BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXII No. 2/2010	72 - 79	Seria Matematică - Informatică - Fizică

Has Open Source Prevailed in Desktop Grid and Volunteer Computing?

Monica Vlădoiu

Petroleum-Gas University of Ploiesti, Bd. Bucureşti nr. 39, Ploieşti
e-mail: monica@unde.ro

Abstract

Grid Computing provides for using resources that are both geographically and administratively distinct, which can be reached over the computer network such as processing power, storage capacity, specific data, input and output devices, etc. Individual users may access computers and data transparently, without having to consider location, operating system, account administration, etc. that are abstracted from them. Desktop grid and volunteer computing systems are one major category of grid systems, in which cycles are scavenged from idle desktop computers, either within an intranet or over the Internet. In this paper we will address the question whether the usage of desktop grid and volunteer computing systems to compute-intensive projects is influenced or not by their open source-ness. We overview here briefly the main such systems and show their main characteristics, emphasizing their most important strengths and weaknesses. Our major concern is the impact of the open source-ness of these desktop grids on their life span and their use scale within projects worldwide. Our opinion is that the answer to the question in paper's title is "yes", and that open source software is a natural choice for desktop grid and volunteer computing, as well as for modern research, because it encourages successful integration, cooperation and boosting of new ideas.

Keywords: *grid computing, desktop grid, volunteer computing, open source, compute-intensive*

Introduction

Grid Computing is a paradigm of distributed computing that utilizes resources that are both geographically and administratively distinct, which can be reached over the network such as processing power, storage capacity, specific data, input and output devices, and so on. Foster's canonical definition of Grid states that *it is a system that coordinates distributed resources using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service* [6]. In grid computing, individual users access computers and data transparently, without having to take into consideration location, operating system, account administration, etc., which are abstracted from the users. The main goal of grid computing aim is to achieve a safe, controlled and flexible sharing of virtualized resources between a range of dynamically created virtual organizations. However, to construct an application that takes advantage of what grid computing has to offer (namely increased execution speed, linking together of geographically disparate resources, interoperation of software, and so on), usually requires the installation of intricate supporting software, and, moreover, of in-depth knowledge of how this software works.

There are available two broad categories of grid computing systems: first, the heavy-weight, feature-rich systems that are concerned mainly with providing access to large-scale, intra- and inter-institutional resources such as clusters or multiprocessors. The second general category of grid computing systems consists of Desktop Grids, in which cycles are scavenged from idle desktop computers. One major advantage is that the costs are very distributed, as each and every volunteer supports the expenditures for his or her resources (e.g. hardware, internet connections etc.), while the benefited entity sustains the necessary infrastructure (e.g. management services, network bandwidth etc.), getting in return a huge computing power.

The typical and most appropriate application for desktop grid systems consists of independent tasks (with no communication with each other) that have a high computation to communication ratio. The execution of an application within a desktop grid environment is managed by a central scheduler node, which distributes the tasks to the worker nodes and waits for their results. An application is completed only when all its tasks have been finished.

Desktop grid systems progress in two major directions: institution- or enterprise-wide desktop grid computing environment and volunteer computing. The former, usually called simply *desktop grid*, describes a grid infrastructure that is restricted to an organization's boundary, and it works by scavenging the spare resources of the organization's desktop PCs to support the execution of internal computationally-intensive applications. Users do not usually participate voluntarily within this infrastructure, they having to comply with the organization's policy with respect to this matter. The later provides for volunteer computing, a paradigm that is based on the users' willingness to provide their temporarily unused resources for external compute-intensive projects. In this framework, the resources are scavenged both from members of the general public or from organizations, provided that they own Internet-connected computing resources. Generally, the projects are academic and are concerned with scientific research.

Besides the source of the resources, the two kinds of desktop grid systems differ significantly with regard to the dimension of the applications that they can tackle. Thus, organizations' desktop grids may approach relatively small size applications compared with volunteer computing projects that are composed of a tremendous number of tasks. Therefore, users of desktop grids are usually interested in a rapid execution of their applications (fast turnaround time), whereas for volunteer computing projects prevails the number of tasks carried out per time unit (high throughput).

In this paper we will approach the question suggested by the title, being concerned with the impact of the open source-ness within desktop grid and volunteer computing world. The paper's structure is as follows: the second section presents briefly the main systems for desktop grid and volunteer computing, the third section includes considerations on the influence of the open source-ness of these desktop grids on their life duration and their use scale within projects around the world, taking into account also the main characteristics of each of the main systems for desktop grid and volunteer computing, and the last section concludes the paper, answering the question from the title.

