International Journal of Advanced Culture Technology Vol.11 No.2 315-322 (2023)
DOI https://doi.org/10.17703/IJACT.2023.11.2.315

IJACT 23-6-36

A Simulation Framework for Wireless Compressed Data Broadcast

[1]Seokjin Im

[1]Prof., Dept. of Computer Engineering, Sungkyul Univ., Korea
imseokjin@gmail.com

Abstract

Intelligent IoT environments that accommodate a very large number of clients require technologies that provide secure information service regardless of the number of clients. Wireless data broadcast is an information service technique that ensures scalability to deliver data to all clients simultaneously regardless of the number of clients. In wireless data broadcasting, clients access the wireless channel linearly to explore the data, so the access time of clients is greatly affected by the broadcast cycle. Data compression-based data broadcasting can reduce the broadcast cycle and thus reduce client access time. Therefore, a simulation framework that can evaluate the performance of data broadcasting by applying different data compression algorithms is essential and important. In this paper, we propose a simulation framework to evaluate the performance of data broadcasting that can adopt data compression. We design the framework that enables to apply different data compression algorithms according to the data characteristics. In addition to evaluating the performance according to the data, the proposed framework can also evaluate the performance according to the data scheduling technique and the kind of queries the client wants to process. We implement the proposed framework and evaluate the performance of data broadcasting using the framework applying data compression algorithms to demonstrate the performances of data compression broadcasting.

Keywords: *Data Compression, Simulation Framework, Scalable Information Service*

1. INTRODUCTION

Intelligent IoT (Internet of Things), the convergence of artificial intelligence and IoT, one of the key technologies of the 4th industrial revolution, is creating new technologies and services. Intelligent IoT services are expanding beyond traditional mobile devices to include energy utilities, healthcare, and future personal mobility, such as autonomous vehicles and PAVs (Personal Air Vehicles) using drones. According to Cisco analysis, by 2030, there will be an estimated 500 billion devices connected to the internet, resulting in a dramatic increase in the number of clients, making data security a critical challenge for autonomous vehicles and PAVs where human lives are at stake [1-3]. To meet the needs of this computing environment, there is a great need for techniques that can provide secure information services regardless of the increasing number of large clients.

Wireless data broadcast is an information service technique that ensures scalability to deliver data to all clients simultaneously regardless of the number of clients [4]. A broadcast server broadcasts data to a wireless channel and clients access the wireless channel to browse and download the data they want. Thus,

Manuscript received: April 7, 2023 / revised: April 25, 2023 / accepted: May 10, 2023
Corresponding Author: imseokjin@gmail.com
Tel:+82-31-467-8168, Fax: +82-31-467-8053
Professor, Dept. of Computer Engineering Sungkyul Univ., Korea

the wireless data broadcast enables to provide information services to an arbitrary number of clients because there is no limit to the number of the clients connecting to the wireless channel [5].

In wireless data broadcasting, a client's data access time is affected by the broadcast period because clients access data linearly on the channel. Therefore, reducing the broadcast period is an effective way to reduce the data access time of clients. A simulation framework that can evaluate the performance of data broadcasting based on effective data compression is very important and essential. It is also essential to be able to apply different data compression algorithms based on the nature of the data and evaluate their performance.

In this paper, we propose a simulation framework to evaluate the performance of data broadcasting based on data compression. The proposed framework is open to the data type and the data compression algorithm and data scheduling schemes. Therefore, the proposed framework allows to adopt various data compression algorithms according to the characteristics of data to be broadcast. This means that the proposed framework enables to evaluate the performances according to the algorithm for data compression and find the effective data compression algorithms to be suited to the characteristics of the data. The proposed framework is also capable of measuring various performance metrics, so it is possible to evaluate performance not only according to data, but also according to data scheduling and the kind of queries the client wants to process.

We organize this paper as follows. Section 2 reviews the data compression as related works. In Section 3, we design the proposed simulation framework for wireless data compression broadcasting. Section 4 implements the proposed framework and evaluates the performance of the data compression broadcast system by applying data compression algorithms to the framework and Section 5 concludes the paper.

2. RELATED WORKS

Data compression is a way to represent data with fewer bits, which can save disk space and speed up data transfer. Data compression can be divided into two types. The first is lossy compression, where some information is lost while compressing the data. The compression of an image in JPEG format, for example, causes the loss of some information in the image. The second is lossless compression, where information is retained while compressing data. ZIP files, for example, use lossless compression to reduce file size.

