Volunteer Computing With Video Game Consoles

DAVID TOTH
Computer Science Department
Worcester Polytechnic Institute
100 Institute Rd. Worcester, MA 01609
U.S.A.
http://www.cs.wpi.edu/~toth

Abstract: - Volunteer computing is a form of distributed computing where projects attempt to accomplish some goal, using volunteered computational resources instead of paying for the resources [1]. Volunteer computing projects are being used for a wide range of computationally intensive scientific and mathematical goals, ranging from searching for evidence of extraterrestrial intelligence to searching for cures to cancer and other diseases, to finding Mersenne prime numbers [1, 2, 3]. Due to the computational demands of volunteer computing projects, it is desirable to find additional sources of volunteer computing power. In order to be a viable source of volunteer computing power, a platform must be able to provide enough CPU cycles to make it worth the effort to port volunteer computing applications to that platform. Video game consoles have become increasingly powerful computers over the last 30 years, and the number of video game consoles sold and their computational power combined with their network capability makes them a potentially good platform for volunteer computing. We devise an experiment to test the potential usefulness of video game consoles for volunteer computing and compare the time it takes a video game console and several different computers to do the same amount of work for an example project.

Key-Words: - Volunteer Computing, Video Game Consoles, Distributed Systems

1 Introduction

Volunteer computing projects allow people to donate the CPU cycles of computers when these computers would be idle, in order to accomplish some goal [1]. Volunteer computing projects enable people to solve problems that are otherwise too computationally intensive to solve. Dividing a large problem into many tasks that can be completed independently allows millions of computers to work on the large problem simultaneously, making volunteer computing a powerful method for solving problems [4].

A volunteer computing project uses two sets of computers: the servers provided by the project's sponsor and the volunteered computers. The owner of a volunteered computer enables the computer to participate in the project by installing a special client program on the computer. The group of servers creates the independent tasks, distributes the tasks to clients, records the results that clients return, and uses the results to determine the answer to the larger problem. When a volunteered computer would be idle, it works on a task that a server sent to the client. When a computer completes a task, the client sends the results to the server and the server sends the client another task. When a person begins to use a volunteered computer, the computer temporarily stops working on the task for the volunteer computing project and executes programs for the user, until the computer becomes idle again. The volunteered computer resumes working on the task for the volunteer computing project when the computer once again becomes idle [4].

Large volunteer computing projects began in the 1990's with The Great Internet Mersenne Prime Search (GIMPS) and Distributed.net [1]. There are a number of volunteer computing projects currently running and several that have been completed.

Volunteer computing has enabled researchers to solve problems that were previously computationally infeasible by decomposing them into smaller problems, distributing the smaller problems to computers, and aggregating the results returned by the computers to form the solution to the large problem. The applications of volunteer computing include solving problems in the medical, scientific, and mathematical fields.

SETI@home searches for extraterrestrial life by processing the signals collected by radio telescopes [5]. The first project Grid.org worked on was a cancer research project that Intel sponsored [2]. In the project, "grid.org was able to screen billions of target molecules against known cancer target proteins" [2]. Currently, Grid.org is working on analyzing human proteins [2]. Many other projects are in progress now.

Volunteer computing projects require a great deal of computational power to make significant progress towards their goals. Some projects end at the completion of a task that is known to be solvable in a finite amount of time. Other projects are designed to continue indefinitely, continually searching for more or better information. Despite the need for large amounts of donated computing power, statistics show that very few people participate in volunteer computing projects [4]. While it is estimated that 300 million computers are connected to the Internet, less than 1% of those computers participate in volunteer computing projects [6]. Due to the low participation rate and the projects' high demand for computing power, it is crucial to increase the amount of computational power that can be volunteered and use the volunteered resources more effectively. Toth and Finkel looked at increasing the effectiveness of the CPU cycles that are donated in [4]. In this work, we address the idea of increasing the amount of computing power available to projects from a technological perspective.

