Entropy of Data Compression Using Object Oriented Data Warehousing

Dr. (Mrs Pushpa Suri)[1] and Mrs Meenakshi Sharma[2]

[1] Associate professor, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, India

[2] Assistant Professor, Department of Computer Science and Engineering, Haryana College of Technology and Management Kaithal, India

ABSTRACT

A data warehouse is a repository of subjectively selected and adapted operational data which can successfully answer any ad hoc, statistical, complex or analytical queries. Data warehousing technology is becoming essential for effective business intelligence, business strategy formulation and implementation in a globally competitive environment where in larger and larger amounts of data are required to be processed faster and faster for comprehension of its real meaning and impact [4].Data warehousing has been around in some form or another since the inception of data storage, people were never able to exploit the information that was wastefully sitting on a tape somewhere in a back room. Organizations across the country and around the world are seeking expertise in this exploding field of data organization and manipulation. It is not a surprise, really, that business users want to get a better look at their data. Today, business opportunities measure in days, instead of months or years, and the more information empowering an entrepreneur or other business person, the better the chances of beating a competitor to the punch with a new product or service. In this paper we have discussed the efficiency of the data warehousing packages so that less CPU time and less Memory is consumed.

Keywords: *Entropy, Data warehousing, object oriented, Data compression*

1. INTRODUCTION

A data warehouse is a mechanism for data storage and data retrieval. Data can be stored and retrieved with a multidimensional structure--hypercube or relational, a star schema structure or several other data storage techniques. The task of transitioning from a procedural mindset to an object-oriented paradigm can seem overwhelming; however, the transition does not require developers to step into another dimension or go to Mars in order to grasp a new way of doing things. In many ways, the object-oriented approach to development more closely mirrors the world we've been living in all along: We each know quite a bit about objects already. It is that knowledge we must discover and leverage in transitioning to object-oriented tools and methodologies. Our research has been from a different point of view – our primary motivating factor is to show how existing applications can be enhanced using object –oriented Technology. Like Many new ideas, object –oriented programming does not have a universally accepted definition [1, 2]. Ideas on the subject do, however, seem to be converging the “best” definition that we have seen to date is “object-oriented = object + classes + inheritance” [3]. OOP can also be defined as an extension of the idea of abstract data type. The task of transitioning from a procedural mindset to an object – oriented paradigm can seem overwhelming: however, the transition does not require developers to step into another dimension or go to grasp a new way of doing thing. In many ways, the object oriented approach to development more closely mirrors the world we’ve been living in all along. [8]. we each know quite a bit about objects already. It is that knowledge we must discover and leverage in transitioning to object- oriented tools and methodology.

2. ENTROPY IN DATA COMPRESSION

Data compression is of interest in business data warehousing, both because of the cost savings it offers and because of the large volume of data manipulated in many business applications. The types of local redundancy present in business data files include runs of zeros in numeric fields, sequences of blanks in alphanumeric fields, and fields which are present in some records and null in others.[1],[2] Run length encoding can be used to compress sequences of zeros or blanks. Null suppression may be accomplished through the use of presence bits. Another class of methods exploits cases in which only a limited set of attribute values exist. Dictionary substitution entails replacing alphanumeric representations of information such as bank account type, insurance policy type, sex, month, etc. by the few bits necessary to represent the limited number of possible attribute values. The problem of compressing digital data can be decoupled into two subproblems: modeling and entropy coding. Whatever the given data may represent in the real world, in digital form it exists as a sequence of symbols, such as bits. The modeling problem is to choose a suitable symbolic representation for the data and to predict for each symbol of the representation the probability that it takes each of the allowable values for that symbol. The entropy-coding problem is to code each symbol as compactly as possible, given this knowledge of probabilities. (In the realm of lossy compression, there is a third subproblem: evaluating the relative importance of various kinds of errors.)

