Short-term studentship report: LHC@home

Leonardo Cristella
University & INFN of Bari
Via Giovanni Amendola, 173 - 70126 Bari, Italy
E-mail: `leonardo.cristella@cern.ch`

Supervisors: **Peter Richardson** and **Laurence Field**
CERN
CH-1211 Geneva 23, Switzerland
E-mail: `peter.richardson@cern.ch` *and* `laurence.field@cern.ch`

LHC@home [1] is a volunteer computing platform for anyone to donate idle time on his computer to help physicists compare theory with experiment in the search for new fundamental particles and answers to the open questions about the Universe. It consists of several projects: *ATLAS@home*, *SixTrack*, *Test4Theory*, *Beauty*, *CMS@home*; each of them uses the BOINC system [2] that enables jobs to be run for the physicists on volunteer nodes, rather than on clusters or commercial clouds. My short-term studentship goal was to include *Test4Theory* jobs in HTCondor [3] and investigate running intensive workloads with *CMS@home* jobs. *LHC@home* has shown an unpredictable success, with 10,000+ simultaneously running jobs and a community of more than 15,000 volunteers from more than 200 countries supporting the project.

1. Introduction

Volunteer computing has the potential to supply scientific research with significant computing power by tapping into the huge number of personal computers in the world. It also encourages public interest in science and offers an easy way for almost anyone to contribute directly to scientific research.

My short-term studentship had two objectives:

1. use HTCondor (HTC) as submission tool for the *Test4Theory* jobs to handle their queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management; HTC has proved to be a very efficient tool, therefore the migration for other LHC@home projects is also foreseen. Furthermore, support for the present CoPilot [4] job scheduler used by *Test4Theory* is finishing.
2. investigate running intensive workloads with *CMS@home* jobs in preparation for the Kansas City challenge [5], which will exploit the performance of Google Fiber in Kansas City [6].

2. HTCondor for *Test4Theory* jobs

The previous *Test4Theory* job queueing mechanism, CoPilot, is described in Fig. 1. In order to replace the CoPilot block with HTC, a script that submits HTC jobs whose executables are taken from the MCPlots [7] input pool has been set up.

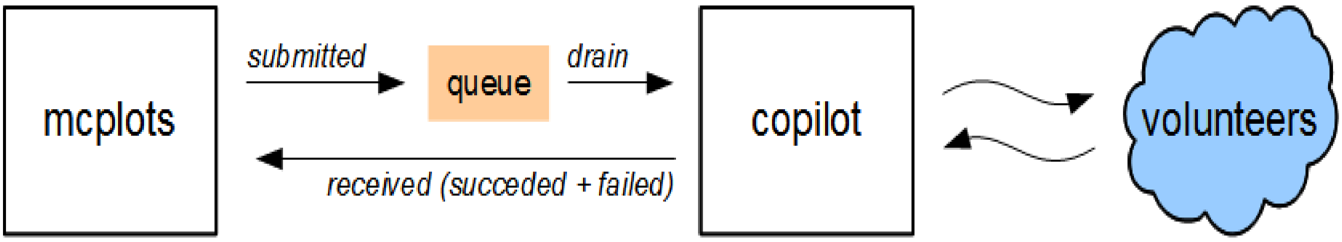

Figure 1: Previous MCPlots jobs queueing mechanism.

HTC jobs land on the volunteer resources available and, once completed, their output is retrieved into the MCPlots output pool by the same script. The script is currently running as a cron job on a Linux virtual machine (VM) launched through the CERN OpenStack resources [8]. Once HTC will be installed on the MCPlots machine, the script can be moved from the VM to the MCPlots machine, allowing for a simpler input/output handling.

Fig. 2 shows the moment (June 6th, 2016) in which the transition from CoPilot to HTC started.

