Performance Analysis of Generic vs. Sliced Tags in HepODBMS

STOCKINGER Kurt, Database Group, Information Technology Division
CERN, Geneva, Switzerland

Abstract

This paper presents a performance analysis of accessing tag data clustered in two different ways, namely event-wise clustering (generic tag) vs. attribute-wise clustering (sliced tag). The results show that especially "prefetch-optimisation" results in an additional performance gain of sliced tags over generic tags when only a subset of all the tag attributes is accessed.

Keywords: clustering, tags, performance analysis

1 Introduction

Tags describe physics quantities that are frequently used to select events for analysis. They maintain a logical connection to the data which they summarise and thus allow keeping them consistent with this data. In addition, they support transparent navigation to the original data after a selection has been performed.

The HepODBMS [1] tag implementation (based on Objectivity) which most frequently uses so-called "generic tags" clusters all attributes of a particular event together (event-wise clustering). For certain selection types, i.e. selections which only reference a small number of attributes from a tag, a different clustering strategy may result in better performance. We call this "sliced tag" since it clusters all values of a given attribute close together (attribute-wise clustering). We implemented the sliced tag in such a way that it has the same interface as the generic tag and can thus be easily used by physicists who currently use generic tags for their analysis.

A previous performance study in [3] described a benchmark with a PAW-ntuple of 81,060 tags with 302 attributes. The results show that the column-wise ntuple (equivalent to sliced tag) is more efficient than the row-wise ntuple (equivalent to generic tag) if less than 10% of all the attributes of a tag are selected. In the worst case, i.e. when all attributes of a tag are selected, the response time for the column-wise ntuple is about 4-5 times higher than for a row-wise ntuple.

2 Performance Analysis

Our "basic" implementation of sliced tags in HepODBMS shows a very similar behaviour to the results presented in [3]. However, by using prefetch optimisation (read-ahead), the relative performance of the sliced tag is improved by a factor of two. Thus, the sliced tag is more efficient than the generic up to an attribute selectivity of 25% (i.e. 25% of all attributes of a tag are accessed) rather than 10% without prefetch optimisation. (We give a brief description of the prefetch optimisation below.) What is more, in the worst case, the sliced tag is only 2.5 times slower than the generic tag. Picture 1 a) shows the response time of accessing various numbers of tag attributes based on generic tags, "basic" sliced tags and sliced tags with prefetch optimisation. All our tests were performed at Caltech's "tier2b" machine (Dual 933 MHz Pentium III Linux server, 900 MB RAM, using a 600 GB 3ware RAID 0 array).

We based our prefetch optimisation on findings presented in [2]. In short, rather than fetching single database pages of parallel data streams, i.e. multiple attributes, we prefetch multiple database pages in chunks of 1 MB. The advantage of this approach over fetching

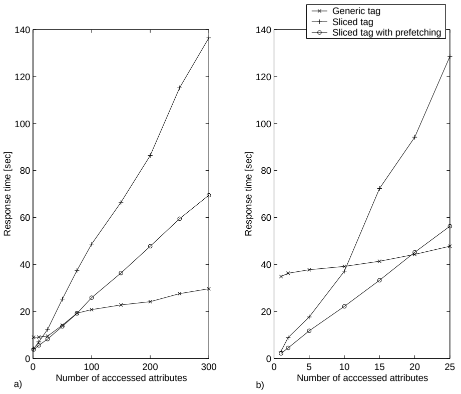

Figure 1: Response times for selecting attributes based on various different tag implementations. a) 81,060 tags with 302 attributes b) 1,000,000 tags with 25 attributes.

single database pages is the reduced number of disk head movements for random access and consequently reduced response times. Figure 2 shows the typical pattern of the disk arm for random access of two parallel streams without a) and with b) prefetch optimisation.

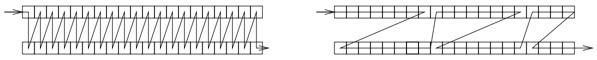

Figure 2: Access patterns of disk head movements for reading parallel streams a) without prefetch optimsation b) with prefetch optimisation.

We also carried out some performance benchmarks for 1,000,000 tags with 25 attributes. In this case, the advantage of sliced tags over generic tags is even more significant (see Figure 1 b)). The sliced tag outperforms the generic tag up to an attribute selectivity of 80%. This is due to the relativ bad performance of Objectivity for handling small objects such as the generic tag with 25 attributes.

3 Conclusions

The sliced tag with prefetching is ”optimial” up to 25% attribute selectivity - above this threashold the generic tag performs better. In short, for ”large” tags with many attributes the sliced tag is to be prefered if only a subset (up to 25%) of the attributes is selected.

References

[1] HepODBMS, http://wwwinfo.cern.ch/db/objectivity/docs/hepodbms

[2] Koen Holtman, Prototyping of CMS Storage Management, Ph.D. thesis (proefontwerp), May 2000, Eindhoven University of Technology, ISBN 90-386-0771-7

[3] RD45, Using an Object Database and Mass Storage System for Physics Analysis, April 1997, http://wwwinfo.cern.ch/asd/rd45/reports.htm