

Resource Availability Prediction in the Grid: Taxonomy and Review of State of the Art

Farrukh Nadeem
King Abdulaziz University
Jeddah, Saudi Arabia.

Mahreen Nasir
University of Hail
Hail, Saudi Arabia.

Abstract - Use of Grid Computing for carrying out cooperative work from distributed resources has been into practice for the past several years. Efficient execution of various tasks on the grid comes with various challenges. One of them is to ensure that a particular resource remains available during the execution of task. The dynamic nature of resources makes it even more challenging to predict resource availability for complete task duration. This paper is an attempt to address this issue by providing a comprehensive review of the existing methods along with a stated taxonomy of the approaches used.

Keywords- resource availability; grid computing; data mining; machine learning; survey

I. INTRODUCTION

In a distributed environment, resource availability prediction is necessary for grid schedulers to have smooth execution of tasks. A resource refers to an asset which is required for task execution e.g. a computing machine, CPU cycle and memory etc. It can also be a group of machines (cluster) working together to achieve a common goal. Before starting task execution, a resource should be ready and free for use. An allocated resource cannot be re-assigned to a new task. To guarantee the task execution without interruption, required resources must be made available for the complete duration needed for task completion. Resources are owned by organizations which have pre-defined policies about resource allocation which can be either on demand (non-dedicated) or made available all the time (dedicated) to the grid. It is vital to determine the resource availability for a specific time period in the near future so that it can be assigned in response to a resource request. Predicting resource availability in grid is a challenging issue mainly because the resources are non-dedicated, dynamic and may not be fully available during the task execution. Also, as the system is distributed, it is hard to record idle or busy resources. Additionally, the node(s) in the network are transient as they join or leave the network without any notice which may result in absence of the node at any particular time. This requires prior investigation of available resources for their fair and uniform allocation to various tasks.

Many techniques had been suggested to address this problem. This survey is an effort to discuss the various proposed methods for predicting resource's availability in a grid.

The survey is aimed to:

1. Provide an extensive overview of the existing resource availability prediction techniques in grids
2. Provide taxonomy for the classification of available techniques
3. Analyze the available techniques by discussing their pros and cons

II. CLASSIFICATION OF RESOURCE AVAILABILITY PREDICTION TECHNIQUES

A. Probability Theory

Prediction approaches under this category estimate probability of resource availability in the future based on its past availability patterns. The main goal is to determine that how much time a resource will spent on each particular state (multistate model).The authors in [4] present a parametric model fitting technique (weibull method) along with two non-parametric techniques(Resample method and Binomial method) to predict machine availability duration. The authors focus on the estimation of a specific quantile for the availability distribution along with a confidence level which is related with each estimate. The goal is to facilitate the schedulers to make dynamic decisions by supporting live availability predictions. The authors verified usefulness of their proposed technique by performing lower bound quantile estimation on a synthetic fixed weibull distribution. Later in the next step, availability traces for individual machines were taken and two sets of training and test data were created respectively. Training data facilitated to find the lower bound of quantile and test data was used for accuracy verification of the estimate. The study in [17] showed the use of probability theory. Various aspects were discussed for resource availability including:

1. Resource online serving time: it is measured by the time when the resource is offline due to some failure, reboot or service with drawl.
2. Resource serving time: Measured by time taken by local task execution in a specific period.
3. Resource availability during task execution: Resource accessibility can be measured by using the length of waiting queue.

Resource may become unreachable because of approaching the maximum limit of number of task acceptance. Prediction of a resource is done by using probability theory and resource evaluation is performed by suggesting four availability metrics. The metrics used for availability were: offline time of a resource, task execution time (local), waiting time and waiting queue length. The authors mention that the resource offline time is almost stable and it is considered as a constant and shows an exponential distribution. The second metric which is local execution time is shown to be a random variable and it shows the pattern of poison distribution. It is assumed that if a resource has a higher idle rate then it can provide service for longer periods.

1) Multistate Model

A multistate availability model is proposed in [13] to determine resource unavailability. Semi Markov process models (SMP) were used. The proposed framework is applied using an ishare production system. The method achieved an accuracy level of 86%. The authors used response time as a performance metric. Additionally, various resource unavailability types were discussed including:

1. Unavailability Due to Excessive Resource Contention (UEC): An unavailability, e.g. due to host and guest process running parallel on same machine. Guest process may lower down host process's execution. To resolve this, there might be a need to decrease the guest process's priority. As a result guest process may be stopped or had to be shifted to another machine which may cause a failure.

