Таким чином студенти можуть під'єднуватись безпосередньо до сервера з віртуальною машиною ОС Kali Linux через встановлений на ПК VMWare Player. Після цього можна використовувати максимально всі можливості дистрибутива Kali з допомогою значно більших апаратних ресурсів самого сервера ніж звичайного ПК.

Висновки

Як показує практика, більшість виявлених вразливостей пов'язана з несвоєчасним оновленням ПЗ і засобів захисту, використанням попередньо встановлених параметрів налаштування ПЗ та мережевого обладнання, недотриманням політики безпеки, помилками в розробці ПЗ доступних з Інтернету сервісів і т.д.

Наше дослідження дозволить в майбутньому застосувати вищезгадане ВВПЗ в навчальних лабораторіях університету для навчання майбутніх спеціалістів в сфері безпеки. Перевага цьому ВВПЗ надається не з огляду на безкоштовність, а через високу зручність і ефективність.

Література

[1] Піскозуб А.З. Використання тестування на проникнення в комп'ютерні мережі та системи для підняття їх рівня захищеності // Матеріали третьої міжнародної науково-практичної конференції FOSS Lviv 2013,. – Львів,2013.
[2] W.Pritchett,D.Smet. Kali Linux Cookbook - Birmingham-Mumbai, Puckt Publishing, 2013
[3] Kali Linux. https://kali.org
[4] Metasploitable2. https://community.rapid7.com/docs/DOC-1875.

Berkeley Open Infrastructure for Network Computing (BOINC) - distributed computing system based on volunteers

Monika Kwiatkowska and Lukasz Swierczewski

I Maria Curie-Skłodowska University in Lublin, II College of Computer Science and Business Administration in Łomża, lswiercz@icm.edu.pl

This work describes BOINC, an open-source distributed computing system. Author explores the specific nature of a computing project which heavily relies on volunteers.

Introduction

Berkeley Open Infrastructure for Network Computing is a system that allows distributed computing [1] [2] with the use of computers connected only global network the Internet. The system is developed on the basis of a completely non-commercial and was originally initiated for the project SETI@Home [3] [4]. Currently, the solution is also adapted by projects in other areas of science such as Word Community Grid (biochemistry) [5] [6] Einstein@Home (astrophysics, the search for pulsars) [7] or OProject (including mathematics) [8]. BOINC is still being developed by a team working at the University of California at Berkeley under the direction of Dr. David Anderson. BOINC is licensed under the GNU LGPL and financially supported by the U.S. government agency, the National Science Foundation (under the grants SCI/0221529 [9] , SCI/0438443 [10] and SCI/0721124 [11]).

BOINC server technology

BOINC server can run on one or more computers which allows good scaling system performance even for very large projects. BOINC server runs on computers based on the Linux operating system. In addition, it is used in Apache, PHP and MySQL.

All major scientific calculations are carried out on computers of volunteers. BOINC server sends only one task to be performed. After completion of the calculation results with the report are sent to the server and placed in a database (or possibly a separate file). The server then verifies the returned results. In case of doubt as to the correctness of the results BOINC server may have to resend the task to another computer for verification.

BOINC server provides, inter alia, certain features:

- homogeneous redundancy - sending data computing tasks only to computers with a specific platform (e.g. Linux 32bit),
- workunit trickling sending information to the server before complete the task,
- division of labor based on the parameters of the host (task that requires 1 GB of RAM will not be assigned to the computer with only 512 MB of RAM).

BOINC server consists of two CGI programs, and usually five daemons written in C++.

The specificity of BOINC

We must see that the BOINC platform, which is a platform for distributed computing does not run efficiently all the parallel algorithms. Distributed algorithms are only a subset of parallel algorithms. The task implemented in a distributed system may not have information about the whole issue under consideration (in the particular case the data is not physically fit into one Internet user on the computer).

Distributed system using computers for calculations volunteers must have certain attributes (based on [12]):

- the ability to work on different hardware and software platforms,
- taking into account the uncertainty of results returned and their corresponding verification because they may be wrong,
- consider the possibility of a total loss of connectivity to computers running and not return partial results carried out by the machine,
- the inclusion of additional preferences of users working on computers (e.g. low CPU utilization)

- difficult to determine the current state of the system (computers interact with various links in various topologies , the current state of the system components may be unknown)

BOINC credit scoring system

For BOINC users can analyze how much they contribute to the calculation in the project. To enter into the competition for points between the volunteers introduced the so-called credits. The scoring system of credit is designed in such a way as to eliminate fraud by verifying the validity of the results before assigning a user of credit.

