INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

STORAGE COST MINIMIZING IN CLOUD - A PROPOSED APPROACH BASED ON LEMPEL-ZEV-WELCH

[1]Priyanka Gidwani, [2]Sreeja Nair

[1]M.Tech Scholar, CSE, OIST, Bhopal, India
[2]Prof. CSE Department, OIST, Bhopal, India

Abstract: - Many people think cloud computing as a new idea but actually the idea itself is almost as old as the computer itself. The concept of cloud computing was introduced in the 1960s. Cloud computing concepts, technical characteristics of cloud computing, online resource management and challenges that Cloud Service Providers (CSP) or vendors face during cloud engineering for providing better resource management and Storage cost management of servers resources. LZW is wordbook primarily based formula, that is lossless in nature and incorporated because the commonplace of the advisory committee on International telegraph and telephone, that is enforced during this paper. LZW compression is one in every of the adaptation wordbook techniques. The wordbook is formed whereas the in four being encoded. Thus cryptography is done on the fly. The wordbook needn't be transmitted. Wordbook is designed up at receiving endwise the fly. If the wordbook overflows then we've to reinitialize the wordbook and add a little too every one in every of the code words. Selecting an oversized wordbook size avoids overflow, however spoils compressions. A codebook or wordbook containing the supply symbols is made. For 8-bit monochrome pictures, the primary 256 words of the wordbook are allotted to the grey levels 0-255. Remaining a part of the wordbook is crammed with sequences of the grey levels. LZW compression works best once applied on monochrome pictures and text files that contain repetitive text/patterns

Keywords— Encoding, Decoding, Compression Ratio

I. INTRODUCTION

Data compression is usually named as committal to writing, wherever committal to writing is general term showing any special illustration of knowledge that satisfies a given want. Scientific theory is outlined because the study of economical committal to writing. Information compression is also viewed as a branch of knowledge theory during which the first objective is to attenuate the quantity of knowledge to be transmitted. Information compression has a vital role within the space of transmission and storage. It plays a key role in data technology. The reduction of redundancies in information illustration so as to decrease information storage demand is outlined as information compression. It used less usage of resources like memory house or transmission capability. Information compression is classed as lossless and lossy compression. Lossless compression is employed for text and lossy compression for image. In 1980, Terry Welch invented LZW algorithm which became the popular technique for general purpose compression systems. It was used in programs such as PKZIP as well as in hardware devices. Lempel-Ziv-Welch proposed a variant of LZ78 algorithms, in which compressor never outputs a character, it always outputs a code. To do this, a major change in LZW is to preload the dictionary with all possible symbols that can occur.

II. RELATED WORK

Cloud computing is the acclaimed word in Information Technology's and computer world. It is becoming very famous day by day. It is the fifth generation of the internet. Cloud computing helps IT business to transform power of computing in a smart, efficient, high efficiency way to solicit with their business solution. Day by day more and more academicians, research scholars, IT industries, SME (Small and Medium Enterprises) gerents is nomadic themselves in the direction of hot chocolaty flavour moorland of "Cloud Computing". As more and more enterprises, government agencies and companies started to explore cloud computing, security issues came out as a gigantic gamut, as every individual preferred to work on a safe environment where privacy and security of their data is a major concern. This research analyses and highlights the model of the cloud computing like service model, virtue of cloud computing, contingency of cloud and application. It also help in better understanding of cloud computing, explain how it works, the different application of Cloud computing, Cloud computing concepts, technical characteristics of cloud computing, online resource management and challenges that Cloud Service Providers (CSP) or vendors face during cloud engineering for providing better resource management of servers resources. This work use bound based algorithms to manage computing resources behind the data canter of cloud work stations while the resource requisite of the cloud application changes time to time. Cloud Computing is a technology that allows anyone to use application software and computing services on demand at anytime, anywhere and any place through internet

The datacenter is the collection of servers where the application is housed to subscribe them by cloud users. It might be a big room in the building anywhere in the world that users access via the Internet. A growing trend in the IT world is virtualization of servers. That is, software can be installed in a server which is allowing to be created multiple instances of virtual servers to be allocated for the users. An end user subscribed different services and connects to the datacenter to interact with the services. A datacenter may be situated at large distance to the clients. Now-a-days virtualization concept is use to create multiple virtual server instances.

