

Tag-based Analysis at the BESIII Experiment

Z Y Deng, J H Zou, S S Sun, B J Liu, L Wang, J Y Shi, X A Xiong and S F Zhang

Institute of High Energy Physics, Chinese Academy of Sciences,
19B Yuquanlu, Shijingshan District, Beijing, China.

dengzy@ihep.ac.cn

Abstract. As more and more data is accumulated from BESIII detector, the reduction of the CPU time of data processing procedures is significant for the experiment to get physics results efficiently with limited hardware resources. In this study, we designed a tag-based analysis method in BESIII offline software system. Typical physics analysis results show the tag-based analysis with reformed DST file can reduce the jobs running time to about 1/10, with high CPU efficiency and low read throughput, without inducing additional disk space occupation.

1. Introduction

BESIII [1] (Beijing Spectrometer) is a new detector at the upgraded BEPCII [2] (Beijing Electron and Positron Collider) that operated in the tau-charm threshold energy region between (about) 2 and 4.6 GeV with a designed luminosity of $1 \times 10^{33} \text{cm}^2 \text{s}^{-1}$. The BESIII experiment implements studies on light hadron physics, charmonium spectroscopy and decays, D and Ds meson decays, and numerous topics in QCD and tau physics. BESIII detector is composed of four subdetectors: multilayer drift chamber (MDC), time of flight (TOF) system, electromagnetic calorimeter (EMC), and RPC muon counter (MUC). BESIII has accumulated the world's largest J/psi, Psi(3686), Psi(3770) data samples. More and more CPU resources are needed to process accumulated data. The reduction of the CPU time of data procedures will allow to get physics results efficiently with limited hardware resources.

BESIII offline software system (BOSS) [3] was developed based on Gaudi [4]. BOSS framework provides standard interfaces for common software components necessary for data processing including simulation, reconstruction and analysis. All the data processing procedures are done event by event. That is efficient and necessary for simulation and reconstruction, but inefficient for analysis since the typical branching fractions for interesting physics channels are of the order of 10^{-3} [5][6]. Tag-based analysis is a more efficient way to enable retrieving only selected events for the analysis. Event tags are event-level metadata which providing basic information for physics users to select interesting events, including run number, event number, charged tracks, neutral tracks, D flags etc. Analysis jobs make pre-selection based on event tags and are allowed to read only the events satisfying the given criteria. Event data for unselected events will not be retrieved and data files containing only unselected events will not be read. The tag-based data analysis jobs can perform much faster than that of full mode analysis (reading event one by one).



2. BOSS framework

The BOSS framework employs Gaudi’s event data service as the data manager. Analysis algorithms can access the DST (Data Summary Tape) event data from Transient Data Store (TDS) via the event data service. As shown in Figure 1, RootCnvSvc is responsible for conversions between persistent DST data and transient DST objects. Reconstruction jobs write DST event data objects to ROOT [7] files and analysis jobs read ROOT files to retrieve the DST objects.

Figure 2 shows the DST data file structure. It is composed of TTree (Event) to hold all event data information for physics analysis, and another TTree (JobInfoTree) saves job history information for generating the DST file. Typical size of DST event data is 3~5kB per event, one DST file contains 100-150k events.

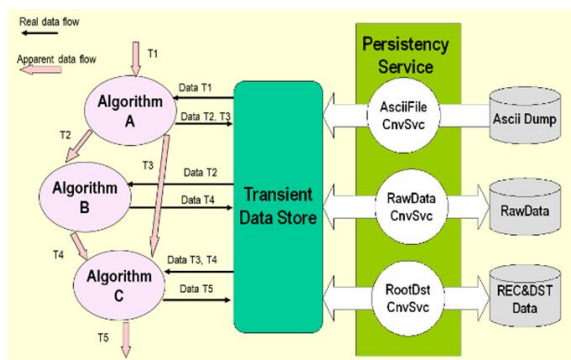


Figure 1. Event data flow in BOSS

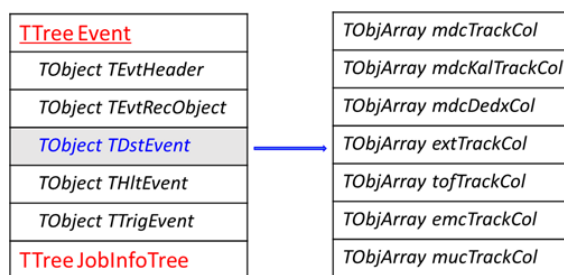


Figure 2. DST data file structure in BESIII

3. Analysis based on data skimming

A possible way to speed up the physics analysis is data skimming. For example, the light hadron physics group can skim 2 prong, 4 prong DST dataset from the full DST dataset. People can choose the specific sub-dataset according to the physics channels they study. Each time a new software version released, data reconstruction jobs produce the full DST dataset, and the sub-dataset needs to be re-generated. Although the analysis based on skimmed sub-dataset will consume less CPU time, more disk space will be occupied due to the additional copies of DST events for each BOSS software release. And it’s inflexible for users to switch between different sub-datasets.

