ITC 2/47
Journal of Information Technology and Control
Vol. 47 / No. 2 / 2018
pp. 236-248
DOI 10.5755/j01.itc.47.2.19738

Big Data Mining Using Public Distributed Computing

Received 2018/03/05 | Accepted after revision 2018/05/08

http://dx.doi.org/10.5755/j01.itc.47.2.19738

Big Data Mining Using Public Distributed Computing

Albertas Jurgelevičius
Institute of Data Science and Digital Technologies; Vilnius University; Akademijos str. 4, LT-08663 Vilnius, Lithuania; e-mail: albertas.jurgelevicius@mii.vu.lt

Leonidas Sakalauskas
Prof. habil. dr., Šiauliai University; P. Visinskio str. 25, LT-76285 Šiauliai, Lithuania;
e-mail: leonidas.sakalauskas@mii.vu.lt

Corresponding author: albertas.jurgelevicius@mii.vu.lt

Public distributed computing is a type of distributed computing in which so-called volunteers provide computing resources to projects. Research show that public distributed computing has the required potential and capabilities to handle big data mining tasks. Considering that one of the biggest advantages of such computational model is low computational resource costs, this raises the question of why this method is not widely used for solving such today's computational challenges as big data mining. The purpose of this paper is to overview public distributed computing capabilities for big data mining tasks. The outcome of this paper provides the foundation for future research required to bring back attention to this low-cost public distributed computing method and make it a suitable platform for big data analysis.

KEYWORDS: Distributed public computing, BOINC, big data mining, cloud computing, computational costs.

1. Introduction

Many companies and organizations in today's world are interested in gathering data for various decision-making tasks. It leads companies to capturing, storing and processing huge data sets, which in turn refers to a term called big data mining. Over time data sets become large and connected to many data points making data difficult to store and process. Big data mining is a computational process of discovering patterns in large data sets. It takes use of methods at the intersection of artificial intelligence, machine learning, statistics and database systems. Unfortunately, internal high-performance computing environments or similar traditional data management solutions are no longer capable of handling such amounts of data [47]. Organizations do not usually have enough internal computational resources to satisfy the demand.

Owning a high-performance grid computing infrastructure may exceed the financial capacities [47]. Even though there are available distributed computing solutions that allow to easily combine internal IT resources into a distributed computing platform, organizations tend to choose other alternatives. One of vastly used solutions is cloud computing, services of which are growing continuously. High performance computational services are being provided at cheaper and cheaper rates and now a number of external companies offer their cloud and big data mining solutions at affordable price. Companies often use cloud computing services from Microsoft, Amazon, Google, Rackspace and others to manage and process the data they have [51]. Research show that small and medium enterprises (SMEs) consider services provided by external cloud computing companies more secure than solutions deployed using their internal infrastructure [34]. This makes SMEs interested in cloud and public distributed computing solutions.

Unfortunately, SMEs are lacking applications that could enable to use their internal IT infrastructures (including desktop computers) as platforms for data analysis and decision-making tasks. To better understand the field, in Section 2 we will present volunteer-based computing solutions (alternatives to cloud computing). Next, we will explore BOINC framework for big data mining applications and review projects based on BOINC in Section 3. We will present a review of adoption issues and possible points of improvement in Section 4. Finally, in Section 5, we will summarize the status of the field and will determine possible directions for future work and research needed to make public distributed computing a desired platform for big data mining tasks.

2. Volunteer and Cloud Computing

Interconnected public computers can be set up to solve a given task in parallel. This method is called public distributed computing. It uses client-server model, where client nodes provide their resources to project server. Network connection is required for the nodes to communicate with the server. This allows them to request server for new tasks and to send back the results. In some cases, nodes can be set up to communicate with each other, however, tasks are usually individual and performed in parallel.

Public computational resource harvesting may be used for distributed computing. This model is called volunteer computing. It relies on people or organizations donating CPU time, network and storage capabilities from the computers that they own. This way, computational resources can be joined to the open dynamic network, where new nodes can be easily added, and old ones removed. Distributed computing is an effective way of taking resources from the compute nodes that would otherwise be wasted. Such an infrastructure does not have any additional costs and may even reduce the already existing ones, making it a great asset. Furthermore, distributed computing model-based infrastructure can be set up on a closed network. This way, the infrastructure will be limited to nodes that are stored inside the organization premises. It can be entirely managed by its staff if needed. These are two major advantages over popular alternatives like cloud computing:

- reduced costs on internal IT infrastructure and external services;
- good solution to some of the data security issues.

We will review security and other adoption issues further in Section 4.

Network connected volunteer Desktop Grids donate resources to solve large computational problems. Such resources include processing power of CPU, GPU, RAM, storage and internet connection. These resources are harnessed from idle machines using a centralized master/worker model. One of the most important points that we must consider in the systems that are based on volunteers is that it is necessary to attract and convince the volunteers to participate. The most popular way of drawing volunteers is to get them rewarded with credit points. Points are estimated by calculating the contribution to scientific progress of the project. However, these credit points are usually just mean of measure and provide no real value to the volunteers. In exchange, the donated resources are contributed to various public projects by executing project-related independent tasks. Most of such projects uses the BOINC framework that we will review in Section 3.1. [7, 8, 17].

Volunteer or crowd computing is a popular method for solving complex research problems from an increasingly diverse range of areas [41]. It permits the user to

forget about certain costs associated with buying and maintaining physical infrastructures and helps disseminate the project to the public [25]. Establishing an own specialized center for data processing might be too expensive or inappropriate. It can be a big issue if the demand for data analysis is irregular [30]. Volunteer computing consists of two aspects: computation and participation [8]. These aspects are related to:

- allocation and management of large computational jobs. Most computing activities that depend on interactive input from the user barely load the machines. This is causing high percentage of resource idleness, making these resources attractive for harvesting [6];
- encouragement and persuasion of individuals to donate their computing resources to the project. A powerful security mechanism must be deployed that makes sure the users are relieved from the security concerns [7]. Trust motivates participants not only to share some of their resources, but to provide more access to their computers [37].