2. Unavailability Due to Resource Revocation (URR): This can be caused due to sudden hardware failure or machine unavailability without any notice. Authors' main contribution is to suggest a method to predict when a resource becomes unavailable. The multistate model combines the above mentioned classes of resource unavailability. In the suggested multistate model, availability and unavailability are modeled by using some observations. These observations are recorded for determining the two resource unavailability states (as mentioned above). The URR state can be caused when there is a failure in accepting service submission. The UEC state is reached in case a host process slows down. Parameters like CPU cycles and memory use are required to monitor the slow down by setting a threshold. The priority of guest process is reduced in such a scenario. The main idea is to fix a threshold which measures the slowdown of host process. A group of host processes having different resource usage were executed

together as part of experiment. A single guest process is permitted to execute at a particular time. The priority of guest process is decreased if it affects host process. If the slowdown still remains, then the guest process is held and is resumed after the congestion is over, else it is terminated. The slowdown of host process is measured by the reduction in the rate of CPU usage going above a threshold which is greater than 5%. To measure the CPU load (Lh), two thresholds Th1 and Th2 were used. The value of Th1 is set as 20% and Th2 as 60%.

The suggested method was implemented within the ishare framework which consist of host (provider) and client(consumer) nodes. On submission of a job request from client, a prediction function is called. The ishare gateway component is responsible to communicate with clients. Resource Monitor component monitors the usage of host process's CPU and memory. On receiving a job submission request, the job scheduler on the client inquires the gateway about the availability of machine during the specified future time window. According to the response, if a machine is selected, guest process execution is initiated and the Resource Manager is informed about the id of new process. Any state transition during the execution of the process is signaled to the gateway in order to take necessary actions regarding process killing or migration to another machine. A multistate grid resource availability characterization is presented in [13]. The authors propose a multistate model to determine the future availability of a resource. The model has five states of resource availability which determines why and how the resource becomes unavailable based on its transition to a particular state. It also categorizes the resources to differentiate between their graceful and ungraceful transitions to unavailability state. According to the authors, use of resource failure information, unavailability information, checkpoint ability and expected run time of a job can be very useful for grid schedulers for scheduling applications on the grid. The multistate model has five states which are: available to Grid, user present, CPU threshold exceeded, job eviction or graceful shutdown and unavailable. A resource may transit between these states from time to time. A resource will be available if the machine is connected to the network with local CPU load less than the threshold and idle time to be non-zero. It transits to user present state if a connected I/O device like mouse or keyboard is activated. CPU threshold exceeded state is reached when the load of local CPU exceeds the threshold. A transition to job eviction state is performed based if any of the following conditions hold:

- a) The resource is suspended for a long time.
- b) The machine is evicted during its execution
- c) The machine shuts down

Lastly, if a machine is unreachable or fails, a direct transition to unavailable state is made. Based on this state model, unavailability types are classified as graceful and ungraceful. A graceful transition is one when a job enters a job eviction state during its execution or suspension, whereas

ungraceful transition occurs when a direct transition is made to unavailable state.

Further enhancement had been proposed to forecast multi-state availability of a resource in [14]. The study shows that analyzing a resources behavior with its history can be a good predictor. The accuracy is enhanced through transition weighting schemes. To decide as when to assess the resource behavior, the predictor uses two approaches. First is to examine the availability behavior for the previous days and the second approach uses the most recent hours of activity preceding the prediction.

The authors in [16] propose an algorithm using transition N-day with equal transition weight is presented in [14]. The algorithm is assumed to have a centralized job scheduler. All jobs are submitted to the scheduler which then forwards them into a job queue. The job scheduler should know in advance the required execution time by each job. After execution the results are sent to the users who submitted the jobs. The scheduler sends the jobs into a queue and uses First Come First Serve for sorting the jobs. If jobs are present in the queue and resources are available then the scheduler searches for the resource by using the TDE prediction method [14]. The purpose of using this method is to determine the reliability of idle resource. A threshold is set for the resource availability. If the resource is suitable and its reliability is above the specified threshold, the job will be assigned to that resource else alternative available resource need to be found by the scheduler. In case of unavailability of a suitable resource, the job will be held by the scheduler.

