Performance Analysis of Distributed Systems using BOINC

Muhammed Suhail T. S.
ASE Tata Consultancy Services
Pooranam House
Nellikkunnu Kasaragod

ABSTRACT
In this paper, we are discussing about the performance of distributed computing systems. Performance is a key parameter which needs to be addressed for any system. Since huge computational power could be employed by using distributed systems, the performance or the complete utilization of such power is also of utmost importance. The key quality performance characteristics such as response time, throughput and scalability are vital to the operation of distributed computing systems. When huge, geographically separated systems are employed the reliability as well as the availability of these systems becomes crucial. We will be looking into the possibility of reducing the downtime of distributed systems and providing a consistent performance throughout its entire life cycle.

Keywords
IT – Information Technology, BOINC – Berkeley Open Infrastructure for Network Computing, API – Application Interface, Chunk – small collection of data, mb – Mega Byte, FLOP – Floating Point Instruction, XML – Extensible Markup Language

1. INTRODUCTION
Distributed computing systems is a software system where computers are connected through networks, where systems are located far away geographically, to achieve a common goal or objective. Distributed systems can provide solutions to many problems which needs immense computational power as well as storage space. There are many challenges that need to be addressed when designing and implementing such a system.

Distributed systems have gained importance as a result of many mergers among different industrial sectors taking place nowadays. The different divisions of a newly merged company have to deliver unified services to their customers. This requires that their IT systems needs to be integrated to allow faster delivery. Since building the systems from scratch is time consuming as well as costly, one of the best alternatives to use would be distributed systems. Since distributed systems are highly scalable, adding a large number of geographically separated nodes will be easier. Adding to this distributed systems are much more secure than other networked systems currently in existence.

There are some critical aspects related to the performance of a distributed system. It includes *Throughput*, which is the amount of work which can be completed by the system in a given amount of time. Throughput of the distributed network matters since a system with lower throughput is not recommended for use in any industry. *Scalability* is another critical aspect related to performance. It is the ability of the system to cope up with increasing number of nodes which can increase both the computational power as well as storage capacity of the existing system. When the scalability of the system is high it will be able to adapt or increase/decrease the throughput of the system according to the requirements. *Latency* is typically the time delay between request and response of an operation which is being performed. As the latency of a distributed system decreases its throughput increases. Both latency and throughput are inversely proportional. Mostly the performance characteristics of distributed systems are measured towards the end of its deployment, i.e. once the system is up and running. But in reality the performance of a distributed system is largely dependent on the base architecture which is followed for its development. Hence performance factors needs to be considered not only towards the end but also during the initial stages of design and development of the system. Only then we will be able to optimize as well as obtain maximum performance from the distributed system which is being deployed.

In this paper we will be taking the example of Berkeley Open Infrastructure for Network Computing (BOINC) API which is the most widely established and reliable source currently existing in distributed computing space. The computational power and storage power which has been achieved for several projects running under BOINC is far much higher than that of the fastest super computer in the world. This proves that with effective regulation of connected computing devices, spread across the globe even complex tasks could be accomplished with ease, which is otherwise considered impossible to achieve. The speed and other performance characteristics of BOINC and its advantage over other computing architecture in detail will be described in this paper. Moreover, it will also include brief ideas about methods which can be used to obtain consistent computing power in a distributed system, as reliability is a crucial aspect which has reduced the effective utilization of distributed computing systems.

2. BERKELEY OPEN INFRASTRUCTURE FOR NETWORK COMPUTING (BOINC)
Volunteer Computing system is one in which the public allows their storage and computational capacities to be used for scientific research. In this system, the computing resources, i.e. phone, tablets, desktops, laptops are plugged into a distributed computing environment and it acts a whole new super computer having immense computational power and storage capacity. The first volunteer computing project was Great Internet Mersenne Prime Search, which was started in 1996. Today this concept is being used in many fields including high energy physics, molecular biology, medicine, astrophysics, and climate dynamics.