2 Increasing Available Computational Power

We attempt to find an additional source of volunteer computing power in order to try to increase the amount of computational power available to volunteer computing projects. Currently, volunteer computing clients run on general purpose PCs. In order to be a viable source of volunteer computing power, a platform must be able to provide enough computational power to make it worth the effort to port volunteer computing applications to that platform. General-purpose computers are the most powerful devices used for volunteer computing and it takes many of them to make progress on a project. Therefore, it is clear that a platform that is not as powerful as a general-purpose computer must have even more units in circulation to produce enough computational power to make progress on volunteer computing projects. Some devices that have the computational ability to perform the calculations used in a volunteer computing project are cell phones and PDAs. However, because performing the calculations for a volunteer computing project is CPU intensive, running a volunteer computing client on a mobile device such as a cell phone will drain the device's battery very quickly. Since people will not want to drain the batteries of their cell phones and PDAs frequently, it does not make sense to port a volunteer computing client to mobile devices. However, video game consoles do not run on batteries and millions are sold. By the end of 2005, Sony had sold 200 million PlayStation2 video game console systems [7]. Video game consoles have become increasingly powerful computers over the last 30 years. The Atari 2600, released in 1977, had a 1.19 MHz CPU and 128 bytes of RAM [8]. Sony's PlayStation2 contains a CPU running at 299 MHz, 32 MB of RAM, an optional Ethernet connector and custom graphics hardware [9, 10]. Microsoft's Xbox has a 733 MHz Intel processor, 64 MB of RAM, an Ethernet connection, and specialized graphics hardware [11]. Microsoft's Xbox360, the newest video game console system, was released in the Fall of 2005. This console contains a triple core CPU with each core running at 3.2 GHz, 512 MB of RAM, an Ethernet connection, and specialized graphics hardware [12]. The number of video game consoles sold and their computational capability combined with their network capability makes them a potentially good platform for volunteer computing.

3 Experiments

We wanted to understand how viable a platform for volunteer computing video game consoles are, so we devised an experiment to test console systems and compare the results to the results from general-purpose computers.

3.1 Hardware

To test the viability of video game consoles for volunteer computing, we conducted experiments using a server, various computers running the a client, and a 10 Mbps Ethernet hub. The server had an Intel Pentium III 450 MHz processor and 256 MB of RAM. It ran Windows 2000 with Service Pack 3 and the Apache web server version 1.3.34. We used a variety of different computers to run a volunteer computing client. We used computers with the following CPU and RAM configurations to run a volunteer computing client:

- Pentium II 233 MHz processor and 320 MB of RAM (general-purpose computer)
- Celeron 400 MHz processor and 128 MB of RAM (general-purpose computer)
- Pentium III 733 MHz processor and 128 MB of RAM (general-purpose computer)
- Pentium IV processor and 512 MB of RAM (general-purpose computer)

- 299 MHz processor and 32 MB of RAM (PlayStation 2 video game console).

Because only the PlayStation 2 had a programming environment available to the public at the time of the tests, we were only able to test that video game console.

3.2 Software

We developed a volunteer computing client to enable it to run on the general purpose computers and the PlayStation 2. The client performs fast Fourier transforms, like the SETI@home client does [13, 14]. We had developed the client like this because unlike most volunteer computing projects, the source code for SETI@home is available, allowing us to see what a volunteer computing client does. To save time, we used freely available fast Fourier transform code for our client [15, 16]. Like other volunteer computing clients, our client downloads a task from a server to work on, performs the computations, occasionally checkpointing to disk to avoid losing all the progress in the event of a system failure, and reports the result of the task back to the server.

3.3 Experiments

We ran nine tests on each computer and game console, one test corresponding to each parameter combination. The parameters were file size (16 KB, 512 KB, and 1 MB) and computation intensity (performing the task 1, 2, or 4 times). For each test, the clients requested a task from the server, downloaded a file to be used in the task from the server, saved the file to the local disk, ran a computation on the contents of the file, and returned the result of the computation to the server 25 times. This was repeated 10 times. In between the tests from the different computers, the web server was rebooted to ensure that the cache was in the same state for all the tests and the computer the client ran on was also rebooted.

3.3.1 Head to Head Comparison

The results of our experiment are shown in Figure 1.

