Resource Management In Volunteer Computing Grids

CS-575 Position Paper

Geoffrey Oxholm
Beata Chrulkiewicz

November 10, 2007

Contents

Abstract

The success of volunteer computing (VC) grid applications depends greatly on the systems ability to effectively manage the resources available to it. The most popular middleware application, Berkeley Open Infrastructure for Network Computing (BOINC), has achieved its popularity in part by succeeding in this task.

The defining characteristic of VC grids is node unreliability. Because nodes can appear and disappear at any time, it is likely that assigned tasks will not be completed. Various VC architectures have been implemented to deal with this issue. The architecture implemented by BOINC is to send each task to some number of worker nodes; this can be called the redundancy coefficient. A large coefficient will yield a reliable network, but will waste resources. A small coefficient will yield an unreliable network, which may never complete important tasks. This redundancy coefficient is in part up to the application developers don't know enough about their constantly changing grid to make an informed decision.

There are many different proposed and implemented techniques to effectively manage the resources in the BOINC architecture. Some designs operate as external tools which help predict grid performance; some offer integrated libraries that draw on other BOINC project data; and others use external means to encourage node reliability. Although all approaches have the same goal in mind - a faster virtual supercomputer - the methods are quite varied. This paper discusses the redundancy dilemma and analyzes a few examples of the current architectures and architecture modifications which aim to improve BOINC performance, but fail to completely address the issue. Ultimately a multifaceted approach is surely the most effective solution, and a more holistic architecture is needed for BOINC to continue to succeed.

1 Introduction

1.1 Volunteer Computing to Date

With the introduction of grid computing as a concept in [1], the idea of decentralized computing has become the paradigm for affordable supercomputing. While the fastest supercomputer, IBM's Blue Gene [2] performs at impressive speeds, it is not an affordable solution for the world's demanding computational problems. Computer clusters are more practical, but these still require space and money which many projects do not have. To address these issues Volunteer Computing (VC) grid research, like the original pioneering work done Distributed.net [3], is proving to be an effective method of providing computing resources to scientific and educational projects with limited budgets. By enlisting volunteers to donate CPU time on their workstations a free, ad hoc, decentralized grid is made.

Currently VC grids, like those implementing the Berkeley Open Infrastructure for Network Computing (BOINC) grid perform, perform beyond supercomputer speeds. Where Blue Gene performs at 360 TeraFLOPS [2], BOINC grids nearly double this speed[1] [4] and its speed is increasing every day. Other VC grids, like Folding@home achieve over 1.1 PetaFLOPS [5].

1.2 Volunteer Computing Challenges

The impressive throughput of VC grids comes at the cost of high complexity in managing resources. The shortcomings of an open, volunteer based grid are numerous and stem directly from the power of the architecture. The primary pitfall of VC grids is node unreliability: nodes can join and leave unexpectedly, they vary greatly in processing power and connection quality. Moreover, In a VC architecture, there is nothing stopping a malicious user from connecting to the grid and supplying false data.

These problems do not exist in a standard computer clusters where the computers are all located in the same place, under the control of one organization and protected by this organization's secure network. Additionally, while in order to increase the power of a commercial grid, an organization can just purchase additional equipment, in a VC grid world there must me some mechanisms to attract volunteers who have relatively little to gain by volunteering their resources.

The most popular middleware system for volunteer computing, BOINC, has several mechanisms to deal with resource management problem, but they do not offer a complete solution. There has been some research on improving performance of the BOINC grid, most notably RIDGE and SimBA projects. Although those projects offer promissing results, they are not a complete solution. Moreover, they can (particularly RIDGE) potentially introduce obstacles in attracting users, therefore actually diminishing the grid power.

The remainder of the paper is organized as follows: Section 2 presents how the BOINC architecture deals with the problem of managing unreliable resources and discusses the shortcomings of BOINC's solution. Section 3 discusses a few strategies to improve resource management in BOINC, their successes, and shortcomings. Section 4 details the authors' proposal for the future. Finally, section 5 provides an overview of the current state of BOINC computing and what impact future work may have.

2 BOINC's Approach to Resource Management

2.1 Overview

The BOINC architecture (as introduced in [11] and further developed in [12]) is a client-server architecture. Volunteer worker nodes connect to the server and request jobs. To address the issue that nodes can leave before they finish the computation, each job, or work unit, is sent to a fixed number of different workers. This redundancy factor is a configured by the manager of the project.

