Monte Carlo production monitoring tool for AMS experiment

Monte Carlo production monitoring tool for AMS experiment

R.Q. Xiong[1], R.L. Shi[1], F.Q. Huang[2], B.S. Shan[3], V. Choutko[4], A. Egorov[4], A. Eline[4], O. Demakov[4], J.H. Zhang[1], F. Dong[1] and J.Z. Luo[1]

[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Baidu, Inc. Shanghai, China
[3]Beihang University (BUAA), Beijing, China
[4]Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA

E-mail: jluo@seu.edu.cn

Abstract. Monte Carlo (MC) simulation production plays an important role in physics analysis of the Alpha Magnetic Spectrometer (AMS-02) experiment. To facilitate the metadata retrieving for data analysis among millions of database records, we developed a monitoring tool to analyse and visualize the production status and progress. In this paper, we discuss the workflow of the monitoring tool and present its features and technical details.

1. Introduction

The Alpha Magnetic Spectrometer (AMS-02) [1] is a multi-purpose high-energy particle physics detector that operates on the International Space Station (ISS). Its goal is to search for antimatter in the universe on the level of less than 10^{-9}, search for dark matter in various physics channels and perform high statistics measurements of cosmic rays composition. AMS-02 achieves its goal by modeling and analyzing the massive cosmic-rays data collected from the space. In each physics analysis, Monte Carlo (MC) production plays an important role, which produces the simulated signal and background events to verify the physical models.

AMS-02 community has begun to perform MC production since six years ago when AMS-02 experiment started. So far, over four thousand different types of MC simulated events have been produced by seven computing datacenter sites around the world. As the MC simulation data are so massive, it is essential to keep monitoring the status of the overall MC production and store all the information of these generated simulated events in the database for the future physic data analysis.

In this paper, we present a web-based Monte Carlo Production Monitoring (MCPM). The MCPM tool was developed in 2012 and has been used by AMS-02 since 2015. The tool was implemented based on the state-of-the-art web development technologies, such as Python, ECharts, JSON, JavaScript [3-6]. During the data analysis of AMS-02 experiment, this tool can help physicists quickly and cleanly to retrieve the simulated events metadata from the millions of database records, by analyzing and visualizing the production status and progress. In the following sections, we discuss the workflow of the monitoring tool and present its features and technical details.

2. Architecture and workflow

2.1. The architecture of the monitoring tool

The MCPM tool is used to display the monitoring information of the AMS MC production. As shown in figure 1, the architecture of MCPM tool is composed of three layers: the user layer, the web application layer and the data layer. The user layer provides a user interface for different users of AMS data production group to monitor various information of the MC production, while the information inquired by users is displayed in form of different kinds of charts or diagrams. The data layer contains a dedicated database and several structured text datasets. This layer plays a role as a central repository for the MC production monitoring data. The data contained in the data layer will be processed by the web application layer and provided to the user layer for displaying. Each of these three layers is described in detail as follows.