LZ compression algorithm is one of the most common data compression algorithms, which compresses data by finding redundant parts of the data and replacing those redundant parts [6]. Gzip compresses data by previewing some of the data before compressing it, looking for duplicates and replacing them [7, 8]. It uses a dictionary to do this, and the more redundant the data, the higher the compression ratio. Gzip is used to optimize data transfer on the internet. For example, when a web server compresses HTML, CSS, JavaScript files, etc. with Gzip before sending them to a client, it saves transmission time and bandwidth.

Thus, data compression offers great benefits for data transmission and storage. In the wireless data broadcast where the broadcasting server disseminates the data on the wireless channel and the clients access the data, the data compression enables to reduce the access time for the clients by reducing the length of the broadcast cycle. For example, the broadcast server disseminates the data compressed with Gzip or LZ algorithm to the wireless channel. This causes that the broadcasting period is shorten by the data compression rate. Therefore, a system capable of evaluating how much a data compression broadcasting reduces clients' access time is very important.

3. SIMULATION FRAMEWORK FOR COMPRESSED DATA BROADCASTING

3.1 Design of Broadcasting Server using Data Compression

We design the server for a compressed data broadcasting that has the structure as shown in Figure 1.

- Broadcast Controller: It handles the entire process for broadcasting in the server. It associates with the index provider and the channel allocator and the transmitter, receiving the configuration of the compressed data and index and transferring it to the transmitter for broadcasting.
- Channel Allocator: It organizes the wireless channel by allocating alternately the compressed data from the data compressor and the index from the index provider.
- Index Provider: It makes the index information that holds the broadcasting times of the data items and the identifiers of them. The provider can adopt various indexing schemes according to the type and features of data items.
- Data Compressor: It compresses the data items to be broadcast that are transferred from the database via the channel allocator. It can adopt a variety of the algorithms for data compression and contributes to reduce the waiting times of the clients for the data items to be queried.

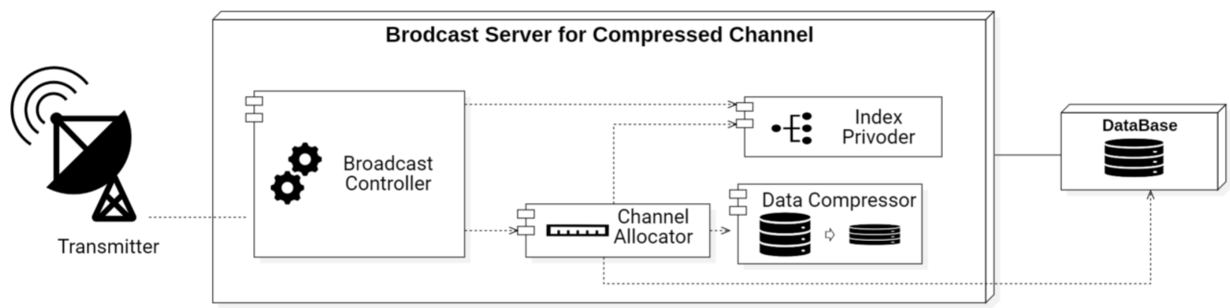


Figure 1. The server using compressed data broadcasting

The server proceeds the entire procedure for broadcasting the compress data on the wireless channel as shown in Figure 2. The figure shows the entire sequences of broadcasting the compressed data items.

1: Broadcast controller initiates the process by requesting for the configuration of the channel that is organize by the compressed data items and the index to be broadcast.
2: Channel allocator calls for generating the index of the data items to be broadcast to the index provider that is one of the elements of the channel.
3, 4, 5, 6: Index provider calls for the information on the data items and the database responses to the request. The index provider generates the index about the data items that carries the broadcasting times of the data items and returns the index to the channel allocator.
7, 8: The channel allocator calls for the data items to the database and the database returns them to the channel allocator.
9, 10: The channel allocator calls for the compression of the data items to the data compressor. The data compressor compresses them by the selected algorithm of data compression.
11: The channel allocator configures the wireless channel with the index from the index provider and compressed data items from the data compressor. The allocator arranges the compressed items and the index according to the selected allocation schemes, e.g., (1, m) indexing scheme or distributed indexing scheme.
12. The channel allocator returns the organized content for the wireless channel to the broadcast controller as the response of the request from the controller.
13, 14. The broadcast controller calls for broadcasting the organized content for the wireless channel

to the transmitter. Then, the transmitter broadcasts the organized channel from the controller.