For example, suppose if it is required to transmit messages composed of the four letters a, b, c, and d. A straightforward scheme for coding these messages in bits would be to represent a by \00", b by \01", c by \10" and d by \11". However, suppose if it is known that for any letter of the message (independent of all other letters), a occurs with probability .5, b occurs with probability .25, and c or d occur with probability .125 each. Then a shorter representation might be chosen for a, at the necessary cost of accepting longer representations for the other letters. a could be represented by \0", b by \10", c by \110", and d by \111". This representation is more compact on average than the first one; indeed, it is the most compact representation possible (though not uniquely so). In this simple example, the modeling part of the problem is determining the probabilities for each symbol; the entropy-coding part of the problem is determining the representations in bits from those probabilities; the probabilities associated with the symbols play a fundamental role in entropy coding. One well-known method of entropy coding is Huffman coding, which yields an optimal coding provided all symbol probabilities are integer powers of .5. Another method, yielding optimal compression performance for any set of probabilities, is arithmetic coding. In spite of the superior compression given by arithmetic coding, so far it has not been a dominant presence in real data-compression applications. This is most likely due to concerns over speed and complexity, as well as patent issues; a rapid, simple algorithm for arithmetic coding is therefore potentially very useful. An algorithm which allows rapid encoding and decoding in a fashion akin to arithmetic coding is known as the Q-coder. The QM-coder is a subsequent variant. However, these algorithms being protected by patents, new algorithms with competitive performance continue to be of interest. The ELS algorithm is one such algorithm

The ELS-coder works only with an alphabet of two symbols (0 and 1). One can certainly encode symbols from larger alphabets; but they must be converted to a two-symbol format first. The necessity for this conversion is a disadvantage, but the restriction to a two-symbol alphabet facilitates rapid coding and rapid probability estimation.