BOINC is a volunteer computing software system which helps scientists to create and utilize public resources by sharing their computational power and storage through a distributed system to accomplish a specific goal. It is a middleware for volunteer computing in which more than 30 projects are currently being operated. In order for the volunteers to participate in projects run in BOINC middleware, they have to download the BOINC client software on their systems. In this system, the bulk of big data (includes both structured as well as unstructured data formats) is stored in server system from which it is broken down into smaller chunks. These smaller chunks of data are then distributed to the peer nodes connected to the distributed environment. These data’s are then individually processed using the connected peers and the result of the same is uploaded back to the server. Since the data chunks are of very small size it is easy for the systems to accept and process the same with lesser time and higher speeds. The result obtained from each node in the distributed environment is then aggregated to form the final result. In effect, the system accomplishes a tedious computational task at high speed and effectiveness. The BOINC client software interface is shown in figure (1).

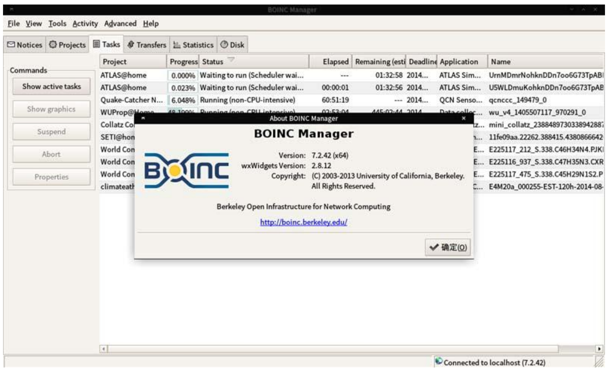

Figure 1. BOINC Manager running 3 different projects.

2.1 Projects Run Using Boinc

The following are some of the interesting projects currently being run on the BOINC manager. Every one of the below listed project is unique.

SETI@home

Search for Extra Terrestrial Intelligence (SETI) is the project where we are analyzing the radio signals in space and trying to identify whether there is extra-terrestrial intelligence. In this project the data is divided to small data chunks of 0.35mb and send across the computers in the volunteer grid from Berkeley to compute and obtain the result. The results are then merged to the SETI@home database and then using various pattern detection algorithms we search for the most interesting signals obtained.

Einstein@home

Einstein@Home is hosted by the University of Wisconsin–Milwaukee and the Max Planck Institute for Gravitational Physics (Albert Einstein Institute, Hannover, Germany). This project studies data obtained from Laser Interferometry Gravitational Observatory (LIGO) and analyses it to search for finding evidence of continuous gravitational wave sources, which probably comes from rapidly spinning non-axisymmetric neutron stars. Einstein@home detected a previously undetected radio pulsar J2007+2722 on Aug 12 2010, which was its first. The project had discovered 49 pulsars till December 2014.

Milkyway@home

MilkyWay@home is a collaboration among the Rensselaer Polytechnic Institute's department of Computer Science and Physics, Applied Physics and Astronomy and is supported by the U.S. National Science Foundation. This project aims to generate accurate three dimensional dynamic models of stellar systems in the immediate vicinity of our Milkyway galaxy. It currently operates at speeds of above 1 petaFLOP.

Folding@home

Folding@home is developed and operated by the Pande Laboratory at Stanford University. This project is used to study the mechanism of protein folding, misfolding, aggregation and related diseases. Protein folding is the process by which proteins finally reach their three dimensional structure. Folding@home also tries to calculate the proteins final structure and determine how other molecules interact with it. This is the first project in which GPUs, Playstation, Sony Xperia smartphones are used in the grid. The current performance of this project is at 40.2 x86 petaFLOPS and 14.4 native petaFLOPS.

The total computational power provided by all these projects combined is equivalent to 10 times the computational power of the fastest super computer namely Tianhe-2. As studies indicate that the number of personal computing devices is only going to increase exponentially in the coming decade, the scope and viability of distributed systems is going to be very high.

