International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 22571-22588

http://www.ripublication.com

Reliable Grid Scheduler With Failure Prediction Using System Logs

S.Sathyalakshmi[1]
Hindustan University, slakshmi@hindustanuniv.ac.in
C.Rajathi[2]
Hindustan University, rajathimeenac767@gmail.com
P.Narayanasamy[3]
Anna University, sam@annauniv.edu
S.Ramamoorthy[4]
Dr. M. G. R. University, srm240759@yahoo.com

Abstract

Grid computing is highly heterogeneous in nature. The availability of resources is dynamic and they may not be available due to failure. Allotted tasks of a job may be disrupted due to failure of the resources. In order to ensure the completion of the job, the tasks allotted to the failed resources have to be reallocated to other available resources. Rescheduling of tasks increases the total makespan of the job and in turn leads to performance degradation of the Grid service. If the resource failure is predicted before allocating the task to that resource, the task may not be allocated to it. System logs are used to predict the failures of the resources. For scheduling, a Genetic Algorithm based scheduling technique is proposed along with Support Vector machine(SVM) based predictor to predict the resource failure. The results show the proposed scheduler with failure prediction minimizes the total makespan and also increases the reliability.

Index Terms: Availability; Genetic Algorithm; Grid; makespan; reliability; Support Vector machine; system log file.

Introduction

Grid computing is a coordinated resource sharing environment where resources from multiple organizations are shared to solve complex problems[1]. Scheduling in Grid is a technique to allocate a task to a suitable resource selected from the available set of resources as per the user's requirement. Heterogeneous and dynamic nature of Grid make the scheduling process complicated and hence Grid scheduling is classified as

NP-Complete. Large number of scheduling algorithms are available to reduce the overall makespan of the job thereby increasing the efficiency of the Grid scheduler.

Due to the dynamic nature of the Grid, the availability of the resources is stochastic - the resources may join and leave the Grid at any time as per their availability for the Grid. The increase in heterogeneity of the resources in Grid is one of the causes for non availability of resources for scheduling. Each type of device shows different availability characteristics as discussed in [2]. Grid is not having any control over the resources as they are owned by other organizations and managed by the policies of those organizations.

The resources are inaccessible due to the failure of hardware, software and communication devices. The failure of the resources may happen in the middle of the execution which makes the jobs fail that leads to disagreement of QoS.

Failures of the hard disks and processors disrupt the execution of the jobs on Grid since these failures require unpredictable time for recovery process. If the failures are predicted well ahead of allocating the jobs, the recovery processes can be planned, the overall makespan can be reduced and the same time the reliability is enhanced.

More efforts have been taken to predict the failures and a number of predictive methods have been discussed in the recent years[3,4]. The failures can be predicted based on the events recorded in the system log files. Each entry in the log file contains the time of occurrence of the event and the message indicating the severity level. With the proper analysis of the log file, the future events can be predicted and this prediction can be given to the Grid scheduler to plan the scheduling. Machine learning and data mining techniques can be applied to analyse the log files to predict the probable resource failures. In this paper, Support vector machine(SVM) is used to predict the resource failure using system log files. SVM is a binary classifier which is used to predict the failure probability of resources. This will enhance the reliability and minimize the makespan.

Once the potential resources are identified, the scheduler has to select from these resources to allocate the tasks in order to minimize the total makespan. First in First Out scheduling technique is used for scheduling the tasks. The performance of the scheduler is compared with and without failure prediction.

The rest of the paper is organized as follows. The previous works related to the system log analysis are discussed in section 2. In section 3, the background of this work is presented. The proposed work is demonstrated in section 4. The experimental results and performance analysis are explained in section 5. Section 6 concludes this paper with the avenue for future work.

Related Works

A. Failure Prediction

Techniques for predicting failure events based on the system log have been proposed by many researchers. Standard machine learning techniques such as Hidden Markov model, Bayesian approach, Support Vector machine, etc., are used to predict the system failures by analysing the system log messages.[3,4,14]. Grag H.et al.,[9]

proposed a Bayesian approach to predict the failures in disk drives. They used the combination of naïve Bayes submodel and naïve Bayes classifier. They tested on real world data and their model's predictive accuracy is higher than the accuracy of the threshold methods used in the hard drive industries.

For accurate failure prediction a three step method is proposed in[8] to prepare the log messages. The processed log messages are fed in to Hidden Semi Markov model for prediction. Log processing before the prediction improves the accuracy of prediction and enhances the speed of fault diagnosis.

Radial basis function networks are used to analyse the hardware sensor data to predict the failure in computer server[11].Customized Nearest Neighbour approach[12] is used to classify the error events of IBM BlueGene/L super computer.

