IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Development of Real Time System for File Compression on Amazon EC2

J. Singh[1], **V. Thapar**[2*]

[1] Department of Computer Science and Engineering, Guru Nanak Dev Engineering College, Ludhiana, India
[2] Department of Computer Science and Engineering, Guru Nanak Dev Engineering College, Ludhiana, India

**Corresponding Author: taurjaswant@gmail.com, Tel.: +91-96468-63063*

Available online at: www.ijcseonline.org

Accepted: 15/Jun/2018, Published: 30/Jun/2018

Abstract— A real time compression on cloud is an activity where compression takes place at the time of uploading a file on cloud and save it on cloud storage space in compressed form. In this paper, we come up with a same activity where compression and decompression take place on Amazon EC2 over the internet. With increasing need and usage of file sharing between user and servers, the demands for storage have been always kept increased, but the problem of storage space is still there. So to reduce the space for large files, compression is always a good idea. We have used Huffman coding and LZ77 compression technique in it to create lossless compression environment The files get compressed and decompressed at real time on Amazon EC2 where compression of files take place at the time of uploading and decompression is taking place at the time of downloading files to the local machine from Amazon EC2. The user gets the file in original size as well as in same file format as it was before being compressed and uploaded. Multiple users can create accounts and can manage their accounts separately by adding their data of any data format and get it compressed on Amazon EC2 cloud. One major advantage of the proposed system is that it is implemented on Amazon EC2 cloud and is used over the internet; it means that multiple users can access it at a particular instant of time from multiple locations and get their files compressed.

Keywords— Cloud computing, Amazon EC2, Real Time compression, Amazon web services

I. Introduction

With increasing need and usage of file sharing between user and servers, the demands for storage have been always kept increased, but the problem of storage space is still there. So to reduce the space usage on servers and devices, and to reduce download/upload times for large files, Compression is always a good idea and is used mostly. We have started working on a compression system, which would be based on Huffman coding and LZ77compression technique, the purpose for this compression system is to reduce the size of larger files and to store in compressed format on Amazon EC2 cloud. Main approach in this to do the things online on the cloud rather the offline like on a standalone system. So, the concept of cloud computing is related to this work as we are using Amazon EC2 to make this compression real-time compression. Meaning of real-time compression is that compression will take place at the time of uploading files to cloud and decompression will take place at the time of downloading files from the cloud.

The rest of the paper is organized as follows: In section II, we explore the Amazon Web Services and different services by AWS like Amazon EC2, Amazon Elastic Container Service (ECS), Amazon RDS, Amazon S3, Amazon Route 53 and Amazon CloudWatch. In section III, we describe data compression where we discuss Huffman coding and Lempel-Ziv Compression techniques. In section IV, the methodology is discussed, where we listed different steps to achieve real-time compression on Amazon Web Services' compute engine Amazon EC2. In section V, results and discussion are listed. Finally, in section VI, conclusion and future scope are listed.

II. Amazon Web Services (AWS)

Amazon Web Services (AWS) is a cloud computing platform provided by Amazon [1, 2]. It is a set of cloud services, providing cloud-based computation, storage and other functionality that enable organizations and individuals to deploy applications and services on an on-demand basis and at commodity prices. Different computing resources are available at low costs and no upfront investment is required for the resources. The users simply need to pay for the resources that he consumes on factor premise. AWS provides infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) contributions [3, 4, 5, 6]. Traditionally it was difficult for a company to provide the performance to the distributed users so they concentrate on the only single geographical region at a time. But with the help of AWS, this problem was solved and no one can deploy its application all around the world and provide a better experience for customers. Different services provided by AWS are written below:

