

Availability based Scoring Model for Resource Grouping in Desktop Grid Computing

Soodeh Peyvandi^{1*}, Rohiza Ahmad² and M. Nordin Zakaria¹

¹High Performance Computing (HPC) Center, Universiti Teknologi PETRONAS, Tronoh, Perak, Malaysia; soodehpeyvandi@gmail.com,

²Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Tronoh, Perak, Malaysia

Abstract

Many scientific projects such as SETI@home are leveraging on enormous numbers of distributed internet-based idle desktop computers to execute their jobs in a Desktop Grid (DG) computing environment. These idle desktop computers (volunteers) are volatile, i.e., they are free to join and leave the DG systems. Besides, the volunteers might also have different resources which are heterogeneous in characteristics and capabilities. Due to the above facts, volunteers in a DG system have different levels of reliability and availability which makes scheduling of jobs more challenging. Thus, this paper presents a grouping method which assigns these heterogeneous volatile volunteers into different levels of reliability groups, according to their availability score. In order to derive the availability score model for the grouping purposes, this study uses statistical analysis of real trace data which are taken from SETI@home project. By having the proper grouping, better assignment of jobs can be achieved in a DG system. Simulation runs of the approach shows that it is feasible to volunteers are able to be grouped using the scoring model.

Keywords: Availability Scoring, Desktop Grid Computing, Grouping, Volatility

1. Introduction

Desktop grid computing systems use computing resources of idle desktop computers simultaneously in order to solve problems of scientific and engineering applications¹. This is achieved by breaking enormous problems into smaller independent units (tasks) and distributing them to some available idle desktop computers for concurrent processing²⁻⁶.

The participating desktop computers, which are also known as volunteers or hosts, have diverse characteristics in their resources such as the number of CPU cores and the amount of memory space available. Besides, the volunteers are also free to join and leave the execution of tasks without any restriction. Hence, volunteers in DG systems have different behavior in execution such as different participation time, availability and volatility. Variation of hardware and behavior of volunteers make it difficult for a resource manager and job scheduler in the

DG system to schedule tasks as well as control the allocated tasks and volunteers^{7,8}. Consequently, this has led to the delay of jobs' execution and also the degradation of the entire DG's performance.

In order to achieve the goals of resource management and job scheduling, availability of volatile volunteers is one of the key issues which need to be addressed in DG systems. Hence, a more reliable resource management and scheduling policy can be achieved by obtaining some predicted availability information of the resources involved in the DG systems⁹. To understand more about the availability of volatile volunteers in DG systems, characteristics of trace data with regards to availability would have to be analyzed¹⁰. For that, statistical analysis is one technique which can be used to analyze the data and predict availability of the volatile volunteers¹¹. The technique makes predictions by analyzing past observations of the volunteers. Hence, determination of similarities among the past observations and somehow relate them to the

*Author for correspondence

current situation are a major challenge in conducting statistical analysis on the DG systems.

Besides availability, reliability is another concern of DG scheduling. Reliability is defined as the probability that a DG system or scheduler performed its task in certain conditions during a specified period of time¹². To improve reliability of DG systems, methods such as reputation or score based scheduling can be applied¹³⁻¹⁵. The method basically chooses the most suitable volunteer (based on its reputation) to perform specific job assignment. Furthermore, resource/volunteer grouping can also be implemented as a method for improving scheduling efficiency in term of reliability^{15,16}. The method classifies heterogeneous resources into homogeneous groups based on certain criteria¹³⁻¹⁵. By having done so, jobs can be assigned to proper group for higher chances of successful execution.

This paper presents our idea of constructing groups of volunteers according to their availability score by using simple grouping method. Ultimately, each group will represent a different level of reliability in a DG system. To elaborate the idea, this paper is organized in the following manner: section 2 begins with related works, section 3 continues with the presentation of an availability scoring model, section 4 provides a technique which can be used for grouping the volunteers, section 5 presents and discusses the results of our work, and Section 6 ends the paper with conclusion and future works.