Similar Events Prediction (SEP) is presented in[13]. Dubious sequence of error events are recognized by SEP to predict the failure. SEP achieved high precision and recall such as 80% and 92% respectively and is better than the other failure prediction techniques. Cox proportional hazard model is proposed in [7,15] to provide a statistical prediction of system failure events. In [16], association rule mining techniques are used to detect patterns in event sequences and prediction is done by combining these sequences into a rule based model.

Automatically generated event logs from fault tolerant systems are analyzed in [17]. Investigation of error and failure dependency among different system components are carried out by applying multivariate statistical techniques. In [7,18] failures are predicted using event logs of IBM's BlueGene/L containing reliability, availability and serviceability data.

SVMs are used to forecast software reliability[19]. In general, prediction/ classification using SVM is performed based on the aggregate features [7,20,21] such as average number of messages during a particular period of time. Spectrum representation of messages from system log files are used in [3]. The results of our predictions are compared with the experimental results of [3].

B. Reliability

Reliability is defined as the ability of a component or a system to perform a required function under a given condition for a given time interval. Reliability of the resources is not guaranteed in Grid because of its heterogeneous nature.

Numerous techniques have been suggested to increase the reliability of grid resources to increase the efficiency of the grid scheduler. Failure rate of software and hardware components are considered in [22] and a priori reliability computation is performed before allocating the job to the grid resources. Reliability factor is considered in [23] based on the failure probability of the resources and rescheduling is being performed if the allotted node fails. Bayesian networks are used[24] to estimate the reliability of the Grid resources.

Task partition and allocation of the subtasks are presented in [25,26,27]. In order to enhance the reliability, partitions of the task are replicated and the replicas are allotted to various resources for parallel execution. Data dependence and failure correlation are modeled using Star topology and tree structure. Minimal Resource

Spanning Tree is used to represent the Grid computing system with nodes and the links to connect the nodes[26].

A replication scheme for cluster computing is presented by mapping tasks to various resources in the form of resource graph[28] and in [29] an algorithm is devised to reduce the communications between tasks. A distributed grid scheduler computes the reliability factor of the grid and also maintains a queue of tasks allocated to a faulty node. These tasks are rescheduled to other nodes [23]. Rescheduling of a task can be done in two ways – task may be reallocated to a resource and start the execution afresh or the execution may be started from the state where the failure occurred. Checkpointing is the ideal mechanism to record the status of the task under execution. If a failed node is recovered soon then the task can be continued in the same node instead of migrating to other. Hence time and cost involved in the task migration is minimized [30].

The existing reliable schedulers for grid are not considered the prediction of the failures of the resources/nodes. The task migration overhead can be minimized by predicting the failure of the resource in advance. With the above observations, it is proposed to design a system which uses SVM as a predictor to predict resource/node failure and Genetic algorithm based scheduler.

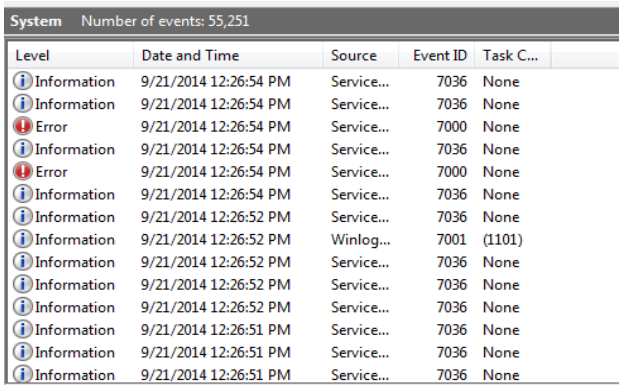
Background For Event Prediction

A. System Log Files

System log files are used to record the events that occur during the execution of the system which provide the history of events for the understanding of the system. The events are recorded by the components of operating system. The changes occurred in the devices attached to the system, device drivers, changes in the system's operations, etc., are logged in system log files. In case of a problem, the events recorded in the log file are analysed and the problem is identified. After the identification of the problem it is fixed. These log files are not only used for detecting the cause of the events/anomalies in the system but also to predict the occurrences of these events/anomalies in future.

System log messages have a fixed structure. The message structure varies with the operating systems. We have considered Windows log file for our investigations. The windows log files are having various fields as shown in figure1. The level field indicates the severity level of an event or message. The various levels are critical which requires immediate action, error that indicates the problem but does not require immediate action by the system administrator, warning gives the indication about a component or an application which is not in proper state that may lead to potential problem, information just sends the non critical information to the administrator and finally the verbose message tells the progress or the success of an event. Date and time field provides the date and time at which the event has occurred and recorded in the log file. The source gives the event or the change in state of a component which generates the message where as task category represents the publisher of the event such as the subcomponent or the activity. The event id is to identify a particular type

of event. User specifies on whose behalf the event has generated. Name of the system on which the event has generated is given in computer.