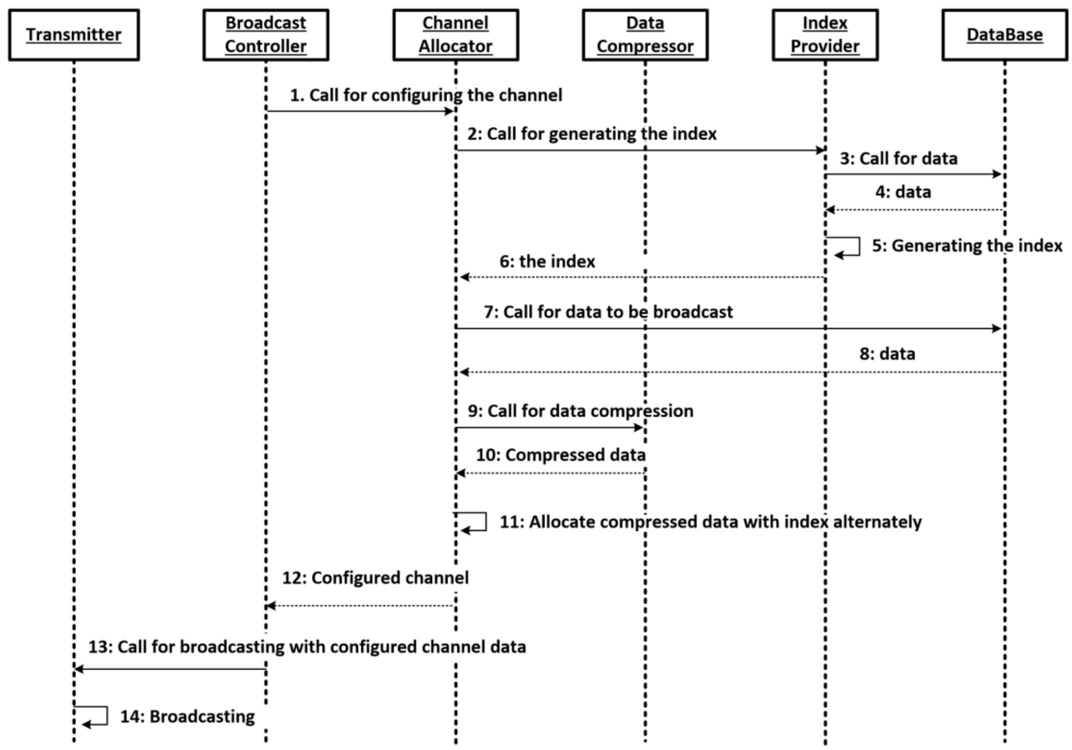


Figure 2. The sequence diagram of the server for data compressed broadcasting

3.2 Design of the Client and Data Search using Data Decompression

The client of the designed framework for compressed data broadcasting is structured with the components as shown in Figure 3, i.e., channel tuner, data searcher, index analyzer, and data decompressor.

- Channel Tuner: It listens to the wireless channel and downloads the compressed data and the index and returns them to the data searcher.
- Data Searcher: This component take charge of proceeding the procedure for searching data items that the user requests for processing queries. It associates with the index analyzer, data decompressor, and channel tuner.
- Index Analyzer: It filters out the queried data items using the index transferred from the data searcher and extracts times when the queried data items are appeared on the wireless channel.
- Data decompressed: It decompresses the compressed data items transferred from the data searcher and returns the data items to be restored to their original form.

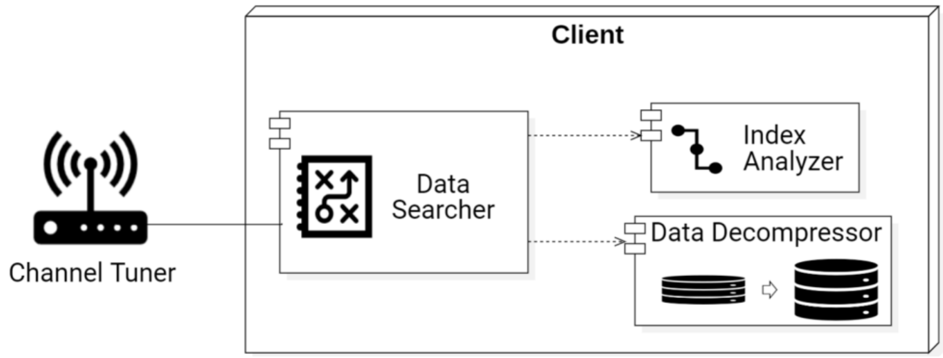


Figure 3. The client for compressed data broadcasting

The client proceeds the procedure to search for the data items queried by the user as shown in Figure 4. The figure shows the sequences for the client to process to filter out the queried data items and to download them using the components within it.

