EFFICIENT AND TRANSPARENT VIRTUALIZATION MECHANISM FOR VOLUNTEER COMPUTING

MOHAMMADFAZEL ANJOMSHOA

UNIVERSITI TEKNOLOGI MALAYSIA

EFFICIENT AND TRANSPARENT VIRTUALIZATION MECHANISM FOR VOLUNTEER COMPUTING

MOHAMMADFAZEL ANJOMSHOA

A dissertation submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

JANUARY 2014

I dedicate my dissertation work to my family and my wife. A special feeling of gratitude to my loving parents, Ali and Mehri Anjomshoa whose words of encouragement and push for tenacity ring in my ears. This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time. I also dedicate this dissertation to my lovely wife and her family who have supported me throughout the process. This thesis is dedicated to my wife for her love, patience, support and understanding have lightened up my spirit to finish this study and this thesis. This thesis is dedicated to my father in law, Mahmoud Pouryazdanpanah Kermani and also my mother in law, Sedigheh Torabi for their support, inspiration, and love.

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to my supervisor Associate Prof. Dr Mazleena Salleh and my co-supervisor Dr. Rodrigo N. Calheiros for the continuous support of my master study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master study.

Last but not the least, I would like to thank my lovely wife: Maryam Pouryazdanpanah Kermani for her support and also believing in me during my master study and research.

ABSTRACT

Virtualization is a technology that introduced long time ago but it has emerged in the last decade as a viable and novel solution for assembling a complete operating system (guest OS) on top of hosting machine (host OS). It has changed computing in many aspects with its unique features like easy deployment of guest OSes, migration, checkpointing and sanboxing. Public resource computing project like SETI@home, Roesetta@home and others that are powered by volunteer resources can benefit from virtualization characteristics in both project developing and volunteers points of view. However wide-scale deployment of virtualized environment for desktop grids impacts on the host performance as virtualization functionalities imposes Central Processing Unit (CPU) and memory overhead into the host environment. Virtualization adoption imposes additional download bandwidth on volunteers machine. This thesis aims to propose an efficient approach to adapt virtualization into volunteer computing platform. The proposed virtualization mechanism is implemented on BOINC. It uses VirtualBox to establish virtualized environment. In order to reduce resource overhead, a centralized virtual machine undertakes the execution process which is created by a symlink virtual machine image file. Evaluation results demonstrate that the proposed virtualization approach, improves BOINC performance in terms of CPU and memory overhead both Random Access Memory (RAM) and storage notions. The proposed mechanism reduced the CPU overhead by 96.17%, 206.5%, 316.85, and 429.47% when executed single job, two jobs, three jobs and four jobs in parallel respectively. In the case of memory overhead, the proposed virtualization mechanism improved the storage overhead by 95.5%, 194.60%, 220.75%, and 286.43%, and declined the RAM overhead by 0.00% , 100%, 200%, and 300% when scaled up from executing single job to four jobs respectively. The proposed virtualization mechanism reduced considerably resources overhead which were occupied by virtual machine environment and depicts the possibility of adapting virtualization functionality into the volunteer computing environments with the acceptable additional overhead.