System Number of events: 55,251

Level	Date and Time	Source	Event ID	Task C...
Information	9/21/2014 12:26:54 PM	Service...	7036	None
Information	9/21/2014 12:26:54 PM	Service...	7036	None
Error	9/21/2014 12:26:54 PM	Service...	7000	None
Information	9/21/2014 12:26:54 PM	Service...	7036	None
Error	9/21/2014 12:26:54 PM	Service...	7000	None
Information	9/21/2014 12:26:54 PM	Service...	7036	None
Information	9/21/2014 12:26:52 PM	Service...	7036	None
Information	9/21/2014 12:26:52 PM	Winlog...	7001	(1101)
Information	9/21/2014 12:26:52 PM	Service...	7036	None
Information	9/21/2014 12:26:52 PM	Service...	7036	None
Information	9/21/2014 12:26:52 PM	Service...	7036	None
Information	9/21/2014 12:26:51 PM	Service...	7036	None
Information	9/21/2014 12:26:51 PM	Service...	7036	None
Information	9/21/2014 12:26:51 PM	Service...	7036	None

Figure 1: Windows System Log File

We have collected more than 1000 system log files from various laboratories of our institution and neighboring institutions over a period of six months. The table I shows the fields of the Windows log files.

Table 1: Fields of Windows Log File

Fields	Description
Level	Severity Level of the message
Time and Date	Date and time when the message was logged
Source	Event or the component of the system which logged the message
Event ID	Number to identify an event type
Task Category	Activity of the event publisher
User	Name of the user
Computer	Name of the computer where the event has occurred

B. Support Vector machine

A support vector machine (SVM) is a classification method introduced in 1995 by Vapnik et al[31]. It is a regression technique that classifies the given data in to two categories. In computer science, SVM is categorized as a supervised learning technique which is used in data mining as a binary classifier to classify the data.

The terminologies used in SVM are attribute, feature, feature selection, hyperplane, vectors and support vectors. An attribute is the predictor variable and a modified attribute is called feature. Selecting an appropriate representation is known as feature selection. A set of features to describe a single sample is called a *vector*. A straight line which is used to separate the set of vectors in to two classes is termed as hyperplane. Number of such hyperplanes can be drawn to separate the vectors. The main objective of SVM is used to find the optimal hyperplane. Support vectors are the vectors which are present near the hyperplane.

The training sample can be represented in the form of a tuple (x_i, y_i) where $x_i \in \Re^N$ ie., the N dimensional attribute and $y_i \in \{-1,1\}$, i= 1,…, n, ie., the label which indicates the classes, here positive and negative and n is the number of samples.

After learning from the samples a general function f: $\Re^N \rightarrow \{-1,1\}$ is devised to classify the unknown sample in to -1 or +1. These general functions can be expressed as hyperplanes of the form

$$(w.x) + b = 0, w \in N / Rb \in \Re \quad (1)$$

The weight vector *w* decides the direction perpendicular to the hyperplane and b is the threshold to fix the distance of the hyperplane from the origin. The function f is defined in (2) to classify an unknown input vector x in a binary fashion by giving either -1 or +1.

$$f(x) = sign\big((w.x) + b\big) \quad (2)$$

SVM is robust even for the biased training sample by selecting appropriate generalization grade [32] and is delivering unique solution. SVM is one of the best classifiers and is suitable for our work.

C. Performance Assessment

The evaluation of the model for Support Vector Machine is assessed with the help of the contingency table otherwise called Confusion Matrix which is depicted in table II.

Table 2: Contingency Table/Confusion Matrix

		Actual values	
		Pos	Neg
Predicted outcome	PP	TRUE POSITIVE	FALSE POSITIVE
	PN	FALSE NEGATIVE	TRUE NEGATIVE

Each row of the matrix represents the samples in a predicted class, while each column represents the samples in an actual class. PP indicates predicted positive whereas PN indicates predicted negative. The values entered in the table show the predictions made by the model. The primary diagonal of the matrix shows the correct classification made for each class by the SVM model and the other values show the incorrect classification made by the model. Further the model's performance is measured with the help of number of performance metrics.

The overall correctness of the model is called Accuracy and is calculated as the sum of all the correctly predicted classifications divided by total number of classifications made by the model and is given in equation (3).

$$Accuracy = \frac{(TN + TP)}{(TP + TN + FP + FN)} \quad (3)$$

Precision of the model is the accuracy in which the model is predicted correctly as positive and is calculated as per equation (4).