2.2 Challenges Faced When Designing Boinc

The BOINC software client has evolved through several stages to reach its current level. There are many issues which needs to be addressed while designing and employing a distributed system.

1. **Different systems used in the distributed environment.**

   Since this project is going to be run using the different types of systems, having different software as well as hardware, it is important that the software component should be independent of the Operating System (OS) and the hardware in use. In order to implement this, BOINC provides a novel framework for distributing application executables. Usually, these are compiled and distributed by the project itself, for the supporting machines. But for some users and platforms this causes issues. Hence BOINC provides an **anonymous platform mechanism** where participants can download and compile the application source code and via an XML configuration file inform the BOINC client of these application versions. Now, when the client communicates with the project server, it indicates that the platform is anonymous and the server provides work units for the corresponding application versions.

2. **How to restructure the data for processing.**

   The BOINC client divides the data into several work units. These work units represent the inputs to a computation. It contains parameters such as compute, memory, storage requirements and a soft deadline for completion. A result represents the result of a computation. It consists of a reference to the work unit and a list of references to the output files.

Since the data needs to be distributed over a network, it is created as units of less size which can be downloaded easily and the corresponding results after processing the same could be uploaded easily. The files used in the BOINC system have unique names and they are immutable. However, files can be replicated. A file includes, list of URLs from which it may be downloaded or uploaded. Files can have associated attributes indicating they must be validated using a digital signature or it should be compressed before transferring it over the network.

3. Checkpoints.

Since the tasks which are currently run in a distributed environment can be interrupted before its completion, a system needs to be in place for handling this scenario. When running a task in distributed network, after a particular time period a checkpoint is created. I.e. results of whatever operation that have been performed till that particular time will either be backed up in the system itself or uploaded back to the server. In case of any failure of the volunteer or the connected system in the grid, once that system is back online, it can resume the remaining calculation from the most recently saved checkpoint. It is also possible for another system to take up or continue the task which has been interrupted with the help of checkpoints. Checkpoints are crucial when we are dealing with critical computing applications that are time consuming.

4. Obtaining Users.

In distributed computing environment one of the greatest challenges that we face is obtaining users. Since the whole architecture is dependent on the users, or the number of systems which could be connected to the grid, users are of critical importance in BOINC. Moreover, BOINC offers community where it is possible to form teams and carry out an entire task. It provides a competitive mentality between the volunteers in the grid and in turn helps to gain more users. The GUI of the BOINC client is also very simple and easy to use. Hence simplicity provides another big impact in gaining users. BOINC also provides a credit rating system where it is possible to see how much time, storage space and computational power we have dedicated till that time, which gives the user information about the position in which they stand in comparison to the whole grid. This creates a competitive mentality which helps to lure more number of systems as well as users to the projects.

In a corporate environment, since volunteers are not necessary, it is possible to reduce such graphical enhancements. But, keeping track of the data values such as computational power, storage are vital to the entire distributed architecture.

5. Retaining Users

Retaining users is just as important as obtaining users. Since it's a volunteer grid, the users need to be attracted so that they will lend the computational power of their devices to accomplish a common goal. If the environment is shifted to a corporate one, we can reduce this dependency of retaining users, since the local machines owned by an industry could be employed easily as compared to a volunteer grid. Moreover retaining users is a tedious task as we always need to add something new so that the current users don't lose interest in the system. There is always a need for the spark which is required to keep the volunteers in the grid.

6. Processor overheating

The initial challenge which was faced by the designers while implementing a distributed computing API was the utilization of the computing power of the system connected to the distributed environment. When the BOINC API was initially released, it used the volunteer's processors computing power to the maximum which caused many hardware failures for the consumer. Since there was no control employed for utilizing the CPU computational power, BOINC API consumed more than 70% - 80% of the CPUs computing power which caused overheating and eventual damage to the hardware. In the later stages, BOINC was optimized to reduce the high usage of the client's computer system to prevent such catastrophic scenarios. The usage of any resource in the distributed environment needs to be governed properly, as uncontrolled usage may cause damage to the hardware system at a very fast pace. The computational capacity of systems connected to the grid should never be used to their maximum for a prolonged period of time. Optimal usage increases the efficiency as well as reduces the electricity consumption and hardware damage.