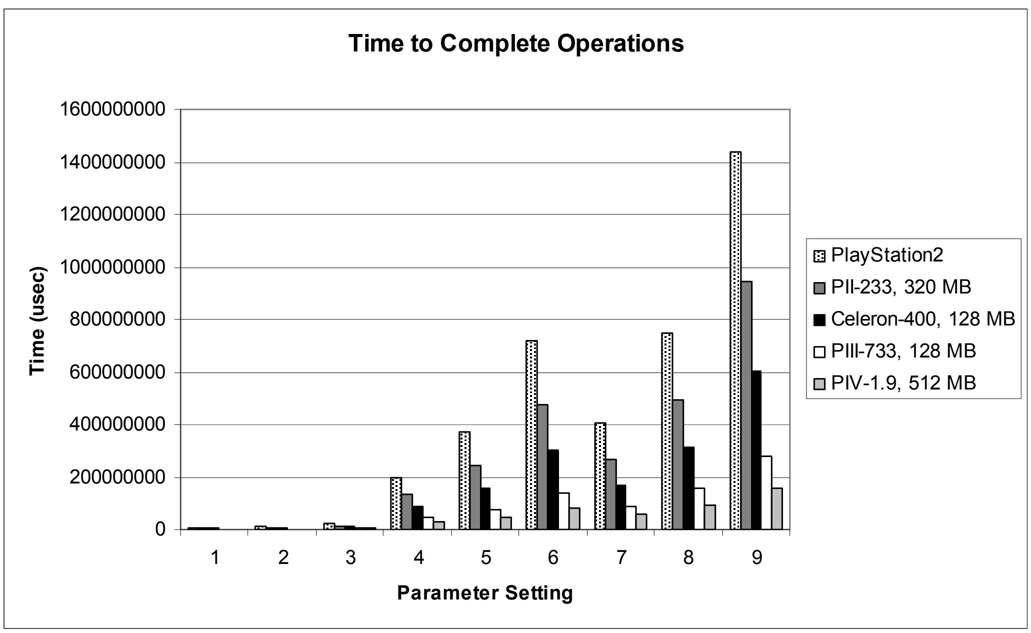

Fig. 1

The PlayStation2 was significantly outperformed by all the computers we tested. We were not surprised by most of the computers outperforming the PlayStation2 significantly, except for the Pentium II 233 MHz computer. This computer was the closest comparison to the PlayStation2 that we had, and just based on CPU speed, we expected the PlayStation would perform roughly the same as the computer. The computer had significantly more RAM (320 MB compared to the Playstation2's 32 MB), so we hypothesized that the computer was keeping things in RAM that the PlayStation2 had to swap out to the disk. Therefore, we tested the computer again with only 32 MB of RAM in it. The results, shown in Figure 2, were almost identical to the ones we obtained when the computer had 320 MB of RAM. Thus, it is unlikely that the small amount of RAM in the PlayStation2 is causing its slower than expected performance relative to the Pentium II 233 MHz computer.

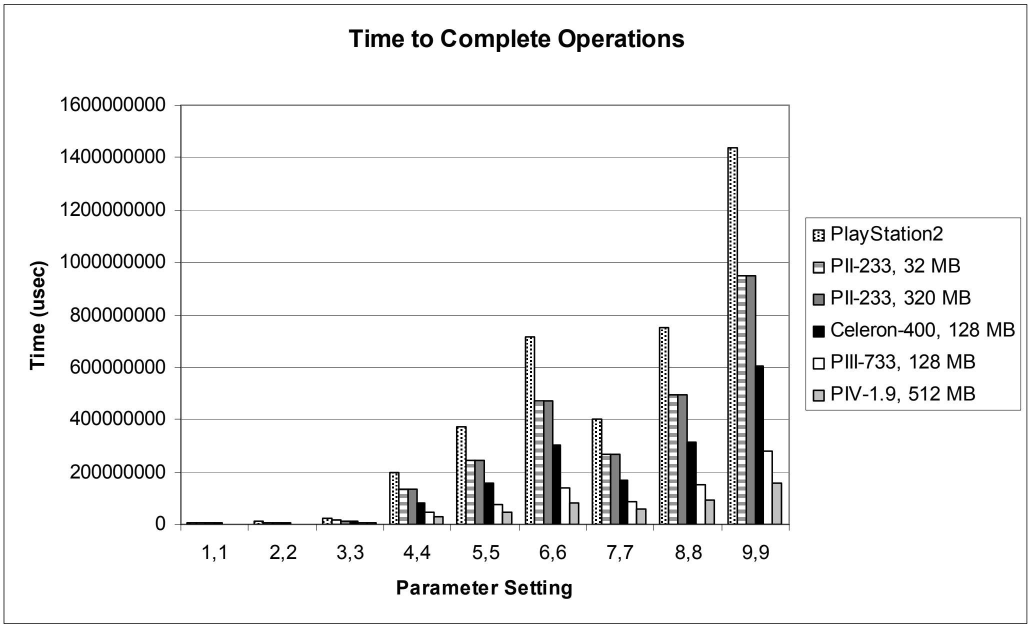

Fig. 2

3.3.2 Processor Only Comparison

The hardware for the PlayStation2 is specialized and the hard drive and network interface adapter are connected in a very different manner than those of normal computers. The hard drive is plugged into an expansion port and the network interface is plugged into the back of the hard drive. Thus any network traffic must pass through the physical casing of the hard drive. We believed that the differences between the hardware connections of the PlayStation2 and personal computers might account for some of the performance differences we observed. In order to test this, we modified our test program to eliminate both the network and disk I/O. We placed all the data that normally needed to be retrieved from the server, saved to disk, and then read in from the disk in memory within the client program and recompiled it. When we ran the modified client on the PlayStation2 and the computer with the Pentium-II 233 MHz processor and 32 MB of RAM, the computer with the Pentium II-233 MHz processor outperformed the PlayStation2 for this program by a significantly greater margin. Therefore, we believe that the PlayStation2's unique hardware connections are not likely a significant cause of any performance differences. We note that the PlayStation2 uses the ext2 file system, as opposed to the other computers we tested which use the ext3 files system. The ext3 file system is a journaling file system, while the ext2 system is not. Thus, the ext2 system may have given the PlayStation2 a little performance edge over the general-purpose computers. Therefore, by removing the file system usage, the program may have performed worse on the PlayStation2 than before compared to how it performed on the general-purpose computers. It is also possible that the cache in the general-purpose computer gave it a very large advantage over the PlayStation2. To test this, we would need to devise some micro benchmarks and run them.