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Recall is otherwise called as true positive rate or sensitivity and is defined as the correctly predicted positive classifications divided by total number of positive classifications and is calculated based on (5).

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

True negative rate or specificity is the ratio between the correctly predicted negative classifications and the total number of negative classifications and is given in (6).

$$True\ Negative\ Rate = \frac{TN}{TN + FP} \quad (6)$$

False positive rate or fall out is given in (7) as the number of wrongly predicted positive classification divided by the total number of negative classifications.

$$False\ Positive\ Rate = \frac{FP}{TN + FP} \quad (7)$$

Performance metrics as described are derived from confusion matrix. They are used to evaluate the prediction model devised using SVM. Receiver Operating Characteristic (ROC) curve is a two dimensional graph used to visually depict the performance of the classification/prediction model. ROC curve is constructed by plotting True positive rate against False positive rate. Using ROC curve we evaluated the performance of the proposed predictor model.

Proposed Work

The reliable scheduler for the Grid is achieved by predicting the failure before allocating the jobs to the resources. The proposed Grid Scheduler is given in figure 2.

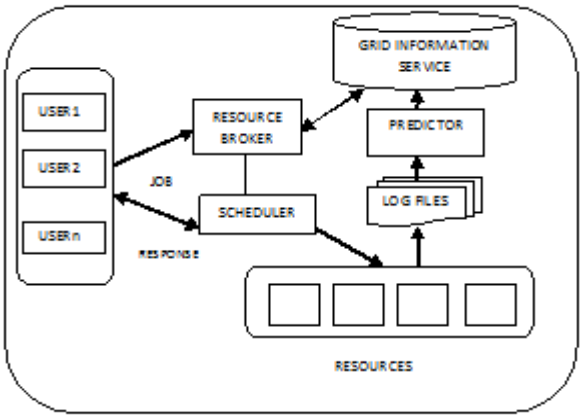


Figure 2: Reliable Grid Scheduler with Predictor

The Resource Broker (RB) receives requests from the users. The Grid Information service (GIS) maintains the resources' configurations and availability information. The SVM predictor predicts the failure probability of the resources by analyzing the log files. After receiving the predicted information, the GIS removes the resources having more failure probability from the potential resource list. Resource Broker performs the mapping of received jobs with the suitable resources which is obtained from GIS. The Scheduler schedules the jobs to minimize the overall makespan. We devised a scheduler using Genetic algorithm.

D. Event Prediction Using SVM

System log events are analysed and system failures are predicted with the help of SVM. Being a binary classifier, SVM predicts the system failure by finalysing the messages of the log files are related to or not related to failure in near future.

The series of log messages of a single computer may be represented as L. The messages are ordered according to time at which the events have occurred. The messages in L are represented by their event-ids. The set L is { 7000,7001,7036} as per the sample log file given in figure 1.

With the help of L, various aggregate features are able to collect such as the number of occurrences of event-ids. For the messages in figure 1: the event-ids 7000 and 7001 occurred once and the event-id 7036 occurred 13 times. This forms a vector describing L as (7000:1, 7001:1,7036:13), where each value is the *event-id*:*count*. This vector can then be used to classify L as belonging to a fail or non-fail system.

E. Reliable Grid Scheduler

The proposed Reliable Grid Scheduler (RGS) is modeled using Genetic algorithm. Let the Grid system is having *m* tasks of a job (T_i where $i=1..m$) and *n* resources (R_j where $j=1..n$). Various symbols used in the scheduler with their descriptions are listed in table III.

Table 3: Symbols Used and Their Descriptions

Symbol	Description
T_i	i^{th} Task of a Job
R_j	j^{th} Resource
ST_i	Size of i^{th} Task(MI)
PR_j	Processing Capacity of j^{th} Resource (MIPS)
EET_{ij}	Expected Execution time of i^{th} Task on j^{th} Resource
ECT_{ij}	Expected Completion time of i^{th} Task on j^{th} Resource
$AV(R_j)$	Available time of j^{th} Resource
$AR(T_i)$	Arrival Time of i^{th} Task T_i for resource R_j
$WTij$	Waiting time of task
$MS(C_k)$	Makespan of k^{th} chromosome

Each resource executes one job at a time. Expected Execution Time for each task on various resources is calculated as per equation (8) and a EET matrix is created.

$$EET_{ij} = \frac{ST_i}{PR_j} \tag{8}$$

The Expected Completion Time of each task on various resources is calculated by using EET_{ij} and waiting time of the task T_i for resource R_j as given in (9) and (10) then the Expected time to complete(ETC) matrix is generated.

$$WT_{ij} = abs\big(AV(Rj) - AR(Ti)\big) \tag{9}$$

$$ETC_{ij} = EET_{ij} + WT_{ij} \tag{10}$$

The initial population of chromosomes are created randomly. The figure 3 shows the sample chromosome which represents a possible solution to the scheduling problem. The length of the chromosome is equal to the number of tasks of a job in the Grid. Figure 3 shows the tasks T_1, T_2, T_3 and T_4 are allocated to resources R_3, R_6, R_{10} and R_1 respectively.