2. Related Works

In order to classify the volatile and heterogeneous volunteers in desktop grid computing system, one should study the dynamic nature of volunteers with uncertain availability. Therefore, one way to get more knowledge from volatile volunteers is by analysing trace data set in DG environment regards to volunteers' availability^{9,10}.

Choi et al.¹⁴ used the grouping method to construct groups of volunteers according to their attributes such as dedication, volatility, host availability and credibility. The authors did not elaborate other state properties such as time zone, RAM size and CPU availability interval. Some existing DG systems provide static based grouping according to disk space or operating system^{2,17-19}.

Furthermore, Neary et al.²⁰, Chakravarti et al.²¹, Zhou et al.²², Montresor et al.²³ and Zhong et al.²⁴ proposed dynamic based grouping in terms of registration time,

time zone, performance and workload irrespective availability, reputation or score, duration of participation and location which are effective on completion time, reliability and result correctness.

In conclusion, due to the volatility of DG's volunteers, both dynamic and static information of volunteers are important in order to ensure higher chance of the job being completed. This information can be combined together as hybrid based grouping which included both dynamic information of volunteers such as availability duration (historical information) and static information such as time zone, hardware capacity and so on.

3. Availability Scoring Model

Before the volatile and heterogeneous volunteers in a DG system can be assigned to various reliability groups, the criteria or factors which can be used to classify them will have to be determined first. In this study, these factors were derived from some statistical analyses which have been performed on real trace data. The real trace data set used was downloaded from <http://fta.scem.uws.edu.au>; and the data collected were related to a project named SETI@home. Since, the study was aimed to find information on availability, as mentioned in our previous published work²⁵, the trace data collected were focusing mainly on CPU availability of the volunteers involved in the project for the period of 8 months (April 1, 2007 to February 1, 2008). In the paper, we have proved using Spearman correlation method, Pearson's chi-square and Kruskal-Wallis test that the availability of a volunteer has an association and relation with some factors of the volunteer. The factors identified were average CPU availability (ACPUA), RAM size, number of processors or CPU cores, and time zone and location of the volunteers²⁵. For each of the identified factors, Frequency Occurrence of Host/volunteer (FoH) was calculated for each possible value and range of each factor. For example, out of the 38166 volunteers involved in the SETI@home project, the numbers of them which have 1 CPU core, 2 CPU cores and so on were calculated. To lessen the gaps between the resulting values, a weight fraction between 0 to 1 was assigned to each range. The value was derived by dividing the range's FoH by the total number of hosts (ToH) as shown in Equation (1).

$$\text{WeightFraction} = \frac{\text{FoH}}{\text{ToH}} \quad (1)$$

In addition, the ranges for each factor were ranked according to their execution capability. For example, a bigger RAM size would be considered better as compared to a smaller RAM size, and therefore has a higher rank. Similarly, higher ACPUA is more reliable in terms of availability as compared to a lower value. Thus, it should be ranked higher. For the ranking determination, the values were calculated using Rank Sum method as in Equation (2). The equation calculates the weight W_k for range k where n is the number of ranges²⁶.

$$W_k = \frac{n+1-k}{\sum_{i=1}^n n+1-i} \quad (2)$$

To further lead to the formulation of the availability scoring model, the derived factors above were grouped according to their dynamicity. In other words, ACPUA is considered as dynamic factor since its value changes over time. RAM size and number of CPU cores on the other hand, are fairly static, hence they were identified as static factors. Meanwhile the dynamic factor, static factor as well as location and time zone when combined together were considered as hybrid factors.

