McPerM – THE McFarm Performance Monitor

Prashant Bhamidipati, Drew Meyer, Jae Yu, Mark Sosebee, David Jenkins, Nirmal Ranganathan

Introduction

McPerM is a performance-monitoring tool based on McFarm. It's main purpose is to display the overall performance statistics of a farm with respect to the number of events processed, the size of the files generated and the relative transfer times. McPerM uses XML as the data transport medium and Ploticus as the interface to display the requested statistics.

Structure

In a nutshell, McPerM performs the following tasks:

1. It converts archived information on a remote farm into XML files on a per-job basis.
2. The XML files are collected by the main-server displaying the information using Globus toolkit.
3. This information is then converted into a McPerM readable database that is again in XML but follows a different structure.
4. This McPerM database is parsed every time there is a request to read information and displays the gathered statistics using Ploticus.

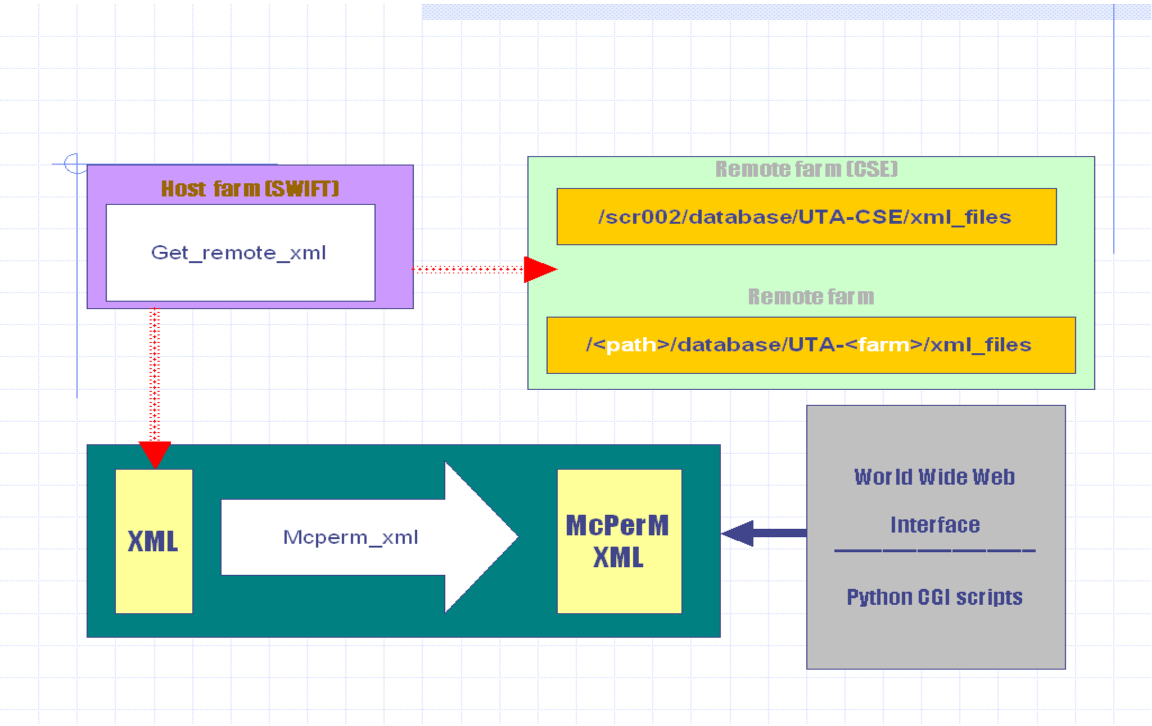

Figure 1 : ***Brief structure and concept of McPerM***

Components

There are two components to McPerM

1. ***McP_remote***
2. ***McP_server***

McP_remote : This unit resides on the remote farm. A remote farm is a farm which is different from the farm where the main server resides. The main server is the server which hosts the web server and the main McPerM database.

The *McP_remote* performs the function of looking at the archives and extracting information from the 'gather.log' files of the archived jobs. These jobs are then stored in a pre-defined location waiting to be picked up by the main server farm.

McP_server : This unit resides on the main farm which hosts the main server (including the web server) that displays the information. Currently this farm is the UTA-SWIFT farm. It determines which files are to be fetched from the remote farms, looks in the pre-defined database location above and fetches those files which are new and which have not yet been fetched from that farm.

