**Proceedings of the 9th IASTED International Conference**
**Parallel and Distributed Computing and Networks (PDCN 2010)**
**February 16 - 18, 2010 Innsbruck, Austria**

# TASK EXECUTION AVAILABILITY PREDICTION IN THE ENTERPRISE DESKTOP GRID

Toktam Ghafarian-M.[1] , Hossein Deldari[2]
Department of Computer Engineering,
[1]Ferdowsi University of Mashhad ,Khayyam Institute of Higher Education
[2] Ferdowsi University of Mashhad,
Iran,
ghafarian@stu-mail.um.ac.ir , hd@ferdowsi.um.ac.ir

**ABSTRACT**
Desktop grid uses an idle cycle of desktop PC's in Intranet and Internet environments for large-scale computation. Heterogeneity and volatility of hosts within this environment are lead to consider issues of availability of resources. Resource availability is critical for the reliability and responsiveness of services. In this paper, two prediction systems have been introduced for task execution availability of resources in the enterprise desktop grid platform. The first one is based on cellular automata and the other one according to Bayesian network. The accuracy of proposed prediction systems is evaluated via four real desktop grids. In spite of highly volatile environment, the meaningful and robust prediction results have been gained. A comparison between two prediction systems indicates that cellular automata have a better behaviour than Bayesian network. It can predict the behaviour of resources in the desktop grid with the prediction accuracy of %95.5.

**KEY WORDS**
Desktop grid, Availability prediction, Cellular automata, Bayesian network

## 1. Introduction

The spread of computational grid [31, 32] and commercial success of peer-to-peer file sharing systems introduce the use of volatile desktop machines for computing and storage platform. There are some differences between classical grids and desktop grids, some of them are as follows:
1. Classical grids such as clusters and multiprocessors are dedicated to the kind of jobs they run, whereas; desktop grids are not [31].
2. Classical grids consist of a limited number of high-end, dedicated resources though there are a huge number of light-weight and volatile resources in the desktop grid platform [31].
3. In contrast to classical grid systems, lightweight desktop grids provide only a thin layer of abstraction over the resources they manage [31].
4. Desktop grid computing is much more cost-efficient than classical grid computing in terms of the cost of hardware and setup [14].
Desktop grid, in which cycles are scavenged from idle desktop computers, can combine power of hundreds to thousands desktop systems to represent a substantial computing resource. In recent years, volunteer computing (desktop grids) and cloud computing have emerged as two cost efficient platforms. Although these two platforms have similar principles, such as transparency, there are some main differences between them as follows:
1. There is a configurable environment in clouds in terms of software stack with the Xen virtual machine (VM) [22] forming the basis of EC2. Whereas; the use of VM's in volunteer computing (VC) systems is an active research topic [23, 24].
2.Clouds represent a homogeneous resource pool; whereas ,the resources in VC systems are heterogeneous and not transparent to VC application developers [21].
3.Dedicated and non pre-emptive resources in clouds guarantee Quality-of-service in this platform. By contrast, with volatile Internet resources in the VC platform, there is no guarantee for data access or storage and it remains as an open research problem [21].
Desktop grids can execute embarrassingly parallel problems. One of the popular projects in this case is SETI@home[15] that uses over 20 TeraFlop/sec provided by hundreds of thousands of desktop PC's. The other one is the Bovine RC5 effort [6] which uses the spare computer cycle. The major drawback of desktop grids lies in the volatility of resources. The owner of desktop resource has control over the process that runs on it, due to its connectivity to network and its reboot cycle. Therefore , in this environment the availability of resources is fluctuated over time. Thus, efficiency of using desktop grid environments can greatly benefit from

prediction of resource availability. However, we have different meanings for term "availability" in different contexts [2]. According to [1, 29], three type of availability can be defined as described below:

1. Host availability: This type indicates whether a host is reachable or not, which corresponds to the definition of availability in [5, 3, 2, 7, 9]. Some factors include power failure, or a machine shutoff, reboot, or crash cause host unavailability.
2. CPU availability: Corresponding to the definition in [4, 6, 10, 8, 11], this is a percentage value that quantifies the fraction of the CPU that can be exploited by a desktop grid application. Some factors such as system and user-level compute-intensive processes affect CPU availability.
3. Task execution availability: according to a desktop grid worker's recruitment policy, this is percentage value that indicates whether a task can execute on the host or not. Some factor such as prolonged user keyboard/mouse activity, or a user compute-bound process can cause this unavailability.