Main Systems for Desktop Grid and Volunteer Computing

In this section we will overview briefly the main desktop grid and volunteer computing systems either still available today or that have played an important role in the field.

SETI@home - BOINC

SETI (Search for ExtraTerrestrial Intelligence) is concerned with establishing the existence of intelligent life outside Earth, mainly by listening for artificial radio signals coming from other stars. Initially, the SETI projects have been using supercomputers to analyze the gathered data. Starting with 1995, another groundbreaking idea came to life, i.e. to use some sort of a virtual supercomputer composed of a large number of Internet-connected computers [23]. SETI@home, which has been developed at the University of California in Berkley, is a radio SETI project that provides for anyone having a computer and an Internet connection to contribute. The systems works very simple by using a screen saver that can get a chunk of data from a central server over the Internet, analyze that data, and then report back the results. The screen saver runs only when the volunteer's computer is idle, and when it is needed back by its user, the screen saver immediately gets out of the way and it only continues its analysis when the computer is idle again. The analysis can be effortlessly divided into small parts, with no dependencies between them, that can all be worked on separately and in parallel.

As for the evolution of the SETI@home project, because it has been devised for a very specific problem, there was no general framework, which could have been used by other types of applications, and, therefore it became SETI@home Classic. With new funding, SETI@home has been rewritten to provide a new general framework and it became the new SETI@home/BOINC in 2005. BOINC is open-source software that can be used for both volunteer computing and desktop grid computing [14].

XtremWeb

XtremWeb is an open source platform that allows building of lightweight desktop grids by gathering the unused resources of desktop computers (CPU, storage, network), which is being developed by IN2P3 (CNRS), INRIA and University Paris XI. With XtremWeb, any participant can volunteer her computing resources, as in BOINC, but she can also use other participants' available resources [5]. The XtremWeb designers place the system between pure desktop grid system, such as Condor and pure Volunteer Computing system, such as BOINC [24]. It is worth to mention that there is also a constant endeavor to create an integrate grid infrastructure that seamlessly put together a variety of desktop grids based either on BOINC, or on XtremWeb, around the world, within the EDGeS (Enabling Desktop Grids for e-Science) project [2, 8, 18].

distributed.net

The distributed.net project runs projects that need tremendous computing power [4, 17], as for example the deciphering of encrypted messages. The commercial company RSA Security has made public a set of cryptographic puzzles, and has offered cash rewards for those who would have solved them, aiming to test the security of their own products and to demonstrate the vulnerability of the encryption schemes they considered inadequate [7]. Another such project is the Optimal Golomb Ruler (OGL). A Golomb Ruler is a mathematical term given to a set of whole numbers where no two pairs of numbers have the same difference. An Optimal Golomb Ruler is just like an everyday ruler, excepting that the marks are placed so that no two pairs of marks have the same distance. OGRs are usfelu in real world situations such as sensor placements for X-ray crystallography and radio astronomy. Golomb rulers can also make an important contribution in combinatorics, coding theory and communications. The complexity of searching OGRs increases exponentially as the number of marks grows, being a NP complete problem [17].

Condor

Condor is a High Throughput Computing (HTC) environment that can deal with very large sets of distributively owned available computing resources. It is being developed at the Department of Computer Science, University of Wisconsin, in Madison. Condor delivers large amounts of computational power over a long period of time, usually weeks or months. In contrast, High Performance Computing (HPC) environments deliver a huge amount of compute power over a short period of time. In a high throughput environment, users are more interested in how many jobs they can finish over a long period of time instead of how quick an individual job can finish. Hence, HTC is more concerned to efficiently harness the use of all available resources [16].

The Condor environment has a layered architecture that provides for a powerful and flexible suite of resource management services for sequential and parallel applications. Condor offers a job queuing mechanism, a scheduling policy, a priority scheme, and resource monitoring and management. Users submit their serial or parallel jobs ant the system places them into a queue, schedule them to be run, monitors their progress, and finally informs the user upon completion [4, 16].

QADPZ

QADPZ (Quite Advanced Distributed Parallel Zystem) is a system for heterogeneous desktop grid computing that provides for a centralized management and use of the computational resources of idle desktop computers from a network. QADPZ users can submit compute-intensive applications to the system, which are then automatically scheduled for execution. Applications can be independent, when their tasks do not require any interaction, or they can be parallel, when the tasks communicate with one another during the computation. Thus, the system provides support for both task- and data-parallelism. Briefly, the main features of the system include native support for multiple operating systems, support for legacy applications, a object-oriented development framework that supports low-level programming languages and high-level language applications, and that provides for using such applications in a computation, an extended C/C++ API, which supports creation of lightweight tasks and parallel computing, low-level optimizations, a refined master worker-model, virtualization of the master, efficient communication mechanisms, and autonomic computing characteristics [4].