ABSTRAK

Pemayaan merupakan satu teknologi yang telah lama memperkenalkan tetapi ia hanya muncul dalam dekad yang lalu sebagai satu penyelesaian yang berdaya maju serta novel untuk membangunkan sistem pengendalian tetamu (guest OS) yang lengkap di atas mesin hos. Pemayaan telah mengubah cara pengkomputeran dalam banyak aspek dengan ciri-ciri uniknya seperti penempatan mudah tetamu OSes, penghijrahan, titik semak dan sanboxing. Projek pengkomputeran sumber umum seperti SETI@home, Roesetta@home yang disokong oleh sumber sukarela boleh memanfaatkan ciri-ciri pemayaan dalam pembangunan projek dan persekitaran sukarela. Walau bagaimanapun penggunaan persekitaran maya untuk komputer meja grid memberi kesan terhadap prestasi hos atau mesin sukarela. Ini adalah kerana fungsian pemayaan memerlukan sumber pemproses dan ingatan hos dan ini akan menyebabkan penambahan penggunaan jalur lebar semasa proses muat turun ke mesin sukarela. Untuk mengatasi masalah tersebut tesis ini mencadangkan pendekatan yang efisien untuk menyesuaikan pemayaan ke dalam pelantar pengkomputeran sukarela. Mekanisma pemayaan dibangunkan di atas Berkeley Open Infrastructure for Network Computing (BOINC) serta menggunakan VirtualBox. Untuk mengurangkan overhed sumber komputeran, satu mesin maya berpusat akan melaksanakan proses yang mana ianya direka oleh fail imej mesin maya symlink. Hasil pelaksanan menunjukkan kaedah pemayaan yang dicadangkan dapat meningkatkan prestasi BOINC dari segi overhed pemproses dan ingatan, iaitu Ingatan Capaian Rawak (RAM) dan juga storan. Overhed pemprosesan menunjukkan penurunan sebanyak 96.17%, 206.5%, 316.85%, dan 429.47% untuk pekerjaan yang tunggal, dua pekerjaan, tiga pekerjaan dan empat pekerjaan yang dilaksanakan selari. Untuk kes overhed ingatan, mekanisme pemayaan yang dicadangkan telah mengurangkan overhed sebanyak 95.5%, 194.60%, 220.75% , dan 286.43% , manakala penurunan overhed RAM sebanyak 0.00%, 100%, 200% , dan 300% untuk pekerjaan yang tunggal, dua, tiga dan empat pekerjaan selari. Mekanisme pemayaan yang dicadangkan telag berjaya mengurangkan penggunaan sumber yang perlukan oleh persekitaran mesin maya. Keputusan menunjukkan kemungkinan penyesuaian kefungsian pemayaan ke dalam persekitaran pengkomputeran sukarela dengan overhed tambahan boleh diterima.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CC	-	Cloud Computing
DG	-	Desktop Grid
DCI	-	Distributed Computing Infrastructures
OS	-	Operating System
PC	-	Personal Computer
VM	-	Virtual Machine
VMM	-	Virtual Machine Monitor
VC	-	Volunteer Computing

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

1.1 Introduction

Technology is the combination of knowledge and working hard. When users want to accomplish something by using special technology, they do not want to know how it works. It means that users only want to employ technology without involving the complexity behind it. Furthermore, technologies are created to solve complex problems in an easy way. Volunteer computing(VC) (Sarmenta, 2001), refers to computing paradigm that resources are provided by public volunteers and is established for open and powerful computing to do scientific project. BOINC (Anderson, 2004) has emerged as the most well-known volunteer computing platform which provides 8.5 petaflops of processing power approximately and gathers more than 2,500,000 users all around the world (states, 2013).

Virtualization (Sahoo *et al.*, 2010) defines as a software abstraction layer between the hardware and the OS. Virtualization improves the BOINC functionality in the terms of application portability and security. Moreover, By enabling virtualization in volunteer computing frameworks it is possible to use the same computing environment including the operating system and all application packages that may be required by the public computing project application, across all participating computing nodes. This configuration eases the task of developers in the way that, they only have to deal with a single platform. In more details, virtualization is a type of sandboxing, where foreign applications execute in a virtualized layer which brings an enhanced security for volunteer hosts. However, there are some overheads in creating and managing the virtualization that apply into the volunteer host machine and affect the overall performance of the system.

This thesis introduces a mechanism for enabling virtualization into a volunteer computing platform by focusing on reduction of the overhead that is applied by virtualization into BOINC. This study is considered as a part of bigger project where by enabling virtualization in desktop grid's environment it is possible to establish Infrastructure as a servive(IaaS) from these volunteer's resources. To achieve this cloud computing service, there is a need to overcome many issues, but this study is considered as the starting point. This new type of cloud computing will be introduced in future works section in more details.