1. *McP_Remote*

- The remote XML module converts job related information at the remote farm into XML format.
- Presently it includes the job specific performance statistics (end time, size, events, file size and file transfer duration) for each phase *(pythia, dØgstar, dØsim, dØreco, thumbnail)*
- There is one important configuration file which has to be created on the remote farm which is located at …/mcperm/conf/<farm_name>.conf
- The following are the contents and the explanation for that file:

```
farm_name=UTA-DPCC
host_name=grid.dpcc.uta.edu
university=UTA
globus_fetch=enable
main_database_location=/blah/mcfarm/mcperm
```

- This file is created when the following command is run at the command prompt

```
make_dir UTA-DPCC grid.dpcc.uta.edu UTA enable
```

where :-

- *UTA-DPCC* is an acronym for the farm name desired by the remote administrator
- *grid.dpcc.uta.edu* is the name of the globus head node that needs to be contacted during file transfers and trouble shooting
- *'UTA'* are the first three characters of the mcfarm-produced jobs on the remote farms. These are the first three characters and may also look like 'LU-' or 'OU-'

- *enable* is an option which specifies whether the XML files on the remote farms need to be deleted after fetching

- This XML module can be modified to include further relevant information.

2. *McP_Server*

- The server application running on the UTA farm server is responsible for fetching the remote XML files from other farms.
- The specifics of the fetch such as the node to contact and the location where the remote XML files are located is controlled by the main.conf file in the ../mcperm/conf/main.conf
- Globus toolkit is used to perform the function of fetching the remote files based on the entries in the main.conf file.
- **The parameters in the main.conf file need to be separated by commas and must have no spaces**
- It stores the XML files in a systematic fashion based on the name of the remote farm and creates a database which can be parsed for various and information extracted for various applications.

The contents of the main.conf file are depicted below in the desired format:-

```
local_host_name=hepfm007.uta.edu
local_location=/scr001/database
farm=UTA-DPCC,/data2/phys/d0/mcfarm/mcperm/database,UTA,grid.dpcc.uta.edu
```

- *local_host* is the name of the globus head node on the server farm
- *ocal_location* indicates where on the 'server' farm the database has to be located it may be any location which has enough space to accommodate all the farms desired
- *Farm* is the name of the remote farm from which information is to be fetched

The line UTA-DPCC,/blah/mcfarm/mcperm/database,UTA,grid.dpcc.uta.edu :

- UTA-DPCC is the remote farm
- /blah/mcfarm/mcperm/database is the name of the database location on the remote farm from which the files are to be fetched
- UTA are again the first three characters of the farm's file names
- grid.dpcc.uta.edu is the name of the globus head node for that remote farm

- The gathered XML information is converted into a McPerM readable format.
- The McPerM readable format is a different format that is based on the dates and the phases of operations contained in each job.

3. *McPerM web frontend*

- The McPerM web front-end uses Python as the CGI scripting tool. It searches for the requested information from the McPerM database based on the farm name and the phase of the process desired to be viewed. Th scripts involved here are :

- *new_m* is the main html page which is the web interface for McPerM
- *new_cgi.py* is the main CGI script that *new_m* refers to for all search operations once the user clicks the 'speed' or the 'events' or the 'size' option
- *new_mpspeed_multi* and *new_mpdata_multi* are the modules which contain scripts for determining which data to search for based on the user request
- *new_xml_function* and *new_mcperm* are the XML parsing modules which are used by the web frontend to parse the McPerM XML database and return the necessary figures for the *new_cgi_multi* to display

Database design

XML database

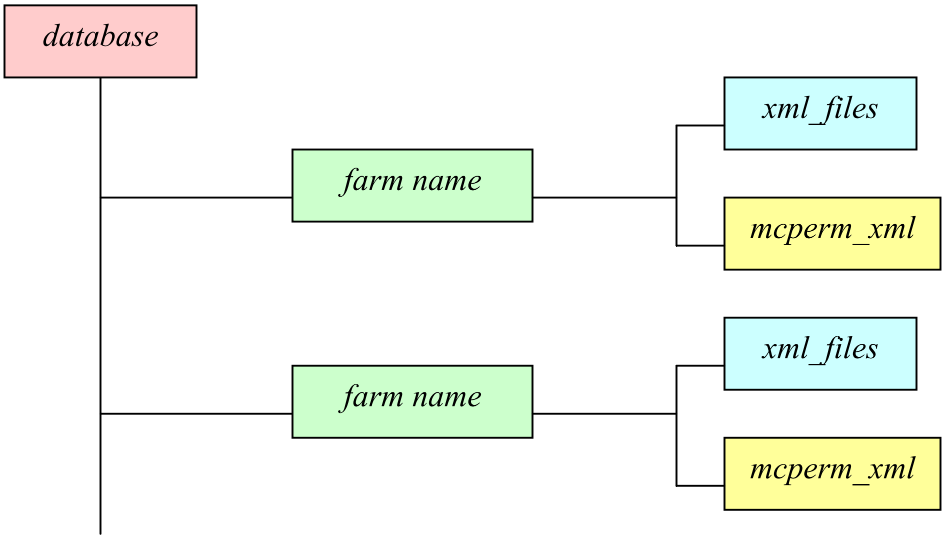

Figure 2 : ***Database structure at main server end***

This is a representation of the main server database. This database resides on the main server that is being used to monitor multiple farms. The main server database is divided into a directory structure according to the farm names of the remote farms.