B. Rule Based

Such approaches handle prediction by deriving useful rules in order to search relevant resources and then assigning tasks to those available resources.

1) Rough Set Theory

The study in [3] uses rough set analysis to efficiently predict a node's behavior. An online announcer approach is used to determine available resources at present or for future thereby eliminating the need for an inquiry from the grid scheduler. The authors intend to provide solution for resource discovery and task assignment. A new algorithm is proposed in cooperation with rough set tool. The main goal of using rough set is generation of useful rules in order to search for relevant nodes and then assigning tasks to the nodes which are available. The algorithm takes Nodes Information Data Table as input and generate appropriate rules as output. Three attributes (start time, final status of task and completion time) are used as decision attributes. Out of these three attributes, when one of the attribute is used as the decision attribute then the remaining two can be used as conditional attributes depending upon the requirements.

C. Machine Learning

Approaches under this category performs prediction by learning from data. The algorithms build a model based on inputs and then use that to make predictions. Pattern

Recognition and classification under machine learning performs pattern matching. It does so by looking at recent resource availability patterns and then examine for similar trace from the past.

The study in [1] focuses on prediction of machine availability by using the techniques of Bayesian methods and Support Vector Machines. Another contribution is the use of a time series framework for the automation of correlation search and selection of attributes which lead to efficient prediction. The authors also mention that availability and user login behavior are important characteristics which can be predicted in such (desktop pool) environments efficiently and accurately. The important contributions of the authors are towards the:

1. Automation of the process of finding correlations between traces.
2. Selection of predictive attributes in an efficient way.
3. Usage of walk forward evaluation techniques and overfitting elimination.
4. Switching between various classifications algorithms based on accuracy and effective demands.

1) Pattern Recognition & Classification

The authors in [8, 9] presented the use of pattern matching technique for resource availability prediction. The predictions are done in two dimensions. One of them as instance based where the availability is determined and the other is duration based which involves making predictions on a specific duration. The Austrian grid data trace is examined for pattern matching purpose. The resources were classified into 3 groups as dedicated, temporal and on demand based on their availability.

To rank grid resources and provide optimized resource selection, static and dynamic comparison of resources is provided by using various metrics. For static comparison among resources Mean Time Between Failure (MTBF) and Mean Time to Reboot (MTR) were used whereas for dynamic comparison resource stability and dependability measures were taken. The authors suggested metrics are different and novel from others as previously the comparisons were made by considering only daily and hourly resource availability patterns which did not prove to be much accurate. After observing various availability patterns, two classes were identified. The first one as having lower availability at weekends and higher availability during working week days and second as vice versa. Boyer Moore string matching algorithm was used for pattern matching which showed good performance than others. The major contribution was to compare resources based not only on their daily and hourly availability but also according to MTBF, MTR dependability for different jobs.

2) Lazy Learning /Instance Based Learning

A study of resource availability prediction in Enterprise resource Grids is presented in [10]. The authors proposed to

use Jacard Index using lazy learning to predict the resource availability. The technique is applied on the data trace taken from Microsoft Corporate Network and Planet Lab test bed.

The trace is divided into two data sets labeled as training and test sets respectively, each comprising of binary values of 0 or 1 which indicates non-availability and availability of resource respectively. The training data set is split into two windows of same size named as window furthest and window recent. A Jacard Index (JI) is computed by taking the ratio of number of shared attributes between the two sets of data. After calculating JI for the window furthest, the prediction availability value of that window is selected which will be the next data item after the window ends. The window is then slided next and the JI is calculated for the next window furthest along with the availability prediction of that window. This process continues till the window slides to the second last element of the training data set. This technique uses a fixed window size of 3 elements. In case, window size is dynamic then the prediction value will also show variations. So, to handle this scenario, the authors proposed to apply majority voting to generate a single prediction value for windows with different sizes but having same JI. The study in [6] address the challenge of predicting the response time of application executing on the grid. According to the authors, response time of an application can be used to identify the availability of a resource in an Enterprise Desktop Grid Computing (EDGC). To improve prediction accuracy, authors suggest to use the state of a resource. Two prediction techniques are presented which are Statistical-Instance Based Learning (S-IBL) and Slow Down of an Application based on Processing and Networking Performance (SdPN). S-IBL is used to consider the state (load conditions) of the resource based on statistical data mining. SdPN is used to dene an applications slow down based on network and resource performance. A simulation engine is used to define that slow down. Additionally a Self-Adjustable Correction Model (SAC) is developed. S-IBL uses past experience for deriving predictions. An experience has input and output features and refers to something happened in the past. Input features identify the conditions under which an experience was occurred and output features refers to the results under those conditions. A data base of all experiences is made. A query is then provided to the data base to match with input features to predict the estimated output features. The query is resolved by determining its similarity with the experiences in the database. The experiences showing relevance to the query, their output features are used to predict query output features.