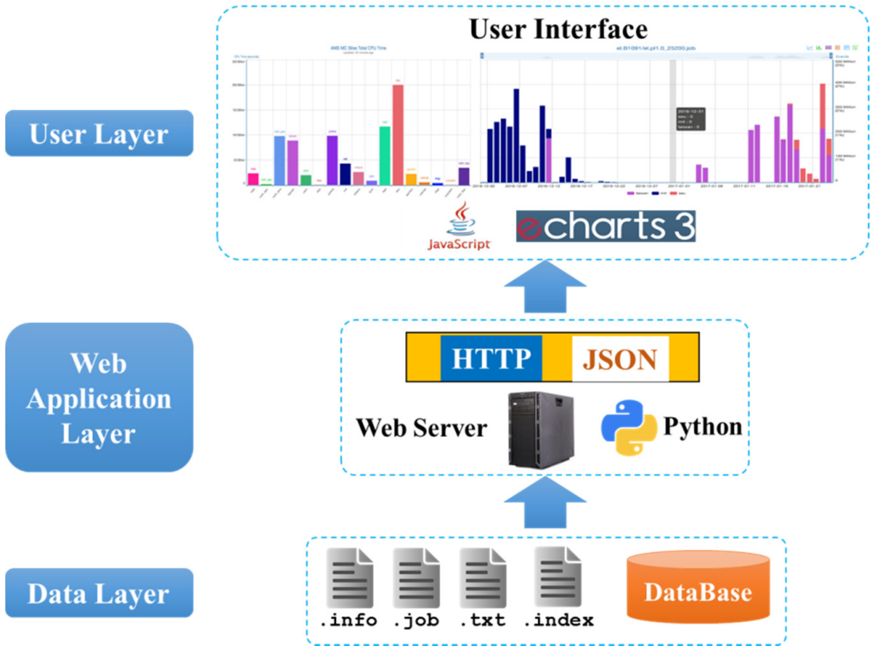


Figure 1: AMS MC production monitoring architecture

In the data layer, the database stores the specific information of MC production in different datacenter sites, such as the number of finished simulated events, consumed CPU time, etc. Besides, this layer also stores detailed information of each MC production template in four different kinds of files. The `.index` file shows the template name of the finished jobs and the location path of the dataset, the `.job` file shows the contents of the jobs, the `.txt` file shows the location path of every root file, and the `.info` file shows the detailed information of every job in the template.

In the web application layer, a python script periodically weak up to calculate statistical number of finished events and the statistical CPU time for different computing centers and different templates. All statistical data are stored in JSON files and provided to the user layer for demonstration. Detailed information of each template, including `.info` file, `.job` file and `.index` file, is provided to the user layer in the form of URLs by POST method.

The user interface of MC production monitoring is a client-side JavaScript application [6]. This monitor communicates with the server, by using asynchronous JavaScript requests from the client browser and expects JSON responses. It utilizes modern web technologies such as JavaScript library, ECharts plotting library [4], etc. The JSON files provided by the JSON responses are parsed and presented by ECharts charting tool. These ECharts and URLs are organized by JavaScript and displayed in the user interface.

2.2. *The workflow of the monitoring tool*

The MCPM tool mainly contains two python programs, which are used to periodically generate the JSON format data. The first python program is used to collect the statistical number of finished events for each template, and the second python program is designed to get the statistical CPU time for each computing center. The processing workflows of these two programs are shown in the figure 2.

