Fast online predictive compression of radio astronomy data

Benjamin V. Hugo
Department of Computer Science
University of Cape Town
bennahugo@aol.com

ABSTRACT

TODO: add at end of writeup

1. INTRODUCTION

In this report we will investigate the feasibility of compressing radio astronomy data using a predictive compression scheme. All compression techniques build on the central concept of reducing redundant data. The exact definition of this redundancy is of course context dependent. It may take the form of repeated values, clustered values, wasteful encoding processes and many others. We also point out the difference between lossy and lossless compression, as well as online versus offline compression.

In a lossless compression scheme the compression is completely invertible with *no* loss of information after a decompression step is performed. Lossy compression on the other hand discards unimportant data and gives much higher compression ratios than lossless methods. Lossy compression is useful in many instances where subtle changes in data is not considered problematic. Some examples of this are the removal of high frequency data from images, sampling voice data at a lower rate than music or to employ a commonly used technique called *quantization* where data is simply binned into consecutive ranges (or *bins*).

An online compression scheme refers to a process of compressing data on-the-fly or as it is being transmitted on the wire. The results are sent off to subsequent processes such transmission over a network or storage to disk. This is in contrast to to an offline scheme where data is compressed as a separate process which doesn't form part of the primary work flow of a system. An online process are normally considered to be fast enough as not to slow the overall data processing capabilities of a system.

Compression performance is measured as a compression ratio described below [2, p. 10]. A value closer to 0 will indicate a small output file and values greater than 1 will indicate that the algorithm inflated the data instead of shrinking it.

$$\text{Compression ratio} = \frac{\text{size of the output stream}}{\text{size of the input stream}}$$

We will now discuss the relevance of our proposed solution and give a breakdown of the most commonly used compression techniques. We will then discuss a detailed design of our predictor, different implementation strategies along with technical details and a section with results.

2. BACKGROUND

2.1 KAT-7, MeerKAT and the SKA

South Africa and Australia are the two primary hosting countries for the largest radio telescope array in the world, known as the Square Kilometer Array. The SKA will give astronomers the opportunity to capture very high resolution images, over a wide field of view, covering a wide range of frequencies ranging from 70 MHz to 10 GHz. Upon completion in 2024 the array will consist of around 3000 dishes in the high frequency range and thousands of smaller antennae to cover the low frequency band. The South African branch of the SKA will be completed in 3 main phases. Phase 1 is a fully operational prototype 7-dish array called the KAT-7. The second phase, known as the MeerKAT, will consist of approximately 90 dishes to be erected in the central Karoo. The final phase add the remaining dishes and increase the baseline of the telescope to roughly 3000 km.

Due to the high signal sampling rate it is expected that each of these dishes will produce data rates up to 420 GiB/s while the lower frequency aperture arrays will produce up to 16 TiB/s. These rates, coupled with the scale of the SKA, will require a processing facility capable of handling as much as 1 Petabyte of data every 20 seconds, necessitating the need for massive storage facilities. Innovative techniques are required to deal with this complex requirement of high throughput rates while effectively reducing the large storage requirements by means of data compression.

2.2 Overview of data compression techniques

There are considered to be 4 broad categories of compression techniques [2]. These are some basic methods, Lempel-Ziv methods, statistical methods and transforms.

2.2.1 *Basic methods*

The more intuitive methods include commonly employed methods such as Run-Length Encoding (RLE) which, simply put encodes runs of characters using some reserved character and a number indicating the length of the run.

Another basic technique which is particularly relevant for application on numerical data a predictive compression scheme. Such a compression scheme encodes the difference between each predicted succeeding value and the actual succeeding value. This can be quite successfully employed to compress data generated from time series [1].

2.2.2 *Lempel-Ziv methods*

Also commonly referred to as LZ or dictionary methods is a class of algorithms with many variants and is one of the more popular adaptive techniques in modern compression utilities. In their simplest form these methods normally uses both a search and a lookahead buffer to encode recurrent phrases using fixed-size codes. An adaptive compression technique is useful in circumstances where the probability distribution of the underlying dataset is not known in advance or may change over time. One example of such an LZ method is the GNU compression utility Gzip which implements the Deflate algorithm.