2.1. Extending the Volunteer Computing Model

A number of research projects have been conducted to tackle the problem of gaining the required amount of resources for certain computations. One of such has been presented by EU FP7 EDGI project, reviewed in [37]. The project combines BOINC and XtreemWeb like desktop grids with cloud computing services. It extends these grids with new resources on demand making this solution to be like SaaS clouds [37]. Users do not have to take costs for additional resources, since they are collected from volunteers. Furthermore, such a solution results in improved response times in volunteer desktop grid systems. Volunteer cloud can be an improvement of the cloud paradigm making cloud resources to be provided by the volunteers [37]. We strongly agree with these claims, since big data mining requires large computational resources that many cannot simply afford. The on-demand extension of volunteer Desktop Grid (DG) resources with cloud resources was also reported in [32]. Of course, volatility and availability issues cause significant difficulty in big data mining applications that we will discuss later.

Clouds@home is another volunteer cloud system that is considered as a new form of cloud computing. The aim of the project was to build a low scale and price cloud computing infrastructure by merging cloud computing and volunteer computing services. It builds Cloud-like-infrastructure from volunteer computers. The idea is based on enabling virtualization technology in volunteer computing resources [21], an approach named "application sanboxing". It isolates the application inside the virtual machine (VM) by using a wrapper for launching VM and managing applications that have run on it [36, 37].

Despite the efforts to improve or extend, volunteer computing is still not always considered the best answer to all problems. The solution does not need to be always volunteer-based. According to [30], there are some cases when attracting computing resources from the outside is hard or inappropriate. Data might be confidential, or there might be big amounts of data to be analyzed. Since big data mining tasks can easily involve datasets containing sensitive data, this suggests that resource harnessing solutions should not rely on volunteer computing only and should either integrate some other platforms or apply some data protection methods. We will discuss this further in Section 4.5.

3. Big Data Mining Using BOINC

Big data mining is a process of discovering new knowledge from large volumes of data using statistical methods or artificial intelligence tools. It is an important process in many industries: automotive, healthcare, banking, insurance, consumer products, oil and gas, energy and utilities, retail, government, telecommunications, travel and transportation [30]. It is expected for the total worldwide data being gathered to reach 39 trillion gigabytes by 2020 [15]. According to survey results, in 2013 about 28% of interviewees working in data mining field have dealt with sets ranging from 1 to 100 petabytes. About 2% have worked with data sets larger than 100 PB [43].

High-performance computing systems are required for solving big data mining tasks. Several solutions are available to get the required resources [30]:

- Computing Clusters (require significant costs for the implementation and support);
- Service Grids (cheaper, but require effort on support);

_ Desktop Grids (cheap and easy to support; harvest available resources from personal computers; not as reliable).

BOINC can be almost considered a standard for running volunteer computing projects. It is the most popular free middleware software used for heterogeneous Desktop Grids [30]. Even though BOINC allows creating low cost enterprise-level computing grids, there are challenges developing a BOINC-based big data mining applications [10, 30]:

_ it might be difficult to decompose the tasks into smaller independent ones;

_ huge data sets to be transferred to the client nodes may overload the network;

_ it is difficult to adapt applications to work on various operating systems and architectures;

_ it is time consuming to keep track of user accounts, deal with redundancy and application errors.

Next, we will examine BOINC framework and the approaches of solving big data mining tasks using it. We will give special attention to performance and quality of service, since these are well solved issues in alternative solutions, such as cloud computing services.

3.1. BOINC

The most popular and standard software for volunteer computing projects is called BOINC (Berkeley Open Infrastructure for Network Computing) [4, 25]. BOINC is designed for volunteer and grid computing. It is ideal in cases where not only low-cost access to massive computing resources is needed, but also for projects having significant public interest in the research being done [41]. It is an open-source middleware system that provides a distributed computing infrastructure. Such an infrastructure does not depend on the scientific computations or experiments. BOINC projects are usually designed for solving challenging scientific problems. Such projects are exploiting opportunistic resources; therefore, project owners must gain public trust and interest. Usually, this is done by providing good and trustworthy appearance [37].

BOINC is based on client-server architecture and may be used as an example of how public distributed computing model works:

_ data are stored on a common database;

_ server divides resource demanding tasks into smaller ones;

_ tasks and data are requested and retrieved by the client nodes;

_ computations are performed without separate node interaction;

_ computation results are uploaded to a server and merged into the final solution.

There are many projects based on BOINC, some of more popular ones are the following:

_ CERN + KC Gigabit Computing Challenge (https://cernkcchallenge.github.io/Cern-KCChallenge/);

_ Gridcoin (http://gridcoin.us/);

_ SETI@home (http://setiathome.ssl.berkeley.edu/).

To better understand BOINC, one can look at a list of special services that usually run on BOINC servers:

_ Transitioner – handles state transitions of work units and results;

_ Feeder – enhances the performance scheduler;

_ Validator – checks the validity of the results received from the work unit;

_ Assimilator – processes the results according to application-specific rules;

_ File remover – deletes input and output files when jobs are completed;

_ Work generator – generates work units and corresponding input files;

_ Database cleaner – moves the result and work unit records from the database to XML-format archive files;

_ Scheduler – assigns jobs to client nodes depending on their characteristics.