7. Memory Overload

Whenever a system is used at its maximum computational capacity, automatically the memory usage also goes high. The 'transactions per second' increases which will utilize all available physical as well as virtual memory, making the device all the more lagging. Constraints on memory are a must as the distributed environment mainly works through fast upload and retrieval of processed data. Hence a user will never use a system which consumes too much of memory and makes his/her system obsolete. Instead the BOINC countered this problem by setting up a limit which could be described by the user as to how much can the API consume at its peak. Further optimization made it possible to sense when the system is idle and make use of the resources appropriately. Whenever the user needs the system the BOINC processes get suspended so that it doesn't affect the user's tasks.

8. Graphical Interface

Since BOINC is a volunteer computing project, the presence of users who can pool in their resources are of utmost importance. In order to attract more users a simple and elegant Graphical Interface is required. The graphical interface shouldn't be too much to handle nor too less. It should be in the right combination so that the users feel it is easy and simple to use. Only by obtaining substantial amount of users into the grid the virtual super computer becomes a reality. Hence to employ a powerful grid the need for obtaining and retaining users are very high. Efficient and simple Graphical Interface helps to gain users as they feel comfortable to use the API.

3. PERFORMANCE FACTORS

The prime performance indicators of distributed systems are throughput, scalability, latency and reliability. As discussed the throughput of the BOINC system is very high as compared to any another networked computing architecture in use today. All the benchmark performance characteristics of this system is above and beyond the expectations.

3.1 Throughput

As discussed, throughput is the measure which shows the amount of work performed by the system. For a distributed super computing system like BOINC, throughput is crucial. Measuring throughput of the system is also a challenge. But the latest data available indicates that for projects such as Milkyway@home and SETI@home, the BOINC system has attained sustainable speeds of over 150 Peta Flops(5 times more faster than the fastest Super computer in the world, aka Tianhe – 2). The possibilities are wide open for tapping into the immeasurable depths of computational power which could be employed at our fingertips from volunteer computing to corporate grid level computing.

Figure 2 depicts the graph which shows the the computational power employed for a single project, viz. folding@home, compared to the fastest super computers around the world. From the graph it is evident that from 2007 to 2011 folding@home project had an operating capacity of over 10000 Tera Flops, whereas the fastest super computers during that time lagged much behind this at rates of 2500 to 3000 Tera Flops. In the early 2012 we could see a rapid climb in the super computing capacity which increased to 32000 Tera Flops or 32 Peta Flops giving rise to the super computer, Tianhe – 2. By 2013, the computing power employed for folding@home also increased considerably to 20 Peta Flops which shows the huge potential which is hidden in the distributed computing space. The distributed networks are not only capable of having immense computational power, but also it has very high dynamic as well as static memory capacity, which it lends from the hosts which are connected onto the grid.

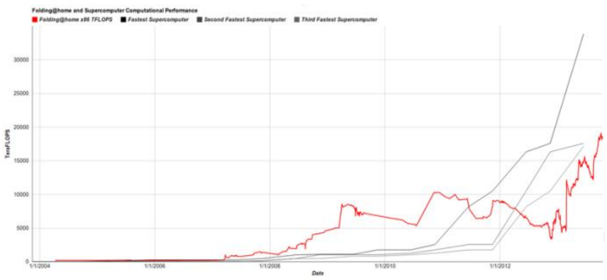

Figure 2. Graph which shows the folding@home computing power in comparison to fastest super computers

The expected increases in number of operational mobile as well as desktop devices are very high in the coming decade. So instead of building high power computing machines the economical way is to utilize the infrastructure that we already have and produce maximum results. Supercomputing is becoming an age old concept which the distributed environment could replace with ease and grace. The BOINC system has proved that the throughput of any distributed network could be made so high as to surpass any super computers. The throughput measure has proven that a BOINC like system can have very high impact on the coming generation.