Using the dynamic factor, i.e., ACPUA value of a volunteer, the availability weight of volunteer_{*i*} at time t was calculated using Equation (3). The equation gives the ranking weight of the ACPUA interval that was extracted from the trace data of volunteer_{*i*} from time 0 until time t .

$$A_i = RWA_i(t) \quad (3)$$

Furthermore, the static factors, i.e., RAM size and CPU cores, were combined together into the calculation of computation power for a volunteer. The formula given in Equation (4) produces the ranking weight of RAM size and number of processors of volunteer_{*i*} at time t , where RWP is the ranking weight of CPU and RWR is the ranking weight of RAM size. Both values have been derived from earlier Equation (2).

$$C_i = RWP_i(t) + RWR_i(t) \quad (4)$$

The weight fraction of volunteer_{*i*} at time t is another value that needs to be calculated. The value which can be derived using Equation (5) comprises both the dynamic and static factors of the volunteer. In other words, the equation sums up all weight fractions which have been calculated using Equation (1). Hence, weight fraction of

CPU cores (WFP), weight fraction of RAM size (WFR), weight fraction of combination of time zone and location (WFE), and weight fraction of ACPUA (WFA) were all included in the formula.

$$F_i(t) = WFP_i(t) + WFR_i(t) + WFE_i(t) + WFA_i(t) \quad (5)$$

Consequently, after each of the values in figure 1 have been derived using Equation (3), Equation (4) and Equation (5) respectively, Equation (6) combines them into a mathematical hybrid equation named as Availability Scoring Model (ASM).

$$S_i(t) = A_i(t) + F_i(t) + C_i(t) \quad (6)$$

The model says that availability of volunteer_{*i*} at time t is the weighted sum of dynamic factor $A_i(t)$ with hybrid factor $F_i(t)$ and static factor $C_i(t)$. The model modifies our preliminary model which was published in²⁷.

4. Grouping Volunteers

Once the availability scoring model has been formulated, each volunteer's availability can now be predicted. Hence, resource/volunteer grouping can organize volunteers with similar weight in terms of their RAM size, number of processors, ACPUA, location and time zone into the same group. This section defines the grouping approach used to group available hosts at time t into four groups according to the availability scoring model.

In order to categorize the available volunteers, we adopted the two levels classification approach which was reported in¹⁵. In the first level, volunteers were classified into four classes according to $F_i(t)$ and $C_i(t)$ as shown in Figure 2. The class FC1 is a set of volunteers that have high computation power and high weight fraction, while the class FC2 is a set of volunteers with low computation power and high weight fraction. Meanwhile, the class FC3

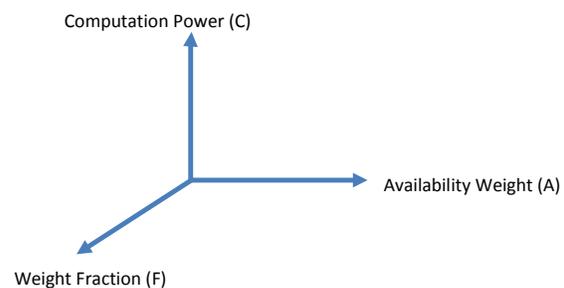


Figure 1. Dynamic, static and hybrid factors of availability scoring model in DG system.

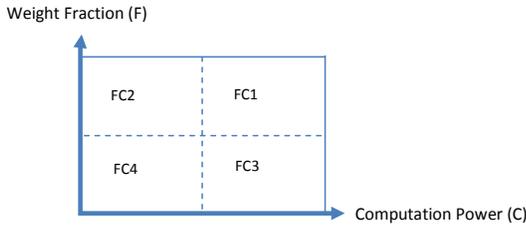


Figure 2. Volunteers grouping according to F_i and C_i .

is a set of volunteers which have high computation power and low weight fraction, and the class FC4 includes the volunteers with low computation power and low weight fraction. Figure 3 shows the grouping algorithm for first level of volunteers' classification.

In the second level, the classified volunteers are further organized into four groups according to dynamic factor $A_i(t)$ as shown in figure 4. The AFC1 group keeps volunteers with high availability fraction, high weight fraction and high computation power. Therefore, this

Classify $Host_i(t)$ in Level 1;

If ($F_i(t) == high$) then

If ($C_i(t) = high$) then

$Host_i(t) = FC1$

Else

$Host_i(t) = FC2$

End;

Else If ($C_i(t) == high$) then

$Host_i(t) = FC3$

Else

$Host_i(t) = FC4$

End; //Finish level 1

Figure 3. Algorithm for volunteers grouping construction according to F_i and C_i