Central server may be distributed among several machines to handle the loads. BOINC API is available to developers not only to make their BOINC applications run on BOINC platform and interact with BOINC-client [24], but also to do reporting, process visualizations and checkpoints to show the project status and help prevent process loss when computational resources become unavailable. There are also solutions allowing legacy application to run in a BOINC-grid without any modifications to the source code [35, 53].

Resources can be donated by downloading a lightweight client daemon and connecting it to the project

URL. This way, the BOINC client manages communication with the BOINC server, download of applications and work units, client-side scheduling and upload of the results [8]. The server in return filters out malicious results from unreliable resources by using custom validation scripts or redundancy checks [37].

Despite its popularity, BOINC still has many drawbacks that will be discussed in Section 4. It also has two limitations [7]:

- applications running on BOINC platform are limited to the architecture and operating system of the environment they are executed in;
- BOINC client does not provide adequate security for users.

These limitations are going to be further discussed in Sections 4.4 and 4.5, accordingly.

BOINC-based Desktop Grid provides means for small scientific groups and up to medium size companies to work on high performance computations demanding problems, like big data mining. This can be done using either own or volunteer computing resources. It may appear that huge additional resources are needed to transfer big amounts of data for analysis. However, according to [30], BOINC-based Desktop Grids can be used on modern networks. According to them, such grids provide the means to process dozens of terabytes of data. A high-speed local network is required for connecting the computational resources.

3.2. Data Mining Tools for BOINC

An observation made by [11] shows that when it turns to platforms that harness resources all over the internet, most of the existing solutions are built and optimized to run Bag-of-Tasks applications. Therefore, solutions such as BOINC, GridBot [49], Bayanihan [46], and many others [2, 9, 14, 33] do not support the execution of MapReduce jobs. However, there has been some research done towards solving this problem. One of such is BOINC-MR [17], a system able to run MapReduce applications on top of BOINC. The goal was to support MapReduce (software popularized by Google [19]) on top of an insecure, unreliable environment. The presented solution works as follows:

1. a client (reducer) requests for a new task from project server;
2. a scheduler appends to each reduce task an IP and port of mappers holding output for the same job;
3. a reducer then has the possibility to download the required input files directly from the mappers. A data copy is stored on for data availability in case of an error;
4. each reducer processes the downloaded data by running its task;
5. the results are sent back to server.

It is worth noting that BOINC-MR client may act as a reducer or as a mapper. This depends on the obtained task and done by running the required executable [17].

"distributedDataMining.org" is one additional project integrating RapidMiner to BOINC [47]. The project goal is the same: to perform data mining tasks. However, the approach is less complex. Data with an assigned task are pulled from the server using the BOINC client. Then, the BOINC client runs data mining tasks by starting an instance of the RapidMiner framework. It performs parallel data intensive computations and sends back the results to server. The results are collected and sent further to be analyzed by researchers.

MapReduce workflows could be enhanced by making project data distribution subsystems use peer-to-peer protocol called BitTorrent [11]. This would allow nodes to download data from multiple sources simultaneously. This would speed up the download process, improve scalability and take some of the load from the central server to the compute nodes. Failing nodes during data transfer would not compromise the task, thus improving the fault tolerance.

A direct approach of extracting association rules from large data sets using BOINC-based Enterprise Desktop Grid was described in [30]. Their solution could be continued in the same direction by implementing other big data mining methods. This would allow small, midsize businesses, scientists and organizations to use it for solving their big data problems. However, in this case, each algorithm would have to be adapted and implemented to work with BOINC.

Another interesting approach is called Ad hoc cloud computing [39]. This platform also allows performing big data mining tasks using BOINC, however, it uses existing user infrastructure as a cloud service. The scale of infrastructure may range from a startup company owning several personal computers to a large organization. The difference from standard cloud service is that Ad hoc clouds gather resources from member hosts. These nodes may also be used by their owners for other tasks and may sporadically become

unavailable. Such a concept improves infrastructure efficiency, utilization and return on IT investments by reducing service costs. However, such a solution may result in an unreliable infrastructure. Despite this, it has a great benefit for users looking to migrate to commercial or private cloud models.

It is a good idea to try out and examine the capabilities of Ad hoc clouds to see if the solution is suitable. If not, then the decision to adopt commercial model can be made. Based on initial evaluation, the concept is feasible. It can be reliable and offer comparable performance to Amazon EC2 [39].

The overviewed big data mining platforms show that public distributed computing solutions can compete with the existing cloud computing solutions. Even though users prefer to employ technology without having to have expert skills, the recent cloud computing model is also not fit with a scientific problem which is complex and needs large computational power. In both cases, technology is a combination of knowledge and working hard. When users are required to solve a certain task using some new technology, they do not want to invest a lot of time and effort to understand how it works [7, 8].

3.3. Resource Availability and Costs

Organizations may increase the number of their computational resources and reduce service costs by using a distributed computing platform, for example, BOINC. This platform performs assigned computational tasks by interconnecting owned computational resources into a dynamic size computation platform. Furthermore, it does not interfere with the employee ongoing work processes. It becomes easier to protect private data, since all computations are done inside the premises with a possibility to add data access levels if required. Such a platform can improve service quality, since services can be customized and adapted for organization specific needs, like big data mining.

CPU usage and power consumption measurements from [31] can be used to support resource availability and cost reduction claims. Measurements were taken during a short 28-day experiment using energy meter device and Microsoft Windows application called "Performance monitor". It was performed in two organizations (organization A and B). Two computers were picked randomly for the experiment and both were configured to run a BOINC client. Both computers were registered to participate in SETI@home project (analyze the radio signals) for two weeks. Afterwards, measurements were taken without any BOINC projects running on the computers for another two weeks. Employees used both computers for work-related tasks during the whole period (except weekends). Fig. 1 and Table 1 show that computers barely performed any computations and were wasting resources during the period when no BOINC projects were running [31].

