ATL-SOFT-PROC-2016-015
28 December 2016

Volunteer Computing Experience with ATLAS@Home

C Adam-Bourdarios[1], R Bianchi[2], D Cameron[3] A Filipčič[4], G Isacchini[5], E Lançon[6] and W Wu[7] for the ATLAS Collaboration

[1] Laboratoire de l'Accélérateur Linéraire, BP 34, F-91898 Orsay Cedex, France
[2] University of Pittsburgh, Department of Physics and Astronomy, 100 Allen Hall, 3941 O'Hara St, Pittsburgh PA 15260, USA
[3] Department of Physics, University of Oslo, P.b. 1048 Blindern, N-0316 Oslo, Norway
[4] Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[5] ETH Zurich, Rämistrasse 101, 8092 Zurich, Switzerland*
[6] Brookhaven National Laboratory, Upton NY 11973, USA
[7] Computer Center, Institute of High Energy Physics, CAS, 19B Yuquan Road, Beijing, China, 100049

Abstract. ATLAS@Home is a volunteer computing project which allows the public to contribute to computing for the ATLAS experiment through their home or office computers. The project has grown continuously since its creation in mid-2014 and now counts almost 100,000 volunteers. The combined volunteers' resources make up a sizeable fraction of overall resources for ATLAS simulation. This paper takes stock of the experience gained so far and describes the next steps in the evolution of the project. These improvements include running natively on Linux to ease the deployment on for example university clusters, using multiple cores inside one task to reduce the memory requirements and running different types of workload such as event generation. In addition to technical details the success of ATLAS@Home as an outreach tool is evaluated.

1. Introduction

Volunteer computing is the concept of using spare computing cycles on computers when they are not in use to perform a computational task for someone else. People typically "volunteer" their computers for public scientific projects in order to contribute to the greater good (such as searching for extra-terrestrial life or large prime numbers). The first big volunteer computing project was SETI@Home [1], where a program installed on volunteers' computers searched for evidence of extra-terrestrial life in radio signals from telescopes. Since then the volunteer community has grown to over 50 recognised scientific projects and hundreds of thousands of volunteers around the globe.

The majority of volunteer computing projects use software called BOINC (Berkeley Open Infrastructure for Network Computing) [2]. The software comprises a server, which hosts tasks or *work units* to be processed and a client which volunteers install and configure to pull and run work units from specified projects. Once a work unit is processed the client sends the result back to the server which validates the result, and if the result is good awards credit to the volunteer.

*This work was carried out whilst a student at the University of Oslo.

Credit is simply a measure of how much computation has been done and has no monetary value, however it provides motivation for many volunteers.

High-energy physics experiments and facilities have been taking advantage of volunteer computing using BOINC for many years, starting in 2004 with the LHC@Home project [3] which was set up as part of CERN's 50th birthday celebrations. In early 2014 a volunteer computing project was started within the ATLAS experiment [4], ATLAS@Home [5]. In this project, volunteers run Monte-Carlo simulation of particle collisions in the ATLAS detector. This type of simulation is a vital part of the overall process of analysing data from ATLAS as it provides both detailed information on the detector itself and a means of comparing the observed data against theoretical models. The computing requirements of simulation tasks match well to volunteer computing compared to other types of ATLAS tasks as they use lower memory and are less data-intensive.

The ATLAS@Home project has two main purposes: to provide extra opportunistic resources and to involve the general public directly in ATLAS data processing. The latter reason is a key part of volunteer computing, allowing people have a direct connection with scientific research. The volunteer base has constantly expanded over the last two years and several changes were made by the project team to improve the experience of the volunteers and to maximise the performance of the platform. The rest of this paper describes these changes and their effects in detail. In Section 2 the implementation of multiple-core tasks is described, then in Section 3 a graphical interface for engaging the volunteers is presented. In Section 4 ongoing work to integrate the ATLAS event display into ATLAS@Home is reported and finally in Section 5 some conclusions about the overall volunteer experience are drawn.