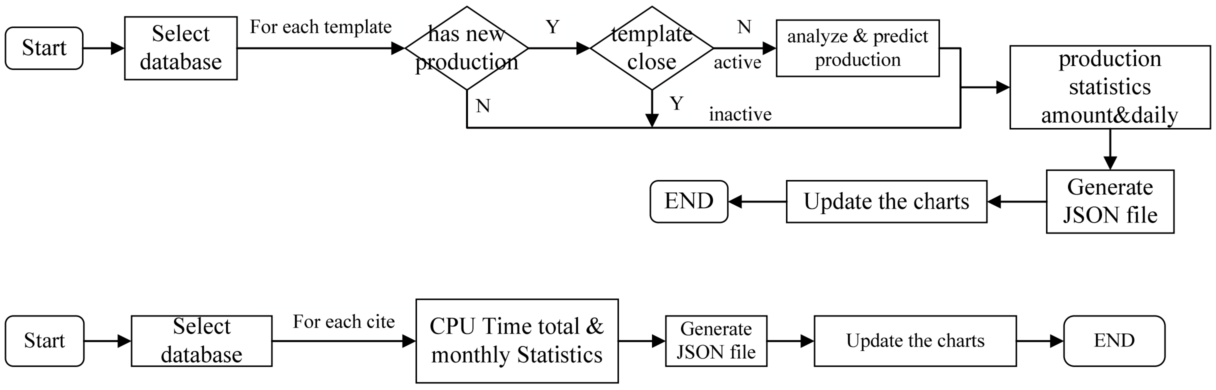


Figure 2: AMS MC production monitoring workflow

The upper part of figure 2 demonstrates the workflow of the python program for generating the JSON format data of the statistical number of finished events for each template. During the operation, this program firstly select databases for all MC production templates. After that, the program will conduct analysis and predict of the production progress, based on the current production status information that collected from active templates. If a template has new production or is not closed, we term it as an active template and vice versa. Then, the program will generate JOSN files based on the production statistics of all active and inactive templates. Finally, the displaying chart will be updated based on these newly generated JOSN files.

The lower part of figure 2 demonstrates the workflow of the other python program. This program is designed to generate the JSON format data of the statistical CPU time for each computing center. The program firstly select databases for each all computing centers. Then it will produce JOSN files based on the information of total and monthly CPU Time. Finally, the displaying charts will be updated based on these newly created JOSN files.

3. Features

3.1. Detailed statistical data
For the AMS MC Production monitoring tool, there are two important categories of detailed statistical data concerned by the AMS scientists, and should be demonstrated, that is, the MC production status and the CPU-Time of each computing center.

3.1.1. MC production status.

AMS MC Production is composed of a variety of datasets, which can be divided into two categories, the active production and the inactive production. For each dataset, it has several templates. The data structure of AMS MC Production is showed as figure 3. Therefore, we show the MC Production statistical data in four levels, that is, the production level, the dataset level, the template level and the event level. For the production level, we divide the MC production into active production and inactive production. For the dataset level/template level, we display statistical information of the dataset/template including the processing status, the percentage of unfinished events and the estimated finish time. For the event level, we present the daily and total events of the selected template that have been finished by the AMS computing centers distributed around the world. Besides, we display the prediction of the production progress in this level as well.

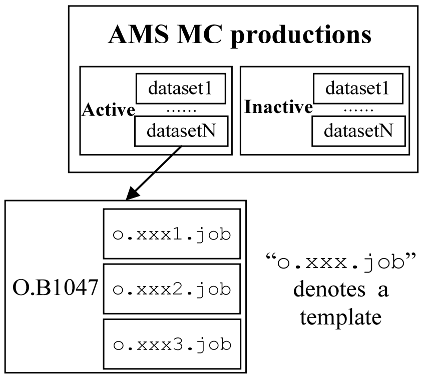


Figure 3: AMS MC production structure

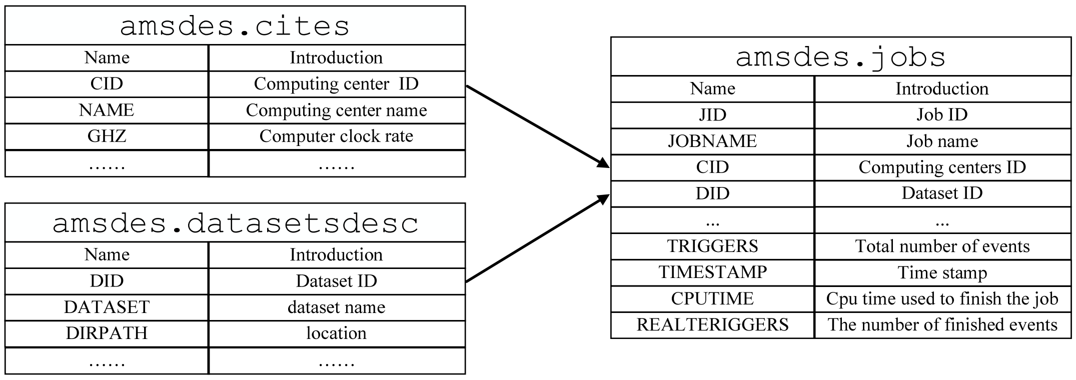
3.1.2. CPU-Time of each computing center.

We calculate the total and monthly CPU time consumed by AMS MC production in each computing center.