Figure 1

Processor idle time with and without SETI@home project running in Organization A. [31]

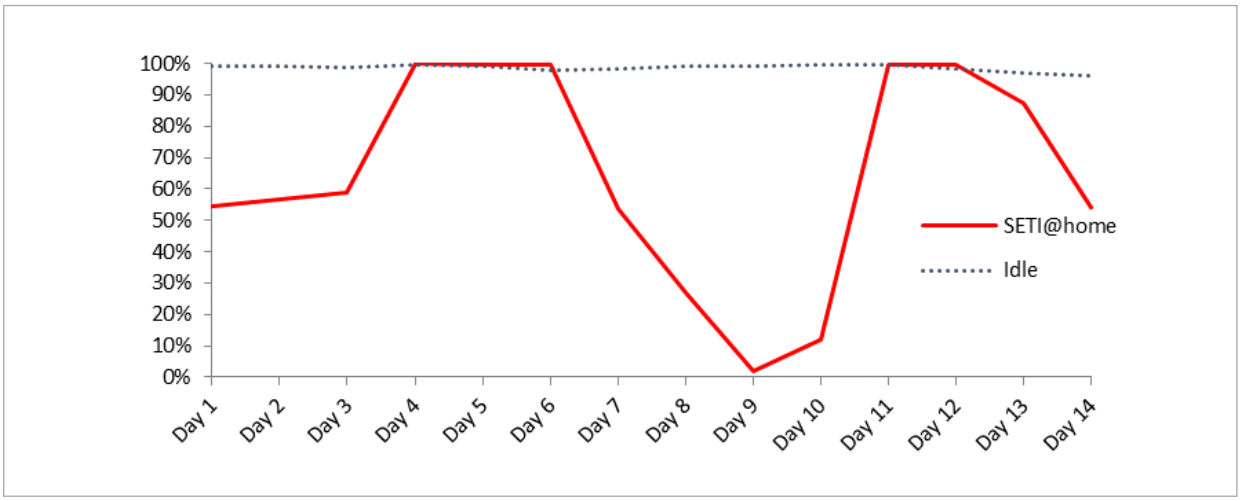

Table 1
Resource and power consumption statistics [31]

#	BOINC project	CPU average idle time	Power consumption
A	Not running	98,77%	16,61 kWh
	SETI@home	65,23%	22,03 kWh
B	Not running	83,49%	1 kWh
	SETI@home	26,86%	2,09 kWh

SETI@home project is designed specifically to handle big data sets. We would like to outline the fact that internal computational resources could be used for other big data mining tasks as well. The question is how large the internal IT infrastructure should be to handle such tasks. This will be discussed in the next section.

Even though a small number of machines participated in the experiment, the results suggest that such a solution might work. Computers were barely loaded by the employees which allowed to perform the additional computation at little cost. Furthermore, it did not interrupt any of the ongoing work processes. It is possible for an organization to perform additional big data mining and similar tasks using public distributed computing platform. However, a follow-up experiment with higher variety of instances should provide more insights.

4. BOINC Adoption Issues

Volunteer computing has some challenging issues: volatility, lack of trust, failure, heterogeneity, voluntary participation. There are certain challenges moving from costly and modern cloud datacenters to volunteer resources [8, 37, 41]:

- motivating donors to donate resources and in some cases provide more access to their computers;
- solving volatility and availability problems of volunteer resources;
- increasing the service migration efficiency (requires forecasting the availability of the volunteer nodes);
- solving the need for new computational and storage resources;
- having on-demand and more predictable return of simulation results;
- rationalizing the costs.

A number of research papers have been published addressing these issues and exploring the combination of volunteer and cloud computing as possible improvements of, or extensions to a real existing project.

According to [52], environmental, organizational, managerial and technological factors can also influence the decision adopting solutions based on distributed public computing. We only reviewed the technological factors. Other factors may present additional adoption issues that might be interesting research topics and reveal new factors that influence the decision.

Next, we will review possible solutions to the challenges above.

4.1. Quality of Service

BOINC provided services are not as reliable due to the public distributed computing model drawbacks. Depending on the project, uncertain amounts of data can be queued for processing at any time. The problem is that it is currently not possible to pre-determine the number of computational resources that are going to be available to execute the task. Furthermore, nodes often can become unavailable making it difficult to predict amount of time required for certain task to complete. As a result, it is difficult to tell if the available compute nodes are going to be enough for a certain task to finish in a decent amount of time. It may be especially relevant when handling emergency data containing tasks.

These issues are not acceptable in big data mining applications. Any delays might cause data to pile up even more and crash the whole process. Quality of Service (QoS) must be supported to help deal with this issue. It would dedicate more resources to process urgent data and give priority to send it to the client nodes.

QoS improvement methods have been researched in [39]. The proposed solution takes advantage of virtual machines by using V-BOINC [38]. This approach not only solves the task continuity problem, but also solves dependency issues that we will discuss further in Section 4.4. The V-BOINC project server sends VM image along with the configuration script to the client. The configuration script allows to set the number

of CPUs to use, memory, disk space on other limitations. Client node then launches the virtual machine, which in turn starts receiving BOINC tasks and returning the results. According to [42], this approach has security concerns, since VMs are under risk of their images being altered with the malicious code injection. This also applies for the offline nodes too. Furthermore, VM images may preserve the original owner information. This data may be passed to the new consumer. Security concerns will be further discussed in Section 4.5.

Since there are yet no reliable methods of ensuring QoS, this could be a good research topic.