Compute - Amazon EC2

Amazon Elastic Compute Cloud is a web-based service which provides the virtual server or we can also say that provides a virtual machine called as instance. This instance is basically a computer machine which is presents on the cloud. There is a number of instance types are present with varying capacity and storage. These instances come up processor, memory, and storage. Selection on this depends upon the environment of application that is going to be deployed on it. Different instances are – t2.micro, t2.small, m5.large etc [3, 7]. These different instances come with different configurations like CPU, memory and storage space. We used a t2.micro instance with Ubuntu operating system installed on it. T2.micro has Intel Xeon physical processor and 1GiB of memory. Amazon EC2 diminishes the time required to get and boot new server instances (called Amazon EC2 cases) to minutes, enabling you to rapidly scale limit, both up and down, as your computing prerequisites change. Amazon EC2 changes the financial aspects of computing by enabling you to pay just for a limit that you really utilize [4]. Different features of the service include - auto-scaling–it enables us to scale Amazon EC2 limit up or down automatically as per conditions characterized by us, elastic cloud hosting services - this empowers to pick the memory, CPU, storage and boot portion appraise, that are perfect for decision of working framework and application and completely controlled - complete control of your Amazon EC2 events as a superuser [8].

Amazon Elastic Container Service (ECS)

Amazon ECS is a very versatile, superior compartment coordination benefit that backings package (programming stage that enables you to construct, test, and convey applications rapidly) holders and enables you to effectively run and scale containerized applications on AWS [3]. Amazon ECS disposes of the requirement for you to introduce and work your own compartment organization programming, oversee and scale a group of virtual machines, or calendar holders on those virtual machines. ECS is associated with two terms one is docker and another one is a container. Docker is basically a type of software or a package that allows us to make, test and modify a different type of applications in within a couple of minutes. These different packages are standardized under one package that is called as a container. The container provides a very good working and development environment to the designers and developers. Different machine learning applications can be deployed with the help of these containers. The whole thing is managed by the identity and security management of AWS that will be covered in next coming points. Utilizing Docker gives you a chance to deliver code speedier, institutionalize application activities, consistently move code, and spare cash by enhancing asset use.

Storage - Amazon S3

Amazon Simple Storage Service (Amazon S3) is a storage service or we can say that it is global infrastructure as a service (IaaS) provided by AWS [1, 3]. It enables highly-scalable, secured and low-latency data storage from the cloud. The interface provided by AWS for storage can be easily used for data storage as well as for retrieval of data from any location or at any time [3]. The space provided by Amazon S3 is scalable for data analytics, data backup and data archival. Allows unlimited information and object storage of most information sorts in an exceeding sort of formats [2]. A keep information set, that is an object, ranges from 1 TB to 5 TB. It offers Reduced Redundancy Storage (RRS), that reduces latency by storing information in regionally un-integrated buckets and protects resources and facilitates application potency for users in geographically spread locations. It also provides different web services like REST (Representational State Transfer) and SOAP (Simple Object Access Protocol) that are built using different web development kits.

Databases, Data Management - Amazon RDS:

It is a database and database management service provided by AWS or it is relational database service that is provided over an internet to make and easy to set up, work, and scale a social database in the cloud. It is very cost-capable and resizable service limit while directing taking care of database organization assignments, freeing you up to base on your applications and business. Amazon RDS is comprised of different database management systems like PostgreSQL, Amazon Aurora, MariaDB, MySQL, Microsoft SQL Server and Oracle [3].

Networking and Content Delivery - Amazon Route 53

Amazon Route 53 is a to a great degree open, accessible, versatile and adaptable cloud Domain Name System (DNS) web service. It translates domain name like www.example.com into numeric IP address like 192.0.4.6 which is how computers connected to each other [3]. An IP addresses referred to as records, listed for domain names in the DNS phone book managed by Amazon 53 route service. Answer requests also known as queries of this service are responsible for to translate domain names into their corresponding IP addresses. Queries are connected to infrastructure in AWS like Elastic Load Balancers and allow developers to map domain names to EC2 instances, S3 buckets and other AWS resources. Latency Based Routing Geo DNS and Weighted Round Robin are an architecture which is used to manage global traffic.

Management and Monitoring - Amazon CloudWatch

It is basically a watching service that will keep track of each and every activity running on Amazon cloud. All the running applications keep tracked by this service. Different users can use Amazon CloudWatch to accumulate and track estimations, assemble and screen log records, set cautions, and instantly react to changes in AWS assets [3]. All the Amazon EC2 instances, Amazon RDS and Amazon DynamoDB table are likewise observed by Amazon CloudWatch, and also moreover custom estimations made by your applications and administrations, and any log reports your applications to create. You can use Amazon CloudWatch to get framework wide visibility into usage of resources, application execution, and operational security [9]. You can use these encounters to react and keep your application running effectively.