4. Tag-based analysis

The motivation of the tag-based analysis is to improve analysis efficiency in BESIII experiment. It should be designed to enable running analysis jobs with less CPU time, high CPU efficiency, negligible additional disk space occupation and flexible user interface.

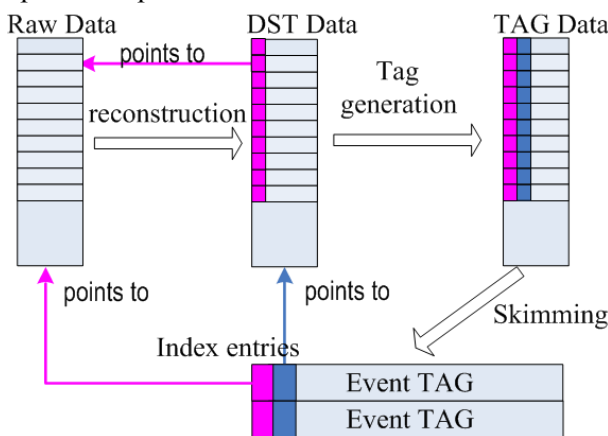


Figure 3. Workflow of tag based analysis

4.1. Workflow

Figure 3 shows the workflow of the tag-based analysis. After data reconstruction, the location of raw data file is recorded in DST data file. With a new algorithm TagWriterAlg, TAG data can be generated from DST data file. The TAG data can be generated during data reconstruction, together with the DST data and independently as well. Location of DST data can be recorded in TAG data. The event tag information can be defined by physics groups. The entries in TAG data are the same as entries in DST data. Compared with the skimmed DST dataset, the size of TAG data can be ignorable, for example, one integer is enough for the tagging of 32 different decay modes. The main advantages of this analysis mode are in the flexible definition of tag metadata and selection criteria, and running analysis jobs without too many changes in jobOptions.

4.2. Software implementation

With TAG data and user-defined criteria as input, together with a new service (TagFilterSvc) and modified RootCnvSvc, the software implementation for tag-based analysis is as the following:

- (1) TagFilterSvc can read entries from TAG data file and detect which entry satisfies the user-defined criteria and saves the selected entries for each TAG data file in memory.
- (2) RootEvtSelector is responsible for the selection of next event to be processed. It gets the next eventID from TagFilterSvc.
- (3) RootInterface is responsible for retrieving branches from ROOT tree. The cooperation of RootEvtSelector and TagFilterSvc enable RootInterface to retrieve specific entries from ROOT tree.
- (4) After getting the branches, various kinds of Converters are responsible for conversion between persistent TObject and transient data object.
- (5) Then analysis algorithms can read DST objects from TDS.

4.3. Performance

The inspection of the tag-based analysis mode bottlenecks requires to test it with more jobs running simultaneously. The figures 4-5 show that with the increasing of the number of jobs, the Tag mode time and disk utilization grow, and CPU efficiency decreases. The computing monitoring system shows the readthroughput of tag mode is larger than the full mode, which is possibly due to the cache strategy of file system. It means that the tag mode reaches I/O bottleneck in the dedicated file system.

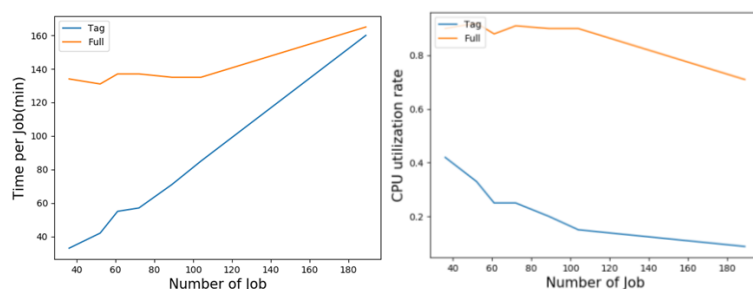


Figure 4. Job time for Tag mode and Full mode(left)
CPU efficiency for Tag mode and Full mode(right)

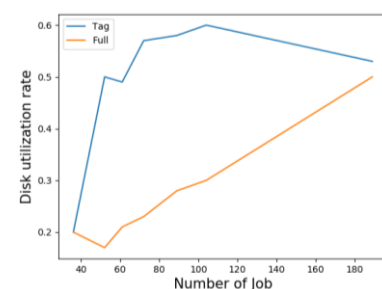


Figure 5. Disk utilization rate for Tag mode and Full mode

5. Analysis based on restructured DST

An efficient method to reduce the total read throughput of analysis job is to put the selected events in a continuous event block, eliminate the cache effect of file system. Different physics analysis jobs can retrieve different event blocks through the tag metadata.

5.1. Workflow

Figure 6 shows the workflow of restructured analysis mode. An algorithm DstReformAlg is designed to re-create a DST data file based on the original DST. The order of events is changed, all information in DST files are kept. And there is an additional TTree “Metadata” to record the [begin, end] position of each event block, and also record the map between original event index and current event index. The event index map can be used to retrieve raw data during the data analysis. Since no information is lost during the restructurization, the original DST can be deleted. So no additional disk space will be occupied for the restructured mode analysis.

