International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IOT Data Compression in Cloud Computing

Arti Jagdish Semwal[1], Dipali Ananda Toraskar[2], Rincy Joy Moyalan[3], Saloni Shankar Atkari[4]

[1,2,3,4] Student, Department of MCA, ASM IMCOST, Mumbai, India.
[1]artisemwal30@gmail.com, [2]dipalitoraskar09@gmail.com, [3]rincjm238@gmail.com, [4]saloniatkari@gmail.com

ABSTRACT:

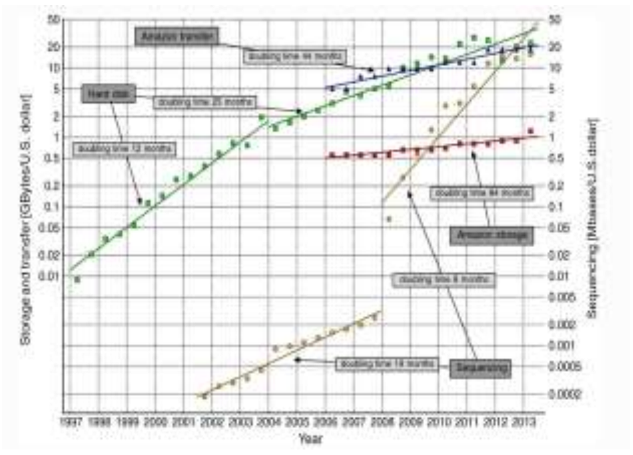
The storage service responsible for data storage faces increasing I/O pressure as the amount of data consumed by large-scale distributed data-intensive applications grows rapidly. Control. Storage has a particularly difficult challenge: maintaining high I/O capacity despite massive data with many simultaneous accesses. Achieving this requires massively parallel data transmission, which inevitably leads to high bandwidth consumption.

With the advent of cloud technology, data-intensive applications have become more attractive to a wider audience who do not have the money to manage expensive large-scale distributed infrastructures to run them. With the advent of cloud technology, data-intensive applications are more attractive to a wider audience that does not have the money to manage expensive large-scale distributed infrastructures to run them. In this context, maximizing storage and bandwidth usage is critical, as these services are compensated based on usage. This paper assesses the aids of plainly applying data compression to except storage space and bandwidth at the expenditure of a small additional computational cost. When dealing with high access concurrency, we want to reduce storage and bandwidth requirements while minimizing the impact on I/O performance. Therefore, data compression is necessary to reduce the need to store IoT data.

A common way to manage large amounts of data is compression. These methods save storage space and at the same time speed up information circulation (eg between research institutes). With roots dating back to the 1800s and early theoretical work published just after World War II, data compression is now used almost everywhere due to large amounts of data being processed and transmitted. There are many important concepts in package software, but the two mentioned below are the most important in bioinformatics. We only provide a brief summary of them here, and more details are available in a supplementary file or monograph.

Background and purpose:

Costs of space, transmission and sequencing have increased in recent years. Prices for cheap hard drives were previously collected from http://www.jcmit.com/diskprice.htm. In the 1990s and around 2000-2004, they were every 12 months. Then suddenly the doubling time increased to about 25 months. The actual sequencing costs calculated by NHGRI [5,] include not only reagent costs, as some studies suggest, but also labour, administration, depreciation of sequencing instruments, submission of data to a public database, etc. The proliferation of second-generation technologies led to major changes in sequencing costs around 2008. Amazon's storage and transfer speed is based on real-time industry contracts from the most prestigious data centers. It is interesting to note that storage costs in data centers are slowly decreasing, as the cost of a bare hard drive is only a small part of the total cost of maintenance. Inflation was not taken into account in the curves.



Huffman coding is a mathematical system that assigns a sequence of bits (a code word) to each alphabetic symbol. It was invented in 1952. Code words are of different lengths and rarer symbols are represented by longer code words according to the golden rule of information compression. After that, each symbol is

replaced by the corresponding code word to encode the specified sequence. The value of Huffman coding comes from its optimum, which means that no other code contributes to a shorter coded sequence.

Keywords: data compression, cloud-based IoT data management and storage optimization

Introduction:

The Internet of Things (IoT) is a network paradigm that connects things to the Internet (such as smartphones, smart TVs, home appliances). , Internet treatment, etc). In recent years, the number of IoT devices has grown 31 percent annually to 8.4 billion in 2017, with 30 billion devices expected by 2020. Therefore, these devices generate large amounts of data every day. The IoT is combined with cloud computing, which has almost unlimited storage and processing power to store and process these massive amounts of data. Cloud computing advances workflow by providing endless tools and helping to create efficient frameworks, such as collecting massive amounts of images for further processing. However, large amounts of data require more storage space and energy when transferred over the network.