III. DATA COMPRESSION

Data compression is a process that modifies, encode or convert the bits structure of data in such a way that it reduces the data via removing redundant information and it consumes less space on disk [9, 10, 11]. Data compression save the capacity of storage, increase file transfer speed and reduces the cost for storage as well as save network bandwidth. It enables quickly send files over the internet. Compression is done with help the specific program called as an algorithm; an algorithm consists of series of steps that define how to compress the size of the data. Data compression methods are classified into two techniques – lossy compression and lossless compression. The algorithm that removes some part of the information from original data is called lossy compression and another one where the compressed file remains same as the original one is called lossless compression. Some of the main data compression techniques are Huffman Coding [12], Run Length Encoding [9], and Lempel-Ziv compression [13]. In this paper, we discuss Huffman Coding and LZ77 compression technique together for real-time compression on Amazon EC2 where compression will take place at the time uploading a file to the cloud, without any change in information, without any data loss as well as without any change in format of the original file.

Huffman Coding

Huffman coding comes under lossless compression technique. The thought is to allocate variable-length codes to input character sequence; lengths of the assigned codes depend on the frequencies of relating characters [13, 14, 15]. The most successive character gets the smallest code and the least frequent character gets the largest code. Another term "Prefix Codes" associated with this technique; these codes are variable-length codes that are assigned to input characters. The distribution of these codes is done in such a way that the code assigned to one character is not prefix of code assigned by any other character. In this way this technique; makes sure that there is no ambiguity when decoding the generated bitstream [13]. Two major parts of this technique are – First one is to build a Huffman tree from input character and the second one is to traverse the Huffman tree and assign codes to characters.

LZ77 (Lempel-Ziv) Compression

A Universal Lossless Data Compression Algorithm created by Abraham Lempel, Jacob Ziv and Terry Welch [11]. It is a dictionary based dynamic compression algorithm which peruses the character substrings and assigns reference codes as in encoded output [13]. It expels redundancy in output file by skipping redundant word references if found in perusing input stream. This algorithm takes each input sequence of bits of a given length 8 bits, 16 bits or 32 bits and creates an entry in a table [11]. This table is a dictionary for that particular bit pattern, consisting of the pattern itself and a shorter code. Input is read, any pattern that has been read before results in the substitution of the shorter code, effectively compressing the total amount of input to something smaller. It includes the look-up table of codes as part of the compressed file. The decoding program that decompresses the file is able to build the table itself by using the algorithm as it processes the encoded input.

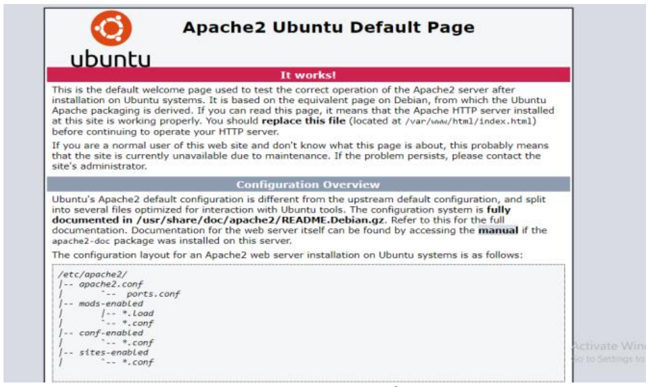
IV. METHODOLOGY

Machine Setup

First and the foremost I have to setup a working environment. It may be either virtual of realistic. In this research of mine I have to make on Amazon cloud, so to create the working environment I have to setup a new virtual machine on Amazon EC2 Cloud. I take a tier from Amazon Web Services (AWS), This tier is free of cost for one year. After this I created an instance of this tier called T2 micro tier which will act as a virtual machine for this whole research. T2 micro machine is basically a computer system that comes up with Intel Xeon processor, 1GB primary memory, and 30 GB storage space it can be scaled up as - Amazon Web Service comes with an option of "pay per use". There is minimal storage space is required according to an operating system that is going to be installed on this machine. According to the proposed work, we need Ubuntu operating system, so minimum 8GB space is required.

