Journal of Theoretical and Applied Information Technology
20th December 2014. Vol.70 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SCORING MODEL FOR AVAILABILITY OF VOLATILE HOSTS IN VOLUNTEER COMPUTING ENVIRONMENT

[1]SOODEH PEYVANDI, [2]ROHIZA AHMAD, [3] M.NORDIN ZAKARIA

[1, 3] HPC CENTER, Department of Computer & Information Sciences, Universiti Teknologi PETRONAS, MALAYSIA

[2]Department of Computer & Information Sciences, Universiti Teknologi PETRONAS, MALAYSIA

E-mail: [1]soodehpeyvandi@gmail.com, [2] rohiza_ahmad@petronas.com.my, [3]Nordinzakaria@petronas.com.my

ABSTRACT

Volunteer Computing systems (VC) use the idle computing resources which are composed of unreliable volunteers whose exhibit volatile behavior and often they are not available because of the autonomy nature of owners. Scientific experiments that use internet-connected computers are internet based VC projects. However, the volatility of volunteer is significant challenge in VC and it is effective on applications that are using these types of hosts. Therefore, investigating resource characterization in order to derive efficient factors of host's availability is needed to improve VC systems. Factors analysis of hosts is fruitful in making better decision for job scheduler in order to improve system's performance. The purpose of this paper is determination the availability score model for volatile hosts according to statistical analysis of host's factors from actual occurrence trace data set. This study is proved the relationship across host's factors then it is proposed Scoring Availability Model.

Keywords: *volunteer computing, resource characterization, CPU availability, scoring model*

1. INTRODUCTION

In recent years, there is growing trend among the scientific societies to use grid computing infrastructures to solve their principal problems, especially in projects that require large computing resources [1]. Grid computing is a type of distributed systems which is often built as a federation of computer systems called hosts [2]. The hosts can be from the same or different domain of management and can have their own kind of software and hardware as well network technology and infrastructure. The goal of grid systems is mainly to let each of the hosts in the grid to share their resources. Each host can offer resources for a specified grid, and each project can dynamically claim specific resources from the grid when needed.

Generally, there are two taxonomy trends observable in the development of grid systems: Service Grids (SG) and Desktop Grids (DG) [3]. In the SG, developers construct a grid service that can be accessible to a large number of hosts; and each host's resources can be made as components of the grid through a grid middle-ware. The structure and operations of a grid middle-ware are usually complex, hence, requires an expert for handling and maintaining them. Due to that, individual or public hosts do not mostly donate their resources for SGs. Instead, SGs are more popular and suitable for institutions and enterprises where there are expert administrators available to attend the software and hardware issues of the grid environment [3]. On the contrary, DG is more for the public hosts. DG refers to Volunteer Computing (VC) system or Public Resource Computing (PRC) system, which relies on the common public resources [2], [3]. Unlike SGs, hosts in DGs have simple architecture, heterogeneous computing resources that are successful in scale extension of idle desktop computers around the world [3].

SGs and DGs have several differences. Firstly, they differ in terms of the service initiation. In SGs, a job submission or a service invocation is the starter activity to acquire a grid resource. Job submission can be considered as a push model where the service requester pushes jobs on resources and resources become active and run jobs. However, DGs work according to the pull model that resources pull tasks or jobs from the application repository which is typically placed in the DGserver. In this way, resources initiate their own activity based on the job pulled from the server [3]. Secondly, SGs and DGs

Journal of Theoretical and Applied Information Technology
20th December 2014. Vol.70 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

differ in terms of the resources that they have. For SGs, the resources might be supercomputers and PCs owned by universities, clusters, research labs, and institutes in SG. These resources are powered on most of the time, and connected to the grid by full-time with high-bandwidth network connections [3], [4]. As for DGs, their resources come from hosts or volunteers who are using personal computers which are connected to the internet using DSL cable, modems or telephone line. Moreover, the computers are frequently turned off or disconnected from the internet. Furthermore, the hosts are typically involved in a small research group and have limited computer expertise; and they are in a project only if they are interested in [3], [4]. In other words, resources of DGs are volatile. From now onwards, VC will be used interchangeably in place of DG.

The first development of VC was for a project named Distributed.net in 1995. The project used VC for acquiring resources of hosts to do scientific supercomputing tasks [5]. After that, several other projects started to follow the same step for acquiring computing power. Among the prominent one was a project named SETI@home. SETI@home was launched in 1999 and it is still an ongoing project until today. According to Anderson (2004), the number of volunteers involved in the project was constantly growing and it was predicted to reach 1 billion in 2015. This scientific project uses computers from all over the globe in order to analyses radio signals from the outer space [6].

