Characterizing Resource Availability for Volunteer Computing and its Impact on Task Distribution Methods

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

Abstract: - Volunteer computing uses computational resources that would otherwise be unused, to solve computationally intensive projects [1]. We have collected data from several different types of computers about the durations of periods when the computers were able and unable to participate in volunteer computing projects. We found that those periods differed significantly, indicating that a single method of task distribution for a volunteer computing project may not be adequate to make the best use of the donated CPU cycles. The data we have collected will also be useful in future work, to analyze portions of volunteer computing clients in an attempt to reduce the wasted resources and increase the productivity of volunteer computing and to compare the amount of work that can be completed by using different types of volunteer computing clients.

Key-Words: Volunteer Computing, Resource Availability, Measurement, Characterization

1 Introduction

Volunteer computing uses computational resources that would otherwise be unused, to solve computationally intensive projects [1]. Over the last 10 years, volunteer computing has become a viable paradigm for solving extremely computationally intensive problems that were previously considered to be infeasible and for making progress on problems that have no specified finite duration such as SETI@home and the Great Internet Mersenne Prime Search (GIMPS) [2]. During this time, the focus of volunteer computing has been on getting projects running and proving that this paradigm is useful. For example, the Berkeley Open Infrastructure for Network Computing (BOINC) was developed to simplify the creation of volunteer computing projects, with the intent that scientists "with moderate computer skills" would be able to construct projects [3]. However, statistics have shown that relatively few people participate in volunteer computing. Of an estimated 300 million internet connected personal computers, fewer than 1% participate in volunteer computing [4, 2]. The low participation rate, the increasing number of projects over the last several years, and the indisputable value of many of the projects, such as Grid.org's cancer research project, provide an incentive to improve performance of volunteer computing systems.

Volunteer computing is still immature and some aspects of it still need to be studied to determine if there are more effective ways of utilizing volunteered resources. A past study showed that in some cases, different methods of distributing tasks can affect how many of the volunteered CPU cycles are wasted [2]. However, the aforementioned work used best guess data values for the durations of the periods that computers were able and unable to participate in volunteer computing instead of actual values, which were unavailable. In this work, we collect actual data about the availability of computers and analyze the data so that it can be used to improve the accuracy of the simulations performed in [2]. Volunteer computing projects typically use a single method of distributing tasks to computers. Although some of the methods can be configured when the volunteer computing client is installed, the configurable methods do not adapt unless the user manually changes them. The variance of the data suggests that a single method of task distribution for a volunteer computing project may not make the best use of the donated CPU cycles. In particular, dynamic task distribution methods that adapt to the usage of computers may increase the effectiveness of the donated CPU cycles. This data will also be used in future work to compare the amount of work that can be completed by using different types of volunteer computing clients.

2 Required Data

We determined that the data we needed had several requirements listed below.

Requirement 1: The data needed to accurately reflect the computers that might be available for volunteer computing.

Four major classes of computers that might be used for volunteer computing are:

1. Home computers: These computers are used for personal and family non-business related activities.
2. Business computers: These computers are used for business related activities.
3. Public computers: These computers are available for the general public in some community to use, such as computers in a public library or computers in a lab on a college campus that is open to the entire student body.
4. Undergraduate student computers: These computers are owned by students and used at their universities or colleges.

We reasoned that the usage patterns of those categories might be significantly different and thus we needed to collect data from computers in each category to see the entire picture.

Requirement 2: The number of computers we collected data from in each category listed in Requirement 1 needed to be significant enough to make some reasonable observations about the data. We used at least 25 computers from each class.

Requirement 3: The data we needed had to reveal when computers were available to participate in volunteer computing, unavailable to participate but powered on, and when the computers were unavailable to participate.

The periods when the computers were available and unavailable will be used for simulations involving the model of volunteer computing clients that run when some conditions are met, such as screensavers. All three types of periods will be used for simulations involving the model of volunteer computing clients that run all the time. In order to collect this data, we needed to define when a computer was available for volunteer computing. We note here that unavailable is simply the complement of available.