Development Environment

After a virtual machine setup, now it's time to install appropriate packages and tools on a virtual machine that is created earlier. Different packages like - installation of the web server, storage and many more things like disk mount directory structure creation and root directory setup etc. Most importantly, an operating system will be required to carry all the proceedings. For that, Ubuntu Operating System installed which is free of cost and has many strong points to use on these cloud services. And also software to support HTTP requests is obviously needed. No one could win the race other than Apache, as it is a free open source and most compatible. I install Apache web server on this virtual machine. After that this machine comes up with a Public IP Address from where everyone can access this server i.e18.188.186.118 (accessible for all end users), as shown in figure 1, a virtual machine created on Amazon EC2 with public IP address and it also comes with Private IP address but this private address is only known to the owner of the aws panel (aws account). This private address is used for terminal login and for directory setup. The private IP address is for this research is 172.31.21.213. It also comes up with a particular key pair (Public and Private Key) that is required to get login into the server with the help of SSH. Without this keypair, we cannot access this machine. There is no any text-based password is present, only one thing is used as a password and as well as for encryption is this public-private key pair. If by chance we lost this key then we cannot access our virtual machine.


Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Figure 1. Public IP address of virtual machine

Security key pair:

Security pair (private and public) is generated. Every chamber has its secret keys. The proposed system developed environment is also protected with public and private key pair. This keypair is only dealt with superuser or we can say that admin of Amazon EC2 cloud because it an entry point to enter your machine. Amazon Web Services provides us Cloud platform, so this platform does not only come with single machine environment. There are numbers of machines present on a single cloud and all these machines are separate from each other with different key pairs. So the main thing is that a single key pair is related to single machine only. One or two machines cannot have the same keypair. In general, to take an access of virtual machine, key pair must be known to admin.

Algorithm Development

Compression is key concept this research. The whole idea behind this research revolves around compression of the data. Compression is always helpful when its results are lossless and is fast. The proposed algorithm followed here is Huffman Coding and LZ77 compression technique. The key feature of this technique is that it compresses all types of files it may be text, photoshop files, illustrator files, and documents anything. This technique is lossless. That means there happens no loss of data after compression of the data and when you retrieve the data back again you get the same amount of data that you firstly uploaded without any loss. The idea is to assign fixed-length codes to input characters; lengths of the assigned codes are based on the frequencies of corresponding characters. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bitstream. The flowchart below represents the entire process involved in the executing of compression using the proposed algorithm.

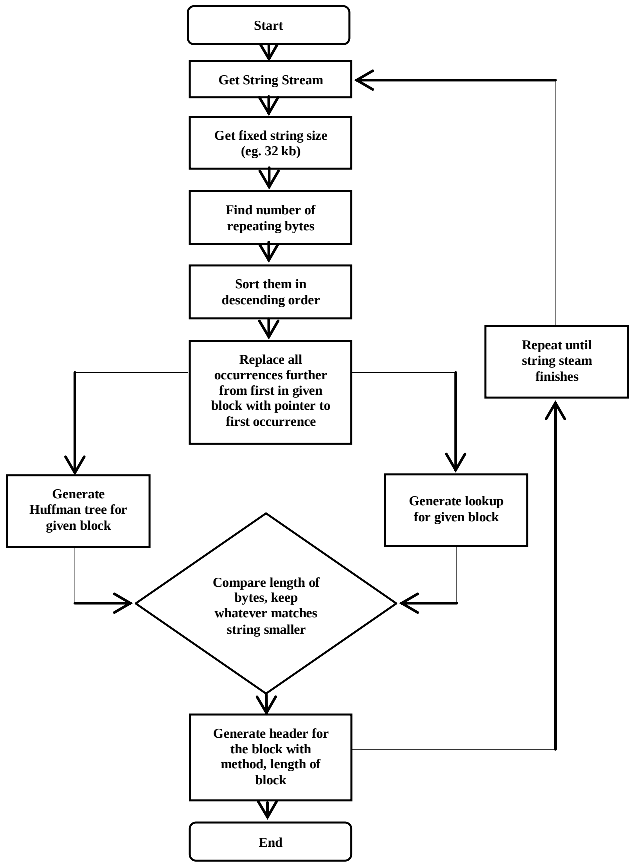

Start
Get String Stream
Get fixed string size (eg. 32 kb)
Find number of repeating bytes
Sort them in descending order
Replace all occurrences further from first in given block with pointer to first occurrence
Repeat until string steam finishes
Generate Huffman tree for given block
Generate lookup for given block
Compare length of bytes, keep whatever matches string smaller
Generate header for the block with method, length of block
End