It should be noted why people keep calculations within BOINC:

- Due to the clean development of science.
  - To support the development of the field of science related to project.
  - For example, to help fight the disease (for example, project World Community Grid).
- Due to the ability to test the computer under heavy load.
  - BOINC full advantage of the computing capabilities of the computer (such as graphics accelerators) because overclockers often use it to test the stability of your system.
- Due to cooperate, compete and receive credits.
  - Users or even entire teams dedicated machine to run it to appear in the top positions in the rankings.
- For personal gain and recognition.
  - BURP and Renderfarm.fi have the potential to make their own calculations on a distributed platform. They can outsource the execution of the tasks which will focus at the moment computers in the system.
  - PlanetQuest to assign the discovered planets which user information using distributed computing of the discoveries made.

References

1. Peleg, David. *Distributed computing*. Philadelphia, PA: SIAM, 2000.
2. Thain, Douglas, Todd Tannenbaum, and Miron Livny. "Distributed computing in practice: The Condor experience." *Concurrency and Computation: Practice and Experience* 17.2-4 (2005): 323-356.
3. Anderson, David P., et al. "SETI@ home: an experiment in public-resource computing." *Communications of the ACM* 45.11 (2002): 56-61.
4. Korpela, Eric, et al. "SETI@ HOME—massively distributed computing for SETI." *Computing in science & engineering* 3.1 (2001): 78-83.
5. Hachmann, Johannes, et al. "The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid." *The Journal of Physical Chemistry Letters* 2.17 (2011): 2241-2251.
6. Cumbaa, Christian A., and Igor Jurisica. "Protein crystallization analysis on the World Community Grid." *Journal of structural and functional genomics* 11.1 (2010): 61-69.

7. Abbott, B. P., et al. "Einstein@ Home search for periodic gravitational waves in early S5 LIGO data." *Physical Review D* 80.4 (2009): 042003.
8. Swierczewski, Lukasz. "OProject@Home - distributed computing". Proceedings, LVEE 2013 Conference
9. Research and Infrastructure Development for Public-Resource Scientific Computing, URL: http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0221529, The National Science Foundation
10. SCI: NMI Development for Public-Resource Computing and Storage, URL: http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0438443, The National Science Foundation
11. SDCI NMI Improvement: Middleware for Volunteer Computing, URL: http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0721124, The National Science Foundation
12. Świerczewski, Łukasz. "Symulacja funkcjonalnego systemu kwantowego na równoległych komputerach klasycznych IV generacji." (2013).

Blue Gene - a parallel computer architecture

Monika Kwiatkowska and Lukasz Swierczewski

[I] *Maria Curie-Skłodowska University in Lublin,* [II] *College of Computer Science and Business Administration in Łomża, lswiercz@icm.edu.pl*

Document briefly describes the project and the Blue Gene supercomputers, Blue Gene/L , Blue Gene/P and Blue Gene/Q. In addition, performance measurement is made simple one node on the computers (Blue Gene/P and Blue Gene/Q) and other structures composed of Intel Xeon X5660 processors.

Introduction

Blue Gene is a project initiated and maintained by IBM. It aims to design a supercomputer capable of achieving speeds of PFLOPS and with low power consumption.

At present (2014) the project has created three generations of supercomputers, Blue Gene/L [1] , Blue Gene/P [2] and the Blue Gene/Q [3]. Class Blue Gene systems often occupy a leading place on the TOP500 list [4] (the fastest supercomputers) and Green500 [5] (the most energy-efficient design). Project in 2009, he received the National Medal of Technology and Innovation (National Medal of Technology and Innovation) [6].

History

In 1999, IBM announced a research project with a budget of $100,000,000 with a view to build a massively parallel computer, which was to be used to study biomolecular phenomena. The project had two objectives:

- broaden the knowledge of the mechanisms of protein folding simulations used in the context of large-scale,
- explore innovative ideas on massively parallel supercomputers and software architectures that could be implemented on them.

Originally Blue Gene project was based on an early version of the architecture Cyclops64 [7], which was designed by Monty Denneau. The initial research and development was carried out by the IBM TJ Watson