1.2 Problem Background

This dissertation tackles the research challenges related to volunteer computing frameworks deployment and also surveys the drawbacks of existing virtualization mechanisms. The problem background of this thesis is divided into three perspectives:

i Desktop grid perspective

ii Combination of virtualization and volunteer computing perspective

iii Parallel computing perspective

1.2.1 Desktop Grid Perspective

The term grid computing refers to a collection of software and hardware infrastructures that allow users to use these resources in a geographically distributed model (Foster, 2002). One of the grid branches is desktop grids (Choi *et al.*, 2007), which rely on harnessing of idle PC's resources that are connected through network to work on a specific computational problem.

Volunteer computing (Sarmenta, 2001) is a type of desktop grid that is established to satisfy the extraordinary growing scientific application's demand that relies on volunteer's PCs. This type of computing uses the idle time of PC's to do research and scientific projects (Amoako *et al.*, 2008). In fact, volunteer computing employs unused CPU cycles to fulfil their workloads. This computing paradigm

provides petaflops of processing power to perform tasks such as formalization of complex mathematical models or predict climate changes. The most famous example of Volunteer computing project is SETI@home (Anderson *et al.*, 2002), which is based on the BOINC platform (Anderson, 2004) that is currently considered as the most popular Volunteer computing platforms since now.

To develop Volunteer computing platforms many issues should be taken into consideration (Choi *et al.*, 2007). The first one is, VC platform should be easy for users to communicate with the environment. This is because the users (donors) are in a wide technical knowledge ranges. As the power of this computing paradigm is based on volunteer's resources and due to the volunteer's natural that is wide range geographically distributed and donors might own different systems that are supported by a variety of OSes with different applications on it so in developing volunteer computing middleware the framework independence should be taken into account to avoid compatibility issues and consequently attract more donors. To encourage more volunteers to participate in the projects, the system should be secure and trusted from the volunteer's perspective. So security is considered as one of the main challenges in volunteer computing platforms.

Applications in BOINC have some considerable drawbacks; they have lack of portability, which means BOINC applications should be overwritten for each platform (windows 32/64, MAC OS, Ubuntu). By applying virtualization in volunteer computing platforms, it is possible to address the problem of portability and also to enhance the security level of the system.

1.2.2 Combination of Virtualization and Volunteer Computing Perspective

The aim of using virtual computing environments are to enhance resource utilization by providing a unified integrated operating platform for users and applications based on association of heterogeneous and autonomous resources. In overall, virtualization has taken into account as a good way to improve system security, reliability and availability, reduce costs and also provide greater flexibility (Figueiredo *et al.*, 2003; Marosi *et al.*, 2012; Ferreira *et al.*, 2011; Krsul *et al.*, 2004).

However, to adopt virtualization in such frameworks, some issues should be taken into account (Marosi *et al.*, 2012, 2010). The VM image file, which is approximately one gigabyte in size, imposes considerable bandwidth to users. Transparency implementation is another issue in virtualization approach. Transparent deployment from user's view means they should not charged by bandwidth overuse, nor notices any slowdown or less responsiveness of the system.

1.2.3 Parallel Computing Perspective

Parallel execution refers to a form of computation in which many computations are performed simultaneously based on the principle that large problems can be divided into smaller ones which are then carried out in parallel (Kumar *et al.*, 1994). Parallel computer programs are more difficult to write than sequential ones because parallel solution introduces many new types of potential software bugs. Communication and synchronization between the different subtasks are typically some of the greatest obstacles to getting good parallel program performance (Patterson and Hennessy, 2008).