Figure 2. Flowchart of proposed compression technique

The flowchart in figure 2 shows that string stream acts as an input in the first step. Secondly, fixed size string is to be prepared or it can be said that block of characters will be created. Each block has string stream with fixed string size e.g 32kb. After that the third step, repeating bytes will be calculated and it will be sorted in descending order. After that, replace all occurrences further from first in given block with a pointer to the first occurrence. Next, a Huffman tree is to be generated for given block and the second thing lookup table is generated. Huffman tree is based on Huffman coding technique; it is a particular type of optimal prefix code. The output from Huffman's tree can be viewed as a variable-length code table for encoding a source symbol (such as a character in a file). A Lookup Table is an array or matrix of data that contains items that are searched. Lookup tables may be arranged as key-value pairs, where the keys are the data items being searched (looked up) and the values are either the actual data or pointers to where the data are located. Now, check the length of bits that are generated by Huffman tree and look up the table, keeps one from Huffman tree and lookup table whatever makes string smaller, either Huffman tree or lookup table via checking the smaller length of bits. Lesser number of bits results with more compression, or we can say that length of bits at this step defines which one can do better compression either Huffman tree or lookup table. After this, a header for the block will be generated, which contains complete of information that is processed like tree information, lookup table information, compression information, and length of the block. This process will repeat until all blocks will be processed that are generated at starting time. Each block is associated with its individual header. After that, we will get the compressed file without repetitive character sequence. At the time of decompression, firstly header will be checked either header information is present or not. If yes, then it will be checked compression was done which method either by Huffman tree or with a lookup table. Then selected method will be applied and decoded string will be there, and the decompressed original file will be there without any data loss. In this way, compression and decompression take place in the proposed system. Algorithm for proposed compression technique is written below:

1. Input : String
2. Find Number of repeating bytes.
3. Sort_DESC(String)
4. Replacing all occurrences with pointer (p) starting from first given block.
5. Generate Huffman tree & Lookup table for given block.
6. Find_string_length = length of bits (storing/keeping whatever makes string smaller either from Huffman tree or lookup table)
7. Generating header (h) & length (l) of block.
8. Goto step 2 until string stream finishes.
9. Output : Compressed String

Combining algorithm and developed environment

After completion of all pre-requisites so now I have to combine developed proposed algorithm and developed environment that have created earlier. One must combine both of the separately created algorithm and the developing environment. So, now two major things will take place a web interface and runtime compression. Runtime compression and decompression is basically a compression and decompression that will take place at the time of uploading and downloading files on Amazon Cloud.

Web Interface

A user interface is the main thing of every system or we can say that interface defines its functionality as well as defines its nature also, similarly our compression system has very user-friendly user interface which is very simple and also easy to use. There are many functions performed with the help of this user interface like adding a new folder, upload files, download files etc to it. And also deleting it afterward when it is not to be used in the future, or it is declared as a dead piece of information. A very interactive feature of this interface is that it not only gives a beautiful design to it and simplifies the operations of adding and deleting new files but a very important feature is that it can add multiple users also.

Multi-User Functionality

Multi-user functionality is like an invocation for our system. It gives the liability of creating more than one isolated accounts in the system which cannot view the information uploaded to the other account. In these personal accounts, users can upload their own files which will be run-time compressed and this compression is lossless. Each user can download these compressed files at any instant of time. All basic operations like upload files, delete files, rename files, create folder etc. options are available for each user. The lossless compressed file - when downloaded, it will give you the exact complete file of the same size as it was off before uploading as well as with same file format. Other compression systems presently cannot compress the file during run-time and cannot give the same file format.