Because most volunteer computing programs consider a computer to be available if the computer's screensaver is running, we also used this criteria to determine when a computer is available. This method contains a shortcoming, however. If a computer is being accessed remotely, the screensaver may still be running locally, even if the remote connection is sending keystrokes or mouse movements [5]. Although the BOINC framework appears to test for remote connections using the Terminal Services, it appears that other volunteer computing projects do not [6]. We note that we expect very few if any of the computers in our study to be accessed remotely, so we do not check for this. Our method also considers a computer idle if the screensaver is running even if it is actively running some intensive task such as a long compilation. Because long compilations that take more than just a couple minutes to complete are usually limited to large software projects, we do not believe that this will occur on the systems in our study.

Requirement 4: The data needed to represent the available, unavailable but powered on, and unavailable periods in a fine enough granularity to make our simulations accurate.

Because we decided to use the screensaver to define when computers were available, the data needed to have a fine enough granularity to reflect the state of the screensaver on computers accurately. Screensavers can be set to come on after a period of idle time that is a multiple of one minute. Sampling the state of the screensaver every 60 seconds would allow the data to be off by as much as 59 seconds, which we deemed was too inaccurate.

If we would need to collect the data instead of just using data from others, we wanted to ensure that a program we would design to collect the data would not impose enough of performance cost on the computer such that the user would notice the program was running. We conducted an experiment to help us decide on a sampling rate, running a prototype data collection program on an old computer with an Intel Celeron 900 MHZ CPU and 128 MB of RAM that was running Windows XP. We expected that if we had to collect the data ourselves, then any computer used in our study would have equivalent or better hardware. Thus, if our program did not have a noticeable performance impact on this computer, then we believed it would not have a noticeable impact on any computer that would participate in our study. The prototype was set to sleep for 60 seconds between every time it queried the operating system to see if the screensaver was running. By monitoring the CPU usage of the program with Task Manager for several minutes, we saw that the program was using 0% of the CPU. We decreased the sleep time to 30 seconds, and upon observing the program was continuing to use 0% of the CPU, we decreased it again to 10 seconds. With this setting, the program continued to consume 0% of the CPU. This setting allowed us to determine when the screensaver starts and stops within 10 seconds of the events actually occurring. This means our measurements would be accurate to the nearest minute, which we felt would be precise enough for our simulations.

Requirement 5: The sampling of the data needed to have been continuous for enough time to get an accurate representation of computer usage patterns and try to avoid the effects of anomalous data.

Although a year's worth of continuous sampling would be nice, we did not expect to be able to find that kind of data in a study and decided that at least two weeks of data would be necessary.

Requirement 6: The data we used for the simulations needed to have been recorded in a consistent manner.

We would only be able to use data from two or more different studies that together had collected data the four categories of computers listed in Requirement 1 if both studies had recorded the same data using the same method and the same sampling intervals.

Requirement 7: The data needed to have been collected relatively recently.

Computer usage has changed significantly during the last 10 years, as computers have gone from being something only a small segment of the population could afford to being a commodity that a huge segment of the population can afford.

We contacted existing volunteer computing projects to see if they had collected this data, but both GIMPS and the BOINC-based projects do not collect this information [7, 8]. None of the other projects we contacted responded. Thus, we decided to review other studies relating to computer usage statistics in an attempt to get the necessary information. Although there were quite a few studies that collected data about the availability of computers, we were unable to find any that had data that was close enough to meeting our requirements. However, for completeness, we discuss the most relevant studies we examined.

3 Related Studies

There have been quite a few studies about the availability of computers. Wolski et al. gathered data from a Condor pool at the University of Wisconsin and several labs at UCSB that are accessible to computer science students. In addition to that data, they also used the data from the 1995 by Long, Muir, and Golding study [9]. They analyzed the data in an attempt to predict the availability of desktop computers. Although Wolski et al collected data from several different sources which is very important to our work, the way they measured availability was not consistent across the different sources and was not consistent with the method of defining availability in [2]. Thus, we were unable to use their data for our work.

Mutka and Livny collected data from three different types of users [10]. They monitored computers used by graduate students, faculty, and systems programmers [10]. Mutka and Livny considered workstations to be unavailable when they are used or when the average user CPU usage was greater than one quarter of one percent within five minutes of being used by the owner [10]. However, the data is almost 20 years old and only 11 computers were monitored for their study [10]. Because of this, we did not feel that the data was representative of the data we needed.