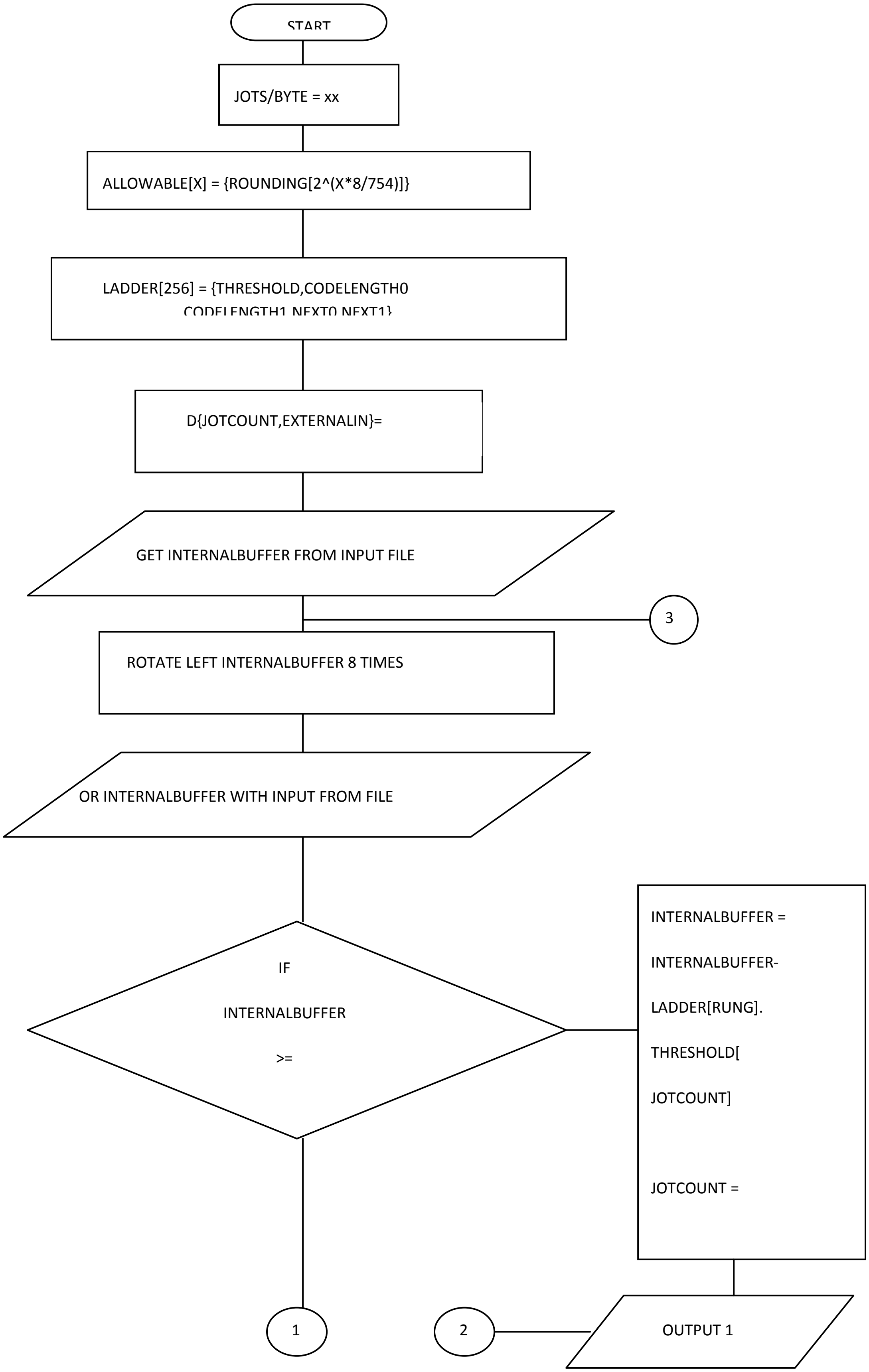
The ELS-coder decoding algorithm has already been described. The encoder must use its knowledge of the decoder's inner workings to create a data stream which will manipulate the decoder into producing the desired sequence of decoded symbols. As a practical matter, the encoder need not actually consider the entire coded data stream at one time. One can partition the coded data stream at any time into three portions; from end to beginning of the data stream they are: preactive bytes, which as yet exert no in seuence over the current state of the decoder; active bytes, which affect the current state of the decoder and have more than one consistent value; and postactive bytes, which affect the current state of the decoder and have converged to a single consistent value. Each byte of the coded data stream goes from preactive to active to postactive; the earlier a byte's position in the stream, the earlier these transitions occur. A byte is not actually moved to the external _le until it becomes postactive. Only the active portion of the data stream need be considered at any time. Since the internal buffer of the decoder contains two bytes, there are always at least two active bytes. The variable backlog counts the number of active bytes in excess of two. In theory backlog can take arbitrarily high values, but higher values become exponentially less likely.