3.2. Python/ECharts/JavaScript
As all our data is stored in an Oracle database [7, 8], we use a python script to select data from the database and generate JSON files to store those statistical data which will be displayed in a JavaScript website with ECharts. ECharts is an open source powerful, interactive charting and visualization library for browser. We use it to organize the statistical data into charts and provide some download interfaces.

4. Technical details

4.1. The design of the database tables
The data statistics involve three tables, that is, `amsdes.cites`, `amsdes.dataesdesc` and `amsdes.jobs`. The table `amsdes.cites` stores the information of computer centers and the table `amsdes.dataesdesc` stores the dataset's information. For the table `amsdes.jobs`, it stores the detailed information of every job including the job ID, the total number of events, the number of finished events, the CPU time used to finish the jobs, etc. The foreign key mapping between those three tables is showed in figure 4.



`amsdes.cites`

Name	Introduction
CID	Computing center ID
NAME	Computing center name
GHZ	Computer clock rate
......	......

`amsdes.datasetsdesc`

Name	Introduction
DID	Dataset ID
DATASET	dataset name
DIRPATH	location
......	......

`amsdes.jobs`

Name	Introduction
JID	Job ID
JOBNAME	Job name
CID	Computing centers ID
DID	Dataset ID
...	...
TRIGGERS	Total number of events
TIMESTAMP	Time stamp
CPUTIME	Cpu time used to finish the job
REALTERIGGERS	The number of finished events

Figure 4: The structure of the database tables

4.2. Core API implementations

We design two python programs, named `datasets_analysis.py` and `cputime_sum.py`, to collect the statistical data described in 3.1. The `datasets_analysis.py` is used to collect the statistical data of MC production and the `cputime_sum.py` is used to calculate the CPU time consumed by each computing center.

5. Demonstration

The MCPM tool is deployed on the AMS website server (http://ams.cern.ch/MC_status). In this section we will demonstrate some results through the screenshots of the MCPM tool. Figure 5 is the AMS MC Production overview. Figure 6 shows an overview of the template level which demonstrates the dataset/template level statistical data. We can get the processing status, the percentage of unfinished events and estimated finish time from those pages. Figure 7 depicts the events level statistical data. Figure 8 displays the CPU Time monthly statistical data.