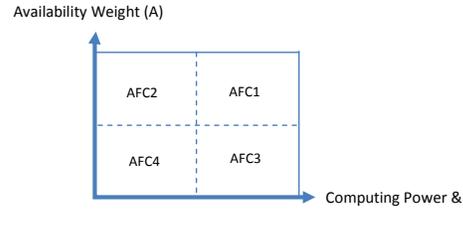


Figure 4. Volunteers grouping according to A_i , F_i and C_i .

group is considered as high reliability level group. The AFC2 group includes volunteers with low computation power, but high weight fraction and high availability weight. Hence, this group is as medium reliability level group. Furthermore, the AFC3 group involves volunteers with low availability weight and low weight fraction, but high computation power. This group is called low reliability level group. Finally, the AFC4 group comprises the volunteers with low computation power, low weight fraction and low availability weight. Therefore, this group is least reliable group and was addressed as very low reliability level group.

The ranking of the groups can be seen in Table 1 where the second column shows the rank order of each group and the third column shows the ranking weight value (RW_i) or reliability level of group. The ranking weight values were derived using the same Equation (2).

The volunteer's groups were constructed using the algorithm of grouping as shown in Figure 5. In the algorithm, we can see that the volunteers are assigned into its group according to their F, C and A scores which can be either high or low.

Furthermore, high value and low value in grouping algorithm will be derived by experimental results which will be discussed in the next section.

5. Result and Discussion

In order to carry out the volunteers' grouping construction algorithm in previous Figure 4, high, middle and low values for average availability, weight fraction and computing power should be defined first. This was accomplished through simulation and testing.

In order to simulate the grouping algorithm, the high and low bounded values for A_i , F_i and C_i were initially assumed to be the maximum and minimum values which were derived from statistical analysis that we have done in

```

Classify  $Host_i(t)$  in Level 2;
If ( $Host_i(t) \in FC1$ ) then
  If ( $A_i(t) = high$ ) then
     $Host_i(t) = AFC1$ 
  End;
Else If ( $Host_i(t) \in FC3$ ) then
  If ( $A_i(t) = low$ ) then
     $Host_i(t) = AFC3$ 
  End;
Else If ( $Host_i(t) \in FC2$ ) then
  If ( $A_i(t) = high$ ) then
     $Host_i(t) = AFC2$ 
  Else
     $Host_i(t) = AFC4$ 
  End;
End; //Finish level 2

```

Figure 5. Algorithm of volunteers grouping construction according to A_i , F_i and C_i .

our preliminary works in^{25,27}. The availability weight was assumed to be zero due to the lack of availability history in the first simulation run.

Based on our works reported in^{25,27}, the maximum value of RWP is 0.3333 and the maximum value of RWR is equal to 0.1111. Hence, using Equation (4), the maximum value for $C_i(t)$ will be 0.4444 (nearly 0.5). Moreover, maximum values of WFR, WFP, WFA and WFE are respectively: 0.4387, 0.5399, 0.6949 and 0.2127. Hence, maximum value of $F_i(t)$ will be 1.8947 when calculated using Equation (5).

Table 1. Ranking of Volunteer's Group (Reliability Level)

Group of Volunteer	Rank Order	Rank Weight (RW_i)
AFC1	1	0.4
AFC2	2	0.3
AFC3	3	0.2
AFC4	4	0.1

Furthermore, the grouping algorithm also assumed that 1.9 is the upper bound and a value which is above zero but lesser than the upper bound as the lower bound of F. As for C, the upper bound is 0.5 and the lower bound should be above zero but lesser than the upper bound. Then, middle bound values for F and C should be extracted from simulation results. The reason for this is to find the most balanced distribution of groups.