2. Implementation of Multiple-core Tasks

ATLAS@Home initially used a single-core application, i.e. each BOINC task spawned a single-core virtual machine via VirtualBox [6] to run a single-core ATLAS simulation task. If a volunteer host allocated multiple cores for the project to use, then multiple virtual machines would be spawned on the host to run multiple single-core ATLAS tasks.

A single-core virtual machine is inefficient in terms of memory, hard disk and network usage. A single core ATLAS task requires around 2.3GB of memory, and that is the amount of memory allocated to each single-core virtual machine. Given the fact that most volunteer hosts are equipped with 2GB RAM per CPU core, this approach excludes full usage of the available CPU cores that many volunteer hosts are willing to contribute to the project. For example, even if a volunteer host allocates 4 CPU cores to the project, but it has only 8GB RAM, given the memory requirement from the ATLAS task, it can only use 3 CPU cores from this host. Also, running multiple virtual machines on a single host requires using more hard disk space for storing both the virtual machine images and the extra hard disk attached to the individual virtual machine. In general, each ATLAS@Home virtual machine uses around 10GB hard disk from the volunteer host. Inside the image, it uses the CERNVM File System (CVMFS) [7] to distribute the ATLAS software, and even though most of the software is already cached in the virtual machine image, each ATLAS task still needs to download files from the CVMFS servers while it is running. As the CVMFS cache cannot currently be shared among different virtual machines running on the same host, each virtual machine has to download the necessary files separately, increasing the network usage.

In BOINC, there is a piece of software named vboxwrapper which runs on the BOINC client side to control the creation of virtual machines, and it supports allocating multiple cores to a single virtual machine. As shown in figure 1, on the BOINC server side, there is already a mechanism called plan-class, in which the project can define different plan-class tags and attributes, then associate the attributes with different application directories via tags. Attributes include the range of CPU cores and memory size for the virtual machine, and minimum and

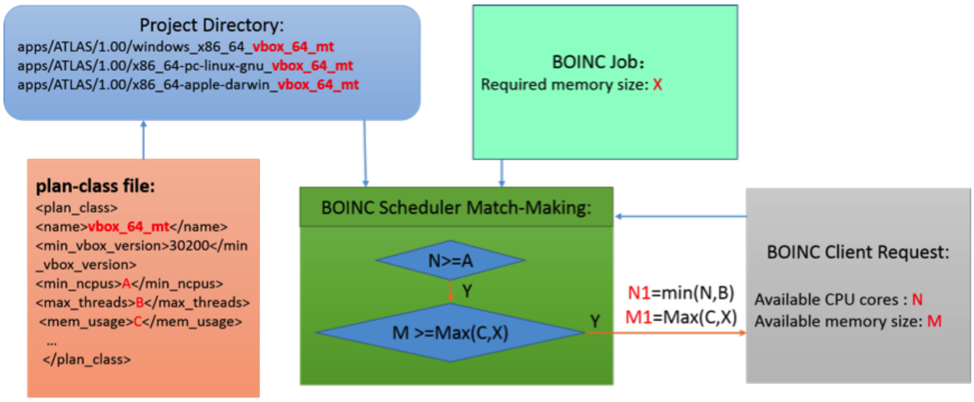


Figure 1: Architecture of multicore scheduling in BOINC.

maximum VirtualBox versions required on the volunteer hosts. All the attributes are passed to the BOINC scheduler which does match-making between the required resources of the task associated to the application and the available resources reported by the client request. However, the original BOINC scheduler only supported a fixed amount of memory in the plan-class regardless of the number of CPU cores, and that means the project had to choose a memory size based on the maximum number of CPU cores. This is not efficient given the memory size is proportional to the number of CPU cores in ATLAS multiple core tasks, as setting up a big memory size will exclude hosts with smaller memory size from running ATLAS tasks.