4.2. Capacity Prediction

BOINC client determines host hardware capacity and characteristics by periodically measuring host availability parameters such as uptime and period client is online. BOINC also regularly runs the Whetstone [18] and Dhrystone [54] benchmarks [5]. However, BOINC only measures storage space on machines it is installed on. It may overestimate available disk space in shared network-accessible volumes, having more hosts running BOINC client. Despite this, a study on the potential resource capacity of volunteer computing platform was run. Tests were based on power, memory, disk space, network throughput, host availability, user-specified limits on resource usage, and host churn [5]. The study showed that this is enough to calculate total capacity of volunteer resources. Resulting expression (1) shows total floating-point computing power X available to a project [5]:

$$X = X_{\text{arrival}} \times X_{\text{life}} \times X_{\text{ncpus}} \times X_{\text{flops}} \times X_{\text{eff}} \times X_{\text{onfrac}} \times X_{\text{active}} \times X_{\text{redundancy}} \times X_{\text{share}} \quad (1)$$

As defined in [5], X_{arrival} is the average arrival rate of hosts, X_{life} is the average lifetime of hosts, X_{ncpus} is the average number of CPUs per host, X_{flops} is the average FLOPS per CPU, X_{eff} is the average CPU efficiency, X_{onfrac} is the average on-fraction, X_{active} is the average active-fraction, $X_{\text{redundancy}}$ is the reciprocal of the average redundancy, and X_{share} is the average resource share (relative to other CPU-intensive projects).

However, this solution may not be sufficient. Prediction methodologies can significantly improve efficiency of volunteer resource task distribution management. It could be done by estimating the short-term resource behavior patterns. The Climateprediction.net project (CPDN) was created in 1999 (Allen, 1999; CPDN, 2015) as a distributed computing initiative to address the uncertainties [41]. Urgent simulations, real-time and unexpected events may require greater resources than volunteer computers may provide at a certain time. The behavior of volunteers cannot be clearly anticipated or measured. If predicted, it is then possible to take proper precautionary actions [6]. One way of fixing this is to re-engineer client part of a volunteer computing model to an infrastructure as a service (IaaS) that would be based on cloud computing (e.g. Amazon Web Services, AWS) [41].

Organizations worry about cloud computing service availability [8]. Not only Desktop Grids, such as BOINC or XtremWeb [13], have centralized architectures causing a potential bottleneck in the continuing evolution of volunteer computing systems, but there are also worrying signs of stagnation of active users and projects. This causes problems that are related to data storage and distribution [3, 16, 17]. Efficiency of use of desktop grid environments can greatly benefit with prediction of resources availability. As shown in Table 2, all major cloud providers offer high availability.

Table 2
Public cloud resource availability [8]

Cloud vendor	Name of services	Monthly uptime percentage
Google	Google apps	< 99,9% -> = 99,0%
Amazon	Amazon EC2	< 99,5% -> = 99,0%
Amazon	Amazon S3	< 99,9% -> = 99,0%
Microsoft	Cloud services	< 99,95%
Cloud flare	CloudFlare	100%

The major drawback of Desktop Grids lies in the volatility of resources. Such issues are caused by the system control policies [6]. Frameworks for automated behavior prediction may help solve the problem. Such frameworks may include calendar methods [29] or classification algorithms known from data mining [55]. The main drawback of these methods is that the period is determined in an arbitrary way. For exam-

ple, the most advanced framework for resource prediction in computer systems is the Resource Prediction System [20], but the main disadvantage is that it focuses on very short-term predictions. A prediction study has been performed on institutional desktop pool using three prediction targets: CPU idle time, memory load and machine availability [6]. The conclusion of the study has shown that even highly dynamic environments such as desktop pools allow for meaningful predictions of a variety of metrics using the Support Vector Machines classifier (SMO).

4.3. Cost Estimation

Volunteer computing can create noticeable energy savings by reducing the amount of internal resources needed to perform the computations. It allows to acquire the hardware from multiple users, thus diverging the costs. This is especially significant if GPUs are used for computations, which may raise the power requirements up to 30% per each instance [25]. Current high-end GPUs have high energy consumption rates, making it a big issue having large clusters. Energy supply costs can be a significant portion of the total expenses for the property [22, 23].

There are cost estimation models for local GPU-based infrastructures using equations for calculating computation costs per each instance [25]. The equations may consider energy consumption costs, machine market price and machine collocation costs. However, despite the research efforts there is still no reliable way of determining the data processing time costs that would help for capacity prediction as mentioned before.

4.4. Virtualization

Pre-installing other dependencies (libraries, etc.) for regular BOINC applications is not possible or is hardly solvable [37]. Virtualization is used for hiding physical resources of system from the operating system and many issues can be solved using it [38]. Virtualization is defined as a technology that introduces a software abstraction layer between the hardware and the operating system and applications running on top of it [7, 45]. Virtualization technology makes application porting to volunteer desktop grids a lot easier [37]. Otherwise, to make an application work on different architectures, project developers must:

- compile application to target architectures;
- preserve execution progress upon termination;
- either not use dependencies or compile them into application if possible;
- gain volunteer trust that the project application is trustworthy and not malicious.

Solving such problems takes valuable time from project developers. Virtualization eliminates unnecessary application of porting step for BOINC systems. Parameter sweep applications are a good case example [37]. Without virtualization, every parameter sweep application required a porting effort and hence the service grid users would not be interested in the Desktop Grid extensions. Applications traditionally running under BOINC initially had to be compiled for each different client operating system. Developments at CERN [12, 26] (pioneered by the Test4Theory LHC@home project during 20102011 [12, 26, 28]) provided means of distributing a virtual machine [50] to the volunteer computers via BOINC [7]. CernVM [48] has employed BOINC VBoxWrapper tool [7]. Later, BOINC developers enabled virtual machine functionality by implementing interface (VBoxWrapper) between BOINC client and VirtualBox [44]. They integrated VirtualBox by storing application and its data in a shared folder between the host operating system and virtual machine, executing the computations.