Acharya et al examined traces of three different sets of workstations [11]. For the trace from the University of Maryland Computer Science department's cluster of public computers, a computer was considered available if the CPU utilization stayed below 0.3 for 5 minutes [11]. In the trace from a Condor pool of roughly 300 workstations at the University of Wisconsin, a computer was considered to be available when the Condor software deemed it so [11]. The remaining trace came from a group of computers at UC Berkeley [11]. Again, the inconsistency of determining when computers are available and the data only coming from one type of computer rendered this data unusable for our work.

Kondo et al published results of a study that provided a data set that most closely resembles the data that we needed for our simulations [12]. This study measured the availability of over 200 computers in the San Diego Supercomputer Center (SDSC) [12]. Kondo et al recorded whether each computer was powered on and reachable over the network and the percentage of the CPU time that was available for a distributed application at 10 second intervals [12]. Their recorded data did distinguish between a host being available, unavailable and powered on, and unavailable and powered on in a way that we might have been able to use [12]. In his dissertation, Kondo used a data set gathered in the same manner from a set of student lab machines [13]. However, because their measurements came from only 2 types of computers (what we deem business computers and public computers, as opposed to a student, and home computers), we still needed to collect data from the other two types of computers [12, 13]. In order to keep the data we collected completely consistent with one measuring scheme, we chose not to use the business computer data they had collected.

4 Methodology

Because the data from the studies in the Section 3 could not be used to refine the work in [2] as explained, we developed a way to collect the data we needed. The lab computers at our University and the computers at the company that participated in our study run Windows 2000 and Windows XP. Also, so many home and student computers run Windows that to collect data from enough computers in a consistent manner, it made the most sense to write our data collection program to run on the Windows operating systems. We had to choose to write the data collection program as a normal application, a screensaver, or a windows service. We chose to write the program as a Windows service for several reasons. We wanted the program to be as invisible to the user as possible so it would not influence the users' behavior. Running the program as a service means it does not show up as the user's screensaver and also it does not show up in the *Applications* tab of the Windows Task Manager. Running the program as a service also minimizes the chance that a user disables the program intentionally or by mistake. Finally, in order to ensure that the data collection would run even when nobody was logged into computers and collect data about when the computer was powered on but the screensaver was not on, it needed to run as a service.

The service recorded the data it collected to files, starting a new file every 24 hours to minimize the amount of data that would be lost if a file was destroyed by accident. Every 10 seconds, the service would determine whether the computer's screensaver was running. If the screensaver was running and had not been running in the previous interval, the service recorded the time. If the screensaver was running for the second or more consecutive interval, the service recorded a *. If the screensaver was not running and had been running in the previous interval, the service recorded the time. If the screensaver was not running for the second or more consecutive interval, the service recorded a @. The service ran for 28 days on the four different types of computers, sending the data it had collected to a dedicated server every 24 hours, except the home user data which was collected manually at the end of the experiment.

We designed the service to keep the impact on the computers' performance small enough so the user would not notice. Sampling at 10 second intervals kept the CPU utilization low; according to the Windows Task Manager, on a Pentium III 450 MHz computer with 256 MB of RAM, the service used 0% CPU utilization. According to the Task Manager, the service used approximately 7 MB of RAM. Our method of recording the data limited the size of each file to approximately 9 KB, which kept the network bandwidth to transfer the files to the server extremely low. Once the data had been collected, we analyzed it, discarding any period of available or unavailable time that started before the data collection began or ended after the data collection finished.

5 Results & Analysis

We obtained traces of 68 public computers from our university which were split between 3 computer labs, 38 undergraduate computers, and 25 home computers, and 26 business computers. We found that the average of the total percentage of time during the studies that different types of computers were available for volunteer computing varied greatly between the different types of computers, ranging from 73% for public computers to 27% for home computers and business computers to 18% for student computers.

We observed that the amount of time computers were available for volunteer computing varied greatly by computer within the undergraduate student, home user, and business types of computers with a range of over 80%, as shown in figures 1, 2, and 3. However, the amount of time the public computers were available for volunteer computing varied significantly less by computer, as shown in figure 4. The average durations for the available, unavailable but powered on, and unavailable periods differed significantly between the different classes of computers, as shown in Figure 5. The cumulative distribution functions for the available periods differed a bit as shown in Figure 6, although we found that the cumulative distribution functions of the other periods were relatively similar for the different classes, as illustrated by Figures 7 and 8.