Technology:

Lossless and lossless data compression are two data compression techniques. With lossless compression technology, compressed data can be restored exactly like the original data. LempelZiv compression methods are one of the most widely used lossless compression algorithms. On the other hand, in a lossy compression system, the extracted data is not identical to the original data and the error rates differ significantly. A two-layer lossy data compression technique has been applied to compress this data, which can be used in any IoT environment. This means that the data is compressed twice: once in the cloud node and then again in the cloud storage. Fog Computing is a highly virtualized platform that connects end devices with traditional cloud computing data centers to provide processing, storage and networking services. It is usually, but not always, at the edge of the network. Because in the fog we compress the data first, we can use less energy when transferring the data from the fog to the cloud. Because we know that sending large amounts of data requires more energy. Another advantage of this package is that the bandwidth usage during cloud-to-cloud communication is reduced.

Problem Statement:

As the Internet of Things (IoT) continues to expand, the amount of data generated by IoT devices has grown exponentially. This surge in data poses significant challenges for storage and bandwidth management, particularly in cloud computing environments where resources are finite and costly. Traditional methods of data storage and transmission are no longer sufficient to handle the immense volume of IoT data efficiently. Hence, there is a pressing need for innovative solutions to optimize storage space and bandwidth utilization while maintaining data integrity in cloud-based IoT systems.

Proposed Methodology:

Our proposed methodology focuses on leveraging data compression techniques to address the challenges of storage and bandwidth optimization in cloud-based IoT environments. The methodology comprises a two-layered approach involving compression at the Fog Computing layer and subsequent compression in cloud storage. By implementing compression techniques at both layers, we aim to achieve significant reductions in storage requirements and bandwidth consumption while ensuring minimal impact on data integrity and accessibility.

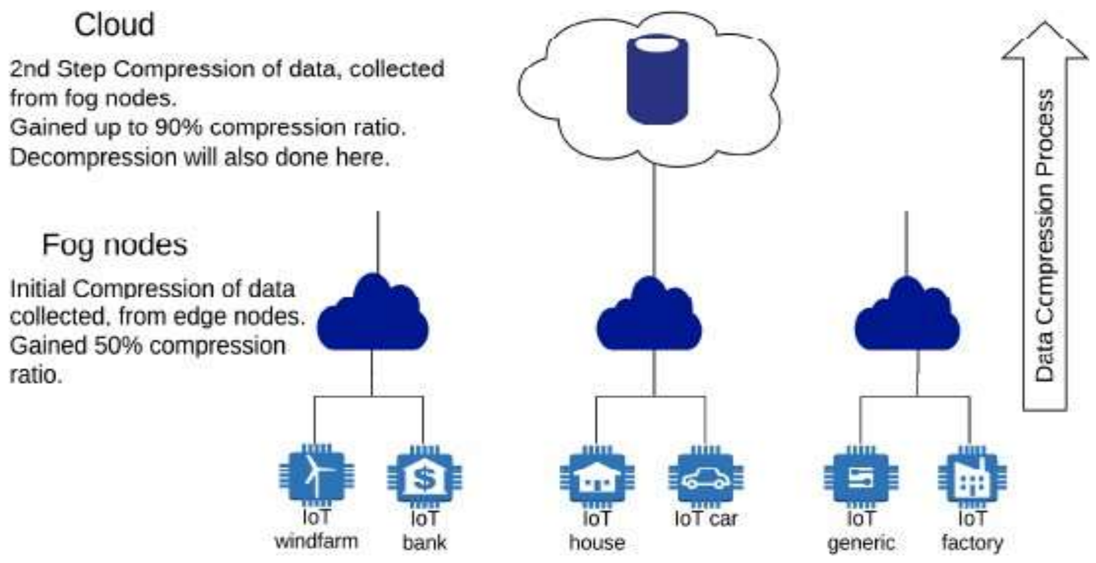

Cloud
2nd Step Compression of data, collected from fog nodes.
Gained up to 90% compression ratio.
Decompression will also done here.
Fog nodes
Initial Compression of data collected. from edge nodes.
Gained 50% compression ratio.
Data Ccmpression Process
IoT windfarm
IoT bank
IoT house
IoT car
IoT generic
IoT factory

Fig: Data compression process

1. Compression in fog

The fog node collects data from the IoT sensor and checks the suitability of the device in the first stage of compression. Data packets are rejected if the system is unauthorized. Otherwise, the table contains data for the specified time. A sorting algorithm is often used to sort data in ascending order. A calculation is then performed to determine the average of two consecutive values in the data set.

