**Volunteer Cloud**

Mayank Tyagi Sunil Parmar

(16CS60R85) (16CS60R59)

**Abstract - Volunteer Cloud is one of the most popular example of Cloud Computing which takes resources from the general crowd who themselves volunteer to participate. The term Volunteer Computing can be easily summarized using two words: “computational” and “participative”. The computational part basically includes the allocation of small computing tasks to different machines volunteered by the general public, whereas the term participative implies the user’s perspective and their contribution in making this venture a success. This paper thus talks about the participation aspect of the users, the factors which influence the users to be a part of volunteer computing. It also includes a Reputation Management Model which helps in the estimation of the reputation of the nodes who serve as volunteers for the system.**

**1 Introduction**

Distributed Computers are nowdays used as major computing power source. These computers volunteer some computing power of their personal computers for common interests and goals. Public computing refers to the contribution of public computers to enhance the computing power for big projects and researches. It is designed for both social as well as scientific benefit. Social benefit, because with time most of the people are losing interest in current scientific researches, by the virtue of public computing, public can be in touch with such researches, they also can give their opinion for the same. Scientific, because the public indirectly gets some control on these scientific researches.

There have been many researches at different physical levels, at atomic level, about earth’s biosphere etc. These researches demand huge computational power which cannot be achieved by a single machine even if the machine is a supercomputer. Such huge demands can be fulfilled by parallelizing the work. The task can be divided into multiple pieces and then can be given to different separate processors at the same time.

Faster computing power is always in demand. So, the development of chips used in computers, have increased rapidly and so does their computing speed. Every 18 months, the speed was doubled, this concept is known as “Moore’s law”.

In 1990[3], the computers became very fast because of “Moore’s law”; in fact as fast as supercomputers just 2-3 years older than them. Increase in speed led to the spread of internet in the consumer market. This resulted into development of large connected networks of fast computers. There were many projects that used the benefit of this concept. Some of them include:

* GIMPS (1997): This project was designed to search large prime numbers.
* Distributed.net (1997): The project was designed to decrypt or decipher the cipher texts or encrypted messages.
* SETI@home (1999): This project was designed to detect and study the radio signals emitted by some intelligent civilization outside the earth. This project was very popular, and millions of people were interested to participate in the project.

We begin by discussing about volunteer computing and its social and technical aspects in section 2, followed by the factors which influence people to volunteer in section 3. Section 4 then talks about the Reputation Management Model, while section 5 gives a conclusion to the entire paper.

**2 Volunteer Computing**

Volunteer Computing is a technique for sharing of resources in a distributed environment. In such a distributed computing scenario, people voluntarily share their resources so that they can be used by other computational tasks. This helps in giving a better utilization of the resources. This form of computing works because of the efforts put in by a community and not any individual body or organization. If a person readily agrees to share his/her resources with others then only he/she can be a part of it, else no force can be implied. If he later on decides to back off then also he is not put under any illegal considerations. Wikipedia is one of the most common applications of volunteer computing[3].

**2.1 Why Volunteer Computing?**

The computing power of volunteer computing is much greater than the supercomputer, cluster or grid. If we take the example of SETI@home, its computing power is contributed by approximately 1 million computers distributed all over the world. The processing rate or speed of SETI@home is 60 TeraFLOPs, which means 60 trillion floating point operations can be done in one second. On the other hand, if we check the processing rate of largest conventional supercomputer, IBM ASCI white, it provides speed of 12 TeraFLOPs.

**2.1.1 Social Aspects**

The most important social aspect of public computing is people get to know about the current scientific researches. If we take the example of SETI@home, the contributors involved had a major interest in knowing the life outside the world. These projects also give credits to contributors by keeping track of their contribution and showing the same on websites in decreasing order. It can be shown as leader boards. In SETI@home leader boards, there is notion of making teams and these teams have their own leader boards.

Incorporating the idea of building team, helped the project in recruiting new contributors for the project.

Sometimes, some participants cheat; they want to take the credit of computation without doing it, or they may do it but, incorrectly. Sometimes some contributors intentionally give wrong results. To avoid such situations, the project needs to have redundant copy of results and then the results can be compared. So, computations are done redundantly to ensure the correctness of the result.