1.3 Problem Statement

Enabling virtualization in VC frameworks provides such framework with enhanced application portability, resource utilization control and security due to the isolation layer. However, in related to the data transfer issue, virtualization imposes the processing and also bandwidth overhead which affects the overall computing performance. So does a transparency virtualization mechanism with the capability of running multiple of BOINC applications in parallel performed by a single VM, decrease the overhead that is forced to machine's resources by virtualization functionalities?

In particular,the following research problems are investigated:

i How can a virtualization mechanism in BOINC be formulated to effectively improve the performance issues in comparison with the related works?

ii What are the factors that should be taken into consideration to implement virtulization mechanism into BOINC framework?

iii What is the impact of proposed virtualization mechanism on the overall of computing performance?

1.4 Objectives

The main goal of this study is to reduce the overhead of processing and bandwidth that are imposed by virtualization into system. To deal with the challenges associated with the research problems mentioned in Section 1.3, the following objectives have been delineated:

i To reduce the CPU and memory overhead imposed by virtualization functionalities into the BOINC client machine.

ii To design a transparent implementation of virtualization mechanism from the user perspective.

iii To evaluate the functionality of proposed virtualization mechanism by comparing it with the BOINC virtualization approach that is named vboxwrapper.

1.5 Scope

This dissertation investigates the shortcomings in wide-scale adoption of using virtual machines for desktop grid computing by considering the performance impact of virtualization functionalities into the desktop grid computing frameworks in order to propose an efficient mechanism to enable VM environment in such frameworks. The scope of this study is defined as follows:

i The BOINC framework is used in this study as a desktop grid computing middleware and the VirtualBox is chosen as virtualization tool.

ii First version of the proposed system works on Linux OSes and VirtualBox software package should be installed on Linux host OS.

iii The proposed system components works on BOINC version 7.0.28+.

iv The programming language that is used for developing the proposed system components is C++.

v The Server and client components are assembled in the same physical machine.

vi The BOINC virtualization method named vboxwrapper is compiled to be compared with the proposed virtualization mechanism in evaluation phase. Furthermore uppercase BOINC sample application is used for creating BOINC job workunit.

1.6 Significance of Study

Greenberg (Greenberg *et al.*, 2008) argues that building cloud based datacenters require large amount of investments; considering approximately $53 million each year only for servers or about $10 million for powering. Thus, less expensive alternatives to deploy cloud like infrastructure are attracting. Instead of building such costly data centers, it is possible to establish a cloud infrastructure on a donor's resources that are almost free. In the other hand, in volunteer computing platforms application need to be written for each platform. Moreover, portability is one of the problems in nowadays volunteer computing platforms. By using virtualization, it is possible to eliminate these problem and provide heterogeneous-supported platforms.

By building cloud-like-infrastructure from volunteer computing resources, new computing paradigm named volunteer clouds is established. This new powerful computing infrastructure returns the control of resources from commercial companies to users who can make decisions which and how much resources needed to be used in geographically distributed manner (Aversa *et al.*, 2011). In order to achieve this goal, it is essential to adapt virtualization technology into the volunteer computing frameworks.

A virtual appliance may range in size from hundreds to thousands of megabytes,considering the 1.3 GB of BOINC VM, that is added to hypervisor software package. This amount of size, highlights the problem of bandwidth and also CPU overhead on volunteer side and it may become an important obstacle to encourage volunteers to participate in these projects. Consequently, finding proper solutions to decrease the CPU and memory overhead of virualization approach plays an important role to enable adoption of virtualization technology into VC.

1.7 Definitions

The followings are some important definitions,with relevant references provided where appropriate.

Computing

Computing paradigm has emerged as a solution to address complex computers or mathematics related issues. Shackelfor (Shackelford *et al.*, 2006) defined computing as a process that uses computers both hardware and software to do many computer related purposes; processing, scientific research, gathering information to extract some beneficial information and so on.

Distributed Computing

Distributed computing is a technology that aims to solve large computational problems with using distributed systems. Distributed systems (Lynch, 1996) are a collection of computers that are physically and geographically distributed and connected to each other to solve a common problem.

