Application of Data Compression to HEP Data

Michael Schindler

Vienna University of Technology, Austria; changing to Intelligent Compression Techologies, Boston, MA

Abstract

This article will present some aspects to be considered when applying data compression to high energy physics data. Using simple examples it shows possibilities and limits of data compression in this field.

Keywords: Compression; Reduction; Data Acquisition

1 Introduction

Data compression [1][7][8] is a field of ongoing research in computer science. Traditionally compression comes in two categories; lossless compression which was originally developed for text compression and lossy compression for specific applications like image, video or sound recording. Scientific datataking, for example in HEP, is predestinated for data compression. This article will present some points to be considered in data compression decisions.

2 Compression basics

There is one very basic fact about data compression: Either there must be redundancy in the data (the data must not be evenly distributed random numbers), or some unimportant aspects of the data have to be thrown away. Sometimes this redundancy is hidden; for example a sequence of random addresses might not be compressible, but the same addresses seen as set are compressible like in the CERN LHC ALICE pixel detector [2].

Other times it is not clear where the signal ends and the noise begins; if one uses a 10–bit linear ADC after an analogue device with 1% error lossless compression is a problem. The reason is that nearly all standard methods for lossless compression rely on pattern repetition. In the above case a signal with

Preprint submitted to Elsevier Science *26 April 1997*

amplitude 0.1 would produce noise of 1 digitization count; this noise is enough to destroy pattern repetitions even if the analogue signal is periodic. It should be stressed here that there is no method to compress this noise, even through there are methods to compress the disturbed signal. In some cases a 5% noise accounts for more than half of the compressed data. A possible solution is to include a filter before a lossless compression or within a lossy compression.

3 Lossless compression

Unlike text or program files measurement data will not contain much bit–by–bit identical sequences. A simplified example of addresses in wire chambers might be the following: the next address is random with 0.1 probability or the previous address incremented by 47 with 0.8 probability or by 18 with 0.1 probability and then the last two digits are used. This example will cause standard lossless compression methods to fail (4–6 bits/number for the example). One mayor exception are block–sorting methods [3], they reach 2 bits/number after several thousand numbers. Recent speed improvements [4] and adaptions to non byte–oriented data (current research) make them an interesting alternative to zip and ppm variants. A good customized compression for this example will not encode the numbers directly, but the differences between numbers modulo 100. This method reaches 1.58 bit/number right from the start using arithmetic coding.

4 Lossy compression

Lossy compression acts like a filter followed by a lossless compression. This simple fact explains why a lossy compression is always specific for one datatype; it simply makes no sense to apply FFT audio filters to image data or wavelet image filters to HEP data.

If the data meaning is known there is no problem in designing a customized lossy compression that performs a harmless and in most cases even useful filtering (reduce 50Hz pickup and on-the-air noise as well as excess accuracy) of the data. Thus it can compress the data much more efficient than any lossless compression. The close interaction between compression and filtering in a lossy compression guarantees better results with less total loss and effort than a separate filter followed by a lossless compression.

There is still one big problem with this compression: nobody will call it *filtering compression* just to avoid the term lossy. But if (and only if) the filter function is carefully chosen you will not lose any information when using a lossy compression.

5 Some other aspects

Compression should be transparent to the user; for this reason it is typically included into IO modules [5] or in the output device or device driver. The required throughput and the data to be compressed determine the best method and if to implement in software, hardware or combination of both. The choice of compression has influence on other DAQ components [6], so compression should be considered at an early stage.

6 Summary

Lossless as well as lossy compression work best if customized to the data they should compress. Using general lossless compression methods (blocksort, gzip and others) results in greater CPU usage and larger compressed data. Sometimes the difference may be small, but most times it will be large.

Here are the actual numbers for monte carlo simulation data of the KLOE experiment at DAΦNE, Italy. Event–by–event compression with gzip compressed by 7%; when compressing several events together 15% could be reached. A costumized compression compressed single events by 50%.

References

[1] M.Nelson, J.Gailly *The Data Compression Book* M&T Books, NY 1995. ISBN 1–55851–434–1 http://web2.airmail.net/~markn/tdcb/tdcb.html

[2] H.Beker, M.Schindler *CERN ALICE Internal note 1996–03* http://aldwww.cern.ch/documents/designNotes/ALICE_Data_Compression.html

[3] P.M.Fenwick *Block Sorting Text Compression* Proceedings of the 19th Australasian Computer Science Conference (Feb.1996), Melbourne, Australia. http://ftp.cs.auckland.ac.nz/out/peter–f/ACSC96.ps

[4] M.Schindler *A Fast Block–Sorting Algorithm for Lossless Data Compression* IEEE proceedings of the Data Compression Conference, March 1997, Snowbird, Utah. http://eiunix.tuwien.ac.at/~michael/st/

[5] N.Smirnov, E.Tcherniaev *Implementation of Data Compression in the DELPHI Experiment.* CHEP95

[6] H.Beker et al. *Architectural Impact of Costumized Compression Methods on the Switch Based ALICE DAQ System* 2nd International Data Acquisition Workshop on Networked Data Acquisition Systems, Nov.1996, RCNP Osaka, Japan.

[7] T.C. Bell, J.G. Cleary, I.H.Witten *Text Compression* Prentice Hall, New Jersey, 1990

[8] *Managing Gigabytes*