Entropia's DC Grid

DCGrid, developed by the Entropia company, has been a PC-based grid computing platform that provided HPC capabilities by aggregating the unused processing cycles of existing networked Windows-based PCs. The system is no longer in use; however we have chosen to still present it since it has been a major desktop grid system, which has had significant contributions to the field. DCGrid enabled new and more difficult problems to be solved, by harvesting unused PC resources, based on user's and organization's policies, having settings centrally monitored and managed via a web-based grid management interface. Work was scheduled to PCs based on application's resource requirements, and was monitored and rescheduled as necessary, if there were system disruptions or resource unavailability. Any native Win32 application could be deployed and executed on the DCGrid platform, and applications were enabled for the platform at the binary code level [1].

DCGrid included an isolation technology, which provided complete and unobtrusive protection for the grid as well as the underlying resources. The grid application was not able to accidentally or deliberately access or modify the PC configuration or data files. DCGrid protected applications, proprietary data, and resources distributed to the desktop PCs by using encryption and tamper detection (shread). When an application program was registered or submitted to the Entropia system, it was automatically wrapped inside the virtual machine. This isolation is called sandboxing. The application was contained within a sandbox and was not allowed to modify resources outside this sandbox [1]. Morever, DCGrid remained invisible at all times, never demanding inputs or responses from the PC user, and never impacting the user's performance [20].

To be or not to be open source?

In this section we will discuss upon the impact of the open source-ness of the desktop grid systems to their life span and their use scale within projects worldwide, considering also comparatively the main features of each of the main systems presented in the previous section.

BOINC presents very powerful features such as project autonomy, volunteer flexibility (flexible application framework, security, server performance and scalability), source code availability, support for large data, multiple participant platforms, open and extensible software architecture, and volunteer community related characteristics. BOINC provides for applications that require huge computing resources or large storage, or both [14].

Since it first started, in 1995, SETI@home, and further on, as BOINC, the open source-ness has been one of the most appealing qualities of this system. Especially due to this attribute, today BOINC has about 495,000 active host computers worldwide that process on average about 3 PetaFLOPS per day as of October 2010 [15, 27], which is more than the processing power of the greatest existing supercomputer system (Jaguar - Cray XT5), with a sustained processing rate of 1.759 PetaFLOPS [10, 20]. Moreover, there are many active BOINC-based projects worldwide today. The problems come from various domains: mathematics (e. g. ABC@home, NFS@home, SZTAKI Desktop Grid, Rectilinear Crossing Number, primaboinca etc.), cryptography (e. g. Enigma@home and PrimeGrid), biology and medicine (e. g. GPUGrid.net, POEM@HOME, SIMAP, docking@home, Malariacontrol.net etc.), earth sciences (e. g. Climateprediction.net and Virtualprairie), astronomy (e.g. SETI@home, Cosmology@Home, Orbit@home, Milkyway@home etc.) and so on [14].

XtremWeb allows multi-users, multi-applications and cross-domains deployments, by exploiting volatile resources spread over LAN or Internet and by putting them to work as a runtime environment that executes highly parallel applications. XtremWeb may be customized for a wide range of applications (bag-of tasks, master/worker), of computing requirements (data, CPU or network-intensive) and of computing infrastructures (clusters, desktop PCs, multi-LAN) [24].

XtremWeb has several deployments, for example XtremWeb-CH that is intended to build an effective peer-to-peer system for high performance computing, based on a completely symmetric model where nodes are both providers and consumers, or XWHEP (XtremWeb for High Energy Physics), which is a desktop grid computing platform capable of deploying and running user applications on the available computing resources [24]. There is a set of projects that are based on these platforms, mainly due to their open source-ness. XtremWeb-CH supports NeuroWeb and Virtual EZ Grid from medical domain, and From-P2P from public computing [25], and XWHEP middleware is used by the DGHEP, EDGI, DEGISCO, IDGF, and EDGeS projects [26].

Compared with BOINC and XtremWeb, the focus of distributed.net is on very few specialized computing challenges, and, as the project only releases the clients' binary code and not the server code, it is practically impossible to adapt it to other types of projects, which limits considerably the large scale use of this system.

XtremWeb uses a pull model (workers pull tasks from the server), whilst Condor relies on a push model (the matchmaker selects the best machine to run the job and push the job on it). Moreover, with XtremWeb, each user or node has the ability (if authorized) to submit a job. When using BOINC, only the projects are able to create tasks and insert new applications [24].