V. RESULTS AND DISCUSSION

There are many compression systems available in the market that also compresses the files and upload them onto the cloud, but their drawback is that they cannot compress the files while run-time. They all first compress the file online and send them onto the cloud bit our system compresses all the files while run-time and when downloaded also it gives the complete original size of the file which was before being uploaded onto the cloud. Multi-user environment will not only keep the things sorted but will also keep them private because the users cannot view one other’s account information without their prior permission. Another important thing is that, these are the different types of files associated with their particular file format are used in this research. These files are – cc.pptx, aws.zip, bootstrap.psd, map.psd and book.pdf. These are the different types of files that are compressed using proposed compression technique and gzip. These files are shown in different tables that are associated with this research work. Almost, all the tables that are present in this chapter, have these files. So, rather than to mention these files on each table description, I write all files, their name and their format associated with each and every file like .pptx(powerpoint file), .psd(photoshop file), .pdf(portable document format) and .zip(archive file format). The comparison is taking a place in this research with some parameters. The parameters are written below, Where, SBC: Size before compression; SAC: Size after compression; SD: Size Difference; CR: Compression Ratio; SSP: Space saving percentage.

a) **Size Difference:** The distinction between the size of the unique document and packed record is the manner by which they are not at all like each other.

$$SD = SBC - SAC$$

b) **Compression Ratio:** Pressure proportion is a procedure of ascertaining how much or to what degree a document is compacted.

$$CR = SBC/SAC$$

c) **Space-saving percentage:** It is defined as the reduction in size relative to the uncompressed size:

$$SSP = (SD/SBC)100$$

Compression using Proposed Compression System on Amazon EC2

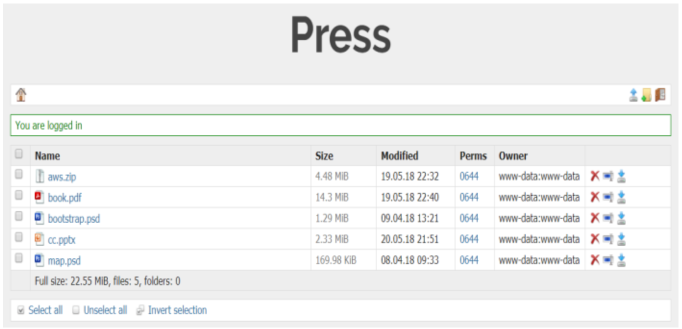
The effectiveness of research can be measured in a particular way in which was examine the size of the uploaded files once before uploading them and once after uploading them on the cloud. There is no change in format at the time of uploading and downloading also. Please check to check different files along with different parameters associated with files like size before compression, size after compression, size difference, compression ratio and space-saving percentage using proposed compression system implemented on Amazon EC2.

Table 1: Compression using proposed compression system

FN	SBC	SAC	SD	CR	SSP
cc.pptx	2.42 mb	2.33 mb	0.09 mb	1.0386	3.72 %
aws.zip	6.07 mb	4.48 mb	1.59 mb	1.3549	26.19 %
bootstrap.psd	30.5 mb	1.29 mb	29.21 mb	23.6434	95.77 %
map.psd	1.33 mb	0.169 mb	1.161 mb	7.8698	87.29 %
book.pdf	16.4 mb	14.3 mb	2.1 mb	1.1469	12.8 %

Where, FN: File name; SBC: Size before compression; SAC: Size after compression; SD: Size Difference; CR: Compression Ratio; SSP: Space saving percentage

In table 1, different files are shown with their corresponding different parameters like file size before compress, file size after compression, size difference, compression ratio and space-saving percentage.