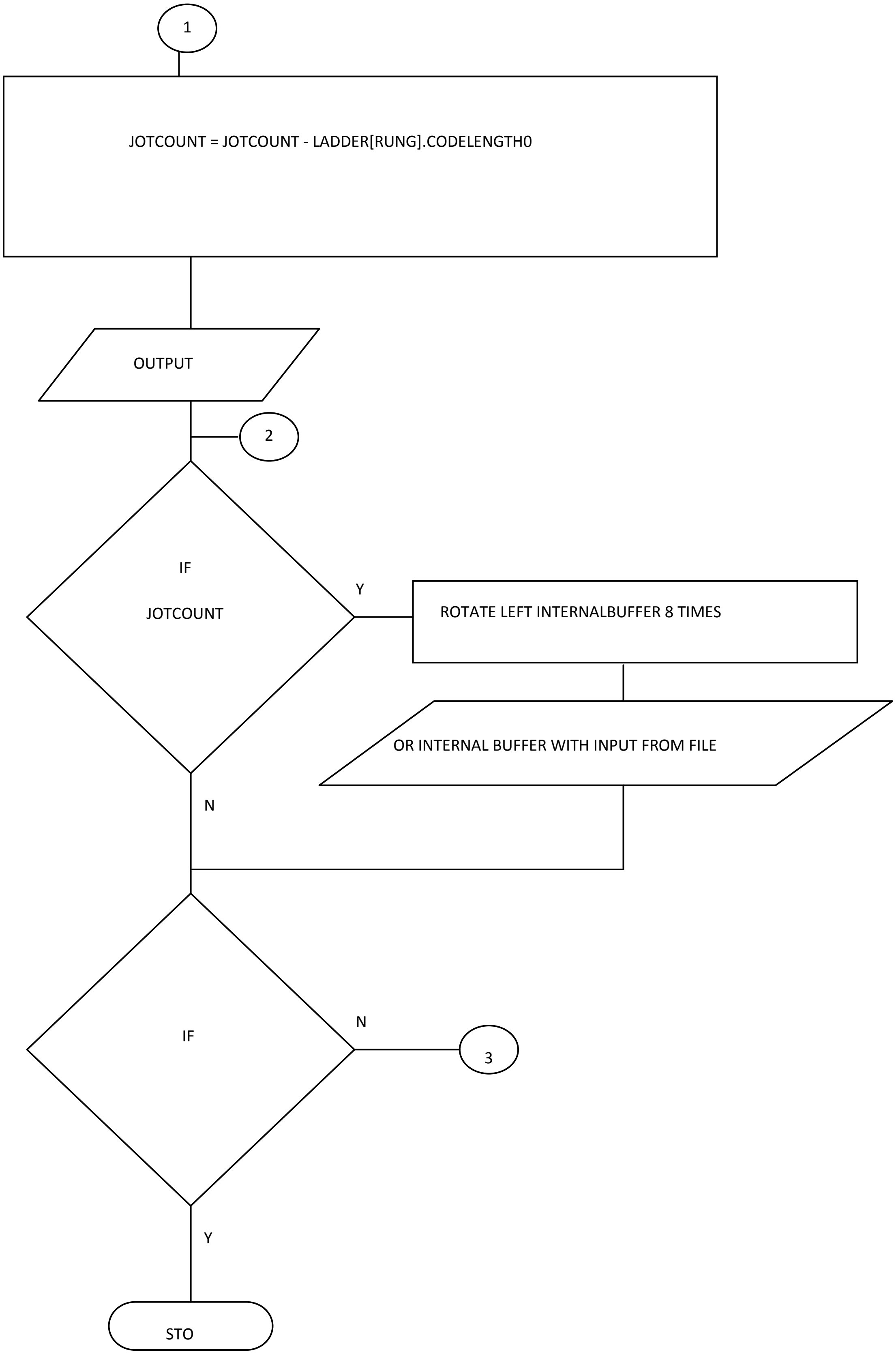
3. METHODOLOGY

Following steps will be taken for the future work

1. Creation of different sizes of databases in oracle
2. Employment of object oriented programming for compression using datawarehousing
3. Further compression of database csv files using C++
4. Comparison of time taken and compression efficiency for different sizes of databases.

4. FLOW CHART OF DATA COMPRESSION


START
JOTS/BYTE = xx
ALLOWABLE[X] = {ROUNDING[2^(X*8/754)]}
LADDER[256] = {THRESHOLD,CODELENGTH0
CODELENGTH1 NEXT0 NEXT1}
D{JOTCOUNT,EXTERNALIN}=
GET INTERNALBUFFER FROM INPUT FILE
3
ROTATE LEFT INTERNALBUFFER 8 TIMES
OR INTERNALBUFFER WITH INPUT FROM FILE
IF
INTERNALBUFFER
>=
INTERNALBUFFER =
INTERNALBUFFER-
LADDER[RUNG].
THRESHOLD[
JOTCOUNT]
JOTCOUNT =
1
2
OUTPUT 1


1
JOTCOUNT = JOTCOUNT - LADDER[RUNG].CODELENGTH0
OUTPUT
2
IF
JOTCOUNT
Y
ROTATE LEFT INTERNALBUFFER 8 TIMES
OR INTERNAL BUFFER WITH INPUT FROM FILE
N
IF
N
3
Y
STO

5. CONCLUSION

A data warehouse is an essential component to the decision support system. The traditional data warehouse provides only numeric and character data analysis. But as information technologies progress, complex data such as semi-structured and unstructured data become vastly used[2],[3]. Data Compression is of interest in business data warehousing, both because of the cost saving it offers and because of the large volume of data manipulated in many business application. The entropy is used in many areas such as image processing, document images. But in our research we used the entropy in object oriented data warehousing. Creation of different sizes of databases in oracle. Employment of object oriented programming for compression using datawarehousing. Further compression of database .csv files using C++. Comparison of time taken and compression efficiency for different sizes of databases.