III. LITRATURE SURVEY

1. As per Mr. Alireza Yazdanpanah [1], A new compression ratio prediction algorithm for hardware implementations of LZW data compression: For more data storage space continues to grow at an unprecedented pace, the need for real time data compression systems becomes more prominent. One of the well-known methods among lossless data compression algorithms is LZW. In this paper, a new prediction algorithm has been introduced that is able to predict whether or not a data block is compressible with the LZW method. Furthermore, the prediction algorithm provides a reasonably good estimation of the final compression ratio, hence helping the storage system.
2. As per Mr. Zhou Yan-li [2], Improved LZW algorithm of lossless data compression for WSN: Wireless Sensor Network (WSN) nodes are powered by limited battery and communication consumes most of the energy. Therefore, it's not appropriate to transmit data directly in the networks while compressing data before sending is an effective method to save energy for nodes. This paper proposes an improved lossless data compression algorithm for WSN nodes. In this algorithm, the span of data to be compressed is reduced by calculating increment between two adjacent data of sample sequence

3. As per Mr. S. Kwong and Y. F. Ho [3], An Lidar data compression method based on improved LZW and Huffman algorithm: Lidar raw echo data has characteristics such as huge data quantity, strong discreteness and unpredictability According to the construction of Lidar monitoring network of atmospheric environment [3], the existing network cannot provide enough bandwidth to transmit Lidar data in real time. We propose a novel hybrid lossless compression algorithm to reduce the transmission amount, namely the probability statistics lossless compression algorithm base on the improved LZW (Lempel-Ziv-Welch), which combines Huffman coding.

4. As per Mr. Nan Zhang, Tao Tao, Ravi Vijaya Satya, and Amar Mukherjee [4]. Improving LZW (data compression algorithm): The Lempel-Ziv-Welch compression algorithm is widely used because it achieves an excellent compromise between compression performance and speed of execution. A simple way to improve the compression without significantly degrading its speed is proposed, and experimental data show that it works in practice. Even better results are achieved with additional optimization of 'phasing in' binary numbers.

5. As per Mr. W. Kinsner, R. H. Greenfield [5], The Lempel-Ziv-Welch (LZW) data compression algorithm for packet radio: The authors present the adaptive Lempel-Ziv-Welch general-purpose algorithm and its implementation suitable for packet radio, telephone transmission, and archival storage, while the statistical variable-length Huffman technique compresses text by 20%, the LZW technique can compress data (text, numeric, mixed, and bit-mapped images) by 40 to 60%. The adaptive LZW algorithm has very simple logic, leading to inexpensive and fast implementations

IV. PROBLEM IDENTIFICATION

Data communication is that the most energy-consuming task for sensing element nodes in WSN or at any Communication Network .because the energy supply, process ability, storage capability and communication information measure of nodes and Network are restricted, it's not applicable to transfer original measured knowledge directly usually. Fortuitously, trading computation for communication could save energy. One research shows that the energy consumption for capital punishment 3000 thousands directions is capable the energy consumption for transmittal one thousand bit over a distance of one hundred m by radio. Therefore, press knowledge before causing is a good way to observe use of nodes' restricted resources and reduce energy consumption of nodes and networks. LZW compression became the primary wide used universal knowledge compression technique on computers. when the invention of LZW there square measure a lot of enhancements and improvement tired LZW for knowledge compression that's mentioned during this section. LZW compression works best for files containing a lot of repetitive knowledge particularly for text and monochrome pictures. The LZW algorithmic rule uses dictionary D whereas cryptography and encryption. LZW compression uses a code table common alternative is to supply 4096 entries within the table.