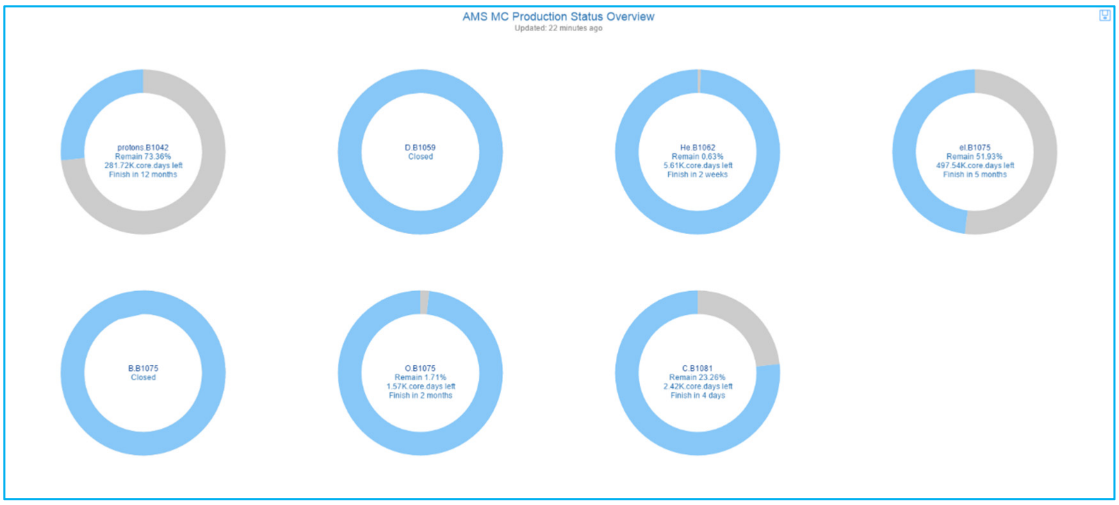

AMS MC Production Status Overview
Updated: 22 minutes ago
protons.B1042 Remain 73.36% 281.72K core.days left Finish in 12 months
D.B1059 Closed
He.B1062 Remain 0.63% 5.61K core.days left Finish in 2 weeks
el.B1075 Remain 51.93% 497.54K core.days left Finish in 5 months
B.B1075 Closed
O.B1075 Remain 1.71% 1.57K core.days left Finish in 2 months
C.B1081 Remain 23.26% 2.42K core.days left Finish in 4 days

Figure 5: AMS MC production overview

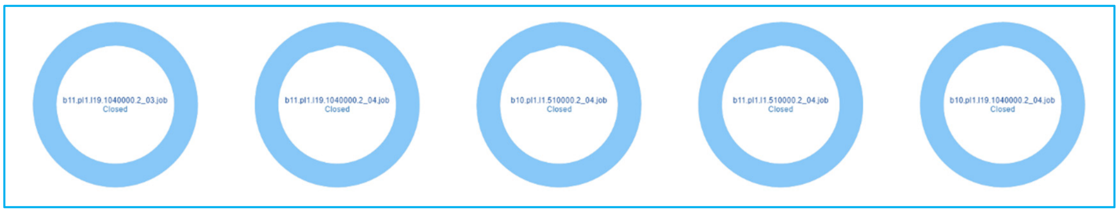

b11.pl1.l19.1040000.2_03.job Closed
b11.pl1.l19.1040000.2_04.job Closed
b10.pl1.l1.510000.2_04.job Closed
b11.pl1.l1.510000.2_04.job Closed
b10.pl1.l19.1040000.2_04.job Closed