Since the ATLAS@Home multicore application requires dynamic memory size according to the actual number of CPU Cores used on the volunteer host, the plan-class definition was extended and the BOINC scheduler code was modified. As shown in figure 2, in the plan-class, there are 2 tags used to specify the memory: *mem_usage_base* and *mem_usage_per_cpu*. The overall memory size is calculated by the equation $M = C + N * D$, whereas M is the overall memory size, N is the number of CPU cores allocated to the virtual machine, C is the *mem_usage_base* and D is the *mem_usage_per_cpu*. In the BOINC scheduler, the actual CPU core number also depends on the available memory size from the client.

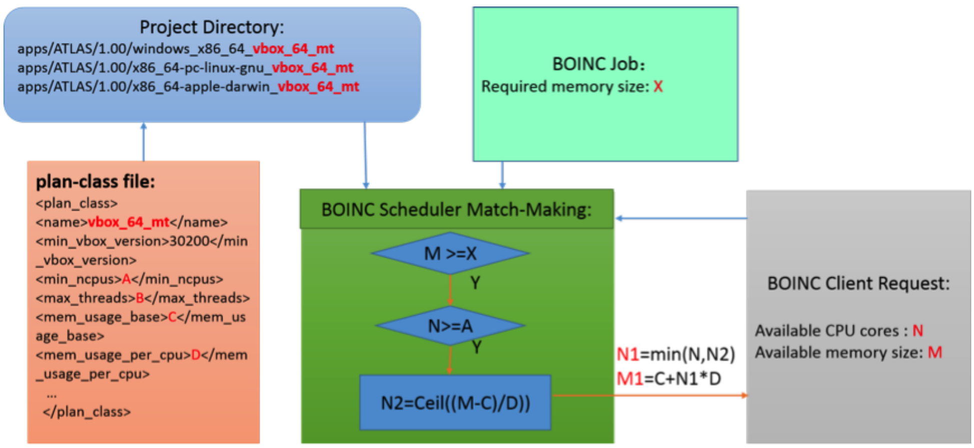


Figure 2: Architecture of multicore scheduling in ATLAS@Home.

The ATLAS@Home multicore application initially set the CPU core range between 1 and 12, so it could efficiently use all available CPU cores that volunteer hosts allocates to the ATLAS@home project. This was especially useful for some of the powerful volunteer hosts.

However experience showed that with the same type of simulation task (simulating 100 events per task) the average CPU time of each event on some hosts was a lot higher than on other hosts. Figure 3 shows the average time in seconds spent processing one simulated event for different numbers of cores. It can be clearly seen that the best performance is obtained with 2-5 cores, while with a higher number the performance is worse than with a single core. Above 8 cores the time increases sharply up to 12 cores where the time per event is almost 4 times longer than with 2 cores. Note that the statistics for unusual numbers of cores such as 3 or 5 are much lower than for 4 or 8 cores so one cannot read too much into the slight differences between 3, 4 and 5 cores.

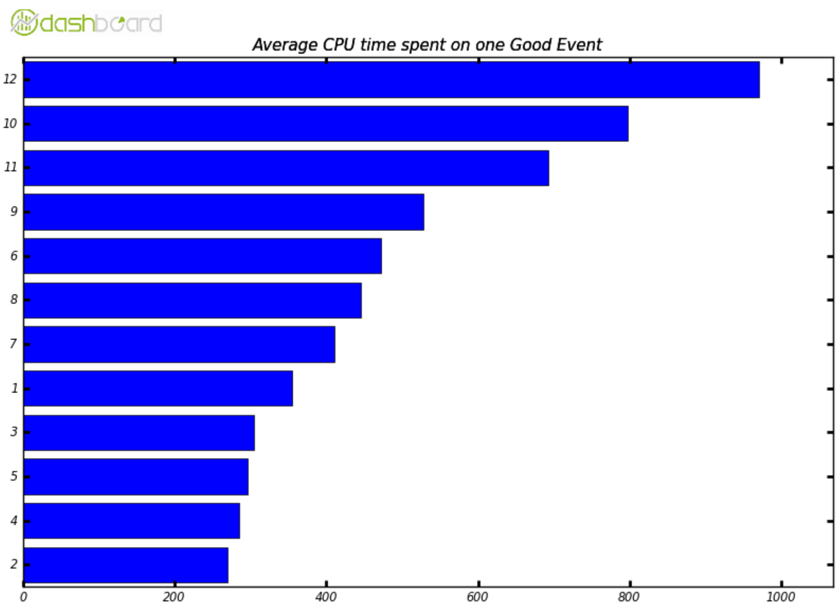


Figure 3: Time in seconds to simulate one event for different numbers of CPU cores.