4 Conclusion

In this work, we attempted to find a platform besides general-purpose computers that could provide enough computational power to make it worth porting volunteer computing clients to the platform. We rejected cell phones and PDAs because they are powered by batteries, but decided that video game consoles had the potential to run volunteer computing clients. We tested a Sony PlayStation2 video game console to see if it would be a viable platform. It is important to note that although the PlayStation2 was significantly outperformed by the general-purpose computers, even the one with a CPU with a slower clock speed, it was still able to perform the required operations for volunteer computing, including network and file I/O. Because of this, although it is clear that the PlayStation2 is not the best resource for volunteer computing (clearly general-purpose computers are better), we believe that video game consoles have potential to be used for volunteer computing. We point out that the Microsoft Xbox is a much faster platform and the new Xbox360 is even faster still, containing 3 processor cores. Thus we believe that the Xbox and Xbox360 have significant potential to be used for volunteer computing. The newest Sony PlayStation, the PlayStation3, is due out in the fall of 2006, and we believe that it may significantly more potential, with 5 years of hardware maturity. We do note, however, that video game console systems are not often left on when they are not being used, which could hinder their usefulness in volunteer computing by limiting the number of CPU cycles that the systems might contribute. Future work should examine whether video game consoles are idle enough to contribute enough CPU cycles to make them worthwhile contributors to volunteer computing. Also, future work should examine how much work the Xbox, Xbox360, and PlayStation3 will be able to do in comparison to a variety of general-purpose computers. Finally, both Microsoft and Sony need to be brought on board with this idea, as it would take their support to legally integrate volunteer computing client software within these video game console systems.

5 Acknowledgements

We would like to thank Professors Robert Swarz and Robert Lindeman for their help acquiring a PlayStation 2 Linux kit. We also would like to thank Professors David Finkel and Craig Wills for their comments.

References:

[1] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, SETI@home: An Experiment in Public-Resource Computing. *Communications of the ACM*, Vol. 45, No. 11, 2002, pp. 56-61.

[2] Grid.org, "GRID.ORG – Project Overview", http://www.grid.org/projects/, Accessed 4/1/05.

[3] GIMPS. "Mersenne Prime Search". http://www.mersenne.org/prime.htm Updated 3/6/05. Accessed 7/13/05.

[4] D. Toth and D. Finkel. A Comparison of Techniques for Distributing File-Based Tasks for Public-Resource Computing, *Proceedings of the 17th IASTED International Conference on Parallel and Distributed Computing and Systems* – PDCS 2005, November 14-16, 2005, pp. 398-403, Phoenix, Arizona, USA.

[5] Seti@home, "About SETI@home," http://setiathome.berkeley.edu/sah_about.php, Accessed 7/13/05.

[6] J. Bohannon, Grassroots Supercomputing. *Science 308*, 2005, 810-813.

[7] Sony Computer Entertainment, Inc., "Sony Computer Entertainment, Inc.," http://www.scei.co.jp/corporate/data/bizdataps2_e.html, Accessed 3/24/06.

[8] "Atari 2600 – Wikipedia, the free encyclopedia" http://en.wikipedia.org/wiki/Atari_2600 Updated 3/25/06. Accessed 3/25/06.

[9] Sarah Ewen and Lionel Lemarie, Sony Computer Entertainment Europe, http://www.technology.scee.net/files/presentations/agdc2002/ConsoleYourself.pdf. 2002. Accessed 3/01/06.

[10] "PlayStation 2 – Wikipedia, the free encyclopedia," http://en.wikipedia.org/wiki/PS2, Updated 3/25/06, Accessed 3/25/06.

[11] "Xbox – Wikipedia, the free encyclopedia" http://en.wikipedia.org/wiki/Xbox. Updated 3/25/06, Accessed 3/25/06.

[12] Microsoft. "Xbox.com | XBOX 360 – The System," http://www.xbox.com/en-US/hardware/xbox360/default.htm, Accessed 3/25/06.

[13] "Porting and optimizing SETI@home," http://setiweb.ssl.berkeley.edu/sah_porting.php, Accessed 10/1/05.

[14] http://setiweb.ssl.berkeley.edu/sah/seti_source/nightly/seti_boinc-client-cvs-2005-10-01.zip, Accessed 10/1/05.

[15] "Fast Fourier Transform C Code," http://www.csua.berkeley.edu/~emin/source_code/fft/, Accessed 9/30/05.

[16] http://www.csua.berkeley.edu/~emin/source_code/fft/, Accessed 9/30/05.