Figure 6: Template level overview

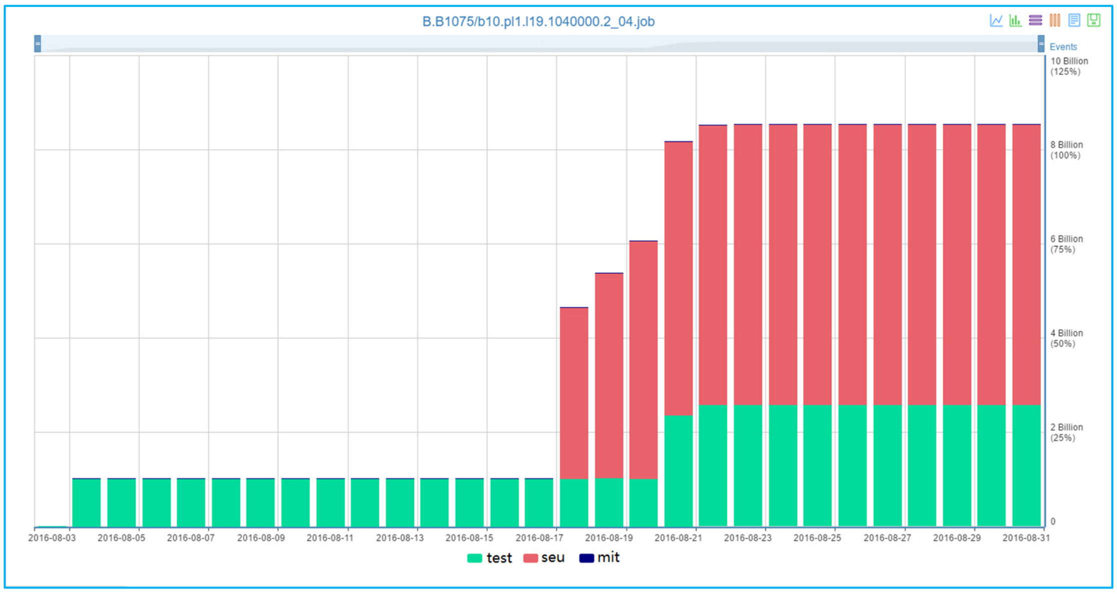


Figure 7: Events level overview

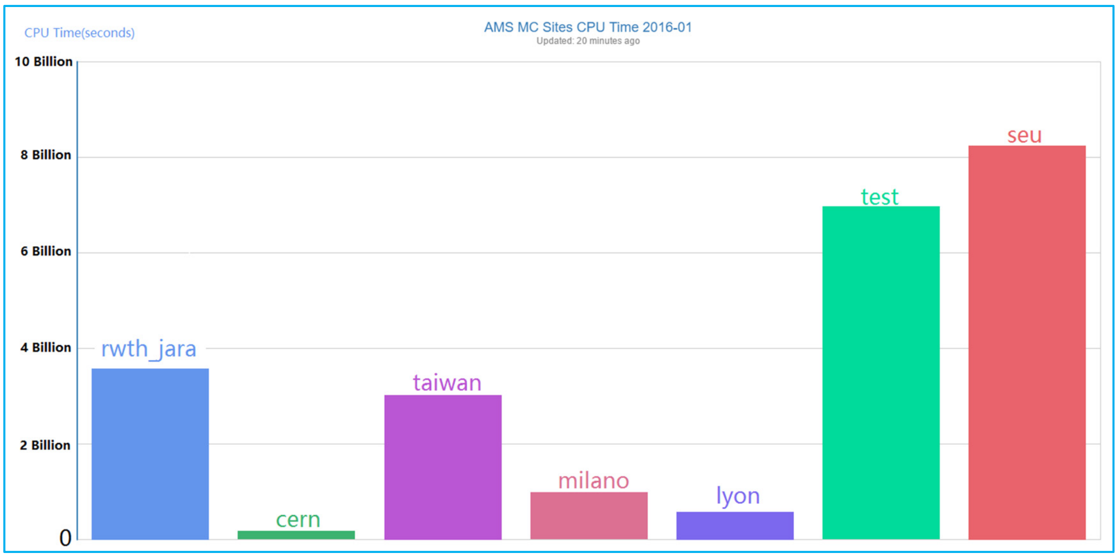


Figure 8: CPU time monthly (2016-01)

6. Conclusion

In order to analyze and visualize the Monte Carlo production status and progress for the Alpha Magnetic Spectrometer (AMS-02) experiment, in this paper, we described the detailed implementation of a web-based Monte Carlo production monitoring tool, which is developed by some recent popular web development technologies, and we also discussed the workflow of this monitoring tool and present its features. This tool can help physicists of AMS-02 experiment quickly and cleanly to retrieve the simulated events metadata from the millions of database records.

References

[1] AMS-02: http://www.ams02.org
[2] Choutko V, Egorov A, Eline A and Shan B 2015 Computing Strategy of the AMS Experiment *Journal of Physics: Conference Series* **664** 032029
[3] Python: https://www.python.org
[4] ECharts: http://echarts.baidu.com
[5] JSON: http://json.org
[6] JavaScript: https://www.javascript.com
[7] Oracle: https://www.oracle.com/index.html
[8] AMS Computing Website: https://ams.cern.ch/AMS/Computing/computing.html

Acknowledgement

This work is supported by the National Natural Science Foundation of China under grants 61320106007, 61632008, 61602112, 61572129, 61502097 and 61370207; the International S&T Cooperation Program of China 2015DFA10490; the National Science Foundation of Jiangsu Province under grant BK20160695; the Jiangsu Provincial Key Laboratory of Network and Information Security under grant BM2003201; the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China under grant 93K-9; and partially supported by the Collaborative Innovation Centre of Novel Software Technology and Industrialization and the Collaborative Innovation Centre of Wireless Communications Technology.