1: The user initiates the procedure of searching data items by issuing various type of query, e.g., window query or NN (Nearest Neighbor) query.
2: Data searcher calls for broadcasting times for the queried items to the index analyzer.
3, 4, 5: Index analyzer requests for downloading and returning the index to the channel tuner. The tuner accesses the wireless channel and downloads the index from the channel and returns it to the index analyzer.
6, 7: Index analyzer filters out the queried items using the index and extracts the broadcasting times of them. The analyzer returns the extracted broadcasting time to the data searcher.
8: Data searcher calls for downloading the filtered data items by transferring the broadcasting time to the channel tuner.
9, 10: Channel tuner downloads the queried data items selectively at the broadcasting times from the data searcher. Then, the tuner returns the downloaded items to the data searcher.
11, 12: Data searcher calls for decompressing the downloaded data items to the data decompressor. The decompressor decompresses the items according to the algorithm of the compression.
13: Data searcher displays the decompressed data items to the user.

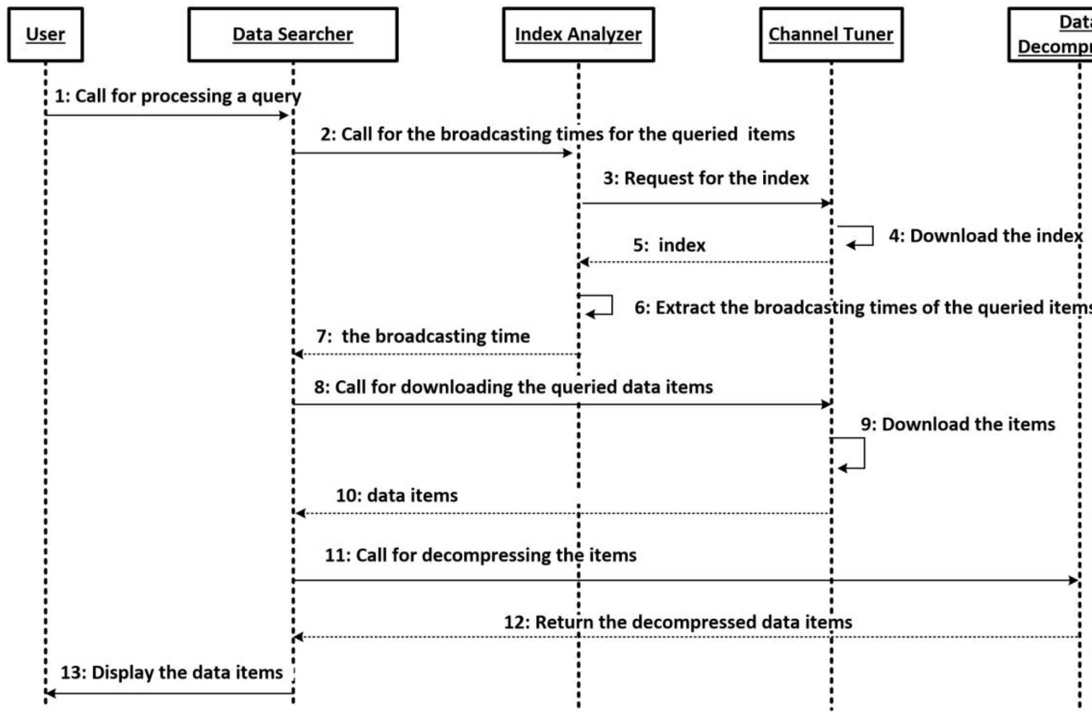


Figure 4. The sequence diagram of the client for the compressed data broadcasting

4. IMPLEMENTATION OF THE FRAMEWORK AND EXPERIMENTS

We implement the proposed framework for the compressed data broadcasting to reduce the access time of the clients. The proposed framework consists of the broadcast server, the client, and the wireless channel that are implemented with a discrete time scheduler according to the sequences as shown in Figure 2 and Figure 4. Figure 5 is the UI of the implemented framework. The button DATA is for selecting and loading the data set for simulation of the compressed data broadcasting. The COMPRESSION button allows to select one of compression schemes and SETTING button is for initiating the condition of the simulations. The RUN button triggers the simulation with the configured conditions.