Grid Computing

Grid computing technology is a branch of distributed computing that enables collaborating and resource sharing where resources are geographically distributed and autonomous. These resources can consist of process,storage and specific data and can be found on the network. Foster (Foster, 2002) argued three important elements that define grid systems clearly. The dream of grid technology is that computing becomes a common utility such as water and electricity. The computational power of grids attracts scientists and researchers to fulfil and implement their researches on it.

Foster (Foster *et al.*, 2001) highlighted that resources for sharing are not only files, whereas it can be processing power, software and applications, storage

capacity, data and any other possible resources. Resources that work on grid technology benefit from its features; faster execution speed, interoperation of software, geographically distributed resources.

Cloud Computing

The term cloud computing is coming to a seat of attention as a revolution in distributed computing technology by enabling lots of attractive features. The NIST (National Institute of Standards and Technology) defined cloud computing as " a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources(e. g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction " (Mell and Grance, 2011). Although state of art cloud computing offers on demand application, Elastic Resource Capacity and Pay-per-used resources, it has been considering as a new enhancement in computing paradigm but it may arise technically and economically concerns for scientific or open usage of cloud which requires large amount of computing or storage resources. These required amount of resource, usually is not applicable by single cloud provider (Cunsolo *et al.*, 2010).

Virtualization

Virtualization (Sahoo *et al.*, 2010) is commonly defined as a technology that introduces a software abstraction layer between the hardware and the operating system and applications running on top of it. This abstraction layer is called virtual machine monitor (VMM) or hypervisor which hides the physical resources of the computing system from the operating system (OS).

Desktop Grids

Desktop Grids (@home) (Choi *et al.*, 2007) has emerged as a type of computing technology where the term computing and storage are enabled by donating individual's computers. The idea of desktop grids is that computational resources are provided by idle desktop computers. In desktop grids, big tasks are divided into small tasks and those small tasks are distributed among worker nodes and the result ready only when all tasks are done.

Volunteer Computing

Volunteer computing differentiates from desktop grid computing in the way that resources are provided by public volunteers (Nouman Durrani and Shamsi, 2013).

1.8 Dissertation Organization

The rest of this thesis is organized as follows. Chapter 2 provides an overview of the related works. It covers the details of computing forms, virtualization technology and combination of virtualization into computing frameworks. Chapter 3, which is research methodology, presents the theoretical design methodology of the system. It highlights the preparation of the research design and procedure, road map to achieving the research objectives and also performance evaluation. Chapter 4 presents the proposed system details. It depicts the system architecture and components and how the components work with each other. Chapter 5 presents the final results and a discussion over results. Chapter 6 concludes the thesis with a summary of the main findings, discussion of future research directions and final remarks.

REFERENCES

Abels, T., Dhawan, P. and Chandrasekaran, B. (2005). An overview of xen virtualization. *Dell Power Solutions*. (8), 109–111.

Adams, K. and Agesen, O. (2006). A comparison of software and hardware techniques for x86 virtualization. In *ACM SIGOPS Operating Systems Review*, vol. 40. ACM, 2–13.

Amoako, G., Amoako-Yirenkyi, P. *et al.* (2008). Volunteer Computing: Application for African Scientist.

Anderson, D. P. (2004). Boinc: A system for public-resource computing and storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*. IEEE, 4–10.

Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M. and Werthimer, D. (2002). SETI@ home: an experiment in public-resource computing. *Communications of the ACM*. 45(11), 56–61.

Anderson, D. P., Korpela, E. and Walton, R. (2005). High-performance task distribution for volunteer computing. In *e-Science and Grid Computing, 2005. First International Conference on*. IEEE, 8–pp.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. *et al.* (2010). A view of cloud computing. *Communications of the ACM*. 53(4), 50–58.