While earlier versions of Condor, since the project started in 1998, have been restricted by a closed license, recently it has made a huge shift and nowadays it is open source. While being closed, despite its powerful resource management mechanism, which was able of matchmaking resource owners with resource consumers, and its other strengths, Condor has not been as widely used as its power would have allowed. It has even seemed to disappear for a while from the desktop grid computing reality, but lately it has made a spectacular comeback, and currently there are various Condor-based grids around the world, such as Greedy@EFPL, a desktop grid platform that runs at Federal Polytechnic School of Lausanne, and supports various mono-processors jobs [9, 12, 21], and a very large condor pool available at University College of London, which is used for several UK eScience projects, such as eminerals within molecular simulations of environmental issues domain [13, 19]. Condor is intensively used also at his "home university" from Wisconsin in several scientific projects such as genomics, IceCube Neutrino Observatory, Collider Detector at Fermilab etc. [22].

The free source code of the QADPZ system provides for its flexible installations and modifications based on the particular needs of various research projects and institutions. In addition to being a very powerful tool for computationally-intensive research, the open source-ness makes QADPZ a flexible educational platform for numerous small-size student projects in the areas of operating systems, distributed systems, mobile agents, parallel algorithms, etc., the system being used for this purpose in universities around the world [4].

The three-layered architecture of Entropia's DC Grid provided a lot of benefits with regard to system capabilities, ease of use, and internal implementation. The physical node layer managed the complex communication, security, and management issues, allowing the layers above to operate with simpler abstractions. The resource-scheduling layer had to handle only with the breadth and variety of resources, but had not needed to handle lower-level issues. The next layer had to deal mostly with conventional job management issues. Finally, the higher-level abstractions presented by each of the layers were simplifying application development [3]. In spite of all these powerful features, DC Grid has not passed the time test, and it is no longer in use due mainly to the fact that it has been developed as proprietary software. Of course, we have to consider also the limitation of not supporting heterogeneous systems, as Entropia's DC Grid was available only for Windows. However, this flaw would have been fixed much easier fueled by the extreme resourcefulness of the open source world.

Conclusions

In this paper we have addressed the issue of the open source-ness of the most important platforms that provide for desktop grid and volunteer computing. After summarizing in a few words the main characteristics of these platforms, we have tried to establish whether there is a correlation between the free availability of the source code and the success of each system with regard to its use in various projects. In our opinion, the open source-ness has determined the broadness of use for these systems and it has hugely influenced their success. Therefore, we are strongly convinced that the answer to the question in paper's title is "yes", and that open source software is a natural choice for modern research as well, because it encourages successful integration, cooperation and boosting of new ideas.

References

[1] Calder, B. et al., The Entropia Virtual Machine for Desktop Grids, *Proceedings of the 1st ACM/USENIX International Conference on Virtual Execution Environments (VEE '05)*, pp. 186-196, Chicago, IL, USA, 2005

[2] Cardenas-Montes, M. et al., EDGeS: bridging Desktop and Service Grids, *Proceedings of the 2nd Iberian Grid Infrastructure Conference (IBERGRID 2008)*, pp. 212-226, Porto, Portugal, 2008

[3] Chien, A. A., Architecture of the Entropia Distributed Computing System, *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2002)*, pp. 55-59, Fort Lauderdale, FL, USA, 2002

[4] Constantinescu, Z., *A Desktop Grid Computing Approach for Scientific Computing and Visualization*, PhD Thesis, Norwegian Univ. of Science and Technology, Trondheim, Norway, 2008

[5] Fedak, G., *Recent Advances and Research Challenges in Desktop Grid and Volunteer Computing,* CoreGrid workshop with Europar (CoreGrid'09), Delft, Nederland, 2009

[6] Foster, I., Kesselman, C., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, Boston, USA, 2004

[7] Hayes, B., Collective Wisdom, *American Scientist*, 86(2), March, April, pp. 118-122, 1998