R_3	R_6	R_{10}	R_1	…
T_1	T_2	T_3	T_4	…

Figure 3: Sample Chromosome Representation

The chromosomes(C_i) are evaluated with the help of a fitness value which is achieved by using a fitness function *f*. The fitness function *f* is defined based on the makespan of the chromosome which is nothing but overall completion time of a schedule.

$$Minimize\ f = (f_1, f_2) \tag{11}$$

$$f_1 = min.\ of\ makespan\ of\ the\ i^{th}\ Job \tag{12}$$

$$f_2 = min.\ of\ cost\ of\ completing\ the\ execution\ of\ i^{th}\ Job \tag{13}$$

The main aim of our work is to minimize the makespan of the scheduling process. To produce the next generation, chromosomes are selected as members of the mating pool by using Tournament selection process.

Reshuffling of chromosomes otherwise called offspring are produced by the crossover operation. In this work a two point cross over is used and for mutation bit-flip type is employed. The parent chromosomes in the previous generation are replaced by their offspring and this process will be repeated till the stopping criteria is met. From that generation, the chromosome with least fitness value is selected as the optimal schedule for the Grid. The RGS-GA algorithm is given in figure 4.

ALGORITHM

```
procedure Generate Initial Population
 begin
   for i=1 to pop_size
    for j=1 to length_of _chromosome
     c_ij = random(R_1,R_m)     //  m – no. of resources
   end
 end
end
procedure fit(c_i)
 begin
   for i=1 to pop_size
     fc_i = makespan(c_i)
   end
end
procedure selection
 begin
  for i=1 to pop_size
  p_1 = random(c_i)       // select pair of chromosomes randomly
  p_2 = random(c_i)
  if fit(p_1) ≤ fit(p_2) then
     select p_1 for matting pool
  else
     select p_2 for matting pool
  end if
 end
end
procedure crossover
begin
 for i=1 to pop_size
   choose two parent chromosomes randomly(c_k,c_l)
   check the crossover probability
   if yes then
    select two points x,y randomly within the length of chromosome
    for j = x to y
      swap the j^th bits of  c_k and c_l
    end
   end if
 include c_k,c_l  to the new population
end
procedure mutation
begin
 for i=1 to pop_size
  check the mutation probability
  if yes then
   select a point z randomly within the length of chromosome
```

flip the z^{th} bit in the i^{th} chromosome
end if
end
end

Figure 4: RGS-GA Algorithm

Experimental Results and Performance Analysis

This section elaborates the experimental setup and results obtained by the proposed work. For predicting system failures using log files we used the real world systems. Simulation environment is used for scheduling using GA.

A. Experimental Results and Performance Analysis for SVM predictor

To conduct experiment, actual log files are used for predicting the system failure using SVM. System log files of around 1000 computers from various laboratories of our institution and other institutions were collected for six months. We used SVMlight tool to analyse the log files. The log files are preprocessed and prepared according to the format required by SVMlight.

All the systems are Windows based systems. The figure 5 shows the logged events for a single computer over a six months period and the circle represents the event-ids for each message and is plotted on y-axis against the dates at which the events occurred. The log files recorded the events on an average of 28 messages per day in a single computer. The figure 3 gives the distribution of the event-ids. There were 53 unique event-ids ranging from 1 to 51047 . The distribution of event-ids for a single system is given in figure 5. Some events occurred more frequently as shown in the figure 6.

The failure events predicted from the log files are System service failures with event-ids 7022,7023,7024,7026, 7031,7032 and 7034 which may cause the windows service failure. This failure may disrupt the availability of the system for Grid. The event-id 41 may lead to disk failure. This event is also to be predicted.