D. Time Series

These methods in time series analysis are used to extract useful statistics from the data. Models are used for the prediction of future values based on previously observed values. The model performs prediction by first learning which includes resource categorization based on past history and then make predictions for the availability of a resource at a particular time interval.

A Local User Pattern Analyzer (LUPA) architecture is proposed in [5] which has three subsystems namely, data collection, pattern analyzer and predictor. Data collection gathers CPU and RAM usage for every five minutes. Pattern analyzer implements clustering and predictor is responsible for run time predictions based on resource usage. This module is initialized on each machine to allow its resources to be used in the grid. For this architecture, Usage Pattern Analysis (UPA) was used to discover the local usage pattern of a resource. To discover patterns of use, cluster analysis is done on the resources past record. UPA considers resource usage as an object which is a vector of values representing time series of machines resource use. The main resources considered for machine allocation decisions are CPU use and RAM availability. The process is divided into learning and prediction phases. The learning phase categorizes resource use based on recent history. This is done by collecting large amount of data objects. Fixed numbers of Clusters (k) are made from this data. A prototypical object is calculated for each cluster which represent the class. The output will be k prototypical vectors. UPA requires many parameters which are: no. of clusters, data normalization, and computation of prototypical element, clustering algorithm and similarity measure.

E. Data Mining

Clustering and classification are the main prediction approaches under this category. Resource availability is determined by first making groups (clusters) of the availability patterns according to the common resource usage. This information is then used by the classifier and then predictions of resource future availability are made by supplying test data to the classifier. A study to determine and assess predictive methods for ensuring resource availability is presented in [2]. The main goal is to predict that a number of N hosts will be available for a time T. The authors also focus on determining the factors affecting prediction error and determining resource predictability indicators. Naive Bayes and Decision trees predictive model were used for prediction purposes. A sample in training and test data is represented as a binary string (01) as it refers to one hour duration. A prediction is computed at time T. The author consider the interval $[T, T+p]$ to determine the complete availability against partial or complete unavailability where p refers to prediction interval length and its values are [1,2]. The prediction accuracy is quantified by a ratio called prediction error. The prediction interval length strongly affects the prediction accuracy.

The study of resource availability prediction in P2P network in [11] showed that the major problem is lack of central server management for keeping track of available or assigned resources. Therefore, the authors proposed an architecture for resource selection by using group availability data. The main idea is to classify the resources in groups (based on their common usage patterns) and then predicting their availability.

F. Function Approximation

These techniques simulate a target value, which is expected to be the output of an unknown function of measured system variables as input data.

1) Stochastic Model

A computing system using grouping of resources to achieve reliability is proposed in [15]. The study presents the idea of grouping resources for availability in grid. In literature, to provide reliability for application execution, a replicated execution model is adopted in which one version of application executes on primary resource and the other is executed on back up or standby resource. The authors provided a comparison of their proposed computing model having resource repairing facility with the existing replicated model for resource reliability and availability. The availability of both computing systems is modeled using Markov Model. As there is a probability of failure of both primary and back up resources, this may lead to a failure of application execution. As an alternative, the application may be scheduled to execute on multiple grid resources in parallel where all of them will be considered as primary. If the execution completes on one of the resources, it will be considered complete. The problem with this approach is underutilization of grid resources, as majority of resources are kept to execute the same application. Additionally, in such replicated systems, no mechanism of resource repairing is provided. The resource might degrade or become unavailable with the duration of application execution.

So, for reliable execution of application in the grid [15] presents a computing system with resource repairing to ensure application execution. The repairing is done by replacing failed resources with matching available resources. Initially all resources are made available in the grid. A resource manager is responsible to manage the resources by receiving resource request and then matching the requested resources. Resources are organized in a primary back up manner for application execution. A primary resource executes the application and the other resources act as back up. During application execution, resources transit between the states of 1(available) and 0(unavailable). If a primary resource becomes unavailable, the application is migrated to the backup resource, which is then designated as primary. Application execution is not affected if a backup resource turns unavailable. The repairing of a resource is done by replacing failed resources with matching available resources.