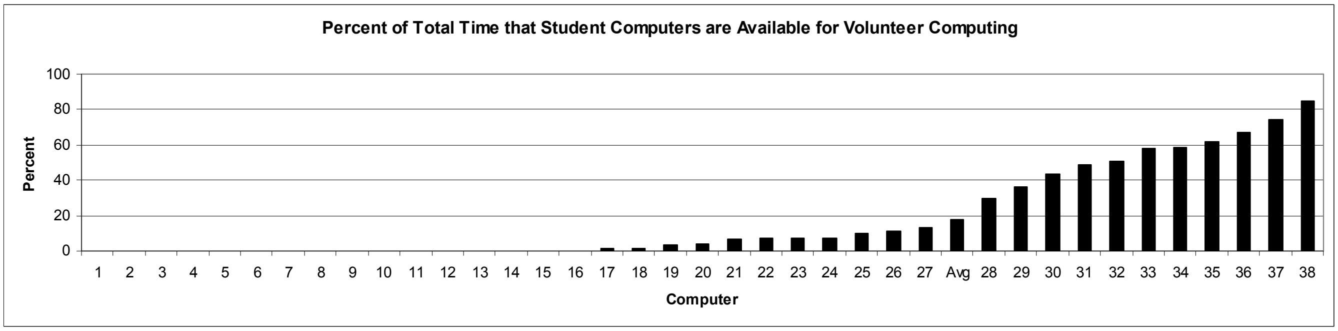

Fig. 1 - Student Computer Availability

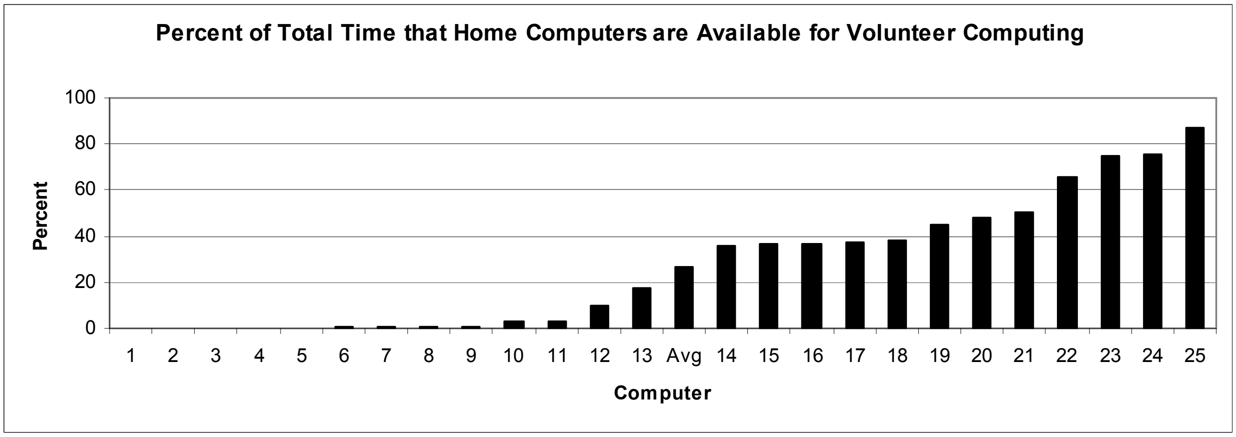

Fig. 2 - Home Computer Availability

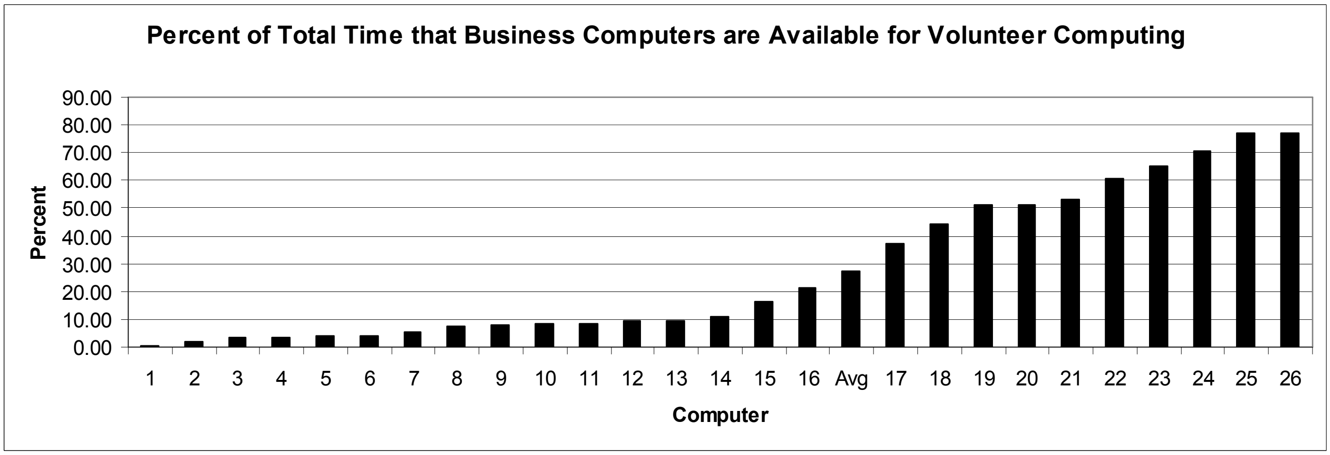

Fig. 3 - Business Computer Availability

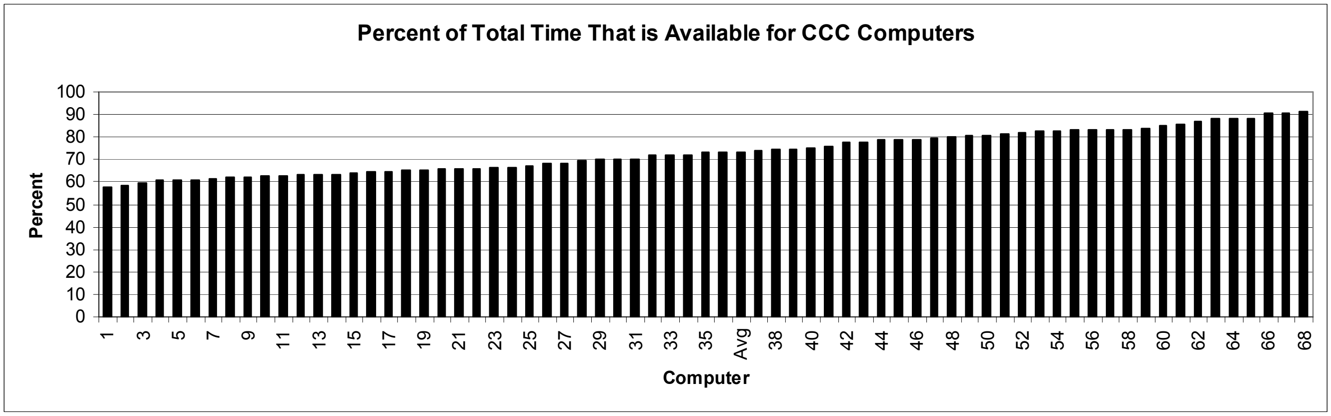

Fig. 4 - Public Computer Availability

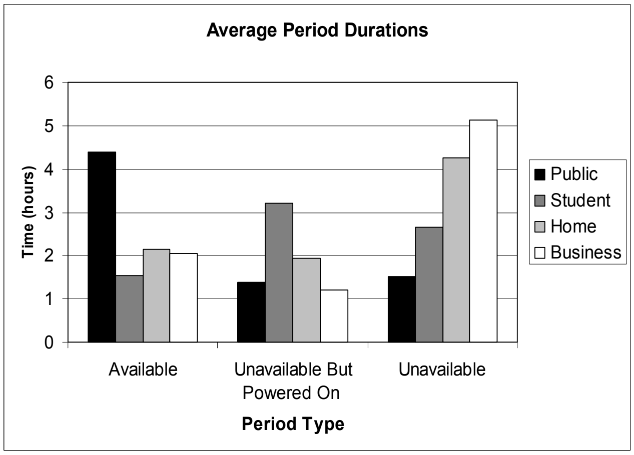

Fig. 5 - Average Period Durations

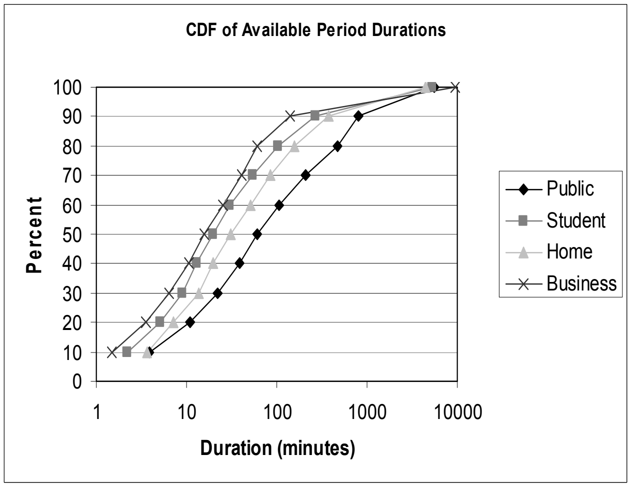

Fig. 6 - CDF of Available Period Durations

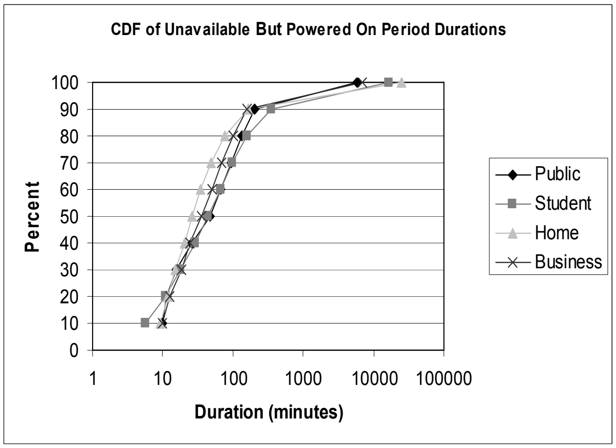

Fig. 7 - CDF of Unavailable but Powered On Period Durations

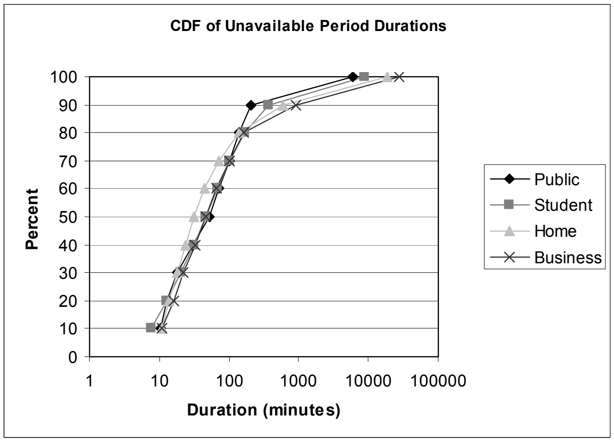

Fig. 8 - CDF of Unavailable Period Durations

Although the CDFs of the different durations appear similar between the types of computers, closer inspection of the data on a per computer basis shows that for the student, home user, and business classes of computers, some computers in the same class that exhibit very different usage patterns. In particular, we found there were computers that were only powered only for short periods of time throughout the data collection period. These computers were almost never available for volunteer computing. However, there were some computers that were powered on for almost the entire duration of the data collection period. Many of these computers were available for volunteer computing for a significant portion of the day.

Using the Arena program from Rockwell Automation, we calculated the most likely distribution for the available periods for each computer. We found that the best match for almost all of the lab computers from two labs was a Weibull distribution. However, despite this being the best match, Arena indicated it was not a good match at all, as the p-values from the Kolmogorov-Smirnov Goodness-of-Fit Test were very large. The best match for almost all of the computers in the third lab was a Beta distribution. The p-values were much better for this lab, but although that doesn't indicate the Beta distribution is a bad characterization, it also does not show conclusively that it is an accurate characterization. The best matches for the available periods from the home, student, and business computers were very inconsistent, being split between five, six, and four different distributions respectively. In addition to this, once again many of the p-values Arena produced for the Kolmogorov-Smirnov test indicated that the distributions were generally not good matches for the data.