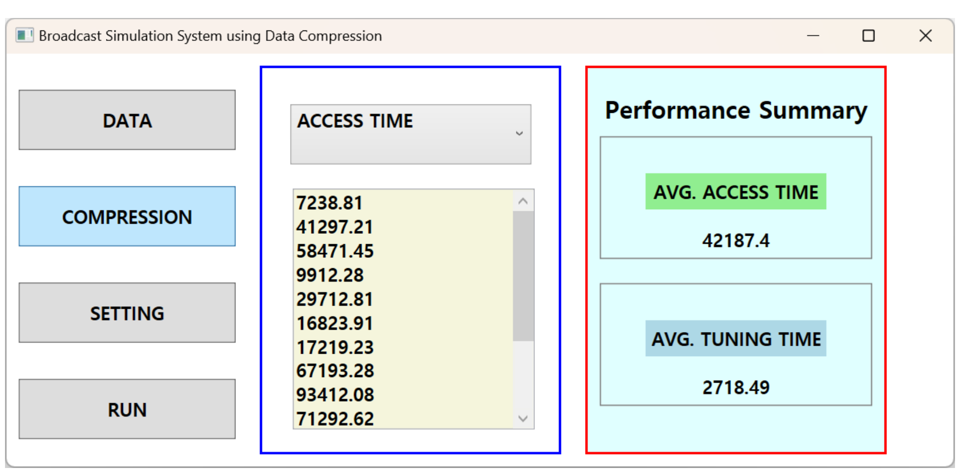


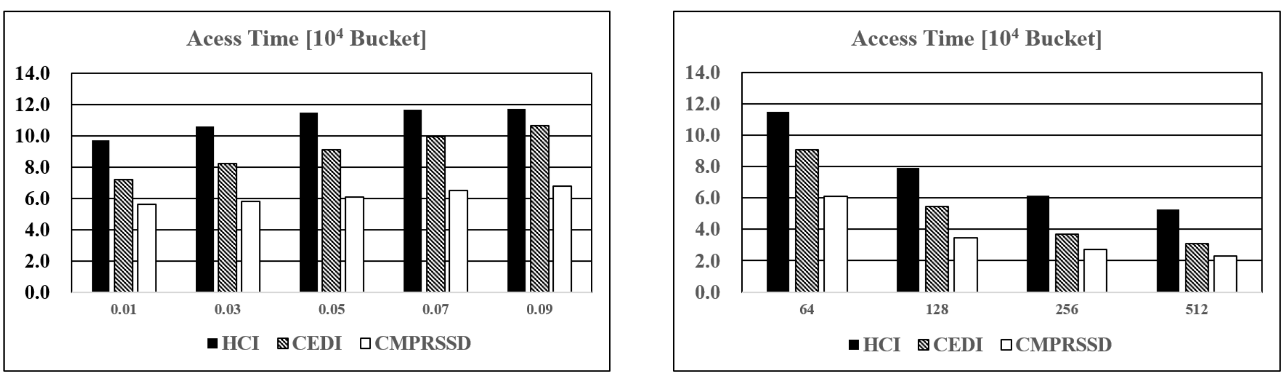
Figure 5. The UI of the implemented simulation framework

For the simulation, we use 6719 America historic sites data that is provided from US National Park Service in [9]. Using the implemented simulation framework, we compare the compressed data broadcasting and the normal broadcasting without compression. For the compressed data broadcasting we adopt Gzip compression algorithm for data compression. For the normal data broadcasting, we adopt two index schemes, HCI and CEDI [10, 11]. Through the simulation with the framework, we evaluate the access time of the compressed data broadcast and normal data broadcasting in the two indexing schemes.

The access time according to the query size shows in Figure 6(a). In Figure 6(a), CMPRSSD means the broadcasting with the data compression and HCI and CEDI mean the normal data broadcasting adopting the indexing schemes, respectively. The access time of the compressed data broadcasting is shorter than the two normal data broadcasting adopting HCI and CEDI, respectively. This results from the reduced length of the broadcast cycle by the data compression than the two normal broadcasting.

Figure 6(b) depicts the access time according to the size of buckets in the broadcasting. In the variety of the bucket size, the compressed broadcasting allows the clients to access data items than the two normal broadcasting. This also results from the reduced length of the broadcast cycle by the data compression.

