Journal of Network and Computer Applications 39 (2014) 369–380

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Volunteer computing: requirements, challenges, and solutions

Muhammad Nouman Durrani *, Jawwad A. Shamsi

Systems Research Laboratory, Department of Computer Science, FAST NU, Karachi, Pakistan

ARTICLE INFO

Article history:
Received 3 August 2012
Received in revised form
3 May 2013
Accepted 18 July 2013
Available online 29 July 2013

Keywords:
Volunteer Computing
Performance
Task retrieval
Scheduling
Voting
Classification

ABSTRACT

Volunteer computing is a form of network based distributed computing, which allows public participants to share their idle computing resources, and helps run computationally expensive projects. Many existing volunteer computing platforms consist of millions of users, providing huge amount of memory and processing. Since the rapid growth in the volunteer computing projects, more researchers have been attracted to study and improve the existing volunteer computing system. However, the progress of concurrently running projects has slowed down due to the increasing competition of volunteers. Moreover, because of high computational needs and low participation rate of volunteers, attracting more volunteers and using their resources more efficiently have become extremely important, if volunteer computing is to remain a feasible method. In order to competently use the huge number of volunteered resources, workers' analysis and efficient task retrieval policies are important. The purpose of this paper is to assess the strengths and requirements of current volunteer computing platforms. The paper analyses different issues relating to volunteer computing such as analysis of workers, the effectiveness of workers, how their communication and computation can be modeled and how the effectiveness of task distribution and results verification policies are analyzed. At the end, some research directions in the form of partial results, and their intermediate verification have been shown, which may improve the performance of the overall system. Moreover, this survey will enable the research community to study the available schemes used in volunteer computing and help them fill gaps in existing research.

Contents

1. Introduction

Volunteer computing (VC) is a form of distributed computing, which allows public participants to share their idle computing resources and helps run computationally expensive projects

* Corresponding author. Tel.: +92 33 33887366; fax: +92 21 34100549.
E-mail addresses: muhammad.nouman@nu.edu.pk, nouman_durrani@yahoo.com (M. Nouman Durrani), jawwad.shamsi@nu.edu.pk (J.A. Shamsi).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2013.07.006

(Volunteer computing, 2013; Desell et al., 2009; Watanabe et al., 2011). VC provides bulk of inexpensive resources that can supply more computing power to science than does any other type of computing. It boosts public interest in scientific research and provides the scientists with voice in determining the directions of scientific research that could not be performed otherwise. A research project that has limited resource funding but large public appeal can catch the massive computing power of VC. Many existing VC platforms consist of millions of users, providing huge amount of memory and processing (Volunteer computing, 2013).

VC offers a scalable, low cost, reliable, and powerful computing platform to its users where expensive tasks are disintegrated into small chunks for parallel execution. Volunteers (also known as workers, hosts or nodes) provide a portion of their storage or computational resources to form a resource-cloud. In order to run computationally expensive projects, a middleware is needed, which collects chunks (a portion of data that needs to be processed – workunit) from servers, and sends these workunits to volunteers for storing or processing (Toth and Finkel, 2009; Toth, 2008).

In VC, users have heterogeneous resources with varying capabilities (Schmohl and Baumgarten, 2008). The foremost challenge is proper distribution of tasks according to the capacity of each volunteer in order to achieve timely completion of jobs. Further, this would ensure efficient utilization of resources, reduced cost of operation, and enhanced user satisfaction (Anderson et al., 2005; Toth and Finkel, 2008). Heterogeneity-aware distribution of workload would also permit priority-based execution of tasks and improve real time behavior. In addition, it may increase the pool of applications that may be executed on the VC platform.

Various task scheduling policies such as earliest-deadline-first, work fetch, buffer multiple tasks, weighted round robin, work send and job completion estimation policies have already been practiced in VC (Toth and Finkel, 2009; Toth, 2008; Toth and Finkel, 2008; Kondo et al., 2007; Anderson, 2011). However, most of these policies are used to reduce the overall execution time of parallel computation, without consideration of the correctness of computational results. Furthermore, VC projects may vary in terms of job length, errors sensitivity, and result verification requirements (Estrada et al., 2009a; Sarmenta, 2001; Sarmenta, 2002).

