Eurogam Sorting Parameters

John Cresswell

August 1990

1 Introduction

The Eurogam project will involve a significant increase in the rate, volume and complexity of data collected. At this point it is worth noting some of the problems of sorting in general so that future analysis requirements can be related to problems associated with the hardware and software that could be used. This document attempts to point out some of the parameters relevant to sorting in order to show where cost/performance boundaries occur. This is a first attempt and so some arguments and possibilities may not be included, but its a starting point.

Upto now, essentially all analysis has been based on 'doubles' gamma-ray event data from TESSA like devices. The sorting analysis creates several 2D matrices. The Eurogam project will involve higher fold events, opening up the possibility of 3D matrices with the corresponding dramatic increase in disc space required for storage.

2 VME Sorting

The Eurogam online sorter will be based on a design that has evolved from systems currently in operation offline in the UK. These systems are VME based and are operational at Daresbury Laboratory, Manchester University and Liverpool University.

The current offline processing power is approx. 10 Motorola Mips. This is sufficient for processing TESSA type events at 5k events/second. A typical event processing will include gain matching, some filtering and spectrum updating. Each system contains 64 Mbytes of memory for spectra. The system performance may be changed by adding more cpus and more memory.

The Eurogam online design will have approx. 100 Motorola Mips (approx. 70 VAX equivalent Mips) of processing power and is calculated to be sufficient for 100% sorting at 10k events/second. This calculation assumes that the high fold data is sorted as combinations of triples (something has to be assumed!). The spectrum memory space will be 64 Mbytes initially. Hence it is clear that 2D matrices will be supported online but not large 3D matrices. This example could produce a spectrum update rate of 100k updates/second for each spectrum involved if filtering did not cause a significant reduction.

It is difficult to quote sorting speeds, since no two sorts are exactly the same. All the calculations are based on 'typical' sorts observed over a period of time. Hence any numbers will be averages and can not be guaranteed for every sort required.

The limitations of the current design occur in several places. The first limitation is the size of a VME crate. The Eurogam online design will occupy approximately half a crate. A multi-crate solution is possible. The ACP project of Fermi-Lab is one current example of a large number of processors occupying several crates.

The second limitation is the amount of processing power that can be accommodated in a single crate. The current design can be enhanced with more of the same type of cpu or with other cpu types in future.

The third limitation is the spectrum memory space. As time goes by the memory capacity of a crate will increase as 4Mbit and later 16Mbit memory chips become widely available. This will lead to a limit of the order of 1 Gbyte, which may be too expensive to achieve in practice.

The fourth and possibly the most important limitation is the number of spectrum updates achievable per second. The spectrum memory is currently global VME-addressable standard memory. There is a maximum rate of incrementing/decrementing spectrum channels which occurs at the point when VMEbus becomes saturated. Since spectrum access for display purposes will consume some of VMEbus bandwidth, then the limitation is currently somewhere in the range 500k to 1M spectrum updates per second. This rate is largely independent of cpu speed, as most of the time is taken by VMEbus and onboard arbitration.

The only way to improve on this update rate is to use locally accessible memory. This requires some thought as other parameters (processor power, multi-processor bus contention, spectrum access, etc ...) then become important.

3 Disc sorting

Disc sorting is usually attempted when there is not enough memory available to sort into directly. Certainly, for large spectrum space then disc based sorting is more cost effective than a large block of memory.

To increase the performance of disc-based sorts to reasonable rates of updates per second, it is necessary to buffer the spectrum updates into memory. The usual procedure is to allocate a memory buffer for every disc block. Many updates are then accumulated in the buffer and only written to disc when the update buffer is full. This technique reduces the number of disc transfers. It is useful to try and parameterise this process ...

Assume ... (quantities in bytes)

`s = spectrum space on disc`

`b = disc block transfer size`

`u = memory update buffer size`

```
m = total memory space used for update buffers

f = factor of spare update buffers (algorithm dependent).
    This number is typically between 1.0 and 1.5
```

$$m = fsu/b$$

To be useful, a disc sort must have

$$m << s$$

and therefore

$$fu << b$$

To first approximation this means that the memory update buffer space associated with one disc block must be significantly less than the size of the disc block. Otherwise a memory-based sort must be the better option.

A list of some choices of the above parameters is shown in Table 1. It is possible to vary all the parameters, but to illustrate the point, I have assumed a spectrum space of 1 Gbyte, disc blocks of 8 kbytes and discs capable of a sustained transfer rate of 100 blocks/second (50 reads and writes). All these values may be discussed. For example, the number of disc blocks transferred per second will be somewhat dependent on the cost of the disc and controller.

Table 1: Possible disc-based sorting parameters

s	u	b	m(f=1)	upds/sec
1G	256	8k	32M	6400
1G	1k	8k	128M	25k
1G	8k	8k	1G	200k

The third line of Table 1 shows the case where the spectrum space is equal to the update buffer space. In this case it is no longer sensible to sort to disc. The update rate can be compared with the figures quoted above for the VME memory-based design.

The figures in the table also show that the memory usage of a disc-based sort can be the limitation. Assuming disc transfers to and from memory are 32bit wide, then 200k read/write instructions per second are required. Add to this the memory operations involved with filling the memory update buffers, and then unloading the updates to each disc block, there may not be much bandwidth left for the sorting itself!

Increasing the performance of the discs and controllers in order to increase the number of disc block transfers per second is therefore not so simple. The discs and controllers will be more expensive. The memory contention problems will become worse.

4 Tape Storage

At the present time tape storage is the real limiting factor to data rates and volume. One double density Exabyte tape drive will store or supply about 500 kbytes/second or 125 kWords/second. The important number to be extracted is the number of spectrum updates per second that are contained in a given data volume.

Assume for now that the spectrum updates are a consequence of processing the high fold Germanium energy words into one 2D matrix per event. Clearly in this case the event size is variable. If only suppressed Ge energy words exist in the event, then each event may only be 6 words, resulting in a data rate of approx. 20k events/second. If the maximum number of words collectable is stored, then each event may occupy 100 words, resulting in a data rate of approx. 1k events/second.

Assuming 5 Ge words there are 10 combinations of triples which could be processed as a filter on one word followed by a 2d update. The filter may just select a channel and may not reduce the data by a significant factor. The spectrum update rate will therefore lie between 10k and 200k updates per second per spectrum.

5 Summary

I have attempted to highlight some of the problems associated with sorting large volumes of data into a large spectrum space. Some of the limitations I have quoted will prevent some sorts from

operating at all (e.g. running out of memory for a memory-based sort). Some of the limitations will just slow down some sorts whilst allowing others to perform with maximum efficiency (e.g. disc speeds and memory bandwidth). In the latter case it is appropriate to consider typical cases and maybe say that sorts consuming the most resources will just go slower.

For 3D sorts occupying disc space in the 1Gbyte region there are further questions to be asked. Such a large matrix will take quite a long time to analyse. Whether the matrix is created by a memory-based or a disc-based sort, it will still have to live on disc for the period of analysis. It will not be feasible to read from tape every time the physicist requires to do some work on the matrix. Therefore medium-term storage accessible to the analysis computer is essential. One possible solution would be to use laser disc drives and treat the laser disc in the same way as we currently use tapes for file archive.