**2.1.2 Technical Aspects**

Technical requirements of public computing:

* The application program should be platform independent.
* The project requires server systems.
* The project requires databases.
* The project needs to keep track of participants/users and their contributions/credits.
* It needs to deal with redundant results.
* It needs to do proper handling in error conditions.

**3 Factors Contributing to Volunteer Computing**

* 1. **Individual Motivations**

The factors which motivate an individual to participate in volunteer computing can be broadly classified into intrinsic and extrinsic factors. Intrinsic factors include mutual benefits of the individuals such as enjoyment and reciprocity [1]. On the other hand, the extrinsic factors are concerned with the after effects of the process. These parts play a major role in increasing the number of volunteers and include factors like gaining a good social reputation, improved skills, etc.

**3.2** **Social and Temporal Factors**

Team affiliation and Tenure are two more important factors which contribute to volunteer computing. Team affiliation emphasizes on the importance of identification within a group and how it affects the participation and involvement of a volunteer. [1] Team cohesion is also taken care of by promoting competition between teams. On the other hand, the effect on tenure on volunteer contributions is debatable. Some evidence suggests that after the initial phase is over and the participants become less enthusiastic towards the work, their participation decreases. On the contrary, there is evidence that as the volunteers get comfortable with the process, start getting feedbacks on their work and start bonding with others, their participations gets increased notably. However, it was found that tenure had a positive effect on second-order contributions and a negative one on first order (information artefact) contributions. Volunteer computing being a first order contribution, tenure of membership corresponds to a decreased level of participation and contribution. However, team affiliation’s positive impact somewhat compensates for the negative effects that come with increased tenure.

The computational [1] aspects of volunteer computing have been studied comprehensively but the participative aspects still lie in the uncharted territory. Using a combination of survey and system data, various factors that determine the contribution levels of participants associated with volunteer computing were identified. The model was tested and regression analysis was performed to determine the actual effect of individual, social and temporal factors on the contribution levels. The results offer a framework for efficient handling of volunteer computing projects.

However, the combination of tenure and team affiliation was found to have a statistically positive and significant impact on contribution levels. The negative impacts of long tenures were statistically less for volunteers associated with teams. This finding is very important because it shows that even though the contributors can’t be kept engaged with only resource contribution over long periods of time, they can still be kept involved and engaged through secondary channels like team affiliation.

**4 Reputation Management System for Volunteer Computing**

Volunteer Computing Paradigm provides a huge amount of processing power to large scale of computing applications. But, to use the huge computing power with this potential, there comes a need of a design which is robust and is able to handle failures. Volunteer computing, in order to deploy services with a high reliability; there is a need to for a robust *Reputation Management System*[2].

**4.1 Reputation Model and Parameters**

This model helps in calculating the slave’s reputation using the feedback obtained for each of the transaction with its master.

Basic important factors of the model are:

* The amount of crash experienced by a slave node.
* The correctness of the returned result by the slave node.
* Performance of the slave node in terms of the performance.

The slave might not always return the result. There can be various causes the slave node is not able to return the result in provided slot of time.

We will discuss in brief about the following two causes:

* Network Failure
* Time Outs
  + 1. **Network Failure**

There can be wide range of reasons for network failure. It can be due to physical link failure or due to the severe congestion in the network. This may cause the slave to come up with an incorrect result or with inability to compute the result. Because of this failure, it is also possible that the result is computed but the slave node is not able to transfer the result back to the master node. If in case a particular node is crashed then the job assigned to it will be reassigned to some another node[2].

* + 1. **Time Outs**

There can be also cases when the slave is able to compute the result but is unable to return the result in the stipulated amount of the time. The cause for this could be limitation of the resources allocated to it. Some malicious slaves can also purposefully get crashed in order to get fake credit or destroy the service provider reputation.

**4.2 Understanding Correctness of the Returned Result**

The master ensures the correctness of the returned result using a two-fold replication. The results calculated and returned by the slave nodes are compared with each other in order to compare for correctness. If in case the results returned by them don’t match with each other than job assigned is resubmitted to other slave nodes. But, this proposal also has some implementation flaws. This may lead to unwanted delay in computation. Moreover, if both the nodes chosen to do the same job belong to the malicious collective, then it can make a severe effect on the results of computation, and master node won’t be able to determine it by any other way.

