Monte Carlo Production Monitoring Tool for AMS Experiment

RQ Xiong[1], RL Shi[1], FQ Huang[1], BS Shan[2], V Choutko[3], A Egorov[3], A Eline[3], O Demakov[3], JH Zhang[1], F Dong[1], JZ Luo[1]

[1]*Southeast University* [2]*Beihang University* [3]*Massachusetts Institute of Technology*

Abstract

Monte Carlo simulation production plays an important part in physics analysis of the Alpha Magnetic Spectrometer experiment. To facilitate the metadata retrieving for data analysis needs among the millions of database records, we developed a monitoring tool to analyze 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.

Architecture and workflow

- The architecture of the monitoring tool is as follows:

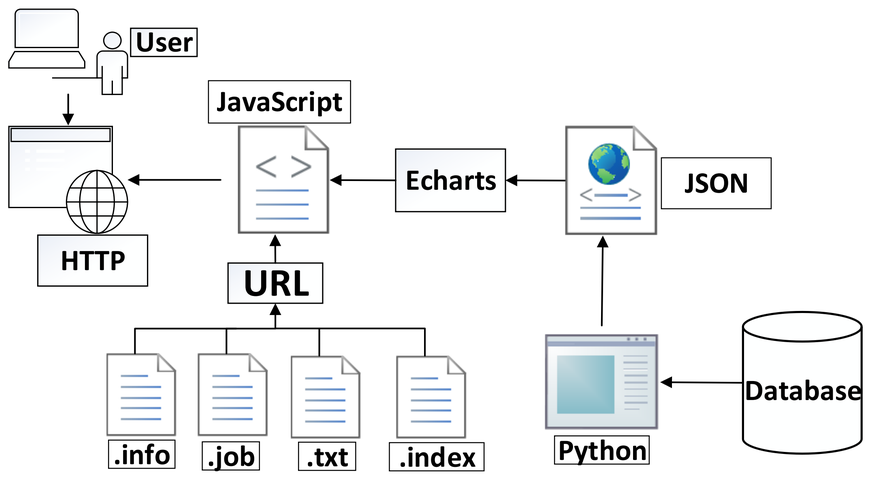
Python program is performed regularly to calculate the total number of finished events, which means REALTERIGGERS in the database, and the statistical CPU time, depending on different computing centers and template, stored the number in JSON. Those JSON files are presented by Echarts. Detailed information of each template, including info file, job file, index file, is showed by URL. Echarts and URL both are organized by JavaScript showing in Website.

Index file: shows the template name of the finished jobs and the location path of the dataset.

Job file: shows the contents of the jobs.

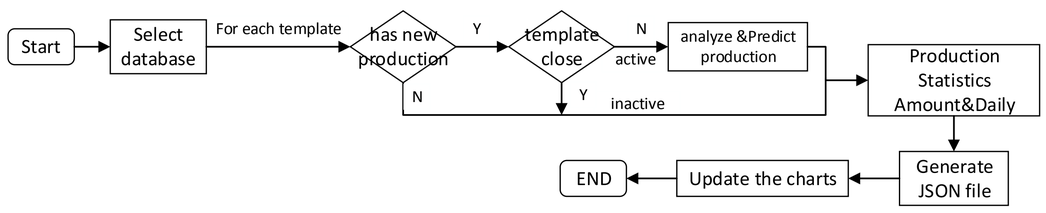
Txt file: the location path of every root file.

Info file: detail information of every job in the template



- The workflow of the monitoring tool is as follows:

There are two python programs used to prepare the JSON format data, statistical number of finished events for each template and statistical CPU time for each computing center, which is refreshed periodically.



Features

- Detailed statistical data:
  - ➢ remaining amount, CPUtime left and predictive completion time of each dataset
  - ➢ remaining amount, CPUtime left and predictive completion time of each template
  - ➢ the finished events of each computing center for each template.
  - ➢ the prediction of production progress for each template.

- The Python/Echarts/JavaScript:

Python：Select data from database and generate JSON files.

Echarts: Generate charts to show the data.

JavaScript: Generate the web page, using AJAX to retrieve JSON format data

Technical details

- The design of the databases tables:

The total CPU time of each computing center and the finished events of each template are selected from amsdes.jobs.

amsdes.jobs

Name	TYPE	Introduction
JID	NUMBER(38)	Job ID
JOBNAME	VARCHAR2(255)	Job name
CID	NUMBER(38)	Computing centers ID
DID	NUMBER(38)	Dataset ID
...	...	...
TRIGGERS	NUMBER(38)	Total number of events
TIMESTAMP	NUMBER(38)	Time stamp
CPUTIME	NUMBER(38)	Cpu time used to finish the job
REALTERIGGERS	NUMBER(38)	The number of finished events

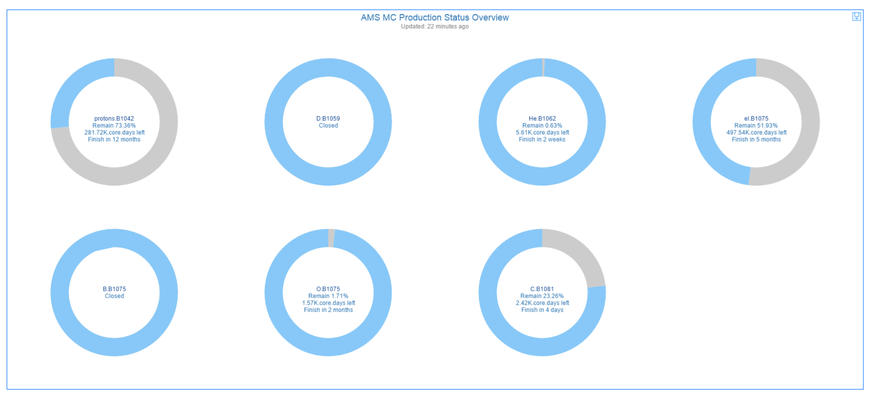
- Core API implementations:

We use Echarts to generate the charts. Here is an example which is the core code for building the Histogram of CPU time statistics with Echarts' API.

```
toolbox: {
 show : true,
 feature : {
  mark : {
   show: false,
   title: {
    mark : 'Marking line',
    markUndo : 'Delete marking line',
    markClear : 'Clear marking line'
   }
  },
  dataView : {
   show: true,
   title : 'Data view',
   readOnly: true,
   lang: ['Data View', 'Close', 'Refresh'],
   optionToContent: function(opt) {
    axisData = opt.xAxis[0].data;
    var series = opt.series;
    var table =
    '<table class="table table-striped"><tbody>'
    +'<tr><th>Site</th>CPU Time</th></tr>';
   for (var i=0,l=axisData.length;i<l;i++) {
    table+="<tr><td>"+axisData[i]
     +"</td>"+series[0].data[i]+"</td></tr>";
   }
   table += '</tbody></table>';
   return table;
  }
 },
 restore : {
  show: false,
  title: 'Restore'
 },
 saveAsImage : {
  show: true,
  title: 'Save as image'
 }
},
```

Demonstration

- AMS MC production Overview



- Active Production & Inactive Production

- Site Summary- CPU Time