III. CONCLUSION

This survey paper was an attempt to discuss in detail the various approaches and methods to address the challenge of resource availability prediction in the grids. An extensive review of various techniques with their accuracy levels along with their pros and cons has been shown in Table.1. Additionally, the techniques are classified according to a taxonomy as exhibited in Figure.1.

REFERENCES

- [1] Artur Andrzejak, Patricio Domingues, and Luis Silva. Predicting machine availabilities in desktop pools. In Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, pages 14. IEEE, 2006.
- [2] Artur Andrzejak, Derrick Kondo, and David P Anderson. Ensuring collective availability in volatile resource pools via forecasting. In Managing Large-Scale Service Deployment, pages 149161. Springer, 2008.
- [3] Asgarali Bouyer, Ehsan Mohebi, and Abdul Hanan Abdullah. Using self announcer approach for resource availability detection in grid environment. In Computing in the Global Information Technology, 2009. ICCGI'09. Fourth International Multi-Conference on, pages 151156. IEEE, 2009.
- [4] John Brevik, Daniel Nurmi, and Richard Wolski. Automatic methods for predicting machine availability in desktop grid and peer-to-peer systems. In Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on, pages 190199. IEEE, 2004.
- [5] Marcelo Finger, Germano C Bezerra, and Danilo R Conde. Resource use pattern analysis for predicting resource availability in opportunistic grids. *Concurrency and Computation: Practice and Experience*, 22(3):295313, 2010.
- [6] Josep L Lerida, Francesc Solsona, Pordio Hernandez, Francesc Gine, Mauricio Hanzich, and Josep Conde. State-based predictions with self-correction on enterprise desktop grid environments. *Journal of Parallel and Distributed Computing*, 73(6):777789, 2013.
- [7] James W Mickens and Brian D Noble. Exploiting availability prediction in distributed systems. *Ann Arbor*, 1001:48103, 2006.
- [8] Farrukh Nadeem, Radu Prodan, and Thomas Fahringer. Characterizing, modeling and predicting dynamic resource availability in a large scale multi-purpose grid. In Cluster Computing and the Grid, 2008. CCGRID'08. 8th IEEE International Symposium on, pages 348357. IEEE, 2008.
- [9] Farrukh Nadeem, Radu Prodan, Thomas Fahringer, and Vincent Keller. Availability-based resource selection risk analysis in the grid. Technical report, CoreGRID Technical Report, Number TR-0169, 2008.
- [10] Mustazur Rahman, Md Raul Hassan, and Rajkumar Buyya. Jaccard index based availability prediction in enterprise grids. *Procedia Computer Science*, 1(1):27072716, 2010.
- [11] Karthick Ramachandran, Hanan Lutyia, and Mark Perry. Decentralized approach to resource availability prediction using group availability in a p2p desktop grid. *Future Generation Computer Systems*, 28(6):854860, 2012.
- [12] Xiaojuan Ren, Seyong Lee, Rudolf Eigenmann, and Saurabh Bagchi. Prediction of resource availability in ne-grained cycle sharing systems empirical evaluation. *Journal of Grid Computing*, 5(2):173195, 2007.
- [13] Brent Rood and Michael J Lewis. Multi-state grid resource availability characterization. In Proceedings of the 8th IEEE/ACM International Conference on Grid Computing, pages 4249. IEEE Computer Society, 2007.
- [14] Brent Rood and Michael J Lewis. Resource availability prediction for improved grid scheduling. In eScience, 2008. eScience'08. IEEE Fourth International Conference on, pages 711718. IEEE, 2008.
- [15] Major Singh and Lakhwinder Kaur. Resource grouping in grid environment towards the availability and reliability of computing service. *Journal of Advanced Computing*, 1:18, 2013.
- [16] Jun Zhang and Chris Phillips. Job-scheduling with resource availability prediction for volunteer-based grid computing. In London Communications Symposium, LCS. Citeseer, 2009.
- [17] Hu Zhoujun, Hu Zhigang, and Liu Zhenhua. Resource availability evaluation in service grid environment. In Asia-Pac Service Computing Conference, The 2nd IEEE, pages 232238. IEEE, 2007.