In addition to this, another project called V-BOINC was introduced by [38]. V-BOINC also uses virtualization to run applications on volunteer computers and sends small virtual machine images to volunteers. This way, BOINC applications run within the virtual machines rather than on host operating system directly. This in turn enabled developers to use dependencies under V-BOINC and increased variety of applications volunteer infrastructure can run. More science and business projects could be carried out this way, since applications become available to wider public [38]. Furthermore, V-BOINC solved additional problems like:

- application trust;
- progress preservation upon termination.

Even though execution time is longer due to virtualization (due to user setting of the virtual environment and hypervisor [38]), a lot of volunteer computing projects have used V-BOINC for its benefits. It is worth noting that not all CPU cores and memory can be used because of VirtualBox limitations. Furthermore, virtual

image sizes can vary from a couple of hundred megabytes to tens of gigabytes. This also makes management and deployment a major challenge [37].

4.5. Security

Information and system security, in general, has always been a concern. These concerns must be addressed to make users and enterprises at least consider public distributed computing for adoption and use. Similar security issues have been addressed and successfully solved in the cloud computing model that is currently widely used. This is anticipated, as one of the major requirements for any information system is security. Users must be confident and trust their data and resources to the environment they use. Data must be protected from any unauthorized access, including system attacks.

One approach that can help to protect the data is cryptography. Many cryptographic algorithms can be used in cloud computing providing greater security [40]. Such a solution can also be deployed in public distributed computing applications. According to [40], cloud computing solutions must offer on-demand self-service and resource pooling. Infrastructure must support rapid elasticity and have broad network access. Finally, the service and its costs must be measurable. BOINC already meets most of these requirements, including measured service. However, it fails to measure costs and delivery time. We believe that these limitations require research and need to be solved.

Both distributed public computing and cloud computing models have similar security concerns [40]:

- system availability;
- data and system integrity;
- user authentication;
- data backups and recovery;
- data confidentiality;
- privacy and access control.

Distributed public computing and cloud computing models share many issues. However, a lot of them are already solved in cloud computing model. As we know, distributed public computing is a cheap solution for solving resource demanding tasks. Solving these issues would make it a good alternative to cloud computing solutions and could open many new opportunities.

The main challenge in volunteer computing is security, since entire computing jobs are done using resources from volunteers [8]. BOINC runs under two less privileged accounts. The first one, which has the more privileges, is for the BOINC client. The BOINC client continuously monitors the running applications that are executed under an even more restricted account. However, according to [37], malicious application might escape this supervision. Cloud computing solutions take this problem particularly serious. Ensuring data security is a lot more difficult in cloud environment than traditional information systems [51]. Traditional data access protection methods that rely on identity management and authorization are not a viable solution anymore for protecting data on clouds. Furthermore, research shows that cloud computing solutions provide additional risk by requiring outsourcing essential services to third party service providers, making it harder to demonstrate compliance, maintain data privacy and provide required service availability. Cloud computing applications have already solved private data protection issues that public distributed computing applications still must deal with. The same privacy issues apply for big data mining tasks. It can be especially difficult to prevent unauthorized access to data, which is distributed throughout different kinds of environments (servers, personal computers, smart devices). Cloud computing solutions present many benefits to adopting the technology, however, they also bring some challenges to adoption, such as: security, compliance, data protection and legal issues, related to outsourcing data to third party. It also involves some of the greater security concerns for data storage, execution environment, and networks [27]. Emerging new technologies will generate even more data that may be used for big data mining tasks. This additional data will also have to be protected from unauthorized access, modification and forgery, denial of service and other attacks [40]. Cloud of things (CoT) and other similar technologies will create new ways for hackers to get access to data [1]. Data privacy protection will especially be important in hybrid clouds (combination of private and public clouds) used for big data mining and other data related tasks. Security concerns and prerequisite must be solved first to create a trustworthy compute environment and win user confidence and make them to consider adopting such technology [27, 40, 51]. Data

privacy issues on cloud computing platforms can be grouped into the following categories [51]:

1. data access control to prevent unauthorized resale of the data;
2. data replication control to prevent data loss and unauthorized modifications;
3. supervision of personal information requirement compliance;
4. supervision of cloud subcontractors level of involvement in data processing.

Cloud computing model also covers resource security, management and monitoring issues. Even though all these issues are solved, the solutions are not standardized and have no regulations. Each cloud service provider has its own rules on how to deploy the applications to the cloud environment. Data are any organization's one of the most valued assets and their security is a major concern. Even though trusting the data to a third party platform can save organization's time and money, it is an additional risk to take. To use the required cloud computing services, data must be imported into the cloud by putting them either into the third-party database directly or through an application [51]. The lack of security, as a result, may prevent from adopting distributed computing model. Companies are required to host data on public infrastructures and outsource the security management. These actions further reduce control of IT assets and increase the probability of an attack. Cloud computing opponents also mention some of other issues, such as: vendor lock-in, network bandwidth, system availability and legal consequences [42].

It is clearly visible that businesses and organizations are interested in using external services such as cloud computing only if there is no serious threat to data security. However, even then, as cited in [1], organizations cannot be sure their data are secure: on Jan 30, 2013, The Independent published an article, stating, "British internet users' personal information on major 'cloud' storage services can be spied upon routinely by US authorities". Private data should not be stored in untrusted environment and a virtual storage server located inside the user's country or trusted geographical domain should be used [1].