V. PROPOSED WORK

A primary wordbook contains less variety of entries than secondary wordbook. We have a tendency to assign smaller code size to primary and bigger code size to secondary dictionary. Primary wordbook contains the often used entries, presumptuous they're any used and therefore probability of finding a match in primary word book is quite secondary wordbook

* We have a tendency to assign larger code size to secondary word book, thus having additional entries than primary wordbook

*Secondary wordbook will address all entries in primary wordbook. Once primary wordbook gets filled up, and then on top of replacement strategy is called. Here the removed node from primary dictionary is further to secondary wordbook. This theme initialized primary wordbook with normal character set. Once input string is searched in primary wordbook, if the input string is in primary dictionary, code is transmitted and if not, longest string that matches the input string and character of input stream that provides match is further to dictionary. If free house is accessible in primary wordbook then it is concatenated with the matched string and mismatch character is searched in secondary dictionary if mismatched character is gift there, than it's transferred to primary wordbook and if browse then on top of replacement strategy is employed to supply space to mismatches character. a bigger index of length 1+ log2 (size (dprim)) bits is generated, otherwise associate degree index of length 1+log2 (size (d sec)) is generated .A single bit is employed to differentiate lengths of tokens transmitted. dprim denotes primary dictionary and d sec denotes secondary wordbook and size is code size of wordbook. When it finds single bytes, it sends them in uncompressed mode and once it finds sequences of bytes, it sends them in compressed mode to output. In uncompressed mode knowledge is distributed because it is and in compressed mode wherever knowledge is compressed to LZW knowledge compression.

VI. PROPOSED METHODOLGY

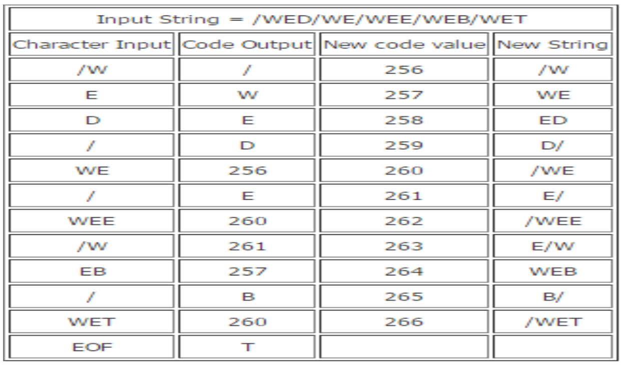
In our System, We have used LZW formula technique to beat house quality and Storage management. Following are the modules of our system:

If you were to require a glance at nearly any file on a pc, character by character, you would notice that there square measure several continual patterns. LZW may be a information compression technique that takes advantage of this repetition. The first version of the tactic was created by Lempel and Ziv in 1978 (LZ78) and was additional refined by Welch in 1984, therefore the LZW signifier. LZW may be a "dictionary"-based compression formula. Thus, to inscribe a substring, solely one code range, such as that substring's index within the wordbook, has to be written to the computer file. though LZW is usually explained within the context of pressing text files, it may be used on any form of file. However, it usually performs best on files with continual substrings, like text files.

Compression: The LZW compression formula in its simplest type is shown in below .a fast examination of the formula shows that LZW is often making an attempt to output codes for strings that are already acknowledged. and every time a brand new code is output, a brand new string is supplemental to the string table.

```
STRING = get input character

WHILE there are still input characters DO

   CHARACTER = get input character

   IF STRING+CHARACTER is in the string table then

           STRING = STRING+ character

  ELSE

           Output the code for STRING

      add STRING+CHARACTER to the string table

           STRING = CHARACTER

             END of IF

     END of WHILE

     Output the code for STRING
```

