International Journal of Research Publication and Reviews

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

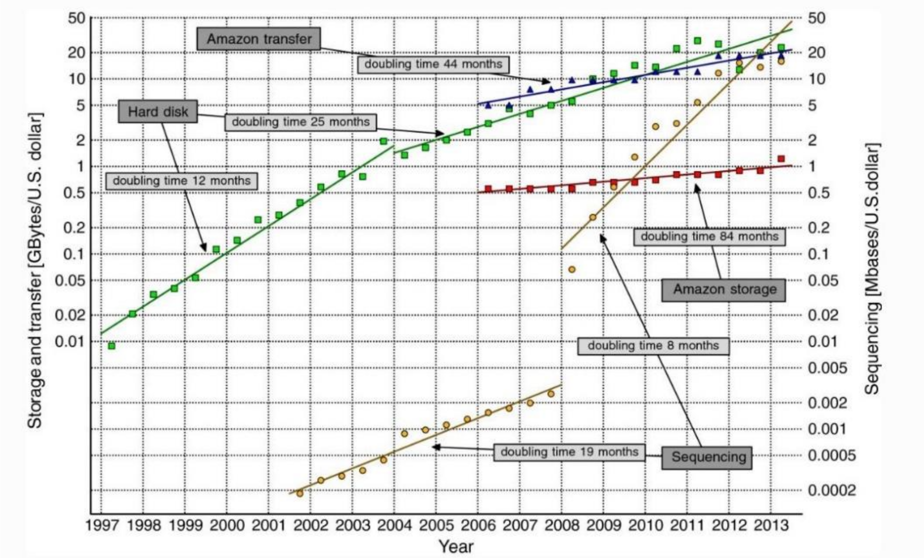
IoT Data Compression in Cloud Computing

Mayur Bhope[1],Bhavesh Gawade[2]

bhopemayur@gmail.com
bhaveshg4523@gmail.com
Department of MCA In ASM IMCOST, Thane
Mumbai University

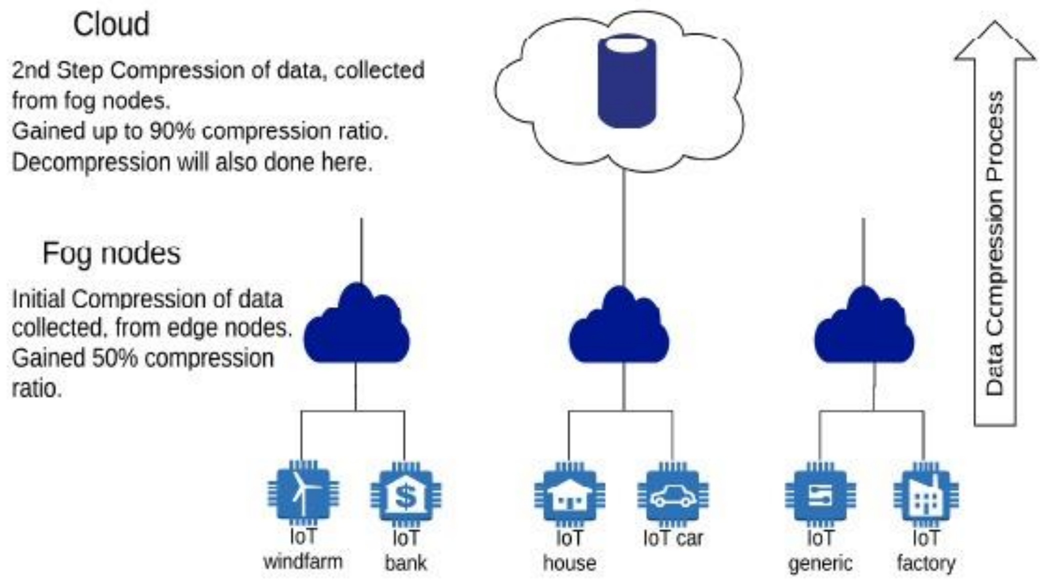
ABSTRACT :