Generally, in VC, majority of the time, not all of the hosts are in use. Hence, many idle computing cycles are available with the hosts. Due to this, one of the objectives of VC is to achieve high throughput computing by utilizing the idle computing cycles available at the volatile volunteers [7]. This is necessary since volunteers are not guaranteed to stay long in VC. The owner of resources has the right to make decision that when to donate idle resources. Therefore these resources are volatile and are unavailable without confirmation at any time [8]. Thus, performing a job as fast as possible using as many volunteers as necessary, is the way to go in VC.

The uncertainty of resources availability in VC can be caused by several factors. One of them could be due to host application patterns, or defective hardware or software [9]. Whatever the reasons might be for the unavailability of resources, this condition impacts the running of jobs. If resources become unavailable suddenly, running jobs will fail to complete. This will reduce job throughput and increase application completion time (makespan) [8]. Hence, resources availability has significant influence on quality of service metrics such as performance. Therefore, it is crucial to use available resources to run distributed and parallel algorithms (tasks) of projects whenever possible [10].

To support the above, job scheduling is important component in VC systems. The scheduler is responsible to assign jobs to volatile volunteers. The scheduling policy is designed to optimize performance requirements. Also, the heterogeneous nature of volunteers (desktops with various resources: CPU, memory, disk space, etc.) have impact on performance of job scheduling.

We believe that, to better cope with volatility and heterogeneous nature of volunteer, hybrid of group-based and reputation-based job scheduling policy can be efficient. This policy will group hosts according to resources' availability values of hosts that is derived by availability scoring model. The hybrid Job scheduling policy will improve performance. However, in order to propose a job scheduler as such, the availability score model must be extracted. The factors which influenced availability score will have to be identified first. Hence, statistical analysis of hosts will be considered in order to extract efficient factors of host in availability degree. This research will provide answers to the following questions:

1. How to analyze hosts factors?
2. Which factors of hosts will be efficient?
3. How to give availability score to each host in VC?

Our goal is to find correlation and relation across efficient factors of volunteers that will be used to develop "Scoring Availability Model" based on statistical analysis. In particular, the objectives of this part of the study are: discover correlation between efficient factors; show relationship among factors; and, propose Scoring Model of availability for individual host. This paper is organized as fallow: section 2 discusses on related work, section 3 presents proposed methodology, and section 4 will propose "Scoring Availability Model" as result of methodology. Finally, section 5 is conclusion and future works.

2. RELATED WORKS

Statistical analysis real trace data set in regards to volunteers' availability and consideration on the characteristics of these volatile volunteers is the way to understand more about them in VC systems [11]. The term availability has many different meanings in

Journal of Theoretical and Applied Information Technology
20th December 2014. Vol.70 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

VC systems [12]. Kondo et.al [13], categorized availability into: host availability, CPU availability and task execution availability. Host availability shows whether a host is reachable or not [12, 14-16]. CPU availability is a percentage value of fraction that CPU can be used by other processes [17-20]. Whereas, task execution availability defines a value which is refer to the status of whether a task can be executed on the volunteers or not according to the VC's host requirement policy [13].

This paper is different from [14, 15, 21and 22], in terms of scale (number of hosts) where they have focused on a few hosts. Also, we investigate hosts from university, office, home, school against [15, 23] and [22] that focused on university and office only. Furthermore, this research has difference with [14, 24, 25], [16] and [12] in the type of availability whereas we will investigate CPU availability.

Besides, in terms of modeling some works focused on internet network model; such as [26], [27] and [28] have investigated the internet of hosts with exclusive of host's resources. Authors in [29], [30] and [31] have focused on modeling residential broadband networks without considering on hardware resources. Faloutsos et.al [26] provided novel vision of internet topology and discovered three power laws with high correlation coefficients.

Researches presented in [24], [16] have focused on application of network traffic, topology and its behavior in P2P while they did not mention to the hardware measurement. Several researches such as [32], [33], [34] and [23] investigated modeling clusters or grids while these researches differ from our research in host heterogeneity and effective factors. Also, Javadi et.al [35] described effective method for discovering hosts with similar statistical availability and modeled with similar probability distributions which they focused only on CPU availability regardless of other factors of hosts. However, as defined by Anderson et al [36], the successful job processing and resources are meaningless in isolation. For example, even if the task has been assigned to the host only needs to CPU, but some quota of RAM size have to be assigned in order to job execution. Furthermore, Heien et al [37] presented that there is strong correlation between RAM size and CPU core's number.

Kondo et al [38] specified hosts from home are powered-on for short time than those from office or enterprise, however, when host from home are powered-on, machine will be idler (more idle). This means that hosts have different availability patterns in various regions of the world.