Aversa, R., Avvenuti, M., Cuomo, A., Di Martino, B., Di Modica, G., Distefano, S., Puliafito, A., Rak, M., Tomarchio, O., Vecchio, A. *et al.* (2011). The Cloud@ Home project: towards a new enhanced computing paradigm. In *Euro-Par 2010 Parallel Processing Workshops*. Springer, 555–562.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A. (2003). Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*. 37(5), 164–177.

Bellard, F. (2005). QEMU, a Fast and Portable Dynamic Translator. In *USENIX Annual Technical Conference, FREENIX Track*. 41–46.

Berman, F., Fox, G. and Hey, A. J. (2003). *Grid computing: making the global infrastructure a reality*. vol. 2. Wiley. com.

BOINC (2012). *vboxwrapper*. Retrievable at `http://boinc.berkeley.edu/trac/wiki/VboxApps`.

Buncic, P., Sanchez, C. A., Blomer, J., Franco, L., Harutyunian, A., Mato, P. and Yao, Y. (2010). CernVM–a virtual software appliance for LHC applications. In *Journal of Physics: Conference Series*, vol. 219. IOP Publishing, 042003.

Buyya, R., Beloglazov, A. and Abawajy, J. (2010). Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. *arXiv preprint arXiv:1006.0308*.

Chaudhary, V., Cha, M., Walters, J., Guercio, S. and Gallo, S. (2008). A comparison of virtualization technologies for HPC. In *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*. IEEE, 861–868.

Chiueh, S. N. T.-c. and Brook, S. (2005). A survey on virtualization technologies. *RPE Report*, 1–42.

Choi, S., Kim, H., Byun, E., Baik, M., Kim, S., Park, C. and Hwang, C. (2007). Characterizing and classifying desktop grid. In *Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on*. IEEE, 743–748.

Choudhury, A. (2013). *Google Outage*. Retrievable at `http://webmasterfacts.com/google-outage/`.

CloudFlare, I. (2013). *Business Service Level Agreement*. Retrievable at `http://www.cloudflare.com/business-sla`.

Credit, B. (2013). *Computation credit*. Retrievable at `http://boinc.berkeley.edu/wiki/Computation_credit`.

Cunsolo, V. D., Distefano, S., Puliafito, A. and Scarpa, M. (2010). Open and Interoperable Clouds: The Cloud@ Home Way. In *Cloud Computing*. (pp. 93–111). Springer.

Distefano, S., Cunsolo, V. D. and Puliafito, A. (2010). A taxonomic specification of Cloud@ Home. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*. (pp. 527–534). Springer.

Domingues, P., Araujo, F. and Silva, L. (2009). Evaluating the performance and intrusiveness of virtual machines for desktop grid computing. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. IEEE, 1–8.

Dropbox (2012). *Dropbox Terms of Service*. Retrievable at `https://www.dropbox.com/privacy`.

EC2, A. (2013). *Amazon EC2 Service Level Agreement*. Retrievable at `http://aws.amazon.com/ec2-sla/`.

Fedak, G., Germain, C., Neri, V. and Cappello, F. (2001). Xtremweb: A generic global computing system. In *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on*. IEEE, 582–587.

Ferreira, D., Araujo, F. and Domingues, P. (2011). libboincexec: A generic virtualization approach for the BOINC middleware. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*. IEEE, 1903–1908.

Figueiredo, R. J., Dinda, P. A. and Fortes, J. A. (2003). A case for grid computing on virtual machines. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*. IEEE, 550–559.

Fisher-Ogden, J. (2006). Hardware support for efficient virtualization. *San Diego, USA*.

Foster, I. (2002). What is the grid?-a three point checklist. *GRIDtoday*. 1(6).

Foster, I. and Kesselman, C. (2003). *The Grid 2: Blueprint for a new computing infrastructure*. Access Online via Elsevier.

Foster, I., Kesselman, C. and Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *International journal of high performance computing applications*. 15(3), 200–222.