6. REFERENCES

1. Wei-Chou Chen; Tzung-Pei Hong; Wen-Yang Lin; , "Using the compressed data model in object-oriented data warehousing ," Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on , vol.5, no., pp.768-772 vol.5, 1999
2. Wei-Chou Chen; Tzung-Pei Hong; Wei-Yang Lin; , "A composite data model in object-oriented data warehousing," Technology of Object-Oriented Languages and Systems, 1999. TOOLS 31. Proceedings , vol., no., pp.400-405, 1999
3. Shieh, J. C.; Lin, H. W.,"The Novel Model of Object-Oriented Data Warehouses", Workshop on Databases and Software Engineering, 2006
4. Chen Wei-Chou, Hong Tzung-Pei (2), Lin Wen-Yang "Three maintenance algorithms for compressed object-oriented data warehousing"
5. Boqiang Huang; Yuanyuan Wang; Jianhua Chen; , "2-D Compression of ECG Signals Using ROI Mask and Conditional Entropy Coding," Biomedical Engineering, IEEE Transactions on , vol.56, no.4, pp.1261-1263, April 2009
6. Gong, Y.; Fan, M.K.H.; Huang, C.-M.; , "On entropy-constrained residual vector quantization design," Data Compression Conference, 1999. Proceedings. DCC '99 , vol., no., pp.526, 29-31 Mar 1999
7. De, I.; Sil, J.; , "Wavelet entropy based no-reference quality prediction of distorted/decompressed images," Computer Engineering and Technology (ICCET), 2010 2nd International Conference on , vol.3, no., pp.V3-245-V3-250, 16-18 April 2010
8. De, I.; Sil, J.; , "ANFIS tuned no-reference quality prediction of distorted/decompressed images featuring wavelet entropy," Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on , vol., no., pp.126-131, 8-10 Oct. 2010
9. Liu, L.; Dong, Y.; Song, X.; Fan, G.; , "An entropy based segmentation algorithm for computer-generated document images," Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on , vol.1, no., pp. I- 541-4 vol.1, 14-17 Sept. 2003
10. Tu, C.; Tran, T.D.; , "Context-based entropy coding of block transform coefficients for image compression," Image Processing, IEEE Transactions on , vol.11, no.11, pp. 1271- 1283, Nov 2002
11. Chen, S.; Reif, J.H.; , "Using difficulty of prediction to decrease computation: fast sort, priority queue and convex hull on entropy bounded inputs," Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on , vol., no., pp.104-112, 3-5 Nov 1993
12. Sang Hyun Kim; Rae-Hong Park; , "A novel approach to scene change detection using a cross

entropy ," Image Processing, 2000. Proceedings. 2000 International Conference on , vol.3, no., pp.937-940 vol.3, 2000

13. Scales, A.; Roark, W.; Kossentini, F.; Smith, M.J.T.; , "Lossless Compression Using Conditional Entropy-Constrained Subband Quantization," Data Compression Conference, 1995. DCC '95. Proceedings , vol., no., pp.498, 28-30 Mar 1995
14. Jegou, H.; Guillemot, C.; , "Entropy coding with variable length re-writing systems," Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on , vol., no., pp.1529-1533, 4-9 Sept. 2005
15. Hua Xie; Ortega, A.; , "Entropy- and complexity-constrained classified quantizer design for distributed image classification," Multimedia Signal Processing, 2002 IEEE Workshop on , vol., no., pp. 77- 80, 9-11 Dec. 2002