As a result, the data values in the fog are initially reduced by 50%, which is our main goal. Averages are rounded to eliminate fractional averages in the calculation. After that, a file is created where the average values are written. Finally, the paper is transferred to the cloud. Finally, Sumu is separated from the temporary collection data. One thing to remember is that if the total frequency of the input data set is an odd number, the last value in the data set is kept.

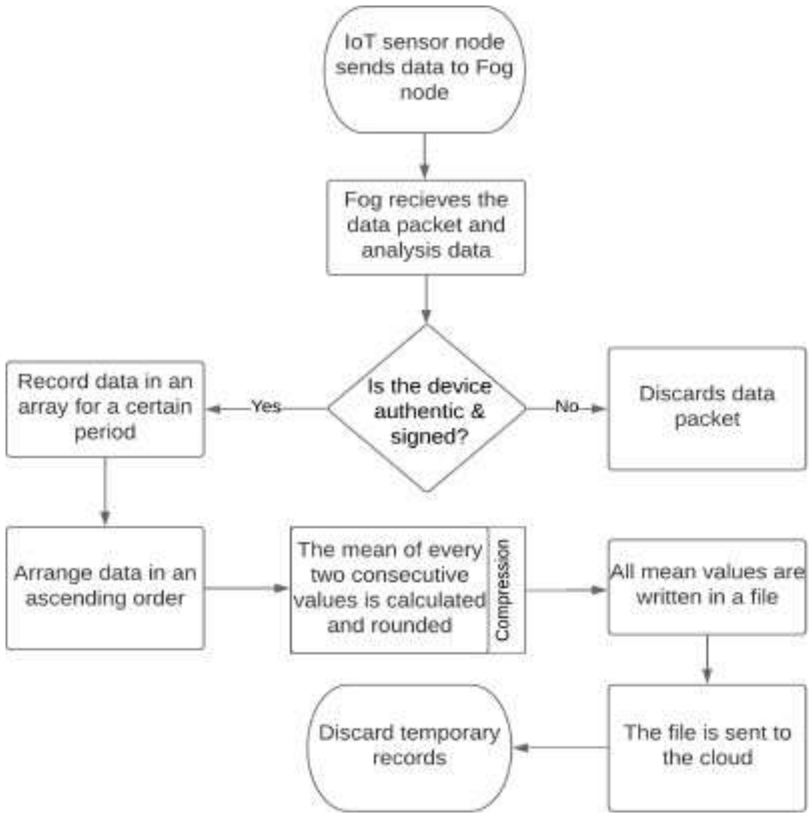

IoT sensor node sends data to Fog node
Fog recieves the data packet and analysis data
Is the device authentic & signed?
Yes
No
Record data in an array for a certain period
Discards data packet
Arrange data in an ascending order
The mean of every two consecutive values is calculated and rounded
Compression
All mean values are written in a file
The file is sent to the cloud
Discard temporary records

Fig: Compression in fog

Proposed Algorithm:

Algorithm 1: Algorithm for data compression in Fog Node

Input: Common number sensor data

Output: Compressed number data

```
1. System initialization;
2. Receiving data packets from end nodes;
3. if device id and IoTcert = true then
4. Set a time period
5. while data receiving do
6. Take data in a certain range;
7. Create array for input data set;
8. for Rangeeach do
9. Categorie the input data in ascending order;
10. while Timeperiod do
```

```
11.   Analyze the mean of each couple of values;
12.   Store mean values in a file;
13.   Send files towards the cloud storage;
14.   else
15.   Discards Packet
```

2. Compress in the cloud

Fog sends the file that was originally compressed in fog in this process. The cloud retrieves the file from Fog and recreates the data values of the file. These newly generated values are then stored in an array. The cloud measures the frequencies of all possible values and stores them in a two-column file.