In this paper, the term availability means task execution availability. The main motivation for this work is to introduce two prediction systems for task execution availability in the enterprise desktop grid platform. The result of two prediction systems is tested over four real desktop grids. Indeed, the resource availability characteristic and future availability prediction of resources can help scheduler to select suitable resources according to the resource requirement of application. In such a way, for the application without check pointing mechanism, scheduler can choose steady available resources; meanwhile, for the heavy check pointing application, longer availability duration resources can be selected. The paper is further structured so that, in the next section, a review of the previous works has been presented. Section 3 defines the proposed prediction systems. In section 4 empirical evaluations and experimental results have been discussed and finally in section 5 a conclusion has been drawn.

## 2. Backgrounds

Some efforts have been made in the context of availability prediction system in the desktop grid platform such as the works that is described in this section:

J. Brevik et al. [19], have examined the problem of predicting machine availability in the desktop and enterprise computing environments. Weibull methods as parametric model fitting technique and Resample, Binomial methods as two non-parametric prediction techniques have been examined in their work. They compared the accuracy of these models in predicting the quartiles of empirically observed machine availability distributions. Their result has been indicated that a non-parametric method based on a binomial approach generates successful predictions 96.0% of the time using only 20 measurements.

J. Wingstrom , H. Casanova [13] have used Log-normal and Weibull methods for modeling resource availability data. They have shown that in many cases Log-normal is a better choice for modeling availability data as it is particularly better at generating data points for small values.

W. Enders [34] has suggested the other approaches for prediction from econometrics and using time series models based on auto-regression and moving averages such as ARIMA and ARFIMA.

Another class of prediction methods (used as one method in this study) has been used classification algorithms known from data mining [15]. In this class, some attributes from past samples together with a correct classification (i.e. prediction value) are first delivered to the classifier and used to build an internal model. Afterward, other requests to the classifier with the values of the same attributes can give a prediction as the response.

Another framework for resource prediction in computer systems is the Resource Prediction System (RPS) [16]. This framework has designed, has built and has evaluated a system that predicts behaviour of resources in the distributed systems. RPS has library and several tools for offering several models like MEAN, NEWTON, ARIMA and wavelet methods. But the main drawback of it has been very short-term prediction and lack of correlation analysis and high computational activity for ARIMA model creation.

F. Nadeem et al. [17] have proposed two methods from pattern recognition (Bayes' Rule) and classification (Nearest Neighbor Rule) for resource instance availability and duration availability predictions on multipurpose grid. These methods have gained more than %90 and %70 accuracy for instance and duration accuracy by using different predicator and different amount of historical data.

In the work have done by A. Andrzejak et al. [18] five classification algorithms (Naive Bayes , complement class Naive Bayes classifier , John Platt's sequential minimal optimization algorithm for training a support vector classifier (SMO) , k-nearest neighbors classifier (IBk) , C4.5 decision tree classifier (J48)) have applied for prediction of resource availability in the desktop grid. In their experiment, for all algorithms, the SMO classifier has been produced the smallest mean squared error, and performing best.

Ren et al. [12] have presented new techniques for predicting the availability of resources in the fine-grained cycle sharing systems. They have exploited daily pattern of resources from history of them. Their model is based on a semi-Markov Process (SMP) and their experimental result is shown an average prediction accuracy of 86.5%.

## 3. Proposed Prediction Systems

In this paper, two prediction systems have been proposed for task execution availability in the desktop grid platform. The former is based on cellular automata and the latter is based on Bayesian networks. In each prediction system, there are only two states for each

resource: available or unavailable and no other observable states between them. In the following subsections, these two prediction systems have been discussed:

### 3.1 Cellular Automata-Based Prediction System

Cellular automata [25, 26, 27] (CA) consist of an array of cells which interact with each other. Each cell has a state that this state can be chosen from a finite number of states. At discrete time steps, all cells update their states in parallel and synchronous or asynchronous way depending on their current state and those of their immediate neighbours. In the update step, all cells can be used the same deterministic update rule, which lists the new cell states for each possible local neighbourhood configuration. This update process is then repeated for a certain number of time steps.

In the simplest case, one dimensional CA (1-D CA) with radius 1 consists of a line of cells with two adjacent cells as neighbours (immediate left and right neighbour). Fig. 1 represents a possible update rule for this kind of CA with radius 1 .It determines possible arrangement of states for the cell and its two neighbours. Since each cell has two possible states (0 or 1),the update rule contains $2^3 = 8$ rules . A value after each arrow determines the next state of each cell, according to the cell state and the state of neighbours.

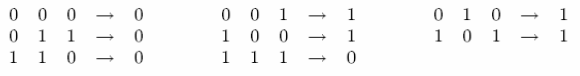


Figure 1. update rule of 1-D binary CA with radius 1

In this particular update rule, four configurations 000, 011,110,111 are mapped to 0 and the other ones are mapped to 1. Of course, the 0s and 1s that indicate new cell states could have been assigned differently; therefore, there are $2^8 = 256$ ways of constructing update rule of this automaton.