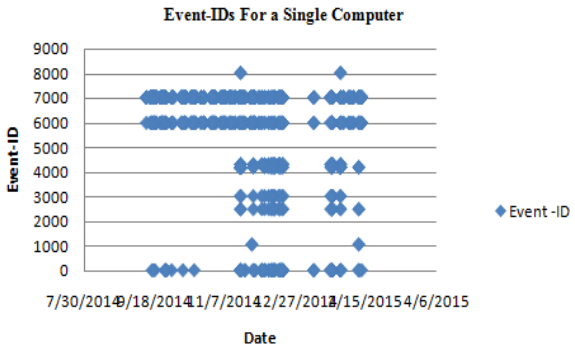


Figure 5: Example of Logged events for a Single computer

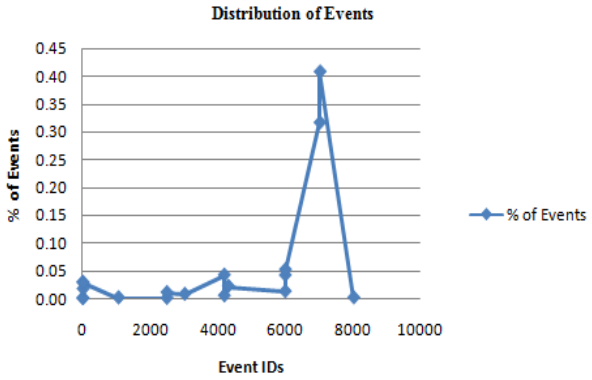


Figure 6: Distribution of event-ids for the system log files

Table 4: Lab Details

Labs	No. of Systems	Predicted		Actual	
		Failures	Non Failures	Failures	Non Failures
Lab1	110	6	104	5	105
Lab2	60	3	57	1	59
Lab3	60	0	60	1	59
Lab4	45	0	45	0	45
…….	……	…….	……	…….	…….
Lab14	90	3	87	1	89
Lab15	45	1	44	1	44
Lab16	60	2	58	1	59
Lab17	100	3	97	2	98
Lab18	90	0	90	0	90

The table IV shows the number of systems in various labs and the predicted failures using the log files and the occurrence of actual failures. There are totally 18 labs and 1010 computers with Windows operating systems- mostly with XP and Vista. The performance of our predictor is analysed by using the performance metrics such as accuracy, precision, recall, true negative rate and false positive rate which is given in table V.

The accuracy of our model is 98%. This shows that our predictor is predicting the failures of the systems by analyzing the log files in an accurate manner. The following table shows the other metrics.

Table 5: Performance Metrics of Svm Predictor

Labs	Accuracy	Precision	TPR/Recall	TNR	FPR
Lab1	0.972	0.666	0.8	0.98	0.019
Lab2	0.966	0.333	1	0.966	0.033
….	……	……	……	…….	……
Lab13	0.95	0.5	0.333	0.982	0.017
Lab14	0.977	0.333	1	0.977	0.022
Lab16	0.983	0.5	1	0.983	0.016
Lab17	0.97	0.333	0.5	0.9795	0.0204
….	……	……	……	…….	……

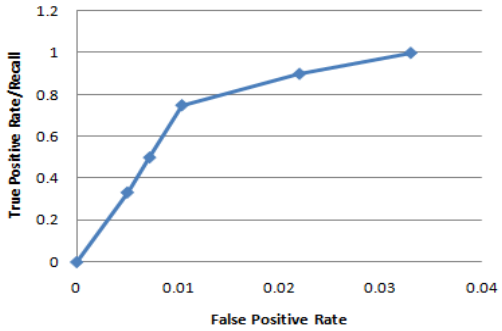


Fig 7: Receiver Operator Characteristics Curve

The ROC curve given in figure 7 shows the performance of our model. The ROC curve shows the relationship between TPR and Specificity. The Specificity decreases when sensitivity increases. If the curve is away from the diagonal the model is fit. Here the ROC curve of our model is away from the diagonal which clearly indicates that our model classifies better.

B. Experimental Results and Performance analysis for proposed scheduling algorithm

In this section , we present the experimental setup and the simulation of the RGS-GA scheduler. The scheduler receives the resources from the resource broker which in turn gets them from GIS. GIS is having only the fail safe resources which are predicted by the SVM predictor. The experimental results of the predictor ensures high reliability of the resources. Hence the scheduler is having only reliable resources. Even though the scheduling is performed using a simulated environment, faults can be induced to check the performance of the proposed GA based Reliable Grid Scheduler(RGS-GA) with failure prediction and without failure prediction. The proposed scheduler's performance is compared with other scheduling algorithms like Min-Min, Minimum Completion Time, First Come First Serve and Opportunistic Load Balancing algorithms.

The parameters selected for GA based RGS scheduler is given in the table VI and the necessary details related to the scheduling is available in table VII.