The results of the work are returned to the server upon completion. As a protection against malicious users, the server verifies them to determine if data is admissible or needs continued effort. If verification fails, additional nodes are used until the work unit passes verification. The two most common verification techniques are majority and M-first

[1]BOINC reports running at 683.675 TeraFLOPS today, November 8, 2007

voting. In majority voting, more than half of the job's worker nodes must agree on the results. For example, if the redundancy factor is seven, then four workers must return the same results for the job to be considered complete and accurate. M-first voting says that a result is complete as soon as M workers agree. For example, if the redundancy factor is set to seven, you might set M to be five.

The verification method and parameters are set by the project administrators, and remain constant. By leaving reliability settings up to the administrators, BOINC allows them to find an optimal configuration. Administrators are thus challenged to come up with settings which eliminate the impact of malicious data and node unreliability while maximizing the speed of their grid configuration.

Additionally, BOINC keeps track of node and grid performance, which administrators can use to evaluate their configurations or relay to the volunteers.

2.2 Success and Limitations

Data has shown that BOINC projects are able to achieve a high level of speed and reliability [6, 4]. The first hurdle is gaining support and a large community of volunteers. BOINC provides some features to help project designers achieve this goal, but ultimately this task must be tackled by the project designers themselves.

Even with a sizeable grid, minimizing redundancy without sacrificing data integrity remains a challenge. It is unreasonable to expect project administrators to predict the performance of their grid. Consequently, choosing a configuration setting is often based on guesswork. Even if a sound configuration is reached, the dynamic nature of a grid means a fixed configuration may lead to an unpredictable decline in performance.

Finally, a generic configuration may make for a reasonable general policy but fails to take full advantage of volunteer nodes which are more reliable than others. A policy which honors reliable nodes would be able to decrease the need for redundancy when possible thereby increasing the throughput of the grid.

3 Maximizing BOINC Resources

Achieving maximum performance from a BOINC grid is difficult and many projects have been designed to achieve this. Although all approaches have the same goal in mind - a faster, reliable BOINC virtual supercomputer - the methods are quite varied. In the next section we will discuss three major approaches: 1) To maximize the number of users and their reliability through feedback and reward; 2) To use simulation tools to test and fine tune configuration settings; and 3) Use nodes' past performance to dynamically modify grid configuration.

3.1 User Encouragement - Feedback and Reward

The power of the VC grid comes from the number of the participants, therefore BOINC must be able to reward volunteers in order to retain them and to attract new users. Credit is an effective way of keeping users interested: each correct computation is rewarded by a credit unit. Credit scores can be posted on websites, such as [4], where participants can compare them. Some people enjoy the sense of competition this provides. Additionally, BOINC gives participants a sense of community. There are websites [7, 8] where users can form teams, chat with other participants and provide technical support to other volunteers.

Screen savers are another way to rewarding volunteers. As mentioned above, BOINC allows the application authors to create a screensaver for their applications. This screensaver can give users instant feedback on their participation by displaying job statistics.

Additionally, to eliminate any inconvenience that running BOINC applications might cause to the users, BOINC provides an interface where users can specify their preferences as to when an how their system may be used by the grid, for instance, only in the night hours, or only when there was not keyboard or mouse input for x time. This is important, because any perceivable diminishment in their computer's performance could discourage its owner from further participation.

3.2 Maximizing Configuration through Usage Simulation

As mentioned in Section 2, a major aspect of maximizing grid performance is to have proper settings. In particular, a good redundancy factor is crucial to balancing performance with reliability. It is often the case that researchers have many hypotheses regarding configuration. Unfortunately testing the hypotheses is difficult for a number of reasons: 1) Making configuration changes then waiting for the grid performance changes to be enumerated and reliable can take considerable time; 2) A live grid is constantly changing and the results of the experiments are consequently unreliable; and 3) Making changes may impact the volunteer nodes adversely, potentially resulting in abandonment and loss of grid resources.

In response to this shortcoming researchers have developed BOINC simulators [9, 10] which provide offline simulation of BOINC grids to help evaluate possible BOINC settings. In particular researchers at the University of Texas at El Passo have developed a tool called SimBA (Simulator of BOINC Applications) [9] which builds on the work of [10].

SimBA starts with a database of trace files from previous BOINC projects. By evaluating node performance from the trace files, SimBA is able to simulate responses which are evaluated by the configured BOINC architecture as successful or failed. This information determines how BOINC will request future data and the cycle is continued until the simulation data is exhausted. When the simulation is over a configuration can be evaluated in terms of grid throughput.