Fig. 2 shows a simple example of how cells in CA lattice change their states according to the above update rule. For example, the second cell in the lattice has a local neighbourhood configuration (011) (first arrow), and according to the given update rule, it will be white (0) at the next time step (second arrow). The fifth cell has a local neighbourhood configuration (001), and thus it will be black (1) at the next time step (second arrow). Similarly, all cells are changed to their new state simultaneously according to the same update rule given above.

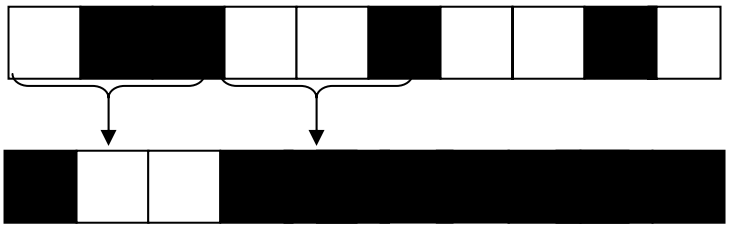

Figure 2. cell state update according to the above update rule of Fig. 1

In case of a finite CA as in the above example, we also need to specify boundary conditions. There are two boundary conditions: null boundary and periodic boundary. In the null boundary condition, additional cell with state 0 is added on either side of the lattice (these boundary cells are not updated) meanwhile, a periodic boundary CA is a CA in which the extreme cells are adjacent to each other.

Nevertheless, many variations on this basic scheme exist, such as higher dimensional (2-D, 3-D, …) lattices, a larger number of states (>2), or larger neighbourhood sizes (a radius of 2, 3,…) the basic principles that are updating of cell state based on local neighbourhood configurations remain the same in all cases.

Proposed cellular automata-based prediction system uses a two dimensional binary cellular automata with periodic boundary conditions. In this system, all resources in the desktop grid have been arranged into 2-D cellular automata as is shown in Fig. 3. In this figure, each resource in the desktop grid is associated with one cell in CA. Since only two states (available or unavailable) have been considered for each resource of desktop grid, thus, cellular automata is binary. Zero indicates resource unavailability and one indicates resource availability. As shown in Fig. 3, Von Neumann neighbourhood configuration is used for defining cell neighbourhood of each cell. Since any resource is assigned to each cell randomly, the neighbourhood of each cell will be random and it is not based on physical connection. In this configuration, the next state of each machine is computed according to the state of cell itself and the state of four neighbourhoods around it. In this prediction system, the update rule of CA is gained via learning phase which is described in section 3.4. Each configuration in the update rule has five bits , one bit located in the middle of configuration ,is the current state of cell and two pairs of bits located in the left and right sides define states of cell neighbourhoods. $2^5 = 32$ rules have been defined in the system according to the different combination of these five bits with two possible values. Each cell changes its state every time-step based on the arrangement of states for the cell and its four neighbours via update rule. The state of every cell in each time-step determines the corresponding resource state. All cells of cellular automata go to the next state synchronously.

### 3.2 Bayesian Network-Based Prediction System

Bayes theorem determines a method to compute the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis and the observed data itself [33]. In this theorem, prior probability *P(h)* describes the initial probability that hypothesis *h* holds, before observing the training data. *P (D)* defines the prior probability in which the training data D will be observed and *P (D/h)* describes the probability of observing data *D* given some world that hypothesis h holds.

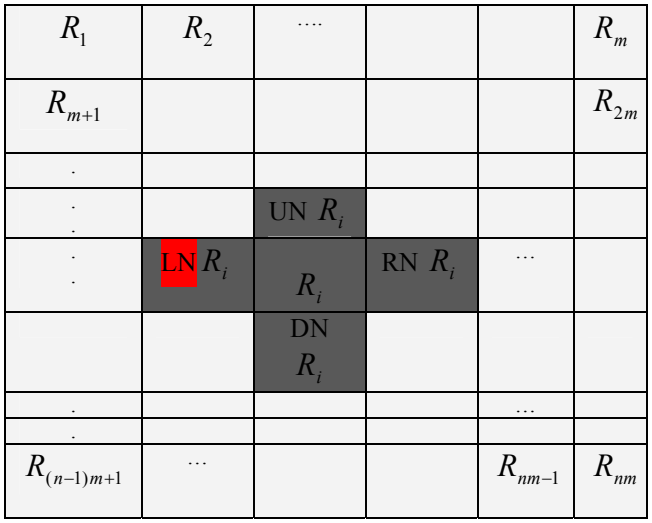


Figure 3. Proposed 2-D cellular automata for prediction system
R : Resource , UN: Up Neighbour , LN:Left Neighbour ,RN:Right Neighbor , DN :Down Neighbour