With further tests and analysis, two causes were found: 1) the older VirtualBox version (older than VirtualBox 5.0.0) performs very badly for multicore virtual machines; 2) Using multicore across physical CPUs leads to bad performance too, namely if a physical CPU has only 4 cores, then creating a virtual machine with more than 4 cores results in using cores across different physical CPUs, leading to a performance drop. As the majority of volunteer hosts have no more than 8 cores on each physical CPU, the maximum CPU cores were reduced from 12 to 8.

Using the multicore application has significantly reduced the memory usage on the volunteer hosts, and also the hard disk and network usage. It enables using more available CPUs from the hosts. As shown in figure 4, more and more ATLAS@home volunteer hosts are running the multicore application after its official launch in July 2016. In mid-November the single-core application was stopped so that only multi-core tasks now run (volunteers can still configure ATLAS@Home to use one core if they wish).

3. Volunteer Graphical Interface

An important goal of the ATLAS@Home project is attracting new volunteers and retaining them. In order to achieve this a graphical user interface was developed, providing attractive and understandable information on the simulations the users process, as well as general information on the ATLAS experiment itself.

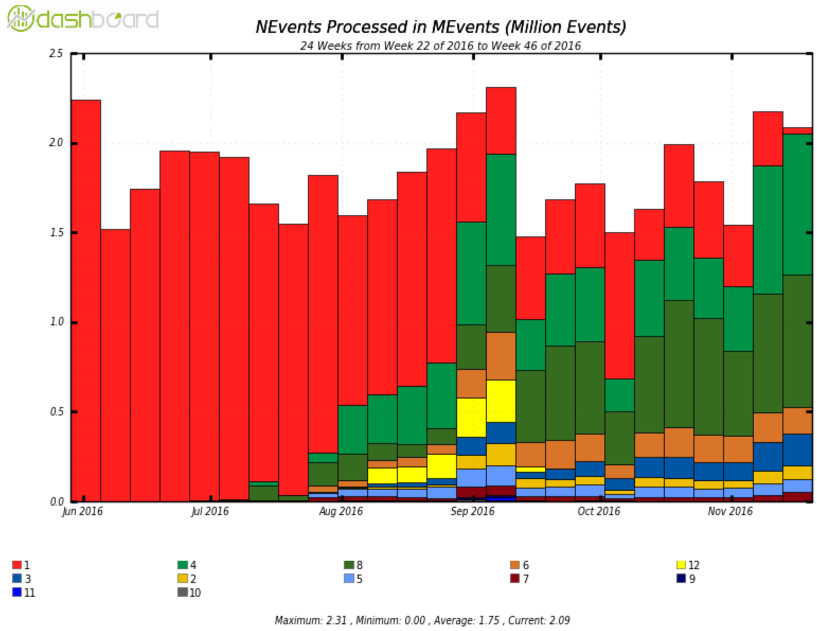


Figure 4: Number of events simulated per week by ATLAS@Home from June to November 2016, grouped by core count.

Many BOINC projects provide visualisation of the running tasks, often through a screensaver which activates when the volunteer's PC is idle. Since ATLAS@Home tasks run inside a virtual machine, the screensaver approach was not possible, but for projects using virtualisation BOINC provides a mechanism to access a web server inside the virtual machine through the PC's web browser. The ATLAS@Home interface was therefore developed as a web service running inside the virtual machine.