A sample string wont to demonstrate the algorithmic rule is shown in Figure 3.2. The input string could be a list of English words separated by the '/' character. Stepping through the beginning of the algorithmic rule for this string, you'll see that the primary experience the loop, a check is performed to check if the string "/W" is within the table. Since it is not, the code for '/' is output, and also the string "/W" is other to the table. Since we've 256 characters already outlined for codes 0-255, the primary string definition may be assigned to code 256. once the third letter, 'E', has been browse in, the second string code, "WE" is other to the table, and also the code for letter 'W' is output. This continues till within the second word, the characters '/' and 'W' square measure browse in, matching string number 256. During this case, the code 256 is output, and a three character string is other to the string table. The method continues till the string is exhausted and every one of the codes are output.

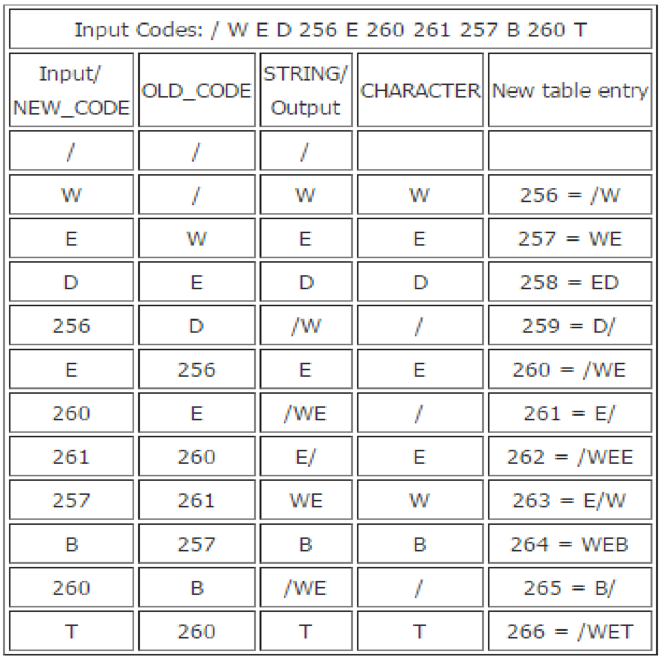


Input String = /WED/WE/WEE/WEB/WET			
Character Input	Code Output	New code value	New String
/W	/	256	/W
E	W	257	WE
D	E	258	ED
/	D	259	D/
WE	256	260	/WE
/	E	261	E/
WEE	260	262	/WEE
/W	261	263	E/W
EB	257	264	WEB
/	B	265	B/
WET	260	266	/WET
EOF	T		

Fig 3.2: Compression Process

Decompression:

The companion algorithmic rule for compression is that the decompression algorithmic rule. It must be ready to take the stream of codes output from the compression algorithmic rule, and use them to precisely recreate the input stream. One reason for the potency of the LZW algorithmic rule is that it doesn't have to be compelled to pass the string table to the decompression code. The table will be designed precisely because it was throughout compression, victimization the input stream as knowledge. this can be potential as a result of the compression algorithmic rule perpetually outputs the STRING and CHARACTER elements of a code before it uses it within the output stream. This implies that the compressed knowledge isn't burdened with carrying an oversized string translation table