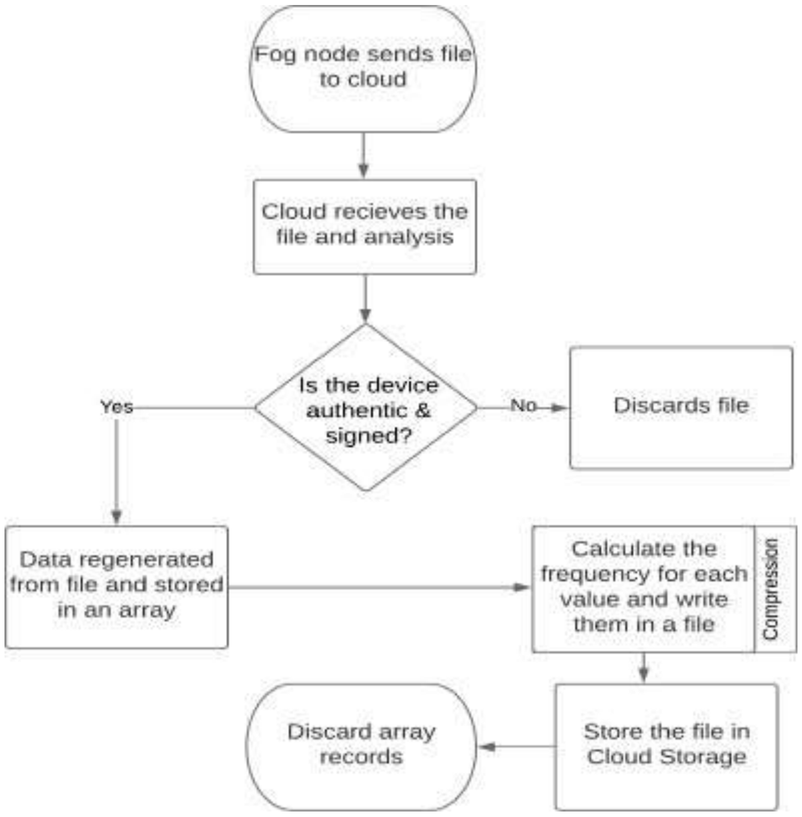

Fog node sends file to cloud
Cloud recieves the file and analysis
Is the device authentic & signed?
Yes
No
Discards file
Data regenerated from file and stored in an array
Calculate the frequency for each value and write them in a file
Compression
Store the file in Cloud Storage
Discard array records

Fig: Compression in the Cloud

***Algorithm* 2: Algorithm for Data Compression in the Cloud**

Input: Originally Compressed Sensor Data

Output: Compressed Output Data

```
1.    System Initialization;
2.    Receive file from Fog nodes;
3.    if DeviceID and IoTcert = True then
4.    Set Timeperiod
5.    while data receiving do
6.    Regenerate data from file;
7.    Create array for regenerated data;
8.    while Timeperiod do
9.    Calculate frequencies for each values;
10.   Store frequencies in a file;
```

```
11. else
12. Discards File
```

1. **Performance Analysis:**

To evaluate the performance of our proposed methodology, we will conduct extensive experiments in a simulated cloud environment using representative IoT datasets. We will measure key performance metrics, including compression ratio, error rates, energy consumption, and bandwidth usage, under various scenarios and configurations. Through rigorous analysis of these metrics, we aim to demonstrate the effectiveness and efficiency of our compression approach in optimizing storage space and reducing bandwidth consumption while preserving data integrity in cloud-based IoT systems. Additionally, comparative analysis with existing compression techniques will provide insights into the superiority of our proposed methodology.

Conclusion:

As the number of IoT devices grows, the big data problem seems to be getting worse. Contrary to necessity, current research is less advanced. As a result, our proposed model compresses IoT data more efficiently and with a lower error rate. We used lossless compression technology to mine IoT data in cloud storage. With the original packaging, we were able to find the fog knot. We were able to minimize energy consumption and bandwidth waste by using precompression in the fog node. We achieved a compression of about 90% with an error rate of 1%, which is very good and shows the efficiency of the process in compressing homogeneous structured data produced by IoT sensors, where approximations are needed for further mining. In this regard, better storage optimization, lower power consumption and bandwidth waste can contribute to more efficient management. In this regard, better storage optimization, lower power consumption and bandwidth waste can contribute to more efficient management.

References:

1. S. A. Awwad, C. K. Ng, N. K. Noordin, B. M. Ali, and F. Hashim, “Second and subsequent fragments headers compression scheme for ipv6 header in 6lowpan network,” in Sensing Technology (ICST), 2013 Seventh International Conference on. IEEE, 2013, pp. 771–776.
2. Y. Li, S. Xi, H. Wei, Z. Zhang, and C. Zhang, “A data compression algorithm for the sea route monitoring with wireless sensor network,” in Information Science and Cloud Computing (ISCC), 2013 International Conference on. IEEE, 2013, pp. 153–159.