Design and implementation of resource and performance monitoring tool for computing cluster

Anirban Das[1], Raman Sehgal[2], Dipanwita Dutta[2]

[1]*Electronics and Communication Engineering Department, Jalpaiguri Government Engineering College, Jalpaiguri-735102, INDIA*
[2]*Nuclear Physics Division, Bhabha Atomic Research Centre, Mumbai - 400085, INDIA*
* email: anirban.dash@gmail.com

INTRODUCTION

Various physics experiment like the Large Hadrons Collider (LHC) will store data corresponding to millions of events per second. A job of High Energy Physics is usually a reconstruction or analysis program running on these events. Such a job can be parallelized on many computing nodes where same program is running on different sets of events. High throughput computing or distributed computing thus plays a great role in high energy physics area. The key to High-Throughput Computing (HTC) is effective management and exploitation of all available computing resources in a fair manner especially if the computing environment is distributed. A large capacity distributed system like clusters will give its best output only if an efficient monitoring tool continuously keeps on profiling the system components. The information thus gathered will be very useful for administrators for troubleshooting system malfunction and also verifying usage of system resources. In this work we are presenting a monitoring tool that can be used for monitoring the computing cluster.

THE LINUX PROC FILE SYSTEM

At the core of our application lies the proc file system of Linux. The proc [1] file system acts as an interface to internal data structures in the kernel. It contains information detailing system hardware and process information. It is possible to access enormous details about the system by using different commands on files within /proc/ directory. In our application we are using this facility to procure our required information.

ARCHITECTURE

The application has been made keeping in mind that clusters are a group of homogenous machines usually connected together by a high speed Local Area Network (LAN) and that we have to display all concerned information on a Web based User Interface (WUI). The first one is absolutely necessary for this application to work and the second one is for better user accessibility. The hierarchy of the application tool has three phases
(1) a backend script for fetching the data (2) storage in a database (i.e. the server-side) (3) frontend part displaying the data in a WUI in graphical format.

The philosophy is to fetch data from the respective nodes (Linux systems) of the cluster using the shell script and these scripts will be scheduled to run periodically using CRONTAB system program that will run a shell script on the background. The aim is not to have any effect on the system performance but at the same time it will fetch data from /proc file system and store/dump it in the database. We have chosen MYSQL as our database. For the server-side data acquisition part from database we are using the well-established PHP scripting language. In our case we are embedding jQueries inside our PHP script to fetch our required information from MYSQL database. jQueries[2] is a Javascript library that immensely simplifies programming with JavaScript providing very easy to use, condensed syntax to achieve even apparently complex tasks with JavaScript. The data is obtained in the JavaScript Object Notation (JSON) format because of its simplicity and easy processing as a data exchange format. Moreover on the client side we are using JavaScript, JSON being a subset of JavaScript so it's very efficient to work on.

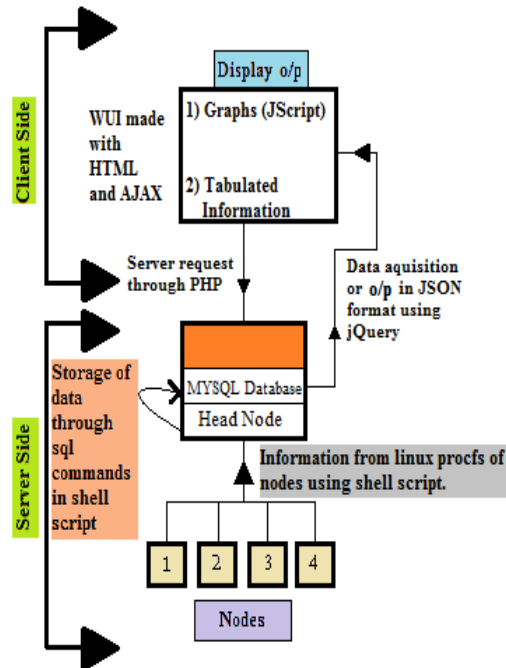Figure 1 below shows the schematic of this monitoring tool

Fig 1: Schematics and Architecture of the cluster monitoring application

WUI FRONT-END

We are parsing the JSON string obtained from the jQuery into a JSON object using JavaScript and feeding the data in our charting library; jQuery is being used to make client side scripting in HTML easy and for better funtionability. A client side charting library AMCHARTS [3] is being used, for displaying both current and historical data. Client side graphing utilities can produce more dynamic graphs as well as reduce the time delay compared to server side graphing utilities. Display is made using HTML, JavaScript and AJAX (Asynchronous JavaScript and XML). AJAX makes the web interface more responsive and dynamic and we can selectively transfer data of a portion of the web page instead of sending the whole page for each request. This results in very little waiting time and also net traffic to and from the server is reduced to a large extent

FUNCTIONING OF THE MODEL

Once inside the interface, the user can see the specification of full cluster. The user can view system status both graphically and in tabular format (in some cases) by clicking on respective fields from a menu. Currently we are displaying information regarding CPU utilization , number of processes running by individual and different users, system memory status, HDD status, core temperatures, network statistics etc. All this information is also getting stored in MySql database. So using this tool user can see the current system state as well as he can also see the historical data and machine usage pattern.

CONCLUSION

In this paper we have described a cluster monitoring tool that will display the status of the cluster onto a WUI. The primary goals were to make the application as efficient and fast as possible as well as easy user accessible; at the same time more features can be easily included by inserting relevant items in the source code. JavaScript, jQuery, JSON, client side scripting and AJAX are all being used keeping the aforesaid goal in mind.

REFERENCE

[1] http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/3/html/Reference_Guide/ch-proc.html
[2] http://www.jscripters.com/what-is-jquery-presentation/
[3] http://www.amcharts.com/javascript/