Each individual 'farm name' folder has two folders called 'xml_files' and 'mcperm_xml' xml_files is the folder which contains the per job XML files fetched from the remote farms

‘mcperm_xml’ is the folder which contains the ‘mcperm_xml’ database
McPerM database

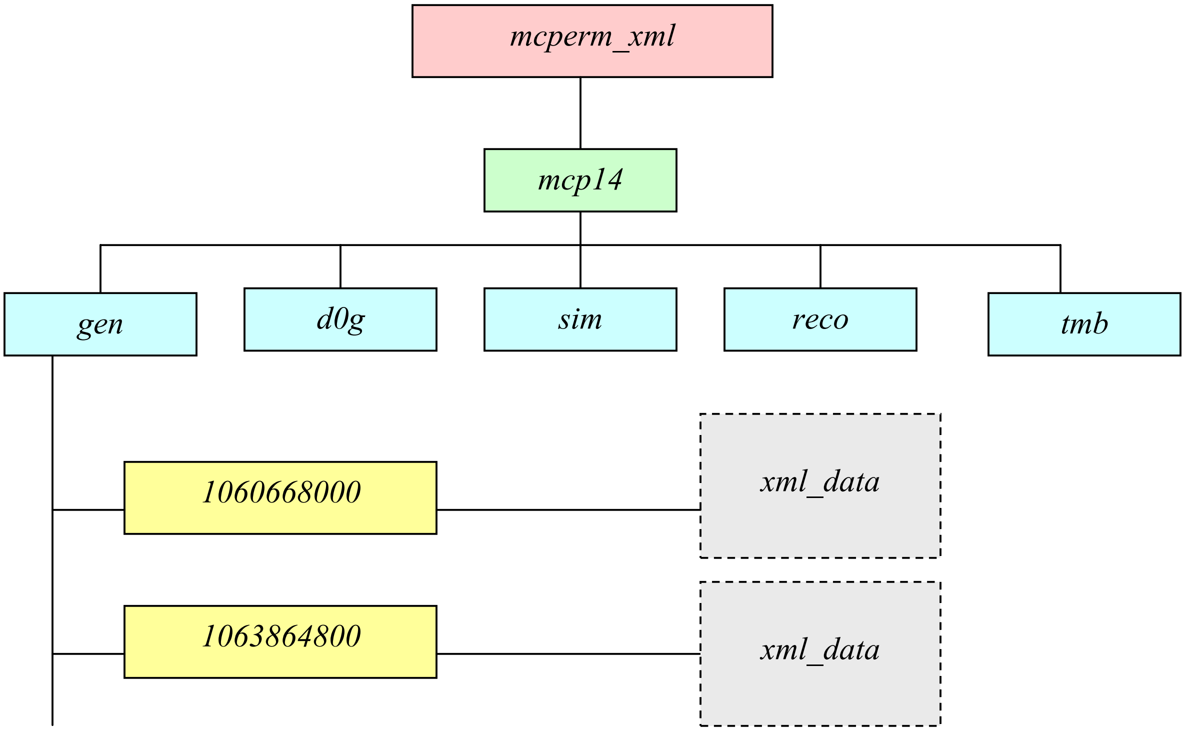

Figure 3 : ***McPerM database structure at main server end***

The McPerM database is based on the phases that the Monte-Carlo simulation jobs go through.

The first segregation is made on the basis of the mcp version of the jobs being reported. There are mcp13, mcp14 etc..

Then come the phases on the basis of which the jobs are split. These are pythia (gen), d0gstar(d0g) d0sim(sim), d0reco(reco) and thumbnail (tmb).

In each of these files there are folders which are named according to the figure generated by the gmtime() function in python.

There is a small file called xmldata in each of these folders and this folder contains the figures and other statistics for that day.

Each of these pythia, gen etc.. folders contains a contents.dat file which is a log of the dates for which there are entries in the database. If there is no entry for a particular date then the value for that date is taken as zero.