3. PROPOSED METHODOLOGY

As mentioned in section 1, due to the volatility of VC's hosts, the availability information of the hosts is important in order to ensure a higher chance of a job to be completed. Thus, in this paper, our method of getting the availability score of the hosts is shared and discussed. Statistical approach is used to implement our method. For that, the research methodology involves data collection and some statistical tests before a scoring formula is able to be formulated.

Figure 1 shows the research methodology, which defines sequence activities. The first step shows characterizations of data set and derived availability factors; second step defines correlation between availability factors of hosts then third phase investigates relationship across availability factors and finally, the forth one proposes scoring model of availability for hosts. In order to do statistical analysis, we use hypothesis testing method by IBM SPSS Statistics 19.

3.1. Characterization

Trace data set on VC environment is needed in order to perform our statistical method. Once the secondary data set is acquired, we obtain trace data set from VC environment, and then in second step, analyze trace data set. Third step includes the clean trace data set and in the last, the step forth is shown descriptive statistics trace data.

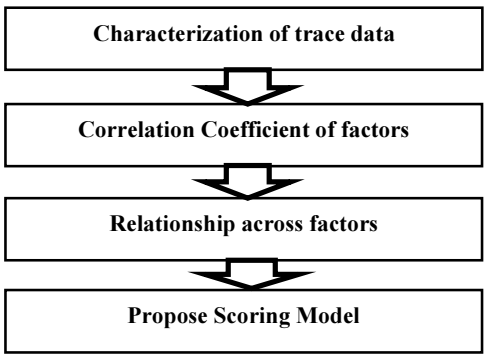

Figure1. Outline of proposed methodology

3.1.1 Obtaining trace data set

Secondary trace data set of a well-known VC project called SETI@home is used in this study. The data set was originally collected by a researcher named Anderson [36] at the Berkeley Open Infrastructure for Network Computing (BOINC) server of SETI@home. The data set is downloaded from a centralized public repository website,

Journal of Theoretical and Applied Information Technology
20th December 2014. Vol.70 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

http://fta.scem.uws.edu.au. The data set contains trace data which were captured between the months of April 1, 2007 and February 1, 2008 of the project. The 10 months data collection contains trace data of 60,883 volunteer hosts from all over the whole.

3.1.2 Analysis of trace data set

Some information is extracted from SETI@home data set for each host. However, for this study, only a few of them, which we hypothesized as having some impacts on the availability of the hosts, are selected. The information or we called them factors, can be categorized into two types: static and dynamic. The static factors are number of processors, RAM size (GB), time zone (GMT) where the host is located and location of the host, i.e., office, home, etc.

As for the dynamic factors, they are CPU availability or unavailability at certain period of time (from now on we refer as event type), and start time and end time of the event type (epoch time). From the epoch time factor, the total duration of CPU availability interval (CPUA) is obtained for each host. In addition, the total number of times or the frequency of host being available is also obtained from the data set. The above total is referred to as Frequency of Event (FE). Using CPUA and FE, Average CPU Availability (ACPUA) is calculated by dividing the two items.

3.1.3 Clean trace data set

In this step, missing values such as null values for location, 0 or negative values for RAM size, null values in time zone and other components are removed from the data set. After completion of the cleaning process, only half of the 60,883 hosts, to be exact 38,166 of them, were having complete data.

3.1.4 Statistical analysis

While step 2 analysis is done to individual hosts, this step analyses or summarizes each of the factors for all hosts. First, for each factor, as an example, host location, each possible value is listed out. Second, an identifier is assigned to each of them. Third, for each different value, the number of hosts which have that value is calculated. This step is referred to as Frequency Occurrence of Host (FOH) for each possible value. Forth, a weight between 0 and 1 (inclusive) is calculated for that value. The weight value is derived by dividing FOH to total number of hosts which is defined weight fraction as ratio of the one component to total (equation (1)).

Weight Fraction = FOH / total number of hosts (1)

Table 1 shows the results of performing the above steps to host location. The last column shows the Weight Fraction of Location (WFL). Table 2 presents the results of similar descriptive analysis for time zone. As for RAM size, number of processors and ACPUA, due to the diverse actual values possible for them, the values are grouped into ranges. Column one of Table 3, Table 4 and Table 5 show the ranges used for RAM size, number of processors and ACPUA respectively.

Another column is added to the Tables 3, 4 and 5, which we called Weight Rank. The Weight Rank of RAM size (WRR) column in table 3 shows the value of weight rank for RAM size range that is calculated by Rank Sum method. Similarly, the WRP column of Table 4 defines the Weight Rank of Processor's number range and the WRA column of Table 5 shows the Weight Rank of ACPUA range. The Rank Sum is a method of Ranking for weight assessment. Rank Sum method generates numerical weights from a rank order of each range by equation (2); this formula calculates the weight W_k for range k where n is the number of ranges [42].

