Gbits/s lossless data compression systems

S. Jones, J.L. Nunez, C. Feregrino-Uribe, *S. Mahapatra

Electronic Systems Design Laboratory
Department of Electrical and Electronic Engineering
Loughborough University
Loughborough, Leicestershire, LE11 3TU, U.K.

I. INTRODUCTION

PPMC [1] is a leading lossless data compression algorithm that produces compression ratios better than many dictionary-based compressors. Its power comes from the use of a statistical model and arithmetic coder that together compress data close to the entropy. However, its excessive resource requirements and its complexity have restricted its practical use. Here we present a new algorithm based on PPMC that produces similar compression ratios and is suitable for hardware implementation. It is called *Shift* algorithm, is division free (opposite to PPMC) and allows parallel operations to speed up the compression process.

II. PPMC REVIEW

PPMC is an effective data compression scheme based on Prediction by Partial Matching. It predicts symbols based on the sequence of symbols that immediately precede it, called the *context*. The number of symbols in the longest context determines the order of the model. A prediction is made by the higher order context (if possible), if not, an 'Escape from order' is transmitted and the context is shortened by 1 symbol. The operation is repeated until the symbol is successfully predicted or the model reaches '-1' order where the symbol is transmitted in literal form. The coding of the symbols is performed by arithmetic coding almost optimally.

PPMC predicts symbols based on method C, where symbol and Esc probabilities, P_{S_i} and P_{Esc}, are computed as follows:

$$P_{S_i} = \frac{c_{s_i}}{t+k} \text{ and } P_{Esc} = \frac{k}{t+k} \qquad (1)$$

where

c_{s_i} = symbol frequency count
k = number of different symbols followed by the present context
t = total number of symbols predicted by the present context

After the prediction of every symbol, the model is updated adding the new symbol to the applicable contexts and increasing their *total* accounts by 1. If the symbol was not new then its frequency count increases by one as well as the context *total* account.

In PPM [2] style models, the longer the context the better the compression, however, the huge amount of space required, even for a model order 3, makes almost impossible to have a practical compression model. A further disadvantage for its hardware implementation is the division operation required to compute every symbol's probability. In next section it is showed how Shift model besides compressing similarly to PPMC reduces model storage needs and is also suitable for hardware implementation requiring very simple arithmetic operations.

III. SHIFT MODEL

Shift model was developed with the aim of implementing a PPM model in hardware. It consists on predicting symbols based on statistics of the data. It maintains a dictionary of data where the most

* On leave from Regional Engineering College, Rourkela, Orissa, India.

recent symbols are stored following their contexts. It predicts symbols as PPMC although both models consider *total*, *t* in (1), in different ways.

In Shift model a *fixed amount of tokens* plays the role of *total* in PPMC and as it suggests, it is kept constant along the compression process. Shift model considers *Esc* as any other symbol that may be added to the dictionary. It represents the number of different symbols 'seen' following the same context, as *k* in PPMC. This simplifies the calculation of the symbol probabilities in Shift model due that is not longer necessary to compute *k*.

Esc symbol, the first one 'predicted' by any context, is assigned a *fixed amount of tokens*, equivalent to the *total* in PPMC. Later, this amount of tokens is distributed among the symbols that follow the context, allowing then the model to adapt.

At updating the model all symbols (or most of them) predicted by the current context contribute with some tokens for the new symbol. The amount diminished from them is proportional to the number of tokens they have. A denominator divides their frequency counts and the results are subtracted from them, added and later assigned to the incoming symbol.

Diminishing all the symbols when a new one comes in have the same effect in Shift model as increasing the *total* in PPMC. In Shift model when all symbols contribute with some tokens for the incoming symbol they are reducing its probability of occurrence while in PPMC increasing the total causes all symbols diminish their probabilities as well.

Having a *fixed number of tokens* has the advantage of avoiding the computation of *total* or its storage and updating. Furthermore, the division operation required to compute the probability is no longer necessary which simplifies enormously the complexity of the model.

In shift model the *number of tokens* and the *denominator* are the key parameters mainly in 0-order model. They are the essential part that dictates the compression ratio that may be obtained from Shift model. In higher order models (from 1 order) there are some other important parameters explained in the next section.

A. Parameters

Shift model depends on the parameters to get compression ratios similar ones to PPMC model. *Number of tokens* and *denominator* are the main parameters of the model. The former rules the 'flexibility' for the adaptation of the model and the latter the adaptation speed. For studying both parameters their behavior has been observed along the compression process in two stages, the start up and in steady state. Among others, two important characteristics identified in PPMC model are a) Big changes at start up and b) Small changes in steady state. The changes refer to the probabilities assigned to the incoming symbols. The changes may be done as desired if the parameters of the model are chosen appropriately.