3.3 Dynamic Redundancy through Reliability Prediction

Reliable Infrastructure for Donation-based Grid Environments (RIDGE) [13] is an enhancement to the BOINC architecture that uses past performance of a node to evaluate its reliability. RIDGE attempts to solve a "redundancy factor problem" of BOINC. In BOINC the redundancy factor - the number of nodes that have to preform the work unit before the result can be validated - has to be set in advance by administrators with no knowledge of grid performance. Setting this number too high will waste grid resources. Setting it too low may give a result that cannot be validated which, in turn, will waste the resources because the computation has to be repeated. By conducting some experiments, RIDGE authors discovered that for a given desired success rate, there is an optimal fixed replication settings. This setting depends on the reliability of the workers [13].

It is a natural conclusion that adjusting the replication setting according to the reliability of the workers could improve grid performance. To achieve that, RIDGE stores the information on the reliability of a node. When the work unit is validated, Reputation Manager (a RIDGE component) calculates node's reliability rating. Ridge has an additional parameter Scheduling-Threshold that specifies the number or workers that a RIDGE server will wait for before scheduling work. In contrast, BOINC assigns work to nodes as they request it. By waiting for a quorum of workers to assemble first, RIDGE can select the ones with the highest reliability, thus maximizing the chances that the work unit will be completed. Additionally, since the reliability of the nodes is known before the task is scheduled, RIDGE can dynamically adjust the redundancy factor, improving the throughput.

4 Evaluation

Each of the methods described above provide valuable contributions to the topic of resource management for VC grids but are incomplete. The methods for attracting new participants by giving them credit and encouraging competition can create unpleasant situations that tarnish BOINC image. In June 2006, a BOINC user called Wate released a Trojan that installed the BOINC runtime on some 1500 computers [14]. Those machines performed work for BOINC projects but reported it as done by Wate. This vastly improved Wate's credit rating. Since the machines owners, unaware that BOINC was running on their computers, did not control BOINC via preference settings, they experienced significant performance degradation on their machines. Incidents like this not only negatively impact the victims' view of BOINC, but also can prove demoralizing to people who are trying to get credit honestly.

The tests that the SimBA authors performed are promising. They have shown that reasonable predictions can be made in a much shorter time. In one project they were within 3% of actual performance in a test that ran for 44 minutes instead of the eight days of data to which it was compared [9]. Unfortunately, SimBA has not yet been used in any real projects, so the success of this concept are not yet fully tested. Nevertheless, the idea has shown promise and may prove useful. What is more, SimBA allows researchers to test settings in a simulation environment. This speeds up the time of testing and eliminates the possible negative effects of testing in a live VC community.

However, these benefits are balanced by a number of shortcomings. Most significantly, the project fails to address the dynamic nature of a grid. By using a standard set of past data, this design ensures each test can be compared to other tests, but fails to ensure the test data can be applied to a real world grid. Further, although researchers no longer have to decide which configuration is exactly right for them, they are still required to come up with a short list of options. Hypotheses are difficult to form with regard to VC grids, and this challenge still reamains, although minimized to some extent.

The work done by the authors of RIDGE [13] has been shown the most promising return for an embedded solution. Grid performance can be expected to increase as reliable nodes will help to minimize the redundancy. Additionally, reliability will not suffer as only reliable nodes will be involved in low redundancy calculations, while unknown or irregular nodes will be involved in higher redundancy calculations, minimizing the potential for invalid data. On the surface, the RIDGE system offers certain protection against malicious users. Nodes that continually turn in incorrect results will be assigned low reliability rating, so they will not be assigned work units. However, a user could build up a good reputation in hopes of minimizing the redundancy factor thereby maximizing the potential for malicious damage.

RIDGE offers a valuable improvement to BOINC scheduling, but it also creates some problems. As the design always selects "the best players" for the team, the slower computers, or new users that do not have reliability rating established yet, are unlikely to be scheduled for a task. This makes it difficult for these nodes to improve their reliability rating. This creates a vicious cycle where the best nodes are constantly picked for jobs and constantly improve their rating, while it gets harder and harder for the weaker nodes to get a task and the gap between the two keeps growing. One side effect of this is that it destroys the system of incentives where volunteers are attracted to the grid by the promise of competition and getting credit. Moreover, it treats new users, users with slow machines and malicious users the same way, which can be discouraging and ultimately result in lower grid throughput.