Table 6: Parameters For Rgs-Ga Scheduler

Parameters	Values
Population Size	25
Number of Generations	100
Number of Crossover Points	2
Crossover Probability	0.6
Mutation Probability	0.01/bit

Table 7: Details For Scheduling

Items	Details
Number of Resources	20
Number of Tasks/job	250
Size of Tasks	1000 to 5000 MIS
Processing capacity of the resources	2000 to 5000MIPS

The sample Expected Execution Time matrix and Expected time to complete matrix are given in table VIII and IX respectively. Using these matrices the fitness value for each chromosome is calculated.

Table 8: Expected Execution Time Matrix

	T_1	T_2	…..	T_{250}
R_1	1150	2000	…..	725
R_2	920	1600	…..	580
	…..	…..	…..	…..
R_{20}	920	1600	…..	580

Table 9: Expected Time To Complete Matrix

	T_1	T_2	….	T_{250}
R_1	1150	1994	….	735
R_2	930	1604	….	600
….	….	….	….	….
R_{20}	910	1584	….	580

The simulation results show that our scheduler with predictor outperforms the other scheduling algorithms with and without predictor. This is clearly depicted in the

figure 8. However the Minimum Completion Time (MCT) scheduler is closely following RGS-GA scheduler.

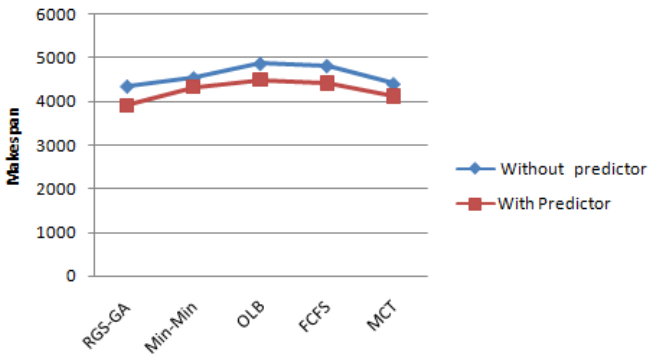


Figure 8: Makespan of Various Scheduling Algorithms with and without Predictor

Conclusion and Future Work

In summary, in this work, we have devised a SVM based failure predictor using system log files to predict the fail safe resources that will be used by the Grid to provide reliable service. Apart from the predictor, we have also created a scheduler based on Genetic algorithm which will use the resources predicted as fail safe by our predictor. For the predictor we used actual system log files to classify the resources in to failure and non failure categories. We used the simulated environment for the RGS-GA scheduler. Furthermore, the scheduler's performance is compared with other heuristic and non-heuristic scheduling algorithms. From the results of the simulation, RGS-GA performed well as against the other scheduling algorithms.

The results of this work paved way for future enhancements. First, the impact of the number of messages in the log files for analysis to be considered for better resource failure prediction. Second, we have not considered the resource utilization that need to be addressed in future. Finally, a suitable fault tolerance technique to be employed if the predicted resource failed after the allocation of a task to it. Since the electronic systems are prone to failure, ensuring reliability is for most important.

References

[1] Foster, I. and Kesselman, C., 1998,The Grid: The Blue print for a New Computing Infrastructure, Morgan Kaufmann.

[2] Brent Rood, Michael J. Lewis, 2009,Grid Resource Availability Prediction-Based Scheduling and Task Replication, Journal of Computing, Springer, vol. 7, pp 479-500.

[3] Errin W. Fulp , Glenn A. Fink,Jereme N. Haack,, 2008,Predicting Computer System Failures Using Support Vector Machines, WASL'08 Proceedings of the First USENIX conference on Analysis of system logs.

[4] Zheng, Z., Lan, Z., Park, B.-H., Geist, A., 2009,System log pre-processing to improve failure prediction. In: Proceedings of DSN'09.

[5] Fu, S., Xu, C.Z., 2007, Exploring event correlation for failure prediction in coalitions of clusters. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Reno, NV, USA, November 15–21.

[6] Gross, K.C., Bhardwaj, V., Bickford, R., 2002, Proactive detection of software aging mechanisms in performance critical computers. In: Proceedings of the 27th Annual NASA Goddard Software Engineering Workshop.

[7] Liang, Y., Zhang, Y., Xiong, H., Sahoo, R., 2007, Failure prediction in IBM bluegene/l event logs. In: Proceedings of the International Conference on Data Mining.

[8] Salfner, F.,Steffen Tschirpke, 2008, Error Log Processing for Accurate Failure Prediction.

[9] Greg Hamerly, Charles Elkan, 2000,Bayesian approaches to Failure prediction for disk drives.

[10] Salfner, F., 2008,Event-based failure prediction: an extended hidden Markov model approach. Dissertation, Verlag, Berlin, Germany.