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

The weights fit the rank order of ranges ($w1 \geq w2 \geq \ldots \geq wn \geq 0$); also all weights are non-negative and add up to 1. Hence, WRP in order to value of Weight Rank for Processors number and lastly, WRA as value Weight Rank for ACPUA. In order to generate weights, it is needed to rank order ranges from Table 3, Table 4 and Table 5 hence can use from second column but assign identifier to each ranges based on rank order of ranges. Assign high rank which started from 1 to high RAM size by descended order of RAM size in Table 3, also did same concept for number of processors at table 4 and ACPUA in Table 5. Investigated sample by Table 5 presents average availability of 69.5% of the hosts are below 1000000 epoch time in the sample period.

Table 1.Frequency occurrence of host's CPUA at each location

Location	IL	FOH	WFL
Home	1	29443	0.7714
Work	2	7296	0.1911
School	3	1360	0.0356
Other	4	67	0.0017
Total	1	38166	1

Table 2.Frequency occurrence of host's CPUA at each time zone

Time zone	IT	FOH	WFT
-8.00	1	3509	0.0919
-7.00	2	1515	0.0396

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

-6.00	3	4614	0.1208
-5.00	4	8231	0.2151
.00	5	3010	0.0788
1.00	6	11179	0.2929
2.00	7	1365	0.0357
3.00	8	274	0.0071
8.00	9	806	0.0211
9.00	10	1585	0.0415
10.00	11	299	0.0078
11.00	12	720	0.0188
Other	13	1059	0.0277
Total		38166	1

Note: Identifier of Time zone (IT)
Weight Fraction of Time zone (WFT)

Table 3.Analysis CPUA of host with different RAM size

Range of RAM size (GB)	IR	FOH	WFR	RWR
18 <= size	1	10	0.7940	0.01818
16 <= size < 18	2	10	0.1752	0.03636
14<= size < 16	3	28	0.0180	0.05454
12 <= size < 14	4	16	0.0071	0.07272
10<= size < 12	5	35	0.0030	0.09090
8 <= size < 10	6	115	0.0009	0.10909
6 <= size < 8	7	273	0.0004	0.12727
4 <=size < 6	8	688	0.0007	0.14545
2 <= size < 4	9	6687	0.0002	0.16363
0 < size < 2	10	30304	0.0002	0.18181
total		38166	1	1

Note: Identifier of RAM size and Rank Order of Range (IR)
Weight Fraction of RAM size (WFR)
Ranking Weight of RAM size (RWR)

Table4.Analysis CPUA with different processors number

Number of Processors	IP	FOH	WFP	RWP
18<=	1	3	0	0.11180
17	2	0	0	0.10559
16	3	19	0.0004	0.09937
15	4	0	0	0.09316
14	5	0	0	0.08695
13	6	0	0	0.08074
12	7	0	0	0.07453
11	8	0	0	0.06832
10	9	0	0	0.06211
9	10	0	0	0.05590
8	11	302	0.0079	0.04968
7	12	0	0	0.04347
6	13	0	0	0.03726
5	14	0	0	0.03105
4	15	2772	0.0726	0.02484
3	16	19	0.0004	0.01863
2	17	20609	0.5399	0.01242
1	18	14442	0.3783	0.00621
Total		38166	1	1

Note: Identifier of #Processors and Rank Order (IP)
Weight Fraction of #Processors (WFP)
Ranking Weight of #Processors (RWP)

Table 5.Analysis host with different ACPUA

ACPUA (100000 epoch time)	IA	FOH	WFA	RWA
10 <=ACPUA	1	183	0.0047	0.1666
9 <=ACPUA < 10	2	65	0.0017	0.15151
8<=ACPUA < 9	3	123	0.0032	0.13636
7<=ACPUA < 8	4	0	0.0000	0.12121
6 <=ACPUA < 7	5	269	0.0070	0.10606
5 <=ACPUA < 6	6	475	0.0124	0.09090
4 <=ACPUA < 5	7	781	0.0204	0.07575
3 <=ACPUA < 4	8	1356	0.0355	0.06060
2 <=ACPUA < 3	9	2859	0.0749	0.04545
1 <=ACPUA < 2	10	5531	0.1449	0.03030
0 <ACPUA<1	11	26524	0.6949	0.01515
Total		38166	1	1

Note: Identifier of ACPUA (IA)
Weight Fraction of ACPUA (WFA)
Ranking Weight of ACPUA (RWA)

3.2 Correlation

Correlation or associations between the factors were found using correlation analysis test. As a result, it is found that ACPUA has small negative correlation with number of processors and RAM size, whereas the RAM size has strong correlation with number of processors. Table 6 shows correlation matrix of host's resources that includes r-value or Spearman rho's for correlation among columns and rows are shown in the entries. Our analysis defined that the correlation coefficient is significant at the 0.01 level and p-values are less than 0.01.