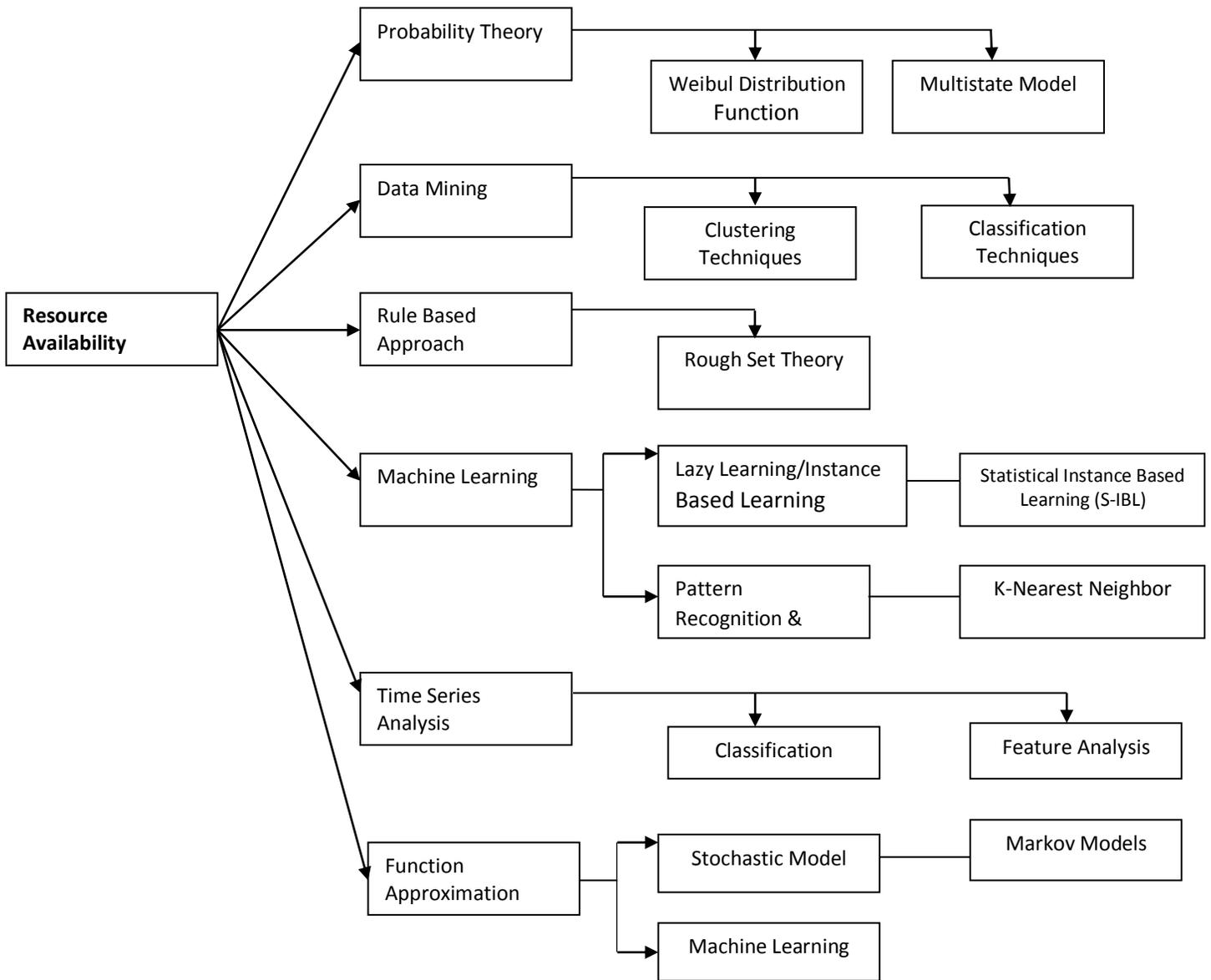


Fig.1. Taxonomy For Classification of Resource Availability Prediction Techniques in Grid