Foster, I., Zhao, Y., Raicu, I. and Lu, S. (2008). Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08*. Ieee, 1–10.

Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A. and Stoica, I. (2009). Above the clouds: A Berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*. 28.

Gens, F. (2009). *New IDC IT Cloud Services Survey: Top Benefits and Challenges*. Retrievable at `http://blogs.idc.com/ie/wp-content/uploads/2009/12/idc_cloud_challenges_2009.jpg`.

Google (2013). *Google Apps Service Level Agreement*. Retrievable at `http://www.google.com/apps/intl/en/terms/sla.html`.

Greenberg, A., Hamilton, J., Maltz, D. A. and Patel, P. (2008). The cost of a

cloud: research problems in data center networks. *ACM SIGCOMM Computer Communication Review*. 39(1), 68–73.

Hans Meuer, J. D. H. S., Erich Strohmaier (2013). *Top 500 Supercomputer Sites*. Retrievable at `http://www.top500.org/`.

HTcondor (2013). *What is HTCondor?* Retrievable at `http://research.cs.wisc.edu/htcondor/description.html`.

Huhns, M. N. and Singh, M. P. (2005). Service-oriented computing: Key concepts and principles. *Internet Computing, IEEE*. 9(1), 75–81.

Inc, A. (2013). *iCLOUD TERMS AND CONDITIONS*. Retrievable at `https://www.apple.com/legal/internet-services/icloud/en/terms.html`.

Kirch, J. (2007). *Virtual machine security guidelines*. Retrievable at `http://www.cisecurity.org/tools2/vm/CIS_VM_Benchmark_v1.0.pdf`.

Korpela, E. J. (2012). SETI@ home, BOINC, and volunteer distributed computing. *Annual Review of Earth and Planetary Sciences*. 40, 69–87.

Krsul, I., Ganguly, A., Zhang, J., Fortes, J. A. and Figueiredo, R. J. (2004). Vmplants: Providing and managing virtual machine execution environments for grid computing. In *Supercomputing, 2004. Proceedings of the ACM/IEEE SC2004 Conference*. IEEE, 7–7.

Kumar, V., Grama, A., Gupta, A. and Karypis, G. (1994). *Introduction to parallel computing*. vol. 110. Benjamin/Cummings Redwood City.

Litzkow, M. J. (1987). Remote Unix: Turning idle workstations into cycle servers. In *Proceedings of the Summer USENIX Conference*. 381–384.

Lynch, N. A. (1996). *Distributed algorithms*. Morgan Kaufmann.

Marosi, A., Kovács, J. and Kacsuk, P. (2012). Towards a volunteer cloud system. *Future Generation Computer Systems*.

Marosi, A. C., Kacsuk, P., Fedak, G. and Lodygensky, O. (2010). Sandboxing for desktop grids using virtualization. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*. IEEE, 559–566.

McGilvary, G. A., Barker, A., Lloyd, A. and Atkinson, M. (2013). V-BOINC: The Virtualization of BOINC. *arXiv preprint arXiv:1306.0846*.

Mell, P. and Grance, T. (2011). The NIST definition of cloud computing (draft). *NIST special publication*. 800(145), 7.

Microsoft (2013). *Service Level Agreements*. Retrievable at http://www.windowsazure.com/en-us/support/legal/sla/.

Nouman Durrani, M. and Shamsi, J. A. (2013). Volunteer computing: requirements, challenges, and solutions. *Journal of Network and Computer Applications*.

Nov, O., Anderson, D. and Arazy, O. (2010). Volunteer computing: a model of the factors determining contribution to community-based scientific research. In *Proceedings of the 19th international conference on World wide web*. ACM, 741–750.

Opensuse (2012). *Chapter 2. KVM Limitations*. Retrievable at http://doc.opensuse.org/products/draft/SLES/SLES-kvm_sd_draft/cha.kvm.limits.html.