Moreover, it is proofed that time zone and location of host have significant correlation which is gained by Chi-square test. The Chi-square test shows different number of hosts in term of CPUA at each time zone with various locations. Table 7 demonstrates results of Chi-square test and shows that +1 GMT zone includes maximum number of hosts which are located at home and office respectively 8445, 2523 hosts. Also, based on frequency of host those are located in different time zones and location, Table 7 shows Weight Fraction for Location and Time zones (WFLT) in matrix with 13*4 values (13 different values for IT and 4 different values for IL). Location and time zone are not tested for correlation to others factors due to their nominal factors. To observe their relation, another method is used and will be explained in the following.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6.Spearman correlation coefficient between scale factors of hosts

	ACPUA	RAM size	Number of Processors
ACPUA	1.000	-.088	-.047
RAM size	-.088	1.000	.598
Number of Processors	-.047	.598	1.000

Table 7.Frequency occurrence of host in different location and time zone

IT		IL				Total
		1	**2**	**3**	**4**	
1	FOH	2886	569	48	6	3509
	WFTL	0.0980	0.0779	0.0352	0.0895	0.3006
2	FOH	1232	225	56	2	1515
	WFTL	0.0418	0.0308	0.0411	0.0298	0.1435
3	FOH	3449	768	385	12	4614
	WFTL	0.1171	0.1052	0.2830	0.1791	0.6844
4	FOH	6750	1271	201	9	8231
	WFTL	0.2292	0.1742	0.1477	0.1343	0.6854
5	FOH	2368	589	48	5	3010
	WFTL	0.0804	0.0807	0.0352	0.0746	0.2709
6	FOH	8445	2523	185	26	11179
	WFTL	0.2868	0.3458	0.1360	0.3880	1.1566
7	FOH	939	409	16	1	1365
	WFTL	0.0318	0.0560	0.0117	0.0149	0.1144
8	FOH	197	64	13	0	274
	WFTL	0.0066	0.0087	0.0095	0.0000	0.0248
9	FOH	460	146	199	1	806
	WFTL	0.0156	0.0200	0.1463	0.0149	0.1968
10	FOH	1199	346	36	4	1585
	WFTL	0.0117	0.0474	0.0264	0.0597	0.1452
11	FOH	249	48	2	0	299
	WFTL	0.0084	0.0065	0.0014	0.0000	0.0163
12	FOH	472	88	159	1	720
	WFTL	0.0160	0.0120	0.1169	0.0149	0.1598
13	FOH	797	250	12	0	1059
	WFTL	0.027	0.0342	0.0882	0.0000	0.1494
To tal	FOH	29443	7296	1360	67	38166
	WFTL	1	1	1	1	4

Note: Weight Fraction for Location and Time zones (WFLT)

3.3 Relationship

By observing the relation across scale variables (ACPUA, RAM size and number of processors) and nominal variables (time zone, location), based on [41] we would use Kruskall-Wallis. The Kruall-Wallis test is a non-parametric method of analysis of variance by ranks and specifies diversity of group [42]. In this study, we compare rank of ACPUA, RAM size and number of processors in different time zones and locations in order to show differences of factor's values in groups of location and time zones. Our hypotheses are displayed in following:

1. Null Hypothesis: There is no difference in ACPUA according to host's location.

Alternative Hypothesis: Hosts from different locations have a different ACPUA if p-value is less than 0.05, so should to reject null hypothesis.

2. Null Hypothesis: There is no difference in RAM size according to host location.

Alternative Hypothesis: Hosts from different locations have a different RAM size if p-value is less than 0.05, so should to reject null hypothesis.

3. Null Hypothesis: There is no difference in number of processors according to host location.

Alternative Hypothesis: Hosts from different locations have a different number of processors if p-value is less than 0.05, so should to reject null hypothesis.

4. Null Hypothesis: There is no difference in ACPUA according to host time zone.

Alternative Hypothesis: Hosts from different locations have a different ACPUA if p-value is less than 0.05, so should to reject null hypothesis.

5. Null Hypothesis: There is no difference in RAM size according to host time zone.

Alternative Hypothesis: Hosts from different locations have a different RAM size if p-value is less than 0.05, so should to reject null hypothesis.

6. Null Hypothesis: There is no difference in number of processors according to host's time zone.

Alternative Hypothesis: Hosts from different locations have a different number of processors if p-value is less than 0.05, so should to reject null hypothesis.