5. Conclusion

In this paper, we have reviewed the distributed public computing model applications for solving big data mining tasks. Research results show that big data mining processes can be run at very low cost using on-premise IT infrastructure. Furthermore, research indicates that public distributed computing model is capable of handling big data processing without interrupting any other ongoing work processes. All the required tools and models are already available. Reviewed solution works well with big data projects like SETI@home that handles anonymous data and has available access to large IT infrastructure. However, security and reliability issues are some of the biggest concerns preventing this computational method from wider adoption. Solving these issues would not only open new research topics, but also make public distributed computing a great asset for any organization performing large computations and data processing.

The focus of our further research will be to tackle public distributed computing reliability and data security issues.

References

1. Aazam, M., Hung, P., Huh, E. Cloud of Things: Integrating Internet of Things with Cloud Computing and the Issues Involved. 11th International Bhurban Conference on Applied Sciences and Technology, (IBCAST 2014), Islamabad, Pakistan, 414-419.
2. Alexandrov, A., Ibel, M., Schauser, K., Scheiman, C. Superweb: Towards a Global Web-Based Parallel Computing Infrastructure. In Parallel Processing Symposium, 1997, 100-106.
3. Anderson, D. Boinc Status Report. In: The 7th BOINC Workshop, 2011.
4. Anderson, D. BOINC: A System for Public-Resource Computing and Storage. Proceedings of Fifth IEEE/ACM International Workshop on Grid Computing, 2004, 410.
5. Anderson, D. P., Fedak, G. The Computational and Storage Potential of Volunteer Computing. Sixth IEEE In-

ternational Symposium on Cluster Computing and the Grid, (CCGRID 06), 2006, 73-80.

6. Andrzejak, A., Domingues, P., Silva, L. Classifier-Based Capacity Prediction for Desktop Grids. CoreGRID Technical Report, (TR-0026), 2006.
7. Anjomshoa, M., Salleh, M. Overview on Clouds@home: Virtualization Mechanism for Volunteer Computing. Parallel and Distributed Computing Systems, (PDCS 2014), Ukraine, Kharkiv, March 4-6, 2014, 11-19.
8. Anjomshoa, M., Salleh, M., Kermani., M. P. A Taxonomy and Survey of Distributed Computing Systems. Journal of Applied Sciences, 2015, 15(1), 46-57.
9. Baratloo, A., Karaul, M., Kedem, Z., Wijckoff, P. Charlotte: Metacomputing on the Web. Future Generation Computer Systems, 1999, 15(5-6), 559-570.
10. Barbalace, D., Lucchese, C., Mastroianni, C., Orlando, S., Talia, D. Mining@Home: Public Resource Computing for Distributed Data Mining, Concurrency & Computation: Practice & Experience, Wiley, 2010, 22, 658–682.
11. Bruno, R., Ferreira, P. SCADAMAR: Scalable and Data-Efficient Internet MapReduce. Proceedings of the 2nd International Workshop on CrossCloud Systems, (CCB '14), Bordeaux, France, 2014, 2:1-2:6.
12. Buncic, P. CernVM – A Virtual Software Appliance for LHC Applications. Journal of Physics: Conference Series, 2010, 219, 042003.
13. Cappello, F., Djilali, S., Fedak, G., Herault, T., Magniette, F., Néri, V., Lodygensky, O. Computing on Large-Scale Distributed Systems: Xtremweb Architecture, Programming Models, Security, Tests and Convergence with Grid. Future Generation Computer Systems, 2005, 21(3), 417–437.
14. Chakravarti, A., Baumgartner, G., Lauria, M. The Organic Grid: Self-Organizing Computation on a Peer-to-Peer Network. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2005, 35(3), 373–384.
15. Clifton, C. Encyclopedia Britannica. http://global.britannica.com/EBchecked/topic/1056150/data-mining. Accessed on November 7, 2017.
16. Costa, F., Silva, L., Fedak, G., Kelley, I. Optimizing Data Distribution in Desktop Grid Platforms. Parallel Processing Letters, (PPL), September 2008, 18(3), 391–410.
17. Costa, F., Veiga, L., Ferreira, P. BOINC-MR: MapReduce in a Volunteer Environment. In: Meersman R. et al. (Eds.) On the Move to Meaningful Internet Systems: OTM 2012, (OTM 2012), Lecture Notes in Computer Science, 7565, Springer, Berlin, Heidelberg, 2012, 425-432.
18. Curnow, H. J., Wichmann, B. A. A Synthetic Benchmark. The Computer Journal, 1976, 19(1), 43-49.
19. Dean, J., Ghemawat, S. Mapreduce: Simplified Data Processing on Large Clusters. Commun, (ACM 51), January 2008, 107–113.
20. Dinda, P., O'Hallaron, D. An Extensible Toolkit for Resource Prediction in Distributed Systems. Carnegie Mellon University CMU-CS-99-138, 1999.
21. Distefano, S., Cunsolo, V. D., Puliafito, A. A Taxonomic Specification of Cloud@ Home. Advanced Intelligent Computing Theories and Applications, With Aspects of Artificial Intelligence, 2010, 527-534.
22. Fan, X., Weber, W. D., Barroso, L. A. Power Provisioning for a Warehouse-Sized Computer. In Proceedings of the 34th Annual International Symposium on Computer Architecture, (ISCA '07), ACM, New York, NY, USA, 2007, 13-23.
23. Feng, W. C., Cameron, K. W. The Green500 List: Encouraging Sustainable Supercomputing. Computer, 2007, 40(12), 50–55.
24. Frequent Itemset Mining Dataset Repository. http://fimi.ua.ac.be. Accessed on November 7, 2017.
25. Guerrero, G., Imbernón, B., Pérez-Sánchez, H., Sanz, F., Garcia, J., Cecilia, J. A Performance/Cost Evaluation for a GPU-Based Drug Discovery Application on Volunteer Computing. BioMed Research International, 2014, 474219.
26. Harutyunyan, A. CernVM Co-Pilot: A Framework for Orchestrating Virtual Machines Running Applications of LHC Experiments on the Cloud. Journal of Physics: Conference Series, 2011, 331, 062013.
27. Hashizume, K., Rosado, D., Fernandez-Medina, E., Fernandez, E. An Analysis of Security Issues for Cloud Computing. Journal of Internet Services and Applications, 2013, 4(5), 1-13.
28. Høimyr, N. BOINC Service for Volunteer Cloud Computing. Journal of Physics: Conference Series, 396, 2012, 032057.
29. Hollingsworth, J., Maneewongvatana, S. Imprecise Calendars: An Approach to Scheduling Computational Grids. International Conference on Distributed Computing Systems, 1999.
30. Ivashko, E., Golovin, A. Association Rules Extraction from Big Data Using BOINC-Based Enterprise Desktop Grid. Journal on Selected Topics in Nano Electronics and Computing, 2014, 2(2), 31-35.

