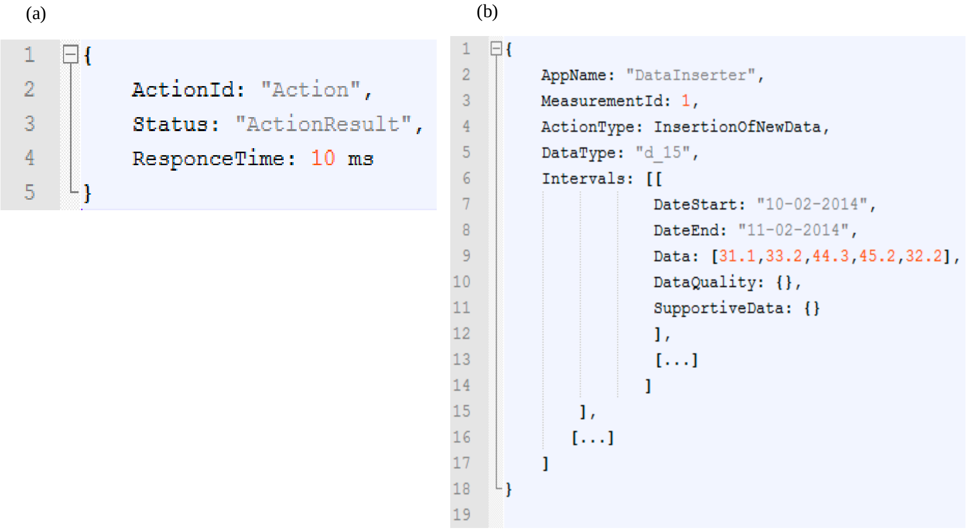
(a)

```
{
Status: "Success",
RespondTime: 10 ms,
RespondData:[
     [
          MeasurementId: 1,
          ObjectId: 1,
          MeasurementType: "Electricity",
          DataType: "d_15",
          Intervals: [
               [
               DateStart: "10-02-2014",
               Data: [31.1,33.2,44.3,45.2,32.2],
               DataQuality: {},
               SupportiveData: {}
               ],
               [...]
          ]
     ],
     [...]
]
}
```

(b)

```
{
     App: "AppName",
     UserId: 1,
     RequestedData:[[
     ObjectId: 1,
     MeasurementType: "Electricity",
     DataType: "d_15" ,
     Interval: [[ 12-11-2013, 12-11-2014],
                [12-11-2013,12-11-2014]],
                ],
                [...]
               ]
}
```

Fig. 4: a) Request of data and b) Requested data



(a)

```
{
     ActionId: "Action",
     Status: "ActionResult",
     ResponceTime: 10 ms
}
```

(b)

```
{
     AppName: "DataInserter",
     MeasurementId: 1,
     ActionType: InsertionOfNewData,
     DataType: "d_15",
     Intervals: [[
                    DateStart: "10-02-2014",
                    DateEnd: "11-02-2014",
                    Data: [31.1,33.2,44.3,45.2,32.2],
                    DataQuality: {},
                    SupportiveData: {}
                    ],
                    [...]
                   ]
          ],
         [...]
     ]
}
```

Fig. 5: a) Data insertion and b) Response on data insertion

- What is average time of data insertion, processings and export how does it change in time?
- What is the average wait time until the page will be loaded?

And others which significantly influences productivity of the system? Therefore, researchers created table to store logs containing information about data manipulation such as: query start time, query end time, objectId, period of extraction, action type performed. All data manipulation queries are logged into this table.

EXPERIMENTS

Experiments are aimed to prove solvency of developed structure for time-critical applications. In the experiment, we compare speeds of insertion, processing and export of data packages of different types (Table 1). In the experiment we used virtual machine placed on Intel Xeon Ei5 Server with 32 Gb of RAM and SSD drive.

For the experiments we used 20 time series of 3 years of data explaining electricity, gas and water consumption from public buildings located in Belgium (Table 2).

Original data has resolution of 15 min, sometimes contained gaps. Researchers measured processing time of conversion of raw data to data package of different resolution, amount of additional storage needed to store data of different resolution, time to export of data in different package and compare it with export of raw data. Results are presented on Fig. 6.

Table 2: Comparison of data packages of different types

Experiment	d 15 m	w h	m d	y m	y y
Preprocess time	28 min, 28 sec, 819 min sec	17 min, 37 sec, 627 min sec	17 min, 17 sec, 770 min sec	16 min, 48 sec, 751 min sec	17 min, 11 sec, 871 min sec
Extra storage needed (Mb)	28.813	5.398	0.703	0.078	0.063
Export time of preprocessed data	05 sec, 679 min sec	01 sec, 906 min sec	00 sec, 750 min sec	00 sec, 531 min sec	00 sec, 406 min sec
Export time raw	17 min, 06 sec, 142 min sec	16 min, 06 sec, 082 min sec	07 min, 36 sec, 818 min sec	07 min, 01sec, 203 min sec	02 min, 47 sec, 821 min sec

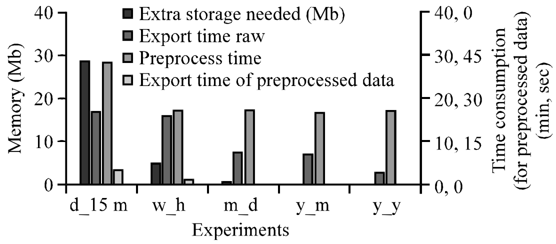


Fig. 6: Comparison of the results

CONCLUSION

The approach significantly improves access time to energy data in different resolution which positively reflected on speed of data visualization and possibility to use it in real-time algorithms.

ACKNOWLEDGEMENT

Researchers would like to thank Adriaan Brebels for providing data for experiments, scholarship of Russian President SP-3164.2013.1, financial support RFBR #14-07-00945.

REFERENCES

Anonymous, 2011. High speed data export from an HP enterprise data warehouse appliance to an SMP Microsoft SQL server: Step-by-step guide using remote table copy (aka. Parallel data export). Technical White Paper, November 2011. http://www 8.hp.com/h20195/v2/GetDocument.aspx?docname=4 AA3-8382ENW.

Khrzhanovskaya, O., A. Tyukov, M. Shcherbakov, A. Brebels and S. Shevchenko, 2014. Data preparation for research on energy efficiency of buildings. Proceedings of the International Scientific-Practical Conference on Innovative Information Technologies, April 21-25, 2014, Praque, pp: 131-135.

Shcherbakov, M. and A. Tyukov, 2014. Mapreduce-based algorithms for outlier detection in energy data. Proceedings of the 8th Multi Conference on Computer Science and Information Systems, June 2014, Greece.

Tyukov, A., A. Brebels, M. Shcherbakov and V. Kamaev, 2012. A concept of web-based energy data quality assurance and control system. Proceedings of the 14th International Conference on Information Integration and Web-based Applications and Services, December 3-5, 2012, Bali, Indonesia, pp: 267-271.

Tyukov, A., A. Ushakov, M. Shcherbakov, A. Brebels and V. Kamaev, 2013. Digital signage based building energy management system: Solution concept. World Applied Sci. J. Inform. Technol. Mod. Ind. Educ. Soc., 24: 183-190.

Vale, S., 2010. Statistical data quality in the UNECE 2010 version 2010. http://unstats.un.org/unsd/dnss/docs-nqaf/UNECE-Quality Improvement Programme 2010.pdf.