The proposed simulation framework enables to evaluate the performances of the broadcasting with the data compression. The framework is possible to adopt various data compression schemes to compare how the compressed data broadcasting enhances the performance of the broadcasting system.



(a) Access time by the query size **(b) Access time by the bucket size**

Figure 6. Comparison of the access time

5. CONCLUSION

In this paper, we have designed the simulation framework for wireless data broadcasting based on the data compression and have implemented the framework and the user interface for handling the framework. The broadcasting server implemented in the framework consists of the channel allocator, the data compressor, and the index provider. The client implemented in the framework consists of the index analyzer and data decompressor. We showed the interaction between the broadcasting server and the clients with the sequence diagrams. The user interface configured in the framework enables to load a dataset to be broadcast and select one of the algorithms to be adopted in broadcasting. Also, through the interface the user can set the simulation condition and trigger the broadcasting simulation with the condition. To prove the effectiveness of the implemented simulation framework, we have evaluated the simulation framework by comparing the compressed data broadcasting adopting Gzip as the compression algorithm with the normal broadcasting without compression. For the normal data broadcasting, we have adopted two index schemes, HCI and CEDI. Through the evaluation we have compared the access time by the data compression broadcasting with that by the normal broadcasting applied the two indexing schemes. As the demonstration, the simulation framework enables to evaluate the impact of data compression on wireless data broadcasting as well as apply a variety of algorithms for data compression.

REFERENCES

[1] S.K. Kim and S.H. Park, "A Study on Intelligent Combat Robot Systems for Future Warfare," *International Journal of Advanced Culture Technology (IJACT)*, Vol. 11, No. 1, pp 165-179, March 2023. DOI: https://doi.org/10.17703/IJACT.2023.11.1.165

[2] G.S. Lee and S.H. Lee, "Study on Real-time Detection Using Odor Data Based on Mixed Neural Network of CNN and LSTM," *International Journal of Advanced Culture Technology (IJACT)*, Vol. 11, No. 1, pp 325-331, March 2023. DOI: https://doi.org/10.17703/IJACT.2023.11.1.325

[3] K. Seo, K. Kim, J. Kim, S. Cho, and S. Park "A Case Study on the Threat of Small Drone and the Development of Counter-Drone System," *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 9, No. 2, pp 327-332, March 2023. DOI: https://doi.org/10.17703/JCCT.2023.9.2.327

[4] I. Imielinski, S. Viswanathan, and B.R. Bardrinath, "Data on Air: Organization and access,", *IEEE Trans. TKDE*, Vol. 9, No. 3, pp. 353-372, 1997. DOI: https://doi.org/10.1109/69.599926

[5] S. Im, M. Song, S. Kang, J. Kim, C. Hwang, and S. Lee, "Energy Conserving Multiple Data Access in Wireless Data Broadcast Environments," *IEICE Trans. Communication*, Vol. E90-B, No. 9, pp 2629-2633, September 2004. DOI: https://doi.org/10.1093/ietcom/e90-b.9.2629

[6] P. C. Shields, "Performance of LZ algorithms on individual sequences," IEEE Trans. On Information Theory, pp. 1283-1288, Vol. 45, No. 4, 1999. DOI: https://doi.org/10.1109/18.761286

[7] O. Plugariu, L. Petrica, R. Pirea, and R. Hobincu, "Hadoop ZedBoard cluster with GZIP compression FPGA acceleration," Proc. of ICECAI 2019. DOI: https://doi.org/10.1109/ECAI46879.2019.9042006

[8] H. Luo, Y. Cai, Q. Luo, and R. Mao, "FPGA-Based Parallel Multi-Core Gzip Compressor I HDFS," Proc. of PDCAT 2019. DOI: https://doi.org/10.1109/PDCAT46702.2019.00017

[9] Real Dataset, available at https://www.nps.gov/subjects/nationalregister/data-downloads.htm

[10] B. Zheng, W.C. Lee, and D.L. Lee, "Spatial Queries in Wireless Broadcast Systems", *Wireless Network*, Vol. 10, No. 6, pp. 723-736, December 2004. DOI: https://doi.org/10.1023/B:WINE.0000044031.03597.97

[11] S. Im and H. Hwang, "A Two-Tier Spatial Index for Non-flat Spatial Data Broadcasting on Air," *IEICE Trans. Communication*, Vol. E97-B, No. 12, pp 2809-2818, December 2014. DOI: https://doi.org/10.1587/transcom.E97.B.2809