3.2 Scalability

The BOINC system is a highly scalable distributed architecture. The most attractive feature of a distributed computing environment is its scalability. It should be able to accommodate increasing number of devices, without any problem, according to changing needs. Most of the applications which run on distributed networks are computation intensive; the rapid increase in requirements calls for rapid addition of systems on to the network. Now, we will have a look at the user and host count for the project milkyway@home.

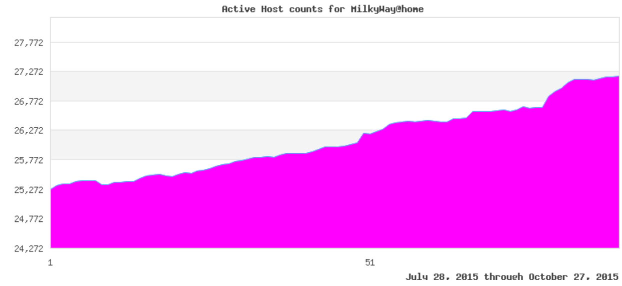

Figure 3. milkyway@home host count

From figure 3, we can see that for the project milkyway@home the host count from July 28th 2015 to October 27th 2015 is following an increasing trend from 25000+ hosts to 27000+ hosts. Here, we can identify that the increasing number of hosts is managed easily and it doesn't cause much problem in the distributed network.

Similar to hosts the number of users in the system has also exhibited a stable level from July to October. From Figure 4, 18000+ users are present in the grid at all time. Handling such huge number of users is also not a problem for BOINC.

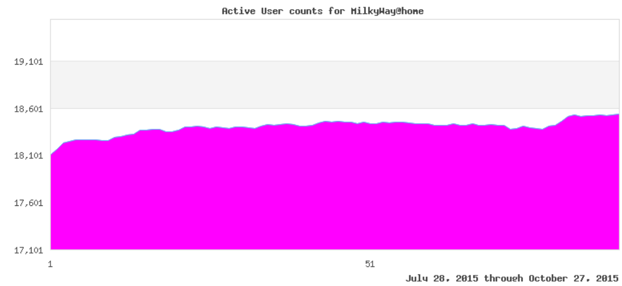

Figure 4. User count for milkyway@home.

As the number of users increases the completed number of tasks also increases. Users are the key resource in a distributed volunteer computing grid. If we are considering a corporate environment, the number of systems which could be contributed from each delivery centers become the key resources.

3.3 Latency And Reliability

The BOINC system has very low latency, since it gives the peers small amount of data to process at a time. The BOINC files which are downloaded from the server are small chunks of data which could be processed fast and the resulting output is uploaded back to the server. BOINC has added preliminary support for 'low-latency' computing. The connecting server defines the time interval in which the system needs to keep contact with the server which is typically in minutes. Earlier the delay was very huge causing a lower throughput. But since the contact between the server and peer happens frequently the resulting latency is low which helps to increase the throughput. Moreover it also helps to synchronize the process. In any distributed network the latency is of considerable importance and keeping it low is always a design challenge. The faster the request and response, higher the throughput. There exists an inversely proportional relationship between latency and throughput.

The reliability of distributed systems is also a critical performance factor. Since we are using volunteers for computing, there is a very high chance that the systems connected in the grid may break off, i.e., they might get disconnected. So while building a distributed network the reliability of the architecture is of utmost importance. In a distributed environment, all the interconnected systems are working in order to achieve a common goal. For that they divide the larger task into smaller ones and accomplish each of these smaller ones and move on to the next. Hence there may be dependencies between the tasks being carried out. Suppose if a system is doing Task A and another system doing Task B, Task A needs to be completed in order to do Task B. The connected system doing Task A dropped off the grid before completing the task hence creating a deadlock situation. In order to eliminate that either we have to divide the tasks as completely independent or we have to employ some redundancy. Like if one system fails the other takes up the task from where it was left off and completes it. In BOINC both these approaches are used as tasks are immutable and if a system doing some task leaves the grid that task is taken up by another system.