The result of applying Kruskall-Wallis test is shown in Table 8, this table defines mean rank of all scale variables (factors) at each location. The means rank for different time zones are illustrated at Table 9. In Kruskal-Wallis test, entire data from all groups have to be ranked, mean rank of groups should be calculated and then assign mean rank to each group.

Table 10 defines the Chi-square value (Kruskall-Wallis H), degree of freedom (df) and significance level for scale factors and location. In addition, Table 11 specifies Chi-square value, df and p-value from Kruskal-Wallis test of scale factors and time zone.

From Table 8 and Table 10, it is concluded that there is statistically significant difference of ACPUA (H (3) = 44.136, p-value= 0.001) among the different groups of location, with a mean rank of 18885.46 for home, 19490.06 for school, 19815.90 for office and 18102.30 for other locations. Also, there are significant differences in RAM size and number of processors at various groups of locations. Similarly, through Table 9 and Table 11, there are significant

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

differences of ACPUA, RAM size and number of processors across various time zones and p-values are less than 0.05. Therefore, for all of hypotheses should to reject null hypothesis and accept alternative hypothesis.

4. RESULT & DISCUSSION

In this section as a result of statistical analysis of hosts, we can propose "Scoring Availability Model". The correlated criteria of hosts and relationship across them in term of CPU availability led to the propose scoring model of CPU availability at each host. As mentioned on proposed methodology, after proved the correlation among factors in term of CPU availability at host and also proving relation across those factors, now we can propose our scoring model in order to give CPU availability weight to each host.

Table 8.Ranks of factors for each location

IL	Number of Host	Mean Rank for ACPUA	Mean Rank for RAM size	Mean Rank for Number of Processors
1	29443	18885.46	19238.15	18857.52
2	7296	19815.90	18743.81	19658.19
3	1360	19490.06	17472.59	20857.46
4	67	18102.30	20814.23	19802.34
Total	38166			

Table 9.Ranks of factors for each time zone

IT	Number of Host	Mean Rank for ACPUA	Mean Rank for RAM size	Mean Rank for Number of Processors
1	3509	20769.34	20513.08	19609.42
2	1515	21425.80	20616.39	19500.42
3	4614	21148.69	18728.25	18500.35
4	8231	20677.37	19109.47	18675.00
5	3010	17923.58	19569.97	19703.87
6	11179	17066.08	19139.17	19132.56
7	1365	17714.36	17941.06	19079.36
8	274	16092.71	16040.61	18075.17
9	806	15427.27	16731.81	20840.57
10	1585	19045.48	19136.31	19337.24
11	299	19905.59	17697.08	19615.68
12	720	21284.39	17707.52	19834.87
13	1059	17003.36	16826.79	18067.04
Total	38166			

Table 10.Kruskal-wallis test for location

	ACPUA	RAM size	Number of Processors
Chi-square	44.136	44.652	86.015
df	3	3	3
P-value	0.000	.000	.000

Table 11 Kruskal-wallis test for time zone

	ACPUA	RAM size	Number of Processors
Chi-square	1091.964	238.477	102.622
df	12	12	12
P-value	.000	.000	.000

In section 3.1 is shown descriptive statistics of CPUA and frequency occurrence of hosts in different criteria and assigned weight to various criteria between 0 and 1. These weights are shown by WFP, WFR, WFA and WFLT. In addition to statistical criteria, values of factors have their weight that is concluded from comparing values regardless of host's frequency occurrence. These weights refer to values of RAM size, number of processors and ACPUA those are scale variables, also these weights are in [0,1] range respectively WRP,WRR and WRA.

Equation (3) shows availability Scoring model through Weight Fraction of host's frequency occurrence (SWF) in $host_i$ based on weighted sum:

$$Score_F(Host_i) = WFP_i + WFR_i + WFA_i + WFLT_i \quad (3)$$

Equation (4) defines availability Scoring model via Weight Rank of scale values (SWR) for $host_i$ based on weighted sum:

$$Score_R(Host_i) = WRP_i + WRR_i + WRA_i \quad (4)$$

Equation (5) is the proposed Scoring Availability Model (SAM) of Volatile Host in volunteer environment which is calculated by Weight Fraction and Weight Rank Score based weighted sum:

$$Score_{FR}(Host_i) = (WFP_i + WRP_i)/2 + (WFR_i + WRR_i)/2 + (WFA_i + WRA_i)/2 + WFLT_i \quad (5)$$

After proposing our Scoring Availability Model for hosts in VC systems, we have applied equation (3), (4) and (5) to five hosts of our samples and results are shown in Table 12. This table defines availability score in last columns.