Table 2 displays different testing results of grouping by various middle bound values. These testing assumed different number of available volunteers at time t. In the first simulation run, i.e., case 1, 99 volunteers were assumed to be available in the DG system, while in the second run, i.e., case 2, 3997 volunteers were assumed available. As for the third run, i.e., case 3, 10002 volunteers were considered. In order to discover the best middle bound values for F and C, the ratio of volunteer's number in each group should be almost balanced in all of the different cases. For the simulation, the testing started with the initial middle bound for F and C to be the middle value between their upper and lower bound. The next subsequent tests decreased the middle bound of either one of F or C until one of the group was assigned zero host. In this case, another run will be done with middle of C is decreased. This is to see if a different value can make the group non-zero. If the result is still the same then the simulation will stop. Then, we will check backwards for the most balanced grouping assignment. This will help us to choose F and C values to find middle bound for both. Therefore, across three cases of simulation, the middle bound values for F and C were derived through 9 tests in case 1, 8 tests for case 2 and case 3. All cases stopped decreasing the middle value of F when the value was 0.4; also check the different two values for middle bounded of C (0.2 and 0.15). Therefore, the best results are derived by 0.15 for middle bound of C and 0.6 as middle bound of F in term of balanced groups. At this time, grouping's algorithm is finalized in order to organize volunteers in suitable group.

6. Conclusion and Future Works

This paper shows the scored-based-grouping algorithm which uses the mathematical availability scoring model based on certain factors of volunteers in order to classify them into four different groups. These groups have different reliability level. Using simulations, it can be seen that a

Table 2. Grouping results by different values of limitation for F and C criteria

		Middle bound of F	Middle bound of C	AFC1	AFC2	AFC3	AFC4
Case1	Test1	0.9	0.2	1	83	11	4
	Test2	0.7	0.2	0	80	12	7
	Test3	0.7	0.15	31	49	15	4
	Test4	0.6	0.2	1	83	11	4
	Test5	0.6	0.15	35	49	11	4
	Test6	0.5	0.2	1	86	11	1
	Test7	0.5	0.15	35	52	11	1
	Test8	0.4	0.2	3	87	9	0
	Test9	0.4	0.15	37	53	9	0
Case2	Test1	0.7	0.2	7	3202	320	468
	Test2	0.7	0.15	1099	2110	649	139
	Test3	0.6	0.2	37	3527	290	143
	Test4	0.6	0.15	1410	2154	338	95
	Test5	0.5	0.2	58	3629	269	41
	Test6	0.5	0.15	1479	2208	269	41
	Test7	0.4	0.2	100	3670	227	0
	Test8	0.4	0.15	1521	2249	227	0
Case3	Test1	0.7	0.2	16	7972	792	1222
	Test2	0.7	0.15	2789	5199	1652	362
	Test3	0.6	0.2	84	8822	724	372
	Test4	0.6	0.15	3584	5322	857	239
	Test5	0.5	0.2	135	9104	673	90
	Test6	0.5	0.15	3768	5471	673	90
	Test7	0.4	0.2	272	9194	536	0
	Test8	0.4	0.15	3905	5561	536	0

balanced distribution of volunteers can be achieved using the scoring model and the grouping approach proposed.

As future work, the reliability level grouping will be used in order to propose a reliable-job-scheduling policy in DG environment to improve performance of the DG systems.

7. References

- Balaton Z, Farkas Z, Gombas G, Kacsuk P, Lovas R, Marosi AC, et al. EDGeS: the common boundary between service and desktop grids. *Parallel Processing Letters*. 2008; 18(03):433–45.
- Fedak G, Germain C, Neri V, Cappello F. Xtremweb: A generic global computing system. 2001 Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid; IEEE; 2001.
- Sarmenta LF. *Volunteer computing*; Citeseer; 2001.
- Kondo D, Taufer M, Brooks CL, Casanova H, Chien AA. Characterizing and evaluating desktop grids: An empirical study. 18th International IEEE Proceedings of Parallel and Distributed Processing Symposium; 2004.
- Desell T, Magdon-Ismail M, Szymanski B, Varela CA, Newberg H, Anderson DP. Validating evolutionary algorithms on volunteer computing grids. *Distributed applications and interoperable systems*, Springer. 2010.
- Kan W, Masaru F, Michitaka K. Adaptive Group-Based Job Scheduling for High Performance and Reliable Volunteer Computing. 2011; 52(2):388–400.
- Choi S, Kim H, Byun E, Hwang C. A taxonomy of desktop grid systems focusing on scheduling. Department of Computer Science and Engineering, Korea University, Tech Rep KU-CSE-2006-1120-01. 2006.
- Foster I, Kesselman C. *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier; 2003.