The interface is built on javascript, mainly through the P5.js library [8], a sister project of the well known programming language, Processing. Within the interface the users can access, through visual and interactive animations, information about high energy physics in general, the ATLAS experiment and the ATLAS@Home project itself. Since the interface runs inside the same virtual machine as the ATLAS@Home task, it can present task- and volunteer-specific information in a highly personalised way.

The welcome screen is shown in figure 5. It greets the volunteer by name and provides a menu of links to further information including: a brief explanation of the motivations behind ATLAS@Home; numbers showing the volunteer's contribution to the project; basic physics information on particles and forces in the Standard Model with links to ATLAS outreach web pages for further information; and links to the ATLAS@Home message boards for help and interacting with the community.

Figure 6 shows a screenshot of the page explaining the meaning behind the badges that volunteers can earn. Badges are a feature of many volunteer computing projects, and can be earned through certain project-defined achievements. They encourage volunteers to participate more and compete with each other in the badges they accumulate. The theme of ATLAS@Home badges is the Standard Model and volunteers can earn different "particles":

- Quarks for the current 1%, 10%, 25%, etc. top contributing volunteers
- Leptons for length of time on the project (up to 6 months)
- Bosons for years spent on the project

Figure 5: Screenshot of the welcome screen of the graphical interface.

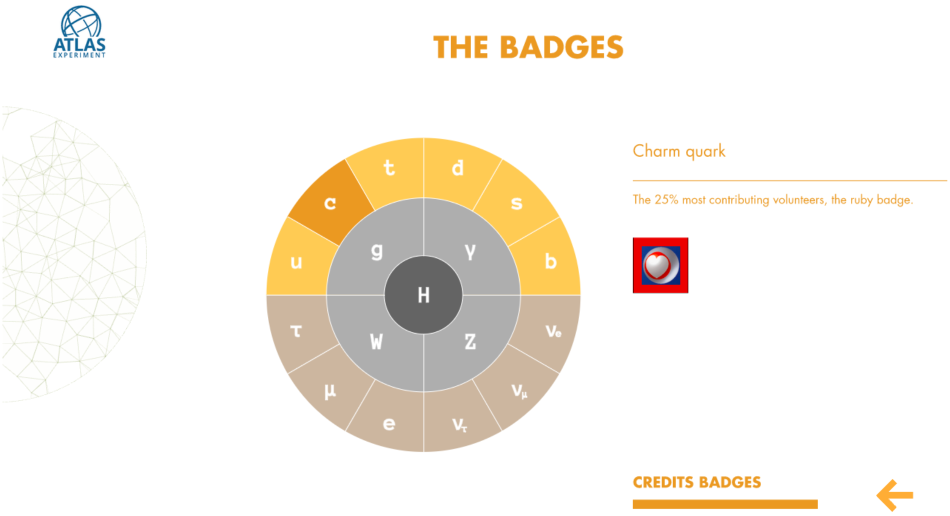
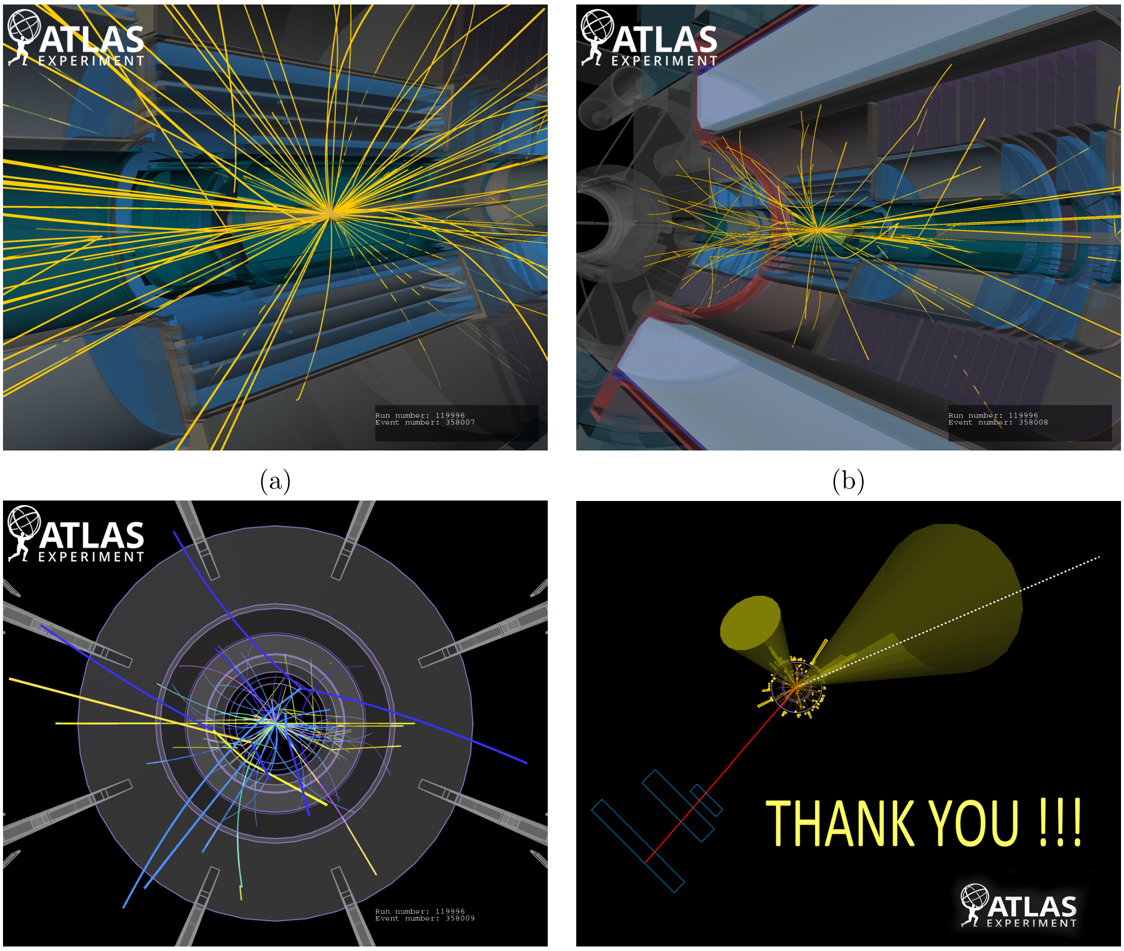