Large numbers of inexpensive resources are available to store and process big-data. Numerous schemes are used to utilize these resources. However, the progress of concurrently running projects has slowed down due to inefficient use of voluntary resources. The purpose of this paper is to evaluate the strengths and requirements of current volunteer computing platforms. The paper analyses different issues relating to volunteer computing such as analysis of workers, the effectiveness of workers, how their communication and computation can be modeled and how the effectiveness of task distribution and results verification policies are analyzed. At the end some research challenges are also identified. Moreover, this paper will enable the research community to study the available schemes used in VC and help them filling research requirements. The sequence of this paper is organized as follows: Section 2 discusses about the *volunteer computing platforms*, Section 3 presents the research issues and challenges, Section 4 open problems and future research directions.

2. Volunteer computing platforms

This section explains the background concept of VC. The section begins with brief introduction leading to detailed discussion of BOINC, which is deemed as the most popular VC middleware.

2.1. Evolution of volunteer computing

In 1990′s, there was a burst of research activities related to VC, including ATLAS, Charlotte, Bayanihan, ParaWeb, Popcorn, Javelin, Gucha, Distriblets, and others (Toth, 2008; Sarmenta, 2001). At the end of 2000, interest in the field was shifted towards the more productive area of grid computing, and only few research projects remained active due to their low-cost, large-scale distributed computing.

In grid computing, resources are owned and shared within and between organizations; where in VC, resources are sporadic and unreliable in nature. At any instance, they may leave the system and make few guarantees about the machine and network. Using this bulk of inexpensive resources, initial goals of projects like GIMPS, distributed.net and grid.org were made successfully possible. Different middleware platforms were designed to build VC projects. However, middleware platforms like BOINC(Berkeley Open Infrastructure for Network Computing) (Volunteer computing, 2013; Yi et al., 2010; Estrada et al., 2006) and Xtremweb continued to move forward, with BOINC having become the leading framework to build volunteer computing projects (Toth, 2008; Toth and Finkel, 2008). Due to the successful development of middleware platforms, the following projects were initiated:

GIMPS: in January 1996, George Woltman started Great Internet Mersenne Prime Search (GIMPS), the oldest of the major VC projects. On December 20, 2012, GIMPS finished double-checking every smaller Mersenne number than M(25964951) – proving that this prime is indeed the 42nd Mersenne prime (Great Internet Mersenne Prime Search (GIMPS), 2013).

distributed.net: founded in 1997, distributed.net was the Internet's first general-purpose distributed computing project. Distributed.net used VC to win several cryptographic challenges of well-known encryption algorithms. Having successfully completed Optimal 26-Mark Golomb Rulers, DES-III, DES II-2, DES II-1, RC5-56 and RC5-64, it is now working on RC5-72 encryption algorithm, RSA prime factoring, fermat numbers and elliptic curve cryptosystem (ECC) (Distributed.net, 2013).

SETI@home: SETI@home (setiathome.berkeley, 2013; Anderson et al., 2002) searches massive amount of radio telescope data for signs of extra-terrestrial intelligence. SETI@home is one of the most well-known VC project started in May 1999. As of January, 2013, over 3.32 million users are participating in this project. It has the ability to compute over 925 teraFLOPS.

grid.org: established in 2001, grid.org was a website and online community services that operated several different VC projects and allowed workers to donate their free computing resource to valuable projects. In this regard, cancer research project was the first project of Grid.org. In the project, it successfully screened billions of target molecules against known cancer target proteins. Following the cancer project, it worked on other projects to identify treatments for smallpox and anthrax. Currently, grid.org is working on analyzing human proteins folding, hidden Markov Modeling and Webload testing (grid.org, 2013).

Folding@home: Folding@home (What is protein folding, 2013), understands protein folding, misfolding, and related diseases, with a minor emphasis on protein structure prediction. The primary purpose of this project is to determine the mechanisms of protein folding, which is the process by which proteins reach their final 3-D structure. It also observes the roots of protein misfolding. The project has founded the use of, PlayStation 3s, GPUs and MPI for distributed computing and scientific research. Folding@home uses statistical simulation methodology that is a complete paradigm shift from traditional computational approaches. Moreover, Folding@home is one of the