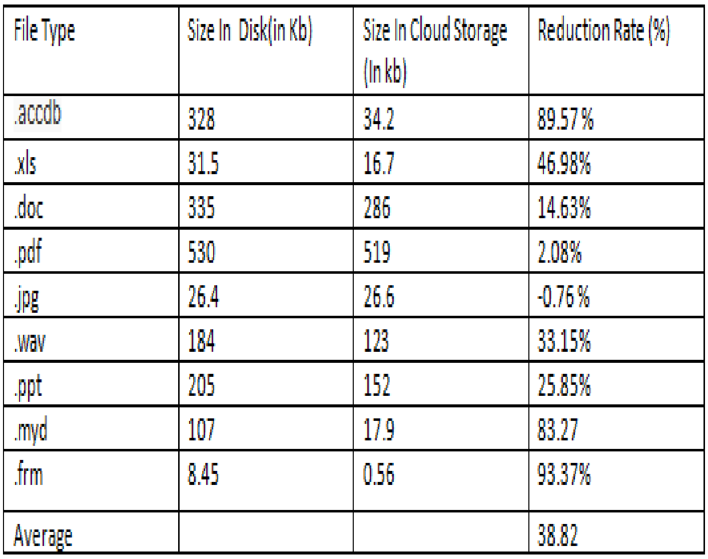


Input Codes: / W E D 256 E 260 261 257 B 260 T				
Input/ NEW_CODE	OLD_CODE	STRING/ Output	CHARACTER	New table entry
/	/	/		
W	/	W	W	256 = /W
E	W	E	E	257 = WE
D	E	D	D	258 = ED
256	D	/W	/	259 = D/
E	256	E	E	260 = /WE
260	E	/WE	/	261 = E/
261	260	E/	E	262 = /WEE
257	261	WE	W	263 = E/W
B	257	B	B	264 = WEB
260	B	/WE	/	265 = B/
T	260	T	T	266 = /WET

Fig 3.2: Decompression Process

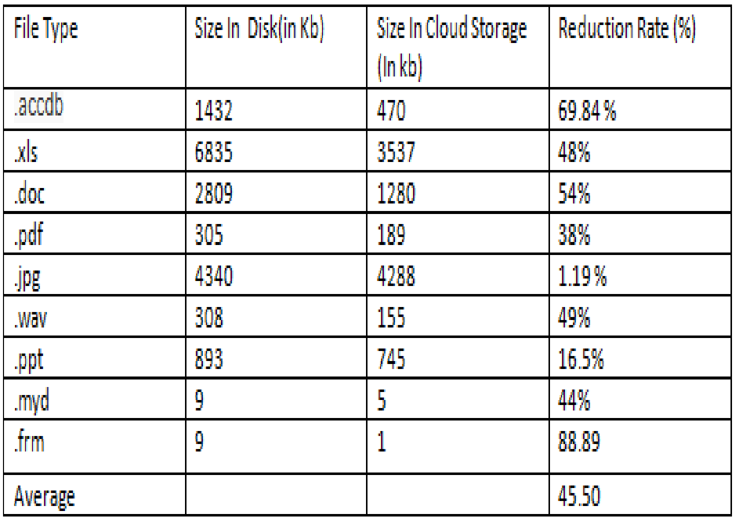
VII. RESULT COMPRESSION

All experiments done on a a pair of.20 GHz Intel (R) Celeron (R) 900 mainframe equipped with 3072KB L2 cache and 2GB of main memory. The machine had no alternative vital mainframe tasks running and solely one thread of execution was used. The OS is Windows XP SP3 (32 bit). All programs are compiled victimisation java version jdk1.6.0_13. the days are recorded in nanoseconds. The time taken of every rule is calculated victimisation the tool compression time quality authority and therefore the graphs are aforethought with MS surpass. All information structures reside in main memory throughout computation.



File Type	Size In Disk(in Kb)	Size In Cloud Storage (In kb)	Reduction Rate (%)
.accdb	328	34.2	89.57%
.xls	31.5	16.7	46.98%
.doc	335	286	14.63%
.pdf	530	519	2.08%
.jpg	26.4	26.6	-0.76%
.wav	184	123	33.15%
.ppt	205	152	25.85%
.myd	107	17.9	83.27
.frm	8.45	0.56	93.37%
Average			38.82

Figure (A) Comparative analysis of Existing Algorithm



File Type	Size In Disk(in Kb)	Size In Cloud Storage (In kb)	Reduction Rate (%)
.accdb	1432	470	69.84%
.xls	6835	3537	48%
.doc	2809	1280	54%
.pdf	305	189	38%
.jpg	4340	4288	1.19%
.wav	308	155	49%
.ppt	893	745	16.5%
.myd	9	5	44%
.frm	9	1	88.89
Average			45.50