*P (h /D)* is called the posterior probability of hypothesis *h* holds after we have observed the training data *D*.

$$P(h/D) = \frac{P(D/h)P(h)}{P(D)} \quad (1)$$

In this paper, Bayesian network (BN) has been used for task execution availability prediction in the desktop grid. In fact, the hypothesis which is considered here is availability prediction. This theorem can exploit availability characteristics of each resource from the past history of it. In order to use BN for availability prediction, we should select different availability feature specific to time from availability traces. Some assumptions have been described below for using this: Let $w_1, w_2$ are two classes determining task execution availability state of resources in the desktop grid. Suppose $P(w_1)$ defines the probability of task execution availability and $P(w_2)$ denotes the probability of task execution unavailability. The prior probability of $P(w_1)$, $P(w_2)$ are determined from the learning phase that is described in section 3.4.

Let $P(x/w_i)$ for $i = 1,2$ defines the class conditional PDFs denoting the distribution of feature vector x in each class and it can be computed during the learning phase. Feature vector contains our availability feature used for prediction. In this paper four features have been considered in the feature vector. The first one is hour of day; a day has been divided into two subsections: business hour and non business hour. The daily time period during which all resources are most volatile has been considered as business hour and the other one as non business hour. The range of hours has been defined these two subsections is different from one desktop grid to other one. The second, third and fourth feature denote the availability history of corresponding resource in the last three time steps. Therefore, x= [hour of day; last time step; second last time step; third last time step]. In this case, all of the features in the feature vector can take only two discrete values (zero or one). For feature hour of day in the feature vector, zero indicates non business hour and one indicates business hour; meanwhile, for all other three features, zero indicates unavailability and one indicates availability in the history.

In this prediction system, for each feature vector x , "2" should be computed in order to predict the next state of resource.

$$P(w_i/x) = \frac{p(x/w_i)P(w_i)}{p(x)} \quad (2)$$

Where *p(x)* is the PDF of *x* and computed according to the "3"

$$p(x) = \sum_{i=1}^{2} p(x/w_i)P(w_i) \quad (3)$$

Since feature spaces can get only discrete values $p(x/w_i)$ becomes a probability and will be calculated as $P(x/w_i)$. Then Resource state for feature vector x is determined using "2" as:

If $P(x/w_1)P(w_1) > P(x/w_2)P(w_2)$ decide $w_1$ otherwise $w_2$ .

### 3.3 Trace Data Set

Proposed prediction systems have been evaluated over four traces collected from real desktop grid platforms [28,29]. these traces includes SDSC trace collected over 275 hosts for a cumulative period of about 1 month from Desktop PC's at the San Diego Super Computer Center (SDSC). This desktop grid platform deployed the commercial Entropia desktop grid software [8]. In this trace, timeslot sampling for task execution availability of each resource is considered every 3 seconds. Our simulation experiments are deployed the longest continuous period of trace measurements, which was the two-week period between 9/3/03 - 9/17/03.

The other traces, LRI and DEUG traces have been collected over 100 hosts from the University of Paris South. These traces were collected using the open source XtremWeb [30] desktop grid software continuously over about a one month period (1/5/05 - 1/30/05). LRI trace attained from a cluster with a total of 40 hosts used by computer science research group and especially used for running parallel application and benchmarks , whereas , DEUG trace obtained from a classroom with 40 hosts used by first-year undergraduates. In the LRI and DEUG desktop grid timeslot sampling is considered 10 and 20 seconds respectively. In these two traces [28] [29],we also have some virtual host. These virtual host is added in order to increase the number of hosts in this platform; traces of hosts on different days were pooled together to create these virtual host.

And prediction system has been evaluated on UCB trace as well. This trace was collected over 85 hosts over a 46-day period (2/15/94 - 3/31/94) obtained from EE/CS department at UC Berkeley used by graduate student.

Although , in simulation experiment, the largest continuously measured period between 2/28/94 and 3/13/94 has been used. The UCB trace originated from an older data set first was reported in [20], but it was processed to show the availability of the hosts for a desktop grid application [29].

### 3.4 Learning Phase

Learning phase in two prediction systems based on cellular automata and Bayesian network needs to select some samples from the traces which is collected on real desktop grid platforms. In order to select samples from each desktop grid, three candid days are randomly selected. In these 3 days, some samples as long as one hour from business hour and non business hour have been selected. According to [1], business hour for SDSC, DEUG and UCB platform are 9AM-5PM, 6AM-6PM and 10AM-5PM respectively. In the LRI platform, there is no distinction between non-business hours and business hours. Therefore, according to the business hour for all desktop grids, 11AM-12AM for business hour and 22PM-23PM for non business hour have been selected as candidate of these two subsections. These samples have been used