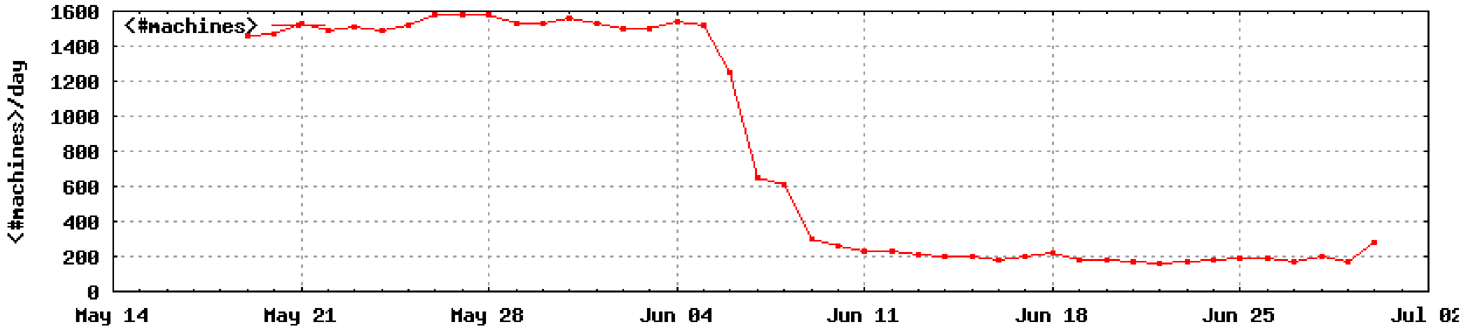

Figure 2: MCPlots jobs trend

3. Running intensive workloads with *CMS@home* jobs

The simulation workflows, a.k.a. Monte Carlo (MC) requests, needed by the CMS Collaboration [9] to compare theoretical models with the data collected through its detector consist of very different jobs in terms of CPU/bandwidth required. Volunteer resources are also largely spread in performances among different hosts, hence a proper matching between job requirements and host resources would result in a more efficient usage of the volunteer devices. For this purpose the VM provided to the volunteers has been equipped with a bandwidth measurement which can be used by HTC to select the most suitable jobs to be run on the host. This feature will be fundamental in the future Computing Challenge that will take place in Kansas City. The goal of this challenge is to exploit the Gigabit network installed in Kansas City by the Google Fiber program. Therefore, by enabling data intensive jobs on the volunteer hosts provided with a high bandwidth connection, the whole network infrastructure performance will be measured in terms of CMS MC completion.

To begin with, an official CPU intensive MC request has been submitted through *CMS@home* (which has been delivered for production in March 2016) for the first time. The particular test workflow was chosen for its very low production efficiency, which prevents it to be run by the standard CMS Computing resources. In this way CMS volunteers can play an important role within the CMS Collaboration and help physicists in the search for new fundamental particles and answers to the open questions about the Universe.
My PhD has benefited a lot from this resource as I was able to generate hundreds of million of MC events in a few days, allowing to reduce the statistical uncertainty in my analysis.

4. Future perspectives

I am about to graduate and I plan to start a postdoc in the same research group of my PhD.
In such a way I will have the possibility to submit many data intensive *CMS@home* jobs thanks to the bandwidth measurement provided to the volunteers. Once a sizeable sample will be collected a paper on this new volunteer computing paradigm is likely to be published.

5. Teaching and Training Activities

During my short-term studentship I participated in the MCNet Network meeting in Göttingen (April 4-6, 2017). I found the meeting very useful as I got in touch with many people involved in the different MCNet areas and had fruitful discussion on how my project could be of most benefit for the network.

References

[1] LHC@home, http://lhcathome.web.cern.ch/
[2] BOINC, http://boinc.berkeley.edu/
[3] HTCondor, https://research.cs.wisc.edu/htcondor/
[4] A Harutyunyan et al, CernVM Co-Pilot: an Extensible Framework for Building Scalable Computing Infrastructures on the Cloud (J. Phys., 2012)
[5] Kansas City Challenge, https://cernkcchallenge.github.io/CernKCChallenge/
[6] Google Fiber in Kansas City, https://fiber.google.com/cities/kansascity/
[7] MCPlots, http://mcplots.cern.ch/?query=frontpage
[8] CERN OpenStack, https://clouddocs.web.cern.ch/clouddocs/
[9] CMS, http://cms.web.cern.ch/content/cms-collaboration