TABLE 1. A COMPARISON OF RESOURCE AVAILABILITY PREDICTION TECHNIQUES IN GRIDS

No.	Year	Prediction Method/Technique	Techniques Taxonomy	Prediction Accuracy	Pros	Cons
1	[6] 2013	S-IBL (Statistical Instance Based Learning)	Statistical /Data Mining	35% increase in prediction accuracy	Potential to adjust to changes in system load	Higher cost of Hybrid Model
2	[15] 2013	Markov Model	Machine Learning	Proposed system showed stable reliability for different application	1. It achieved steady state availability for longer durations 2. Resource repairing feature	None
3	[11] 2012	Resource Usage Pattern	Probability Theory/Data Mining Scheduling Algo's	Accuracy in terms of timings comparison (like availability for hours/days etc)	Peer 2 Peer grid studies of non-dedicated machines	Human Intervention to setup the infrastructure
4	[5] 2010	Use Pattern Analysis (UPA)	Pattern Classification, Machine Learning	UPA showed 75.6%	Use of Usage Pattern Analysis	It is preferred if data collection of more than 21 days is available
5	[10] 2010	Jacard Index Calculation by utilizing Lazy Learning ,Majority Voting	Artificial Intelligence	96.87% for Microsoft data and 99.74% for PlanetLab data	1. Jaccard Index 2. System is adaptable to the newly arrived data	If window size is dynamic, it assumes that each window has same JI and then calculates the Prediction value based on majority voting.
6	[3] 2009	Rough set approach	Data Mining	98.30%	Resource availability detection based on online-announcer with out any inquiry from Grid scheduler	No real data used
7	[16] 2009	Transitional N-Day (Equal transition weights predictor)	Multi-state model (Probability model)	Depends on T (Resource Reliability Threshold) if T is <50% then the accuracy is approx 73%.	New job scheduling algorithm based on Transition N day with equal Transition Day(TDE)	Including more history data for days donot provide better prediction
8	[14] 2008	multi-state availability; simple probability, Transitional N-Day with Equal Transition Weights (TDE) & The Freshness Weighting Scheme (TRF)	Data Mining	78.3% for TDE & 77.3% for TRF	New multi-state availability prediction algorithms	Including more history data for days/hours decreases the performance
9	[8] 2008	Bayes' Rule , Nearest Neighbour	Pattern Recognition and Classification	90% (Instance Based), 70% (Duration Based)	1.Instance & duration based prediction 2.Static & Dynamic comparison of resources	Accuracy decreases if more distant historical data is included
10	[2] 2008	Naïve Bayes & Decision Tree	Data Mining/Machine Learning	Evaluates Success rate in terms of redundancy & no. of hosts, redundancy of 35% can achieve success rate of 95%	1. Investigating factors influencing prediction error like amount of training data, host type & prediction interval length 2.Determining indicators for resource predictability	None
11	[13] 2007	Resource availability states	Multistate Model	Measured in terms of % of time when predictor accurately estimates the next particular state	failure-aware predictive grid scheduling	Linear decrease in accuracy with increase in duration of interval
12	[17] 2007	Strange Theorems	Probability Theory	78%-97% for each metric	Fast Prediction algorithm based on Probability Theory	No. of used parameters is low

13	[12] 2007	Semi Markov Model	Data Mining	86.50%	Proactive management of guest jobs with improved response times	1. Slightly worse predictions on Smaller Window size. 2.Unadaptability of linear time series models for long term predictions
14	[1] 2006	Bayesian Methods, Support Vector Machines	Machine Learning	Mean Squared Error(mse) is the evaluation criteria and shown good results	No power on/off policy makes prediction easy	The machines had been used independently by the users, so no inter-machine correlations were used in predictions
15	[7] 2006	Right Now Predictor, State based predictor, Hybrid Predictors etc	Signal Analysis & Information Theory	PlanetLab accuracy=95% and Microsoft accuracy=87.0%	Use of predictors for tracking uptime state per node	None
16	[12] 2006	Semi Markov Model	Machine Learning	>86.5%	Development of Multistate model by applying SMP for the prediction of temporal reliability	1. Slightly worse predictions on Smaller Window size. 2.Need for deciding a suitable training data set size
17	[4] 2004	Weibull Distribution Function	Probability Theory	95%	Automatic methods to forecast machine availability in peer to peer systems and desktop Grids	Results are useful only for specific application domain



© 2016 by the author(s); licensee Empirical Research Press Ltd. United Kingdom. This is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license. (<http://creativecommons.org/licenses/by/4.0/>).