Figure 6: Screenshot of the page explaining the badges of the standard model, highlighting the charm quark badge.

In addition pieces of the ATLAS detector are allocated for total credit earned.

4. Integration with ATLAS Event Display

In order to show more information to volunteers on what exactly is being processed on their PC, there is work on-going to integrate the ATLAS 3D event display, VP1 [9]. Being part of the experiment software framework, VP1 is able to access all the experiment data, and for that reason, it can be used to provide the volunteers with detailed visualizations of the processed data. Different new packages have been developed extending the event display capabilities, to access event data as soon as it has been processed on the volunteer's host and to produce on-the-fly visualizations. Also, a new mechanism has been developed to randomly pick a configuration file from a provided set, while producing the visualizations: the goal is to provide a quite different image for each event processed on the remote host, letting the volunteers see a different picture each time. Examples of different visualizations produced by the new event display tools are



(a) (b) (c) (d)

Figure 7: a–c) An example of event displays generated from different configuration files, which are randomly picked in order to provide users with always-changing images of the processed data; the pictures show the actual geometry of the ATLAS subdectors while visualizing the particles traveling through them. d) An example of the special event display that can be used by the ATLAS@Home managers to interact with the volunteers.

shown in figures 7a–7c. In the pictures tracks of particles are shown as they pass through different parts of the ATLAS detector, as well as their interaction with the detector. Moreover, the pictures show the actual geometry of ATLAS, letting the volunteers peer at the very inner core of the detector, where collisions happen.