Table 12.Availability Score of 5 Hosts

Host's Number	SWF	SWR	availability score
100021662	1.5259	0.07603	0.92014
100037624	2.3156	0.04575	1.32443
100064880	2.1092	0.0421	1.11803
100101910	1.4788	0.0609	0.81201
100139133	2.258	0.0421	1.26683

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSION & FUTURE WORKS

We analyzed CPU interval availability trace data that achieved from http://fta.scem.uws.edu.au. Characterized trace data set and found statistically analysis result in term of independent and dependent factors frequency occurrence. Then we defined null and alternative hypothesis, next consider significance coefficient of correlation, relation across factors is investigated and finally availability score model for hosts in volunteer computing environment is proposed. As future works, for classification hosts in suitable group based on their availability score, will define five levels of availability (Very low, Low, Medium, Large and Very large). Then, we will extend proposed scoring availability model and groups of hosts based on availability score to scheduling algorithm in order to investigate the effect of model on performance..

REFERENCES:

[1] Foster, I., Et Al," Cloud Computing and Grid Computing 360-Degree Compared", *In Grid Computing Environments Workshop*, 2008.

[2] Fran Berman, G.F., Anthony J.G. Hey, "Grid Computing: Making the Global Infrastructure a Reality", 2003.

[3] Abbas, A., "Grid Computing: A Practical Guide to Technology and Applications" Inc., Rockland, Ma, Ed. C.R. Media, 2003.

[4] Anderson, D.P., "BOINC: A System for Public-Resource Computing and Storage", *In Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, IEEE Computer Society, 2004, pp. 4-10.

[5] Anderson, D.P., Et Al., "SETI@home: An Experiment in Public-Resource Computing", Commun ACM.45 (11), 2002, pp. 56-61.

[6] Naseera, S. And K.M. Murthy, "Prediction Based Job Scheduling Strategy For A Volunteer Desktop Grid", *In Advances In Computing, Communication, And Control*, Springer, 2013, pp. 25-38.

[7] Araujo, F., Et Al., "Edges: The Common Boundary between Service and Desktop Grids", Parallel Processing Letters. 18(03), 2008, pp. 433-445.

[8] I. Foster, C.K., "Globus: A Met computing Infrastructure Toolkit", Supercomputer Applications. 11(2), 1997, pp. 115-128.

[9] Andrzejak, A., D. Kondo, And D. Anderson, "Ensuring Collective Availability In Volatile Resource Pools Via Forecasting, In Managing Large-Scale Service Deployment", F. Turck, W. Kellerer, And G. Kormentzas, Editors. Springer Berlin Heidelberg, 2008, pp. 149-161.

[10] Zhang, J. And C. Phillips, "Job-Scheduling Via Resource Availability Prediction for Volunteer Computational Grids", Int. J. Grid Util. Compute. 2(1), 2011, pp. 25-32.

[11] Heien, E.M., D.P. Anderson, and K. Hagihara, "Computing Low Latency Batches With Unreliable Workers in Volunteer Computing Environments", Journal of Grid Computing. 7(4), 2009, pp. 501-518.

[12] Bhagwan, R., S. Savage, and G.M. Voelker, "Understanding Availability, In Peer-To-Peer Systems", Ii. Springer, 2003, pp. 256-267.

[13] Kondo, D., Et Al., "Characterizing Resource Availability in Enterprise Desktop Grids", Future Generation Computer Systems. 23(7): P. 888-903. (2007)

[14] Acharya, A., G. Edjlali, and J. Saltz, "The Utility of Exploiting Idle Workstations for Parallel Computation", In ACM Sigmetrics Performance Evaluation Review, 1997.

[15] Bolosky, W.J., Et Al, "Feasibility of a Server less Distributed File System Deployed On an Existing Set of Desktop Pcs", In ACM Sigmetrics Performance Evaluation Review, 2000.

[16] Saroiu, S., P.K. Gummadi, And S.D. Gribble, "Measurement Study of Peer-To-Peer File Sharing Systems", In Electronic Imaging. International Society for Optics and Photonics, 2002.

[17] Arpaci, R.H., Et Al., "The Interaction of Parallel and Sequential Workloads on A Network of Workstations", ACM. Vol. 23, 1995.

[18] Casanova, H., Et Al, "Heuristics for Scheduling Parameter Sweep Applications In Grid Environments", *In Heterogeneous Computing Workshop*, 2000. (Hcw 2000) Proceedings. 9th. IEEE, 2000.

[19] Dinda, P.A., "The Statistical Properties of Host Load", Scientific Programming. 7(3), 1999, pp. 211-229.