The proliferation of the use of IoT devices has resulted in data generation that has increased exponentially. Advanced strategies need to be developed to manage and store it in the good environment of cloud computing. Managing and storing data efficiently has become a matter of critical concern. Data compression has emerged as a most important technique for addressing these issues. The importance of data compression designs in cloud based IoT systems concerns different aspects of storage optimization, transmission efficiency, and system performance because data compression designs reduce the data size with little loss of quality. As such, this research paper will investigate the different techniques of IoT data compression, their principles, methods of implementation, and their applications in cloud computing frameworks. This paper is interested in both lossless and lossy data compression techniques for cloud based IoT applications. The current study takes its results one step forward by totaling the consequences of various techniques on reduction in data storage requirements, bandwidth consumption, and processing speed to give a comprehensive outcome of the implications of these techniques concerning cost savings and performance improvement. On the other hand, the paper discusses some practical implementation challenges of data compression in IoT-cloud ecological environments: computational overheads, latency issues, and sustaining data integrity and accuracy. Looking Ahead Emerging trends and future directions in IoT data compression are then discussed, such as the integration of artificial intelligence and learning mechanisms into compression algorithms to render them adaptive and optimal, the development of adaptive compression techniques that alter based on real-time characteristics of the data, and how edge computing can be used to complement cloud-based strategies in IoT data compression. Through this direction, the paper is making a worthwhile contribution of insight for researchers and practitioners to have better ways of managing the data apart from improving the efficiency of data storage in the vastly expanding landscape of the IoT.



Introduction:

The Internet of Things (IoT) is a network paradigm that links things to the Internet (e.g., smart phones, smart TVs, home appliances, online healthcare, and so on). In recent years, the number of IoT devices has risen 31 percent year over year to 8.4 billion in 2017, with 30 billion devices expected by 2020. As a consequence, these devices produce large quantities of data on a daily basis. IoT is combined with Cloud Computing, which has nearly infinite storage space and processing capacity, to store and process these vast volumes of data. Cloud computing promotes work flow by providing infinite tools and assisting in the creation of efficient frameworks, such as the collection of a vast volume of images for further processing. However, vast quantities of data necessitate more storage space and energy during transmission across the network. Lossless and lossy data compression are the two forms of data compression techniques. The compressed data can be recovered exactly to the original data using lossless compression techniques. The LempelZiv compression methods are one of the most widely used lossless compression algorithms. In a lossy compression system, on the other hand, the decompressed data is not identical to the original data, and error rates which differ significantly. A two-layered lossy data compression technique is implemented to compress these data, which can be used in any type of IoT environment. This means that the data will be compressed twice: once at the fog node and then again in the cloud storage. Fog Computing is a highly virtualized platform that links end devices to conventional Cloud Computing Data Centers to provide processing, storage, and networking services. It is usually, but not always, located at the network's edge. Since we compress data in Fog first, we can use less energy during the data transmission from Fog to Cloud. Since we know that transmitting large volumes of data takes more energy. Another benefit of this compression is that bandwidth usage during data transfer from Fog to Cloud would be reduced.



Overview of IoT Data Compression Techniques

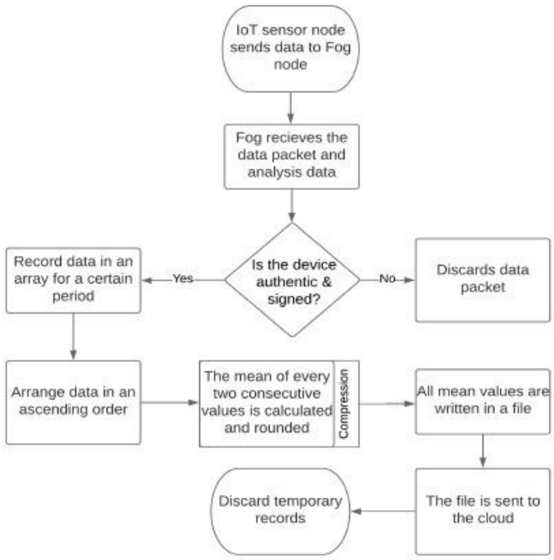
1. Compression in Fog

The Fog node gathers data from the IoT sensor and verifies the device's validity in the first stage of compression. Data packets are discarded if the system is unauthorized. Otherwise, the data is contained in an array for a fixed period. A sorting algorithm is often used to sort the data in ascending order. After that, a calculation is performed to determine the mean values of any two consecutive values in the data set.

As a result, we get one value for each pair of values.

As a result, the data values are initially decreased by 50% in the Fog, which is our primary target. Mean values are rounded to exclude fractional sections of mean values while calculating. After that, a file is created in which the mean values are written.

Finally, the paper is transferred to the cloud. Finally, the Fog detaches itself 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 data set's last value will remain.