A big number of tokens and a big denominator allows the model to adapt completely and produces small changes in steady state. While a small number of tokens and a small denominator are not a good combination since in this way the model is not flexible enough for adapting and always produces big changes in steady state.

To identify how PPMC and Shift models adapt, the probabilities assigned to the incoming symbols have been studied. Next, there is an explanation.

PPMC assigns probabilities to the incoming symbols according to '$1/n$' function where n is the total number of tokens, i.e., the first symbols get a high probability that very soon decreases as symbols come. When less than 1% of the symbols have been processed, the probability assigned to the incoming symbols has almost reached the minimum.

Shift model assigns probabilities according to the *number of tokens* of the symbol divided by the *denominator*. All the results are added and assigned to the incoming symbol. In this way the model achieves similar results to PPMC since symbols with higher frequency counts give more than the ones with a small frequency counts.

For the adaptation of the Shift model 0^{th} order only the parameters *numerator* and *denominator* are required, however, the adaptation of higher order models involves additional parameters that may affect the compression ratio. The number of different contexts of each order in the model may alter the adaptation speed as well as the order of the model.

B. Implementation

The Shift model has been designed with the aim of its hardware implementation. For simplicity it has one dictionary per every order of the model. The dictionary is to be implemented in a CAM array that allows parallel search operations. It is 2K locations in length and each location contains 'dictionary-order' +1 symbols. These sizes have proved to deliver the best compression ratios with the model.

Whenever a symbol comes in, it is codified and the dictionary updated. All symbols predicted by the present context diminish their frequency counts. All these number of tokens goes though an adder's tree where the number of tokens to be assigned to the incoming symbol is obtained. This may be the slower part of the compressor but can be done in parallel with the reading and searching operations for the incoming symbol to speed up the compression process.

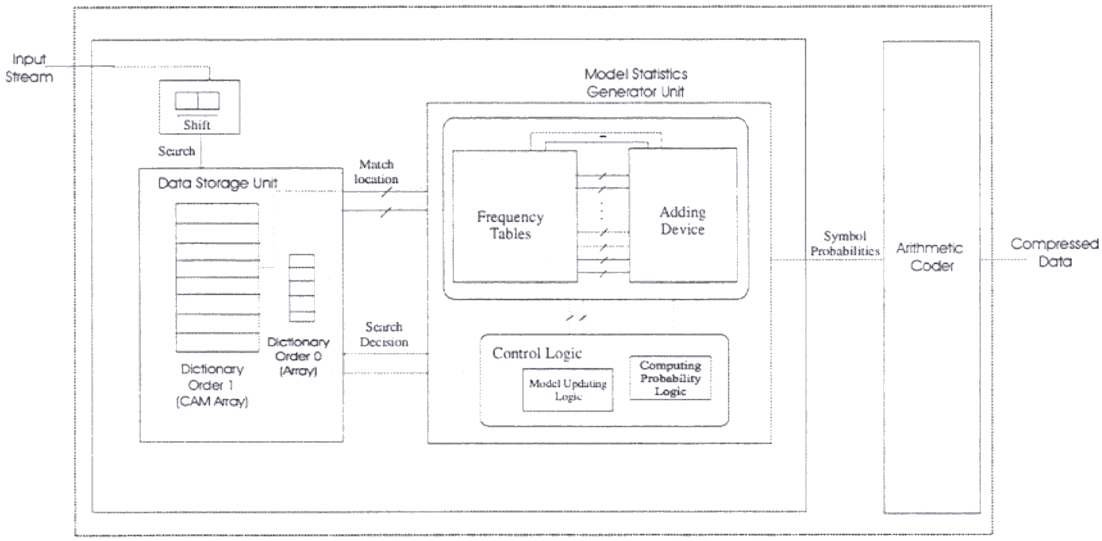

Figure 1. Shift Model Architecture

Shift model simplifies space requirements them. It requires a dedicated dictionary for each order in the model but its size is limited at most to 2K locations.

In terms of complexity, Shift model does not require the complex division operation to compute symbol probabilities nor the arithmetic coding coupled with it. It requires many simple arithmetic operations, subtracts and shifts, that are done in parallel for updating the model. Furthermore, the search operation is done in only one cycle. At the moment further research is being undertaken to identify the possibility of eliminating of the adders tree what may be a remarkable simplification of the model.