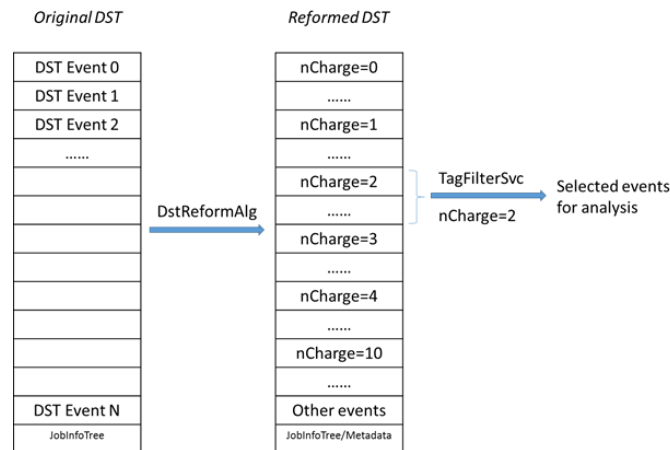


Figure 6. Workflow of reformed analysis mode

5.2. Performance

Figures 7-8 show the total job time is much less than the time of full mode and doesn't change with the number of jobs. CPU efficiency is about 90%, almost the same with full mode, and disk utilization rate is much lower than full mode and tag mode. As shown in Table 1, monitoring system shows the peak read throughput of reformed mode is also lower than full mode and tag mode. These results show the reformed mode can obviously improve the analysis efficiency in current computing environment.

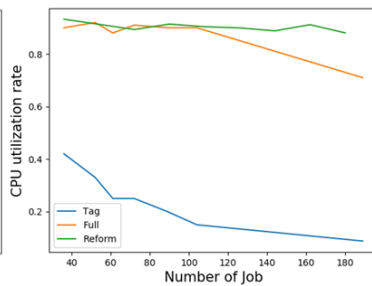
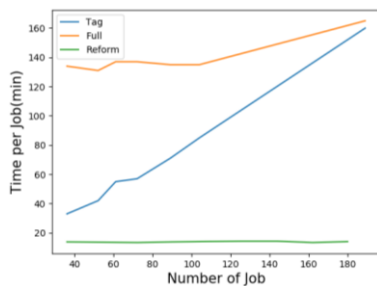


Figure 7. Job time (left) and CPU efficiency (right) for Reform mode, Tag mode, and Full mode

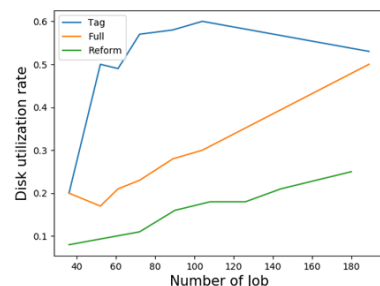


Figure 8. Disk utilization rate for Reform mode, Tag mode, and Full mode

Table 1. Read throughput for different analysis modes

Mode	Full	Tag	Reformed
Peak read throughput	200~800MB/s	600~800MB/s	100~200MB/s

6. Conclusion

In this paper, a tag-based analysis mode is developed in BESIII offline software system, dedicated for reducing the job running time for BESIII physics analysis. Tag based analysis enables users to define event-level metadata (tag) to describe each DST event, and implement pre-selection based on the tag.

Compared with reading all the events in DST data files (full mode), Job running time of tag mode can be less. But the CPU efficiency will be lower when number of jobs increases. The I/O bottleneck can be explained with the cache strategy of file system. The strategy can benefit the full mode analysis. So some improvement on the tag-based analysis mode were implemented by reforming the DST data file with the according tag information, and then made pre-selection based on the tag. This reforming mode will enable analysis job to read a continuous event block in DST data file. Performance test results show the reforming mode can get high CPU efficiency and low disk utilization rates. Study on a typical physics analysis channel in BESIII shows the job running time of reformed tag based analysis mode can be 1/10 of the full mode. It will help BESIII experiment to get physics results more quickly with limited CPU resources.

7. Acknowledgments

This work was partly supported by Youth Innovation Promotion Association, CAS.

References

- [1] Ablikim M et al. (BESIII Collaboration) 2010 *Nucl. Instrum. Meth. A* 614 345
- [2] C. Zhang for BEPC & BEPCII Teams, Performance of the BEPC and progress of the BEPCII, *Proceedings of APAC*, 2004, pp.15-19, Gyeongju, Korea.
- [3] Li W 2006 Proc. Int. Conf. Comput. *High Energy and Nucl. Phys* 225
- [4] Barrand G and others 2001 *Comput. Phys. Commun.* 140 45-55
- [5] Asner D M, Barnes T, Bian J M, Bigi I I, Brambilla N, Boyko I R, Bytev V and Chao K T et al. 2009 *Int. J. Mod. Phys. A* 24, S1
- [6] Beringer J et al. (Particle Data Group) 2012 *Phys. Rev. D* 86 010001
- [7] R. Brun and F. Rademakers, ROOT: An object oriented data analysis framework, *Nucl. Instrum. Meth. A* 389(1997) 81