Press

You are logged in

Name	Size	Modified	Perms	Owner
aws.zip	4.48 MiB	19.05.18 22:32	0644	www-data:www-data
book.pdf	14.3 MiB	19.05.18 22:40	0644	www-data:www-data
bootstrap.psd	1.29 MiB	09.04.18 13:21	0644	www-data:www-data
cc.pptx	2.33 MiB	20.05.18 21:51	0644	www-data:www-data
map.psd	169.98 KiB	08.04.18 09:33	0644	www-data:www-data

Full size: 22.55 MiB, files: 5, folders: 0

Figure 3. Implementation of proposed system on Amazon EC2 with compressed file's size

Figure 3 shows, implementation of the proposed system on Amazon EC2 with particular output that is also used in table 1. It is very amusing to see that the proposed work can compress almost every type of file without any change in format; it may be .zip, .psd, .pdf, .docx, & many more. From the experiments done, it can be inferred that there had been a subsequent amount of compression as it is very clear that the total size of the files is **56.8 mb**, whereas, the combined size of the files after compression is **22.55 mb.** So the total size difference of files is **34.25 mb,** we can say that we save **34.25 mb** space using proposed compression system.

Figure 4 represents original file size and size after comparison using proposed compression system. Basically, it shows how much size is reduced using proposed compression system and show a comparison of original files and compressed files.

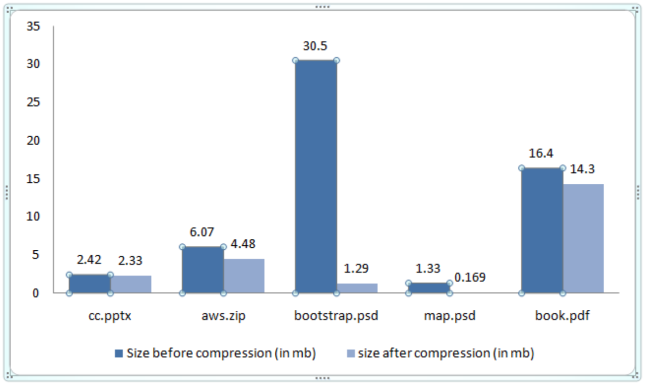


Figure 4. Size comparison of files before compression and after compression using a proposed compression system

Total size representation of all files in a combined manner is shown in figure 5. It includes total size of files before compression, the total size of files after compression using proposed compression technique and total size difference that is saved storage space. It also shows that, how much quantity of data is compressed without any data loss.

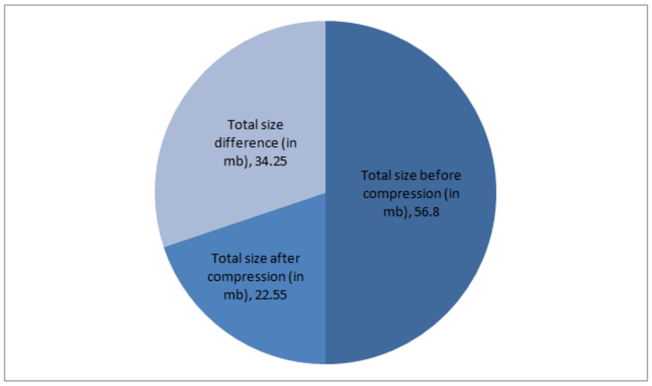


Figure 5. Total size of files before compression, after compression & Total size difference of files using proposed compression system

VI. CONCLUSION AND FUTURE SCOPE

With increasing data on the cloud day by day, there is a need for a better solution through which would upload same data but with using lesser space on the cloud. The research is based on the same thing which would upload the whole data which will consume actually less space on the cloud. So, to achieve our main objective, a real compression - we implement Huffman coding and LZ77 compression techniques together and implemented on Amazon EC2 cloud, to get files compressed online and save it on the cloud. The whole compression system is associated with Amazon EC2 cloud, where a virtual machine is created to get use this compression system over the internet with multiple user environment. How much data get compressed is measured with the help of some parameters like Size Difference, Compression Ratio, and Space-saving percentage. These parameters define how much size of files gets compressed and how much storage space is saved as compared with the original size of the file.

In order to reduce an amount of bandwidth, a new functionality will be added. It will comprise an operating system based client (software) and a control panel (cPanel). This will reduce an amount of bandwidth plus the confidentiality of the data will be maintained i.e location where the data will be saved will not know to the user. The client (software) will have cpanel from where a user can easily handle all files and folder without a worrying about internet speed. Compressed files will be present on user's personal computer but in actual that compressed file will not present on that computer, it is present on the cloud.

ACKNOWLEDGMENT

The author is highly thankful to the teachers and staff of Guru Nanak Dev Engineering College, Ludhiana.