**4.2.1 Performance Issue of the Volunteer Cloud**

Considering the traditional Cloud computing, it is deployed in sophisticated data centres that have very high performance servers. But the volunteer clouds exploit the processing power of the personal computers which are end devices and can’t do the job well.

To estimate the performance of the slave nodes and rank them, a large dataset is divided in equal sizes and submitted to slave nodes. The nodes are evaluated on the basis of time taken to do the job. A highly reputed node will be taking less time to do the same job compared to a low performing node. Malicious node can claim to do more job to ask more credit, but by comparing the time taken by other slaves to do same amount of job, undue the credit to be awarded.

**4.3** **Reputation Metric for Ranking Slaves**

We are focusing over three basic parameters to estimate the reputation of the slave node. We are going to propose a general reputation metric which can combine all these three parameters to rank the node on a scale of [0-1].

For a provided slave, say

Nr: denotes the probability that a node will crash during a transaction

N: denotes that the node will return correct result for a job submitted.

Etavg: average performance of a node in terms of time taken to do a job.

Then, the overall performance reputation of the system will be given by the following formulae. A0 is normalized so that it lies in between a scale on [0-1].

The first component (1 - Nr) contributes to the overall reputation of the node by quantifying the likelihood of a node to successfully return a result to a job submitted to it [2]. Nr denotes the probability or likelihood of a node to crash and is calculated using a straight forward formula:

Where the C factor denotes the number of crashes of the node while doing the job and the S factor denotes the total number of jobs submitted by master to this node. If the value of this unit is large in that case, the node crashes very often. In that case it will be not a good decision to assign this node any critical task. The value of Nr is always going to be within the scale [0-1].

The second comparison aspect, N adds to overall reputation by estimating the likelihood of the slave to return the correct result for the computation task assigned to it. This factor is also estimated by:

C component of the formulae indicates the number of times the node has returned the correct result and the S factor denotes the number of transactions or jobs submitted by the master to this node.

The next component Etavg is average performance of the node in terms of time taken. The formula for the component is:

E component of the equation denotes the performance of node in terms of time taken. For the ith transaction relative to other participating nodes on a scale of 0-1.

Ci is the index of node in ascending order of time taken as Ct is the number of slaves participating for the transaction. Nodes that have poor performance or malicious nodes cannot claim higher credit as they will have a poor value of compared to other nodes.

We now consider the effectiveness of the overall reputation in the scenario of a volunteer cloud. All the parameters have a value between 0 and 1. Since the overall reputation is a multiplication of the three individual parameters, the overall reputation A0 also is a value between 0 and 1 only.

Let us consider a scenario where a slave node is guaranteed not to crash, and always provide accurate results. This scenario can be compared to a situation where the processing is done in the local processor itself. In such a scenario, we usually measure the performance of the system by the time taken to complete the job. Considering the formula for A0 and Nr is zero and N is 1 as the probability for crash is zero and the system always returns correct result. Which implies, in absence of crashes or inaccuracy, the reputation purely depends on the performance of the system.

1. **Conclusion**

The study involves understanding the factors which determine the contribution levels in a volunteer computing project and using the findings to devise more effective and efficient methods which lead to better motivation and hence, better contribution levels by those involved in the project. The difference between other community projects and volunteer computing projects is also highlighted.

The results of the study can also be used to guide such volunteer computing projects by allowing the project managers and decision makers to focus their effort on the factors that have a significant impact on the contribution levels. Some suggestions for making the volunteer computing projects more efficient would be:

* More social interaction opportunities in volunteer computing projects.
* The contribution scores of the contributors be displayed more prominently on the interface to the community. Graphical representations or a dedicated widget can be used for the same.
* The project accomplishments to be reported back to the community. Knowledge of the fact that one’s contribution was scientifically published will always motivate participants to contribute more.

**6 References**

[1] Nov, Oded, David Anderson, and Ofer Arazy. "Volunteer computing: a model of the factors determining contribution to community-based scientific research." *Proceedings of the 19th international conference on World wide web*. ACM, 2010.

[2] Muralidharan, Sreenidhi P., and V. Vignesh Kumar. "A novel reputation management system for volunteer clouds." *Computer Communication and Informatics (ICCCI), 2012 International Conference on*. IEEE, 2012.

[3] Anderson, David P. "Public computing: Reconnecting people to science."*Conference on Shared Knowledge and the Web*. 2003.