3.4 Reliability Improvement

One of the most severe problem faced by distributed computing networks is its reliability. In a network such as BOINC, the users are right away integrated to the grid and given the tasks to process which maximizes the utilization. But the reliability of the volunteers in the grid network varies, like some systems may drop off abruptly. If a huge number of systems happen to drop off from the grid within a minimal time span, it can greatly affect the overall throughput of the network. So while designing a distributed computing network reliability of the same is of very high importance.

In this paper, what I'm suggesting is instead of completely integrating all the systems to the grid we should preserve at 5% - 8% of total systems as failsafe or backup. If we can do that, then even if a particular set of systems drops off from the grid, the backup systems can pick up their tasks from where they have left off and continue working. This will increase the reliability of the grid network considerably and also helps to gain consistency. The grid won't be as volatile as it used to be. But, when we do take such a step, we are reducing the total utilization of the network by 5% +. Considering the reliability a 5% reduction in utilization is acceptable.

In corporate environments if we deploy the grid networks, then it is of utmost importance that we increase the reliability considerably. Throughput and reliability of the whole network should go hand in hand.

The application of grid/network computing is so huge and unexplored, that it can solve many major as well as minor hurdles which we face in our environment today.

4. CONCLUSION

Distributed computing networks still remains as an uncharted area in the computational realm. The potential it holds is very high as compared to other fields. It can provide easy solutions to the complex problems that we face in our routine environment. Hence its application as well as its scope still remains high.

The key aspects in a distributed environment are its performance factors, which will increase its overall efficiency as well as its utilization. If we are able to obtain the correct mix of those factors then the resultant system would be extremely good. There are a lot of key challenges that needs to be addressed while designing this system and getting it right is one of the hardest challenges in distributed computing.

In this paper, we have seen how BOINC systems work and how can we increase the reliability of the whole system by providing a buffer system percentage. This will help to stabilize the entire distributed network and will also help in increasing the reliability of the whole system.

The distributed computing network system still holds the key to solving many problems related to high computational capacity that we face today.

5. ACKNOWLEDGMENTS

My thanks to the experts who have contributed towards development of the template.

6. REFERENCES

[1] Carl Christensen, Tolu Aina and David Stainforth *The Challenge of Volunteer Computing With Lengthy Climate Model Simulations*, Department of Atmospheric Physics University of Oxford Clarendon Laboratory Oxford OX1 3PU United Kingdom(carlc,tolu,das)@atm.ox.ac.uk.

[2] Trilce Estrada · Michela Taufer · David P. Anderson *Performance Prediction and Analysis of BOINC Projects: An Empirical Study with EmBOINC* published on 4 August 2009.

[3] 1 David P. Anderson, 2 Carl Christensen and 3 Bruce Allen *Designing a Runtime System for Volunteer Computing*, 1 UC Berkeley Space Sciences Laboratory, 2 Dept. of Physics, University of Oxford, 3 Physics Dept., University of Wisconsin - Milwaukee.

[4] *Early Performance Testing of Distributed Software Applications*, Giovanni Denaro†, Universit ´a di Milano Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione I20126, Milano, Italy, denaro@disco.unimib.it, Andrea Polini†, ISTICNR, Via Moruzzi, 1, I56124, Pisa, Italy, a.polini@sssup.it, Wolfgang Emmerich, University College London, Dept. of Computer Science, WC1E 6BT, London, UK, w.emmerich@cs.ucl.ac.ukSannella.

[5] B. Clifford Neuman, *Scale in Distributed Systems*, Information Sciences Institute, University of Southern California

[6] Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Distributed_computinghttps://en.wikipedia.org/wiki/Berkeley_Open_Infrastructure_for_Network_Computing