Palankar, M. R., Iamnitchi, A., Ripeanu, M. and Garfinkel, S. (2008). Amazon S3 for science grids: a viable solution? In *Proceedings of the 2008 international workshop on Data-aware distributed computing*. ACM, 55–64.

Patterson, D. A. and Hennessy, J. L. (2008). *Computer organization and design: the hardware/software interface*. Morgan Kaufmann.

Popovic, K. and Hocenski, Z. (2010). Cloud computing security issues and challenges. In *MIPRO, 2010 proceedings of the 33rd international convention*. IEEE, 344–349.

Preimesberger, C. (2013). *Virtualization Will Keep Driving Storage Market in 2013: 10 Reasons Why*. Retrievable at http://www.eweek.com/virtualization/slideshows/virtualization-will-keep-driving-storage-market...

Project, W. (2013). *Wine user guide*. Retrievable at http://www.winehq.org/docs/wineusr-guide/index.

Raphael, J. (2013). *The worst cloud outages of 2013 (so far)*. Retrievable at http://www.infoworld.com/slideshow/107783/the-worst-cloud-outages-of-2013-so-far-221831#slide4.

Rimal, B. P., Choi, E. and Lumb, I. (2009). A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*. Ieee, 44–51.

S3, A. (2013). *Amazon S3 Service Level Agreement*. Retrievable at http://aws.amazon.com/s3-sla/.

Sahoo, J., Mohapatra, S. and Lath, R. (2010). Virtualization: A survey on concepts, taxonomy and associated security issues. In *Computer and Network Technology (ICCNT), 2010 Second International Conference on*. Ieee, 222–226.

Sanchez, C. A., Blomer, J., Buncic, P., Chen, G., Ellis, J., Quintas, D. G., Harutyunyan, A., Grey, F., Gonzalez, D. L., Marquina, M. *et al.* (2011). Volunteer Clouds and citizen cyberscience for LHC physics. In *Journal of Physics: Conference Series*, vol. 331. IOP Publishing, 062022.

Sarmenta, L. F. (2001). *Volunteer computing*. Ph.D. Thesis. Massachusetts Institute of Technology.

Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., LeBlanc, R. and Lunt, B. (2006). Computing curricula 2005: The overview report. *ACM SIGCSE Bulletin*. 38(1), 456–457.

Shankar, S. (2009). *Amazon elastic compute cloud*.

Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A. and Peterson, L. (2007). Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In *ACM SIGOPS Operating Systems Review*, vol. 41. ACM, 275–287.

Staff, C. (2013). *The 10 Biggest Cloud Outages of 2013 (So Far)*. Retrievable at `http://www.crn.com/slide-shows/cloud/240158298/the-10-biggest-cloud-outages-of-2013-so-far.htm?pgno=11`.

states, B. (2013). *BOINCstates*. Retrievable at `http://boincstats.com/en`.

SteffenMoeller (2013). *BOINC Uppercase job application*. Retrievable at `https://wiki.debian.org/BOINC/Server/Projects/UpperCase`.

Team, B. (2013a). *BOINC Software prerequisites (Unix/Linux)*. Retrievable at `http://boinc.berkeley.edu/trac/wiki/SoftwarePrereqsUnix`.

Team, G. (2013b). *Apps Status Dashboard*. Retrievable at `http://www.google.com/appsstatus#hl=en&v=status&ts=1383652254987`.

team, X. (2013). *Introduction to XtremWeb*. Retrievable at `http://www.xtremweb.net/introduction.html`.

Tsai, W.-T., Sun, X. and Balasooriya, J. (2010). Service-oriented cloud computing architecture. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*. IEEE, 684–689.

Vaquero, L. M., Rodero-Merino, L., Caceres, J. and Lindner, M. (2008). A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*. 39(1), 50–55.

Workstation, V. and Player, V. (2002). VMware Inc. *Palo Alto, California, USA*.