2.2.3 *Statistical methods*

This class of algorithms normally uses variable length codes to achieve an optimal (or near optimal) encoding of dataset. In information theory this optimal encoding is described as an entropy encoding. As the name may suggest the techniques uses the probability of occurrence to assign shorter codes to frequently occurring values. The class of statistical methods include two widely employed techniques known as Huffman and Arithmetic coding respectively.

2.2.4 *Transforms*

As the name suggest it can be useful to transform a dataset from one form to another in order to exploit its features for the purposes of compression. Such transformations includes, for example, wavelet transforms. As the name suggests a wavelet is a small wave-like function that is only non-zero over a very small domain and can be used to represent, for example, the high frequency components in an image (JPEG2000 and DjVu are popular formats using wavelet transforms). The coefficients within this transformation can then be further compressed using other techniques, for example, Huffman coding. If lossy compression (loss of accuracy which cannot be recovered after decompression) is tolerable, quantization can be used to discard unimportant values (for example the high frequency features of an image).

Transforms are furthermore particularly useful where multiple levels of detail are desired. An example of this may include the transfer of scientific data over a network for real-time analysis and observation. Low resolution samples can be constantly transferred, while higher resolution samples can be transferred upon request [3].

3. RESEARCH QUESTIONS

We are investigating the feasibility of adding a *online* predictive compression step to the existing KAT-7 / MeerKAT pipeline. Such a step has to meet at least two primary criteria: high throughput and effective compression ratios. These are outlined below:

1. The technique must be fast. The algorithm should be able of achieving throughput rates of at least 40 GiB/s.

2. The technique should be effective. The algorithm should reduce the size of transmissions by several percent and hopefully this reduction can take the form of double digit figures. It has, however, been pointed out that the data may be too noisy to expect great reductions, while maintaining the throughput rate we mentioned above.

3. We will investigate the trade-off between throughput and size reduction.

4. DESIGN AND METHODOLOGY

TODO

5. IMPLEMENTATION

TODO

6. FINDINGS

TODO

7. DISCUSSION

TODO

8. CONCLUSION

TODO

9. FUTURE AVENUES OF RESEARCH

TODO

10. ACKNOWLEDGEMENTS

I would like to acknowledge A/Prof. James Gain and Dr. Patrick Marais of the Department of Computer Science at the University of Cape Town for their continuous, expert, input on the project.

Secondly I would like to thank Jason Manley, a Digital Signals Processing specialist at the MeerKAT offices in Pinelands, Cape Town for providing us with technical information on the MeerKAT project. Jason has also kindly prepared a 100 GB of sample of KAT-7 output data for testing purposes.

Thirdly I would like to note that all tests were performed on the ICTS High Performance (*HEX*) cluster at the University of Cape Town. The cluster has 4 DELL C6145 nodes each boasting 4 16-core AMD Opteron 6274 CPUs, clocked at 2200 MHz with 16 MB L3 cache memory. There are two additional GPU nodes with 4 Tesla M2090 GPU cards each. Each GPU card has 6 GB GDDR5 and 2048 CUDA cores. I want to especially thank Andrew Lewis from ICTS for his assistance and support during testing.

This research is made possible under grant of the National Research Foundation (hereafter *NRF*) of the Republic of South Africa. All views expressed in this report are those of the author and not the NRF.

11. REFERENCES

[1] Vadim Engelson, Dag Fritzson, and Peter Fritzson. Lossless compression of high-volume numerical data from simulations. In *Data Compression Conference*, pages 574–586. Citeseer, 2000.

[2] D. Salomon. *Data Compression.: The Complete Reference.* Springer-Verlag New York Incorporated, 2004.

[3] Hai Tao and Robert J. Moorhead. Progressive transmission of scientific data using biorthogonal wavelet transform. In *Proceedings of the conference on Visualization '94*, VIS '94, pages 93–99, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.