IV. RESULTS

Shift model order 0 achieves excellent compression ratios, similar to PPMC model. Figure 2 shows them as the *number of tokens* and the *denominator* varies. It can be seen that the bigger the *denominator* and *number of tokens* the better, until denominator reaches 256. After that the compression ratio starts to grow up again.

A small denominator, 2, assigns big frequency counts to the symbols causing poor compression ratios as can be appreciated in Figure 2. When a symbol is repeated several consecutive times, the 'incoming symbol' gain almost all the tokens, thus, the amount to give to this incoming symbol is 0. This is proof of no adaptation and happens with a large number of tokens as well as with a small one, but is more common when the number of tokens is small.

Due that a big number of tokens allow the model to adapt better compression ratio is obtained. It can be observed as the denominator grows. The compression ratio improves while with a small number of tokens it is variable.

Higher order models do not compress at about the same ratios as PPMC. 1st order model gives 0.4685 and 2nd order 0.4273, what is 5.6 % and 8.7% worst than PPMC respectively. However this compression ratios are not too far and some good solutions may exist. The difference in compression ratios from the excellent ones obtained with order 0 is due to the involvement of other facts as the number of different context present in the order.

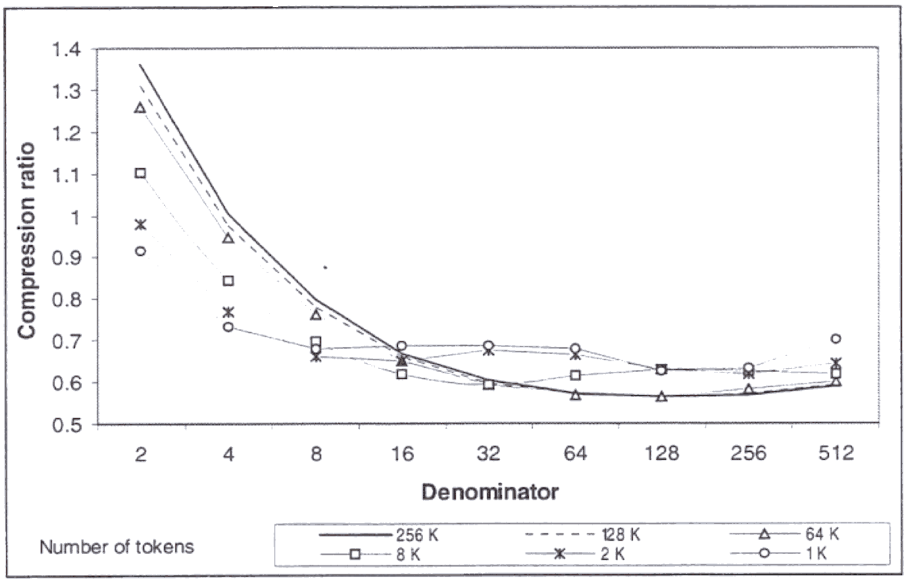

Figure 2. Compression ratio at increasing number of tokens and denominator

The fast adaptation of the model in higher orders may require only 'dividing' at start up by the *denominator* multiplied by 2, what implies only a shift by one bit less. In terms of hardware it does not represent a major change in the architecture of the model and in terms of compression ratios may be as close as 3% of PPMC model. Compressing the entire files, the average compression ratio obtained from Canterbury corpus in 0.4137 with 1st order model and 0.3482 with 2nd order model.

V. CONCLUSIONS

0th order Shift model has proven to produce similar compression ratios to PPMC, higher orders that are affected by other facts as the block size being compressed and the speed of adaptation required by the contexts. Higher orders have showed that adapting *Esc* improves compression ratio considerably while in 0th order is not necessary.

Compression is being done in blocks of 4K symbols that in terms of hardware complexity may be implemented to achieve compression ratios close about 3% to PPMC being not a very significant loss. Compression of the entire files is only about 1% different from PPMC in 1st order and 6.7% in 2nd order.

An alternative model to PPMC has been developed and has showed its compression capacity at achieving similar compression ratios while is suitable for hardware implementation.

VI. REFERENCES

[1] Alistair Moffat, Implementing the PPM Data Compression Scheme, IEEE Transactions on Communications, Vol. 38, No. 11, November 1990.

[2] John G. Cleary and Ian H. Witten, "Data Compression Using Adaptive Coding and partial String Matching", IEEE Transactions on Communications, Vol. COM-32, No. 4, April 1984.

© 1999 The Institution of Electrical Engineers.
Printed and published by the IEE, Savoy Place, London WC2R 0BL, UK.