9. Kondo D, Fedak G, Cappello F, Chien AA, Casanova H. Characterizing resource availability in enterprise desktop grids. *Future Generation Computer Systems*. 2007; 23(7):888–903.
10. Heien EM, Anderson DP, Hagihara K. Computing low latency batches with unreliable workers in volunteer computing environments. *J Grid Computing*. 2009; 7(4):501–18.
11. Lerida JL, Solsona F, Hernandez P, Gine F, Hanzich M, Conde J. State-based predictions with self-correction on Enterprise Desktop Grid environments. *Journal of Parallel and Distributed Computing*. 2013; 73(6):777–89.
12. Pham H. *Recent studies in software reliability engineering*. Springer; 2003.
13. Choi S, Baik M, Gil J, Park C, Jung S, Hwang C. Dynamic scheduling mechanism for result certification in peer to peer grid computing. *Grid and Cooperative Computing-GCC 2005*: Springer; 2005. p. 811–24.
14. Choi S, Baik M, Gil J, Jung S, Hwang C. Adaptive group scheduling mechanism using mobile agents in peer-to-peer grid computing environment. *Applied Intelligence*. 2006; 25(2):199–221.
15. Choi S, Baik M, Gil J, Park C, Jung S, Hwang C. Group-based dynamic computational replication mechanism in peer-to-peer grid computing. *2006 CCGRID 06 Sixth IEEE International Symposium on Cluster Computing and the Grid*; IEEE; 2006.
16. Venugopal S, Buyya R, Ramamohanarao K. A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Computing Surveys (CSUR)*. 2006; 38(1):3.
17. Anderson DP. Boinc: A system for public-resource computing and storage. *2004 Proceedings Fifth IEEE/ACM International Workshop on Grid Computing*; IEEE; 2004.
18. Sarmenta LF, Hirano S. Bayanihan: Building and studying web-based volunteer computing systems using Java. *Future Generation Computer Systems*. 1999; 15(5):675–86.
19. Nisan N, London S, Regev O, Camiel N. Globally distributed computation over the internet-the popcorn project. *1998 IEEE Proceedings 18th International Conference on Distributed Computing Systems*; 1998.
20. Neary MO, Christiansen BO, Cappello P, Schauer KE. Javelin: Parallel computing on the internet. *Future Generation Computer Systems*. 1999; 15(5):659–74.
21. Chakravarti AJ, Baumgartner G, Lauria M. The organic grid: self-organizing computation on a peer-to-peer network. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*. 2005; 35(3):373–84.
22. Zhou D, Lo V. Cluster computing on the fly: resource discovery in a cycle sharing peer-to-peer system. *2004 CCGrid 2004 IEEE International Symposium on Cluster Computing and the Grid*; IEEE; 2004.
23. Montresor A, Meling H, Babaoglu O. Messor: Load-balancing through a swarm of autonomous agents. *Agents and Peer-to-Peer Computing*: Springer; 2003. p. 125–37.
24. Zhong L, Wen D, Ming ZW, Peng Z. Paradropper: A general-purpose global computing environment built on peer-to-peer overlay network. *2003 IEEE Proceedings 23rd International Conference on Distributed Computing Systems Workshops*; 2003.
25. Peyvandi S, Ahmad R, Zakaria MN. Association among independent and dependent factors of host in volunteer computing. *2014 IEEE International Conference on Computer and Information Sciences (ICCOINS)*; 2014.
26. William H, Kruskal WAW. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*. 1952; 47(260):583–621.
27. Peyvandi S, Ahmad R, Zakaria Mn. Scoring model for availability of volatile hosts in volunteer computing environment. *Journal of Theoretical and Applied Information Technology*. 2014; 70(2).