Figure (B) Comparative analysis of LZW Algorithm

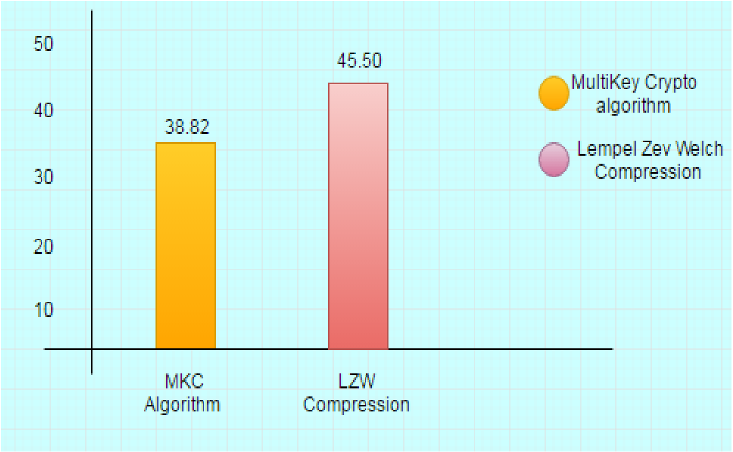


Figure. (C) Comparative analysis graph of LZW Algorithm and MKC

VIII. BENEFITS

Benefit of this scheme is reduction in transmitted number of bits over the communication channel. This results in gain/improvement in compression ratio.

1. LZW algorithm works extremely well with repeated data streams or strings of English text,
2. The original LZW is implemented on4k dictionary is, Defining larger dictionary of course results in greater string capacity.
3. If smaller text files containing large entries of the standard character set, then expansion of the data take place in spite of compression by assignment of 12 bit index to 8 bits data value and thus more bits are transferred over the communication channel. With help of LZW large data can be compressed

IX. COST REDUCTION:

In these days, all the businesses are victimisation the construct of cloud computing. It's a brand new technology. It works supported Pay on Demand model. This model suggests that, during which quantity user needed the resources for a particular time to finish a task, user ought to pay some cash just for that a lot of your time. Attributable to this technology several edges are applicable to urge the higher profit within the market. Profit in terms of your time, cost, load reconciliation, storage then on. During this technology, all applications run on a virtual platform and every one the resources are distributed among these virtual machines. Every and each application is totally different and is freelance. To live the entire allocation cost of every applications, each individual use of resources (like CPU value, memory value, I/O cost, etc.) Should be calculated. Once the entire allocation cost of every individual resources are calculated, correct cost and profit analysis supported it may be obtained, compared to those of the normal programming ways. Attributable to ancient programming construct over budgeting and over valuation is obtained within the market.

STORAGE COST:

As we are able to say the entire Cloud Storage compute the price of cloud meantime, if the minimum storage in cloud, value for resources needed for Cloud would even be minimum .So as If we tend to Compress the The Cloud knowledge storage ,Cost of cloud can minimize mechanically.

Storage value is formed of the price to store VM pictures within the straightforward Storage Service (S3, associate degree object-based storage system), and therefore the value of storing computer file within the Elastic Block Store (EBS, a SAN-like block-based storage system). Each S3 and EBS use fastened monthly charges for the storage of knowledge, and charges for accessing the information, which may vary per the appliance. The rates for fastened charges square measure $0.15 per GB/month for S3 and $0.10 per GB/month for EBS. The most distinction in value is that EBS is charged supported the quantity of disk storage requested, whereas S3 solely charges for what's used. To boot, EBS may be hooked up to just one computing instance, whereas S3 may be access at the same time by any variety of instances. The variable charges for knowledge storage square measure $0.01 per 1,000 place operations and $0.01 per 10,000 GET operations for S3, and $0.10 per million I/O operations for EBS. The 32-bit image used for the experiments during this paper was 773 MB, compressed, and therefore the 64-bit image was 729 MB, compressed, for a complete fixed charge of $0.22 per month. The fastened monthly value of storing computer file for the 3 applications is shown in Table III. For the experiments represented during this study, there have been four,616 S3 GET operations and a couple of,560 S3 place operations for a complete variable value of roughly $0.03. Additionally, there have been three.18 million I/O operations on EBS for a complete variable value of $0.30.

X. ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the communicating editor for their useful comments which improved the presentation of this paper. In addition, they would like to thank for his assistance in implementing the design layout and doing the post-layout simulation of the chip.

XI. CONCLUSION AND FUTURE WORK

The implementation of initial 2 enhancements could also be a challenging task. Replacement strategy eliminates the frequent flushing of the reference book by replacement previous entries with the newer ones. 2 level reference book can increase vary of dictionaries to prevent frequent flushing of reference book. Implementation of these schemes leads to improvement in time interval and efficiency of LZW info compression at expense of plenty of memory house and package quality. Third sweetening is implemented and it's seen that encoder with reduced vary of bits achieves higher compression .After compression of Cloud Storage We can Minimize the Cost Of the Cloud Storage using More Storage for other Stuff too.

When comparison the performance of the higher than data structure, the LZW enchained hash table implementation (for Encoding) offers higher result than Linear array and BST data structure implementation, however once comparison the LZW secret writing formula the Linear array is best than Linear array and BST implementation and this Linear array implementation of Linear array LZW secret writing formula virtually offers to kenish procedure price or it doesn't would like any more optimization so as to cut back the procedure price. This work is often additional increased and enlarged for the authentication of compression techniques to get optimum accuracy in time.

Main purpose of cost minimization is achieved through storage optimization.

XII. REFERENCES

[1] Alireza Yazdanpanah, "A new compression ratio prediction algorithm for hardware implementations of LZW data compression", 1st ed Multimedia Processing Laboratory, School of Electrical and Computer Engineering, University of Tehran, Iran.

[2] Zhou Yan-li, "Cloud Improved LZW algorithm of lossless data compression for WSN", School of Information Science and Engineering, Central South University, Changsha, Hunan, China.

[3] S. Kwong and Y. F. Ho "A statistical Lempel-Ziv compression algorithm for personal digital assistant(PDA)" IEEE Transactions on Consumer Electronics, Vol. 47, No. 1, FEBRUARY 2001 pp 154 161

[4] Nan Zhang, Tao Tao, Ravi Vijaya Satya, and Amar Mukherjee " An Lidar data compression method based on improved LZW and Huffman algorithm " 2010 International Conference on Electronics and information Engineering Year: 2010, Volume: 2

[5] W. Kinsner, R. H. Greenfield " The Lempel-Ziv-Welch (LZW) data compression algorithm for packet radio" – WESCANEX '91 'IEEE Western Canada Conference on Computer, Power and Communications Systems in a Rural Environment'

[6] Nirali Thakkar and Malay Bhatt Two-Stage Algorithm for Data Compression Proc. of the Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering 2012 ISBN: 978-981-07-1847-3 pp 350-354

[7] Liu Feng and Gao Dong-Mei DNA Algorithm of Verifiable Secret Sharing ETP International Conference on Future Computer and Communication -2009 pp 244-246

[8] C Clark et al.,"Live Migration of Virtual Machines" NSDI'05 Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation ,2005.

[9] E Arzuaga, D R Kaeli, "Quantifying load imbalance on virtualized enterprise servers." In WOSP/SIPEW '10: Proceedings of the first joint WOSP/SIPEW international conference on Efficiency engineering, ACM, 2010.

[10] R. Santhosh and T. Ravichandran, "A Survey on Cloud-Based Scheduling Algorithms", *International Journal of Communications and Engineering (IJCE),* Volume 01– No.1, Issue: 01 May2013.