Algorithm 1: Algorithm for data compression in Fog node

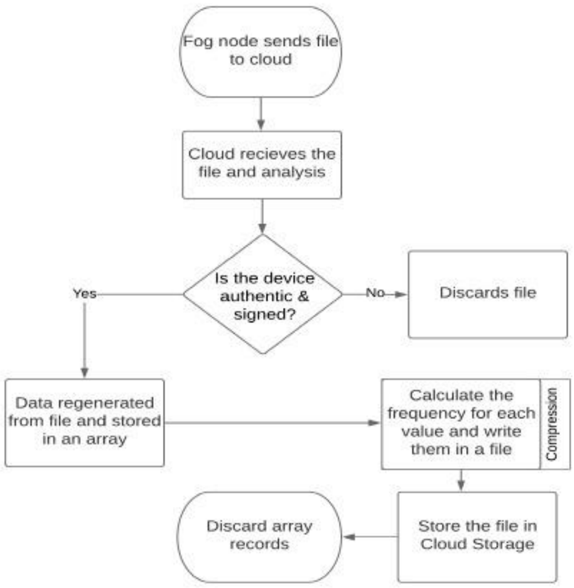
Input: Generic Numerical Sensor Data

Output: Compressed Numerical data

1. System Initialization.
2. Receive data packets from end nodes.
3. if DeviceID and IoTcert = True then
4. Set Time period
5. while data receiving do
6. Take data in a certain range.
7. Create array for input data set.
8. for Range each do
9. Sort the input data in ascending order.
10. while Time period do
11. Calculate the mean of each pair of values.
12. Store mean values in a file.
13. Send files towards the cloud storage.
14. else
15. Discards Packet

2. Compression in Cloud:

The Fog sends the file that was initially compressed in the Fog in this process. Cloud retrieves the file from Fog and regenerates the file's data values. These newly generated values are then saved in an array. The Cloud measures the frequencies for all possible values and records them in a two-column file.



Algorithm 2:

Algorithm for data compression in Cloud Input:

Initially 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 Time period
5. while data receiving do
6. Regenerate data from file.
7. Create array for regenerated data.
8. while Time period do
9. Calculate frequencies for each value.
10. Store frequencies in a file.
11. else

12. Discards File

Impact on Data Storage and Transmission :

Compression is the fundamental technology in cloud computing that influences data storage and transmission. Since the size of the data is reduced to a minimum, it is more efficient in storage and transmission and economical, too.

1. Storage Efficiency

You can save space because the compressed data occupies less space. Cloud service provider can store more data in the same hardware, hence utilizing their storage fully.

Cost Implications: Reduced storage space, translates to reduced expenditure. The cloud service provider is able to conserve its cash concerning hardware, Service, and electricity, while the end-user benefits through reduced storage cost.

Improved Data Management: Dealing with small datasets is quicker and easier. Data migration, replication, and disaster recovery are easily done and quite fast due to compression hence making data management even better.

2. Transmission Efficiency

Faster Data Transfer: Due to compression, less data is transferred which results in faster transfer rates. Faster transfer rate is necessary in many real-time applications such as IoT.

Bandwidth Conservation: It saves bandwidth because smaller data volumes are transferred across the network. This is important when the available bandwidth is limited or expensive.

Improved Performance: Smaller data volumes reduce network congestion and latency, hence better performance. This works wonderful for responsive applications such as video conferencing and online gaming.

3. Cost Reduction

Reduced Storage Costs: Compressed data reduces storage costs by minimizing costly storage hardware and energy. These savings are passed down to the user through lowered pricing models.

Lower Transmission Costs: Compression reduces the amount of data that is transmitted thus cutting down transmission costs. This becomes a critical factor for those who transfer large amounts of data or work within geographical areas that are expensive.

Efficient Data Handling: A smaller amount of data for processing also means lesser compute resource requirements. This lowers the cost of operations and also ensures a longer hardware life.

Challenges of IoT Data Compression

Although data compression has several benefits in the management of IoT in respect to cloud computing, several challenges are involved that need to be addressed so as to offer its full value in such an application. These are discussed below:

1. Data Diversity

Section Heterogeneous Data Types Many data types are generated by IoT devices, such as sensor readings, images, videos, and textual data. Each has different characteristics and compression requirements. For instance, sensor data is generally made of small-size but high-frequency measurements. In contrast, video data makes a large volume of continuous streams.

Versatile Algorithms: It is highly difficult to develop compression algorithms that could deal efficiently with this diversity. What can be effective and efficient for one kind of data may be ineffective or inefficient in another. Therefore, versatile and adaptive techniques of compression are required to meet different requirements of IoT applications.

2. Resource Constraints

Limited Computational Power Most IoT devices are designed to be lightweight and energy efficient, with limited processing power and memory. This limits the possibility of implementing complex compression algorithms requiring significant computational resources.