6 Conclusions

Our work has produced four major results. The first important result is that we have collected data that was previously unavailable and has the potential to be very valuable for volunteer computing research. In future work, we will use the data to quantify the different amounts of work that can be completed by different types of volunteer computing clients.

The second major result is that our analysis of the data that we collected has shown that the data does not appear to conform to any well known distribution. Thus, to make good use of the data, we believe that using the traces we have collected instead of modeling data with a well known distribution will make work using it more accurate.

The third major result is that we have shown that the method used in [2] to generate the durations of periods when computers were available and unavailable for volunteer computing was too simplistic. To compare the results of using different methods for distributing file-based tasks in volunteer computing, the authors of [2] used available and unavailable period durations that followed exponential distributions because no real data was available. Now that we have collected actual data, the authors of [2] can revise their work and improve their analysis.

The final major result is that our analysis of the data we collected indicates that a single method of task distribution for a volunteer computing project may not be adequate to make the best use of the donated CPU cycles. There was a large variation of times that the student, home, and business computers were available for volunteer computing. Computers that are available for small amounts of time may benefit from receiving only one or two tasks at a time while computers that are available for larger amounts of time may be more productive if they receive higher numbers of tasks at once. Therefore, a single method of distributing tasks to computers appears to be inadequate. We also note that some computers may have long periods where they are available followed by long periods of being unavailable. This type of computer may be more productive if the number of tasks it receives at a time adjusts to match its level of availability during a given period. Therefore, an adaptive method of task distribution may be even more effective than simply classifying a computer by its average level of availability and using a task distribution policy based solely upon that.

7 Acknowledgments

We would like to thank the company (that wished to remain anonymous), the Windows System Administrator at our University, Tom Collins, and all of the undergraduate students and home users who allowed us to gather data from their computers for 4 weeks. This is especially appreciated in this day and age where the fear of viruses and spyware run rampant.

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] D. Toth & D. Finkel, A Comparison of Techniques for Distributing File-Based Tasks for Public-Resource Computing, *Proc. 17th IASTED International Conference on Parallel and Distributed Computing and Systems*, Phoenix, Arizona, USA, 2005, pp. 398-403.

[3] D.P. Anderson, BOINC: A System for Public-Resource Computing and Storage, *5th IEEE/ACM International Workshop on Grid Computing*, Pittsburgh, USA, 2004, pp. 4-10.

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

[5] Personal Communication with David Shapiro. 9/20/05.

[6] "Getting Source Code", http://boinc.berkeley.edu/source_code.php. Accessed 2/23/05.

[7] Personal Communication with George Woltman. 9/14/2005.

[8] Personal Communication with David Anderson. 8/16/2005.

[9] J. Brevik, D. Nurmi, & R. Wolski, Automatic Methods for Predicting Machine Availability in Desktop Grids and Peer-to-peer Systems, *Fourth International Workshop on Global and P2P (GP2P) in conjunction with CCGrid04*, Chicago, Il, 2004.

[10] M. W. Mutka, & M. Livny, Profiling Workstations' Available Capacity For Remote Execution, *Proc.12th IFIP WG 7.3 International Symposium on Computer Performance Modeling, Measurement and Evaluation*, Brussels, Belgium, 1987, pp. 529-544.

[11] A. Acharya, G. Edjlali, & J. Saltz, The Utility of Exploiting Idle Workstations for Parallel Computation, *Proc. SIGMETRICS'97*, Seattle, Washington, USA, 1997, pp. 225-234.

[12] Kondo, Derrick, Scheduling Task Parallel Applications For Rapid Turaround on Desktop Grids, Ph.D. Dissertation. University of California, San Diego, 2005.

[13] D. Kondo, M. Taufer, C. Brooks, H. Casanova, & A. Chien, Characterizing and Evaluating Desktop Grids: An Empirical Study, *Proc. International Parallel and Distributed Processing Symposium* (IPDPS'04), 2004.