[8] He H. et al., Extending the EGEE Grid with XtremWeb-HEP Desktop Grids, *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGRID'2010)*, pp. 685-690, Melbourne, Australia, 2010
[9] Jermini, P., Building a Condor Desktop Grid and its Exercises, *Swiss Grid School 2010*, Lugano, Switzerland, available online at http://greedy.epfl.ch/articles/building-condor-grid-sgs10.pdf, accessed September 2010
[10] Meuer, H.W., *Top500 Supercomputer Sites Report, 35th Edition,* June 2010, available at http://www.top500.org, accessed November 2010
[11] Shread P., Avaki, Entropia and Platform Announce New Products, http://www.gridcomputingplanet.com/news/article.php/1433141/Avaki-Entropia-and-Platform-Announce-New-Products.htm, accessed November 2010
[12] Thiemard, M., On the Usage of a Desktop Grid - Greedy@EPFL, *Symposium on High Performance Computing Methods (HPCM),* June, 2008, Lausanne, Switzerland, available online at http://greedy.epfl.ch/articles/hpcm-0608.pdf, accessed October 2010
[13] Wilson, P., Emmerich, W., Brodholt, J., Leveraging HTC for UK eScience with Very Large Condor Pools: Demand for Transforming Untapped Power into Results, *Proceedings of the UK e-Science Programme All Hands Meeting (AHM 2004)*, pp. 308-315, Nottingham, UK, August-September, 2004, EPSRC, Swindon, UK
[14] * * * - BOINC, Open Source Software for Volunteer Computing and Grid Computing, http://boinc.berkeley.edu, accessed September 2010
[15] * * * - BOINCstats, Statistics for BOINC, available at http://www.boincstats.com, accessed September 2010
[16] * * * - Condor High Throughput Computing, http://www.cs.wisc.edu/condor, accessed September 2010
[17] * * * - Distributed.net, http://www.distributed.net/Main_Page, accessed December 2010
[18] * * * - EDGeS - Enabling Desktop Grids for eScience, http://www.edges-grid.eu, accessed September 2010
[19] * * * - eMinerals, Environment from the Molecular Level – a NERC eScience testbed project, available at http://www.eminerals.org, accessed September 2010
[20] * * * - Entropia Announces Release Of DCGrid 5.0 for PC Grid Computing; *Business Wire*, July 2002, http://www.highbeam.com/doc/1G1-89371103.html, accessed November 2010
[21] * * * - Grid@EFPL Greedy Pool Desktop Grid, http://greedy.epfl.ch, accessed September 2010
[22] * * * - Paradyn/Condor Week – Condor Presentations, Madison, Wisconsin, USA, April-May, 2008, available at http://www.cs.wisc.edu/condor/PCW2008/condor_presentations.html, accessed November 2010
[23] * * * - Seti@home Classic, http://seticlassic.ssl.berkeley.edu, accessed September 2010
[24] * * * - XtremWeb, The Open Source Platform for Desktop Grids, http://www.xtremweb.net, accessed October 2010
[25] * * * - XtremWeb-CH, The Volunteer Computing Middleware, http://www.xtremwebch.net, accessed November 2010
[26] * * * - XtremWeb-HEP, Desktop Grid for High Energy Physics, http://www.xwhep.org, accessed November 2010
[27] * * * - Wikipedia, Berkeley Open Infrastructure for Network Computing, available at http://en.wikipedia.org/wiki/Berkeley_Open_Infrastructure_for_Network_Computing, accessed September 2010

A triumfat abordarea open source în sistemele pentru desktop grid și volunteer computing?

Rezumat

Paradigma Grid Computing permite folosirea de resurse care sînt distincte, atît din punct de vedere geografic, cît și din punct de vedere administrativ și care pot fi accesate prin intermediul rețelei de calculatoare, cum ar fi unitățile de procesare, capacitățile de stocare, date specifice, dispozitive de

intrare/ieșire etc. Utilizatorii individuali pot accesa calculatoare și date în mod transparent, fără a fi nevoiți să ia în considerare locația, sistemul de operare, administrarea conturilor etc., care sînt abstractizate față de ei. Sistemele pentru desktop grid și volunteer computing sînt o categorie importantă de sisteme grid, în care ciclurile de calcul sînt recuperate de la calculatoarele desktop care sînt nefolosite temporar, fie în interiorul unui intranet, fie în Internet. În acest articol, ne preocupă întrebarea dacă folosirea sistemelor pentru desktop grid și volunteer computing, în cadrul proiectelor care necesită calcule intensive, este sau nu influențată de faptul că ele sînt sau nu open source. Trecem în revistă aici pe scurt principalele astfel de sisteme, prezentînd caracteristicile majore ale acestora, subliniind cele mai importante puncte forte si neajunsuri ale acestora. Principala noastră preocupare este impactul disponibilității codului sursă asupra timpului de viață și a scării de folosire a acestor desktop grid-uri în cadrul proiectelor din toată lumea. În opinia noastră, răspunsul la întrebarea din titlu este "da", iar software-ul open source este o alegere naturală atît pentru sistemele pentru desktop grid și volunteer computing, cît și pentru cercetarea modernă, deoarece încurajează succesul integrării, al cooperării și al stimulării apariției de idei novative.