Event displays provide the public with an easy to grasp concept of an extremely complex experiment. The images are produced within the virtual machine, then served to the volunteers through the graphic interface presented in the previous section. The graphic interface also serves images taken from an external folder hosted on CERN ATLAS machines, where the ATLAS@Home managers can insert pictures that can act as announcements; they can be used, in fact, during special periods (like Christmas or Easter breaks) or project phases (like data processing campaigns), to interact with the active volunteers or simply to thank them for their contribution, as shown in figure 7d.

At the moment, the integration of the new event display mechanism with the BOINC framework is being developed and tested and it will be included in one of the next releases.

Also, a new mechanism to toggle the event display production on request is being developed, to spare computing resources when event displays are not needed.

5. Conclusion

In this paper the latest developments in the ATLAS@Home project have been presented. These developments have helped to expand the volunteer base and keep existing volunteers interested in participating. One important use case of the original project – to allow an institute to contribute to ATLAS computing without setting up a full Grid site – has also been realised. Two of the top contributors are ATLAS institutes in Munich, Germany and Prague, Czech Republic which are running ATLAS@Home on idle office desktop PCs or common clusters when they are not used by others. There were concerns initially that such volunteers with access to large resources may demotivate the regular volunteers but so far there have been no negative reactions.

However, the vast majority of work is still done by regular volunteers who are not affiliated with ATLAS or CERN. A key part of the project's success is that many of these volunteers are willing to help others in case of problems. The main communication channel with the volunteers is the ATLAS@Home message boards, which is an online forum provided by BOINC software. At the time of writing, there have been more than 5000 posts, mainly concerned with technical issues or questions. More often than not, a problem reported on the message boards will be answered by another volunteer rather than a member of the ATLAS@Home team. Some of these volunteers have much more experience of BOINC projects and provide generous assistance when needed.

The project has grown steadily since the beginning and now on average roughly 2% of all ATLAS simulation events are generated on ATLAS@Home. Compared to an equivalent Grid site the manpower costs or running ATLAS@Home are negligible and the hardware comes for free. In addition the outreach potential is vital for a large scientific experiment like ATLAS.

Acknowledgements

We wish to thank our colleagues in the CERN IT department for providing the hardware and support for integrating ATLAS@Home into the LHC@Home infrastructure. We would also like to thank the developers of BOINC for their helpful responses to our questions and rapid implementation of feature requests. Finally we thank all the volunteers who have contributed to ATLAS@Home, both in terms of computing and invaluable help on the message boards.

References

[1] Korpela E, Werthimer D, Anderson D, Cobb J and Leboisky M 2001 SETI@home-massively distributed computing for SETI *Computing in Science & Engineering* vol **3** pp 78–83

[2] Anderson D 2004 BOINC: a system for public-resource computing and storage *Proc. 5th IEEE/ACM Int. Workshop on Grid Computing, GRID 04* pp 4–10

[3] Herr W, McIntosh E, Schmidt F and Kalchev D 2006 Large scale beam-beam simulations for the CERN LHC using distributed computing resources *Proc. 10th European Particle Accelerator Conference*

[4] ATLAS Collaboration 2008 *J. Inst.* **3** S08003

[5] Adam-Bourdarios, C and Cameron, D and Filipčič, A and Lancon, E and Wu, W on behalf of the ATLAS Collaboration 2015 ATLAS@Home: Harnessing Volunteer Computing for HEP *J. Phys.: Conf. Series* vol **664** p 022009

[6] Watson J 2008 *Linux J.* **2008** ISSN 1075-3583 URL `http://dl.acm.org/citation.cfm?id=1344209.1344210`

[7] Aguado Sanchez C, Bloomer J, Buncic P, Franco L, Klemer S and Mato P 2008 CVMFS-a file system for the CernVM virtual appliance *Proc. XII Advanced Computing and Analysis Techniques in Physics Research* (PoS(ACAT08)012) p 52

[8] p5.js web site URL `https://p5js.org/`

[9] Kittelmann T *et al.* 2010 The Virtual Point 1 event display for the ATLAS experiment *J.Phys.: Conf. Series* vol **219** p 032012 URL `http://atlas-vp1.web.cern.ch/`