To that end, there is a need for the development of lightweight compression techniques that will not consume a lot of computational power as well as memory. In doing so, such techniques must offer a balance between ensuring effective compression ratios, on one hand, and the operational efficiency of IoT devices on the other. Some algorithms, such as Huffman coding and Run-Length Encoding, are examples of this, but even more optimized solutions might be called for.

Battery Life: Compression activities tend to consume more battery power-an extremely important consideration for a majority of the battery-operated IoT devices. Effective compression techniques need to be tiny members of sets such that they are transmitted using as little energy as possible and, at the same time, by كبير enough so that adequate data reduction is achieved.

3. Security Concerns

Data Privacy: IoT devices very easily deal with vital data. It could even be personal health records or confidential business data, etc. Then this compression process should not expose it to security risks. Whatever data is compressed, shall remain secure and confidential in all its lifecycles.

Compression and Encryption: Encryption must be used along with compression for decent data security. This can be difficult to achieve as data, while being encrypted, is random, which cannot be compressed further as such, making compressing of encrypted data inefficient. While compressing data prior to encryption calls for special care in process for it not to compromise on security. Compressive Sensing is an attempt where sampling and compression are integrated together in one secure operation.

Future Directions :

Several new technologies and innovative methods are sure to make the future of IoT data compression in computing on clouds rather advanced. Edge computing, AI, and machine learning and standardization are the key domains of development.

1. Edge Computing

Edge computing brings this processing nearer to the source, smart processing at the IoT device itself or in a proximal local edge server. This implies a tremendous reduction of data load on cloud servers by applying compression techniques at the edge as only the compressed data in transit enables minimized bandwidth usage and speed in terms of data transfer. This is going to yield improved response times and latency.

Real-Time Processing Compression to the edge brings real-time processing of applications like autonomous vehicles or industrial automation since, in these cases, the analysis of data may require the system to perform there and then.

Resource Management Efficient management of edge resources such as computational power and energy consumption is quite critical. In the future more lightweight and energy-efficient compression algorithms will be developed which are implemented on edge devices. At the same time, the algorithm needs to reach the best resource balance between high compression ratios and limited available resources or edge computing capabilities.

2. AI and Machine Learning

Adaptive Compression The whole compression of IoT data is bound to be completely revolutionized by AI and machine learning. Since machine learning algorithms have the capability to learn from real-time data patterns and characteristics, it is quite possible to use adaptive compression techniques dynamic in operation optimization process. For example, through artificial intelligence, redundant information or repetitive patterns of data can be recognized to compress such data more effectively.

Predictive Models: Machine learning can yield predictions of the types of data that will be generated and thus alter compression strategies accordingly. Thus, this proactive approach may help in the efficiency of compression, particularly in environments where there are predictable flows of data, like smart homes or industrial IoT setting.

Automate Optimization: AI-property systems can automatically choose the algorithm depending on data type and usage scenario. Given that most of the process is automated, there is less human intervention involved, which would simplify the whole process of compression and guarantee it that results are at an optimum.

3. Standardization

Interoperability-Implementation, the standardization of the compression protocols is crucial to allow interoperability between different platforms and devices in IoT. That way, in this case, the standardization gives a seamless transmission of the compressed data, decompressed, and used from any device, regardless is the cloud service provider of choice.

Efficiency Standardized protocols would make compression more efficient by giving the developers an effective and common framework. Optimized compression algorithms could be written across the board, and redundancy would decrease, which increases system performance generally.

Compliance and Security: Standardized compression techniques allow organizations to fulfill data security and privacy regulations. Because data is transmitted in a standardized, compressed form, sensitive information will be compactly transmitted; more importantly, organizations ensure ongoing compliance, thereby building confidence in the IoT solutions.

Conclusion :

With the rising number of IoT devices, the big data problem seems to be getting worse. In contrast to the need, existing research efforts are less advanced. As a consequence, our proposed scheme compresses IoT numerical data more effectively and with a lower error rate. To mine the IoT data in cloud storage, we used a lossy compression technique. We were able to rediscover the Fog node with initial compression. To mine the IoT data in cloud storage, we used a lossy compression technique. We were able to minimise energy consumption and bandwidth waste by using initial compression in the Fog node. We achieved a compression ratio of around 90% with a 1% error rate, which is very good and shows the efficiency of the process when compressing the homogeneous structured data produced by IoT sensor networks where approximate values are needed for further mining. In this regard, improved storage optimization, reduced energy usage, and reduced bandwidth squandering can all contribute to more effective management. In this regard, improved storage optimization, lower energy usage, and lower bandwidth squandering can all contribute to more effective 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