[20] Wolski, R., N. Spring, and J. Hayes, "Predicting The CPU Availability Of Time-Shared Unix Systems On The Computational Grid", *In High Performance Distributed Computing*, Proceedings. The Eighth International Symposium On. IEEE, 1999.

[21] Mutka, M.W. And M. Livny, The Available Capacity Of A Privately Owned Workstation Environment. Performance Evaluation. 12(4), 1991, pp. 269-284.

[22] Bolosky, W.J., Et Al, "Feasibility of a Server less Distributed File System Deployed On an Existing Set of Desktop Pcs", In ACM

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Sigmetrics Performance Evaluation Review, 2000.

[23] Kondo, D., Et Al., "Characterizing and Evaluating Desktop Grids: An Empirical Study", *In Parallel and Distributed Processing Symposium*, Proceedings, 18th International. IEEE, 2004.

[24] Chu, J.C., K.S. Labonte, and B.N. Levine, "Availability and Locality Measurements of Peer-To-Peer File Systems", *In ITCOM 2002: The Convergence of Information Technologies and Communications*. International Society for Optics and Photonics, 2002.

[25] Stutzbach, D. And R. Rejaie, "Understanding Churn in Peer-To-Peer Networks", In Proceedings of the 6th ACM Sigcomm Conference on Internet Measurement. ACM, 2006.

[26] Faloutsos, M., P. Faloutsos, And C. Faloutsos, "On Power-Law Relationships of the Internet Topology", Sigcomm Comput. Commun. Rev. 29(4), 1999, pp. 251-262.

[27] Floyd, S. And E. Kohler, "Internet Research Needs Better Models", Sigcomm Comput. Commun. Rev. 33(1), 2003, pp. 29-34.

[28] Caida,"The Cooperative Association for Internet Data Analysis", Available From: Http://Www.Caida.Org/Home/, 1996.

[29] Simpson, C., Jr. And G. Riley, "Neti@Home: A Distributed Approach To Collecting End-To-End Network Performance Measurements", In Passive And Active Network Measurement, C. Barakat And I. Pratt, Editors. Springer Berlin Heidelberg, 2004, pp. 168-174.

[30] Dischinger, M., Et Al,"Characterizing Residential Broadband Networks", In Internet Measurement Conference, 2007.

[31] Shavitt, Y. And E. Shir, Dimes," Let the Internet Measure Itself", Sigcomm Comput. Commun. Rev. 35(5), 2005, pp. 71-74.

[32] Sulistio, A., Et Al., "A Toolkit for Modeling and Simulating Data Grids: An Extension to Gridsim", Concurrency and Computation: Practice and Experience. 20(13), 2008, pp. 1591-1609.

[33] Lu, D. and P.A. Dinda, "Synthesizing Realistic Computational Grids", In Proceedings of the ACM/IEEE Conference on Supercomputing, 2003.

[34] Kee, Y.-S., H. Casanova, and A.A. Chien, "Realistic Modeling and Synthesis of Resources for Computational Grids', *In Proceedings of The ACM/IEEE Conference On Supercomputing*, IEEE Computer Society, 2004, pp. 54.

[35] Bahman, J., "Discovering Statistical Models of Availability in Large Distributed Systems: An Empirical Study of SETI@home", IEEE Transactions on Parallel and Distributed Systems. 22(11), 2011, pp. 1896-1903.

[36] Anderson, D.P. and G. Fedak, "The Computational and Storage Potential of Volunteer Computing", *In Cluster Computing and Grid International Symposium*. CCGRID 06. Sixth IEEE, 2006.

[37] Heien, E.M., D. Kondo, and D.P. Anderson, "a Correlated Resource Model of Internet End Hosts", *In Parallel and Distributed Systems*, IEEE Transactions. 23(6), 2012, pp. 977-984.

[38] Kondo, D., A. Andrzejak, and D.P. Anderson, "On Correlated Availability in Internet-Distributed Systems", In Proceedings of the 9th IEEE/ACM International Conference On Grid Computing, IEEE Computer Society, 2008, pp. 276-283.

[39] Javadi, B., Et Al., "the Failure Trace Archive: Enabling the Comparison Of Failure Measurements And Models Of Distributed Systems", In Journal of Parallel and Distributed Computing, 2013.

[40] Papadimitriou, C.H., "Computational Complexity", John Wiley and Sons Ltd, 2003.

[41] Slate, J.R. and A. Rojas-Lebouef, "Calculating Basic Statistical Procedures in SPSS: A Self-Help and Practical Guide to Preparing Theses, Dissertations, and Manuscripts", P. 152.

[42] William H. Kruskal, W.A.W., "Use of Ranks in One-Criterion Variance Analysis", In Journal Of The American Statistical Association, 47(260), 1952, pp. 583–621.