31. Jurgelevičius, A., Sakalauskas, L. BOINC from the View Point of Cloud Computing. CEUR Workshop Proceedings, 1973, 2017, 61-66.
32. Kacsuk, P., Kovacs, J., Farkas, Z., Marosi, A., Balaton, Z. Towards a Powerful European DCI Based on Desktop Grids. Journal of Grid Computing, 2011, 9, 219-239.
33. Lo, V., Zappala, D., Zhou, D., Liu, Y., Zhao, S. Cluster Computing on the Fly: P2p Scheduling of Idle Cycles in the Internet. In Peer-to-Peer Systems III, Springer, 2005, 227–236.
34. Lu, Y., Xu, X., Xu, J. Development of a Hybrid Manufacturing Cloud. Journal of Manufacturing Systems, 2014, 33(4), 551–566.
35. Marosi, A. C., Balaton, Z., Kacsuk, P. GenWrapper: A Generic Wrapper for Running Legacy Applications on Desktop Grids. In 3rd Workshop on Desktop Grids and Volunteer Computing Systems, (PCGrid 2009), 2009, 1–6.
36. Marosi, A. C., Kacsuk, P., Fedak, G., Lodygensky, O. Sandboxing for Desktop Grids Using Virtualization. Parallel, Distributed and Network-Based Processing, (PDP), 18th Euromicro International Conference, 2010, 559-566.
37. Marosi, A., Kovács, J., Kacsuk, P. Towards a Volunteer Cloud System. Future Generation Computer Systems, 2013, 29(6), 1442-1451.
38. McGilvary, G., Barker, A. D., Lloyd, A., Atkinson, M. V-BOINC: The Virtualization of BOINC. In 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, (CCGrid), IEEE Computer Society, 2013, 285-293.
39. McGilvary, G., Barker, A., Atkinson, M. Ad Hoc Cloud Computing: From Concept to Realization. IEEE 8th International Conference, (CLOUD 2015).
40. Mishra N., Siddiqui, S., Tripathi, J. A Compendium Over Cloud Computing Cryptographic Algorithms and Security Issues. International Journal of Information Technology, 2015, 7(1), 810-814.
41. Montes, D., Añel, J. A., Pena, T. F., Uhe, P., Wallom, D. C. H. Enabling BOINC in Infrastructure as a Service Cloud System. Geosci. Model Dev., 2017, 10, 811-826.
42. Morsky, M., Grundy, J., Müller, I. An Analysis of the Cloud Computing Security Problem. 17th Asia-Pacific Software Engineering Conference, (APSEC 2010), Sydney, Australia.
43. Poll Results: Largest Dataset Analyzed/Data Mined. https://www.kdnuggets.com/2013/04/poll-results-largest-dataset-analyzed-data-mined.html. Accessed on November 7, 2017.
44. Running Apps in VirtualBox Virtual Machines. http://boinc.berkeley.edu/trac/wiki/VboxApps. Accessed on November 7, 2017.
45. Sahoo, J., Mohapatra, S., Lath, R. Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues. Second International Conference on Computer and Network Technology, (ICCNT), 2010, 222-226.
46. Sarmenta, L. F., Hirano, S. Bayanihan: Building and Studying Web-Based Volunteer Computing Systems Using Java. Future Generation Computer Systems, 1999, 15(5-6), 675-686.
47. Schlitter, N., Laessig, J. Distributed Data Analytics Using RapidMiner and BOINC RapidMiner. RapidMiner Community Meeting and Conference, (RCOMM 2013), Porto, Portugal, 2013, 81-95.
48. Segal, B., Buncic, P. LHC Cloud Computing with CernVM. PoS, (004), 2010, 28.
49. Silberstein, M., Sharov, A., Geiger, D., Schuster, A. Gridbot: Execution of Bags of Tasks in Multiple Grids. In Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, (SC '09), New York, NY, USA, 2009, 11:1-11:12.
50. Smith, J., Nair, R. Virtual Machines. Morgan Kaufmann, Elsevier, 2005.
51. Sun, Y., Zhang, J., Xiong, Y., Zhu, G. Data Security and Privacy in Cloud Computing. International Journal of Distributed Sensor Networks, 2014, 10(7), 1-9.
52. Tehrani, S., Shirazi, F. Factors Influencing the Adoption of Cloud Computing by Small and Medium-Sized Enterprises (SMEs). International Conference on Human Interface and the Management of Information, 2014, 8522, 631-642.
53. The BOINC Wrapper. http://boinc.berkeley.edu/trac/wiki/WrapperApp. Accessed on November 7, 2017.
54. Weicker, R. P. Dhrystone: A Synthetic Systems Programming Benchmark. Communications of the ACM 27 (10), October 1984, 1013-1030.
55. Witten, I. H., Eibe, F. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, 2000.