REFERENCES

[1] J. Murty, *"Programming amazon web services: S3, EC2, SQS, FPS, and SimpleDB"*,O'Reilly Media, Inc, USA,pp.51-251, 2008.

[2] R.P. Padhy, M.R. Patra, S.C. Satapathy,*"X-as-a-Service: Cloud Computing with Google App Engine, Amazon Web Services, Microsoft Azure and Force"*, International Journal of Computer Science and Telecommunications, Vol.2, Issue.9, pp.8-16,2011.

[3] J. Varia,S. Mathew,*"Overview of Amazon Web Services"*, Amazon Web Services, pp.1-22,2017.

[4] Q. Zhang, L. Cheng, R. Boutaba,*"Cloud computing: state-of-the-art and research challenges"*,Journal of internet services and applications,Vol.1, Issue.1, pp.7-18,2010.

[5] V. Patidar, M. Kumbhkar, *"Analysis of Cloud Computing Security Issues in Software as a Service Vishnu"*, International Journal of Scientific Research in Computer Science and Engineering, Vol.2, Issue.3, pp.1-5, 2014.

[6] U.Kaur, M. Mahajan, D. Singh, "A Comparative Analysis of Trust Models in Cloud Computing", International Journal of Scientific Research in Network Security and Communication, Vol.6, Issue.2, pp.19-23.

[7] P. Kokkinos, T. A. Varvarigou, A. Kretsis, P. Soumplis, E. A. Varvarigos,*"Cost and utilization optimization of Amazon EC2 instances"*, In the proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing Cost, USA, pp.518-525,2013.

[8] G. Kulkarni, R. Sutar, J. Gambhir, *"Cloud computing-Infrastructure as service-Amazon EC2"*, International Journal of Engineering Research and Applications,Vol.2, Issue.1, pp.117-125, 2012.

[9] M. Sharma, *"Compression using Huffman coding"*, International Journal of Computer Science and Network Security, Vol.10, Issue.5, pp.133-141,2010.

[10] D. Mathpal, M. Darji, S. Mehta, *"A Research Paper on Lossless Data Compression Techniques"*, International Journal for Innovative Research in Science & Technology, Vol.4, Issue.1, pp.190-194,2017.

[11] K. Sharma, K. Gupta, *"Lossless Data Compression Techniques and Their Performance"*, In the proceedings of the International Conference on Computing, Communication and Automation (ICCCA2017), India, pp.256-261,2017.

[12] S. Porwal, Y. Chaudhary, J. Joshi, M. Jain, *"Data compression methodologies for lossless data and comparison between algorithms"*, International Journal of Engineering Science and Innovative Technology, Vol.2, Issue.2, pp.142-147,2013.

[13] S. Kreft, G. Navarro, *"LZ77-like compression with fast random access"*, 2010 Data Compression Conference, Snowbird, Utah, USA, pp.239-248,2010.

[14] J. Shoba, S. Sivakumar,*"A Study on Data Compression Using Huffman Coding Algorithms"*, International Journal of Computer Science Trends and Technology, Vol.5, Issue.1, pp.58-63,2017.

[15] A. Shahbahrami, R. Bahrampour, M.S. Rostami, M.A. Mobarhan, *"Evaluation of huffman and arithmetic algorithms for multimedia compression standards"*,International Journal of Computer Science, Engineering and Applications,Vol.1, Issue.4, pp.34-47,2011.

Authors Profile

J Singh studied Bachelor of Technology in Computer science and engineering from Guru Nanak Dev Engineering college, Ludhaina (India) in 2014. He is currently pursuing Master of Technology in Computer Science & engineering from Guru Nanak Dev Engineering College, Ludhaina (India). His main research work focuses on Real time compression, Cloud Computing, web routing services and cryptography.

V Thapar pursed Bachelor of Technology in Computer science & engineering in 2001 and Master of technology in Information technology in 2003. He is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computer Science & engineering, Guru Nanak Dev Engineering College, Ludhiana (India) since 2005. His main research work focuses on Network Security,Web Technologies, and Cloud computing. He has 17 years of teaching experience and 15 years of Research Experience.