[11] Doug Turnbull, Neil Alldrin, 2007,Failure Prediction in Hardware Systems,.

[12] Yinglung Liang, Yanyong Zhang, Hui Xiong, 2007, Failure Prediction in IBM BlueGene/L Event Logs, Seventh IEEE International Conference on Data Mining, pp. 583-588.

[13] Salfner, F., Schieschke, M., Malek, M., 2006,Predicting failures of computer systems: a case study for a telecommunication system. In: Proceedings of the 20th Inter-national Conference On Parallel and Distributed Processing Symposium, Rhodes Island, Greece, April 25–29.

[14] Li, Z., Zhou, S., Choubey, S., Sievenpiper, C., 2007,Failure event prediction using the Cox proportional hazard model driven by frequent failure sequences. IEE Transactions 39 (3), 303–315.

[15] Illenia Fronza, Alberto Sillitti, Gian Carlo Succi, Jelena Vlasenko, 2011 Failure Prediction based on Log files using the Cox Proportional Hazard Model, 23rd International Conference on Software Engineering(SEKE 2011).

[16] Vilalta, R., Ma, S., 2002,Predicting rare events in temporal domains. In: Proceedings of the International Conference on Data Mining.

[17] Lee, I., Iyer, R.K., Tang, D. 1991,Error/failure analysis using event logs from fault tolerant systems. In: Proceedings of the 21st International Symposium on Fault- Tolerant Computing.

[18] Prahasta Gujrati, Yawei Li, Zhiling Lan, Rajeev Thakur, John White, 2007,A Meta-learning failure predictor for Blue Gene/L System.

[19] Pai, P.F., Hong, W.C., 2006,Software reliability forecasting by support vector machines with simulated annealing algorithms. Journal of Systems and Software 79 (6), 747–755.

[20] Xue, Z., Dong, X., Ma, S., Dong, W., 2007, A survey on failure prediction of largescale server clusters. In: Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 733–738.

[21] Yamanishi, K., Maruyama, Y., Dynamic 2005,Syslog mining for network failure monitoring. In: Proceedings of the International Conference on Knowledge Discovery in Data Mining, pp. 499–508.

[22] Zahid Raza, Deo Prakash Vidyarthi, 2011,Reliability Based Scheduling Model (RSM) for Computational Grids, International Journal of Distributed Systems and Technologies, 2(2), pp. 20-37.

[23] Kovvur Ram Mohan Rao, Ramachandram S, Vijaya Kumar Kadappa Govardhan A, 2011, A Reliable Distributed Grid Scheduler for Independent Tasks, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, pp. 296-301.

[24] Ozge Doguc J., Ramirez Marquez E., 2008,Estimating Reliability of Grid System using Bayesian Networks, Proceedings of IPVC'- The 2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition.

[25] Gregory Levitina, Yuan-Shun Dai, 2005,Service Reliability and performance in grid system with star topology, Reliability Engineering and System Safety, vol. 92, pp. 40-46.

[26] Yuan-Shun Dai, Gregory Levitin, 2006,Reliability and Performance of Tree-Structured Grid Services,IEEE Transactions On Reliability, Vol.55, No.2, pp 337-349.

[27] Yuan-Shun Dai , Jack Dongarra, 2010,Reliability and Performance Models for Grid Computing, Handbook of Research on Scalable Computing Technologies, IGI Global Disseminator for Knowledge.

[28] Nabil Tabbaa, Reza Entezari-Maleki, and Ali Movaghar, 2011, A Fault Tolerant Scheduling Algorithm for DAG Applications in Cluster Environments ,The International Conference on Digital Information

Processing and Communication (ICDIPC 2011), Communications in Computer and Information Science (CCIS), Vol. 188, Springer press, pp. 189-199, Ostrava, Czech Republic.

[29] Nabil Tabba, Reza Entezari-Maleki, Ali Movaghar, 2012,Reduced Communications Fault Tolerant Task Scheduling Algorithm for Multiprocessor Systems , International Workshop on Information and Electronics Engineering (IWIEE),.

[30] Suchang Guo, Hong-Zhong Huang, Zhonglai Wang, and Min Xie, 2011,Grid Service Reliability Modeling and Optimal Task Scheduling Considering Fault Recovery, IEEE Transactions On Reliability, Vol. 60, No. 1, pp.263-274,.

[31] Vapnik, V., 1995,The Nature of Statistical Learning Theory, New-York, Springer-Verlog,.

[32] Laura Auria, Rouslan A Moro, 2008, Support Vector Machine(SVM) as a Technique for Solvency Analysis, Discussion Papers, Berlin.