5 Conclusion

The founding work done by the BOINC developers at Berkeley has proven to be a reliable and powerful starting point. Although BOINC has provided supercomputer level performance to educational and scientific projects, it left some crucial aspects of resource management up to the project administrators. They were given an impossible task of selecting settings for their project that would maximize grid resources, without having any data on how the grid performs. In BOINC, selecting the right redundancy level remains guess work.

The projects discussed in this paper attempt to maximize resources in the BOINC environment, but ultimately, they have all proven to be incomplete. Currently, the solution posed by RIDGE uses only the statistics of completed work units to calculate node reliability. By utilizing a more complex model system, a node's reliability could be determined not just on the sum of past performance, but by determining during what time periods the node is most effective. This data should be used to determine how reliable a given node is at the time of its assignment. If, for example, node A is online every Monday night for six hours, it can be assigned a work load which can be quite long. Conversely, nodes with more erratic pattern of activity will be assigned smaller tasks that can be completed in a shorter time.

Another important issue that is not directly addressed is the issue of malicious nodes. A reliability rating should be split into two parts: results that were correct but turned in late, and incorrect results. This could help to differentiate between the malicious nodes which supply wrong results and slow nodes that may take longer to perform computations. Additionally, some system for "blacklisting" the malicious nodes should be created.

While there is no one solution that can fix the obstacles that currently exist for BOINC volunteer computing grids, a combination of simulation environments and node reliability projects such as those in [9, 13] could create an architecture that leverages the strengths of these two methods. An initial redundancy factor could be calculated by running simulation. This factor would be further refined dynamically using node performance tracking.

Volunteer computing grids have long since proved their power and their place in the future of supercomputing. Although each day more volunteers are willing to donate their resources, the task of maximizing these resources while maintaining data validity remains an important topic. The research presented in this paper has shown promise and important reasons for continued development. Combining efforts and developing intelligent resource management techniques such as those proposed by this paper will ultimately lead to higher grid throughput.

References

[1] Ian Foster, Carl Kesselman *The Grid: Blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers Inc. San Francisco, CA 1998

[2] International Business Machines *IBM Rsearch — Blue Gene* http://www.research.ibm.com/bluegene/, Accessed November 10, 2007

[3] Distributed.net Community *distributed.net: Node Zero* http://www.distributed.net, Accessed November 10, 2007

[4] BOINC Community *BOINCstats — BOINC combined - Credit overview* http://boincstats.com/stats/project_graph.php?pr=bo, Accessed November 10, 2007

[5] *Folding@home Statistics* http://fah-web.stanford.edu/cgi-bin/main.py?qtype=osstats, Accessed November 10, 2007

[6] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, Dan Werthimer *SETI@home: An Experiment in Public-Resource Computing* Communications of the ACM, Volume 45 Issue II, ACM Press, November 2002

[7] SETI@home Community *Teams* http://setiathome.berkeley.edu/team.php, Accessed November 10, 2007

[8] Team OcUK Community, *Team OcUK Predictor@home* http://teamocuk.com/PAH.php, Accessed November 10, 2007

[9] Michela Taufer, Andre Kerstens, Trilce Estrada, David Flores, Patricia J. Teller *SimBA: a Discrete Event Simulator for Performance Prediction of Volunteer Computing Projects* Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation, IEEE Computer Society, 2007

[10] D. Kondo. *Scheduling Task Parallel Applications for Rapid Turnaround on Enterprise Desktop Grids* PhD thesis, UCSD, 2004

[11] David P. Anderson, Carl Christensen, Bruce Allen *Designing a Runtime System for Volunteer Couting*. Proceedings of the 2006 ACM/IEEE conference on Supercomputing, ACM Press, 2006

[12] David P. Anderson *BOINC: A System for Public-Resource Computing and Storage*. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, ACM Pess, 2004

[13] Krinshnaveni Budati, Jason Sonnek, Abhishek Chandra, Jon Weissman *RIDGE - Combining Reliability and Performance in Open Grid Platforms*. Proceedings of the 16th international symposium on High performance distributed computing, ACM Press, 2007

[14] BOINC Community *Trojan boinc installation by rogue member* http://boinc.berkeley.edu/dev/forum_thread.php?id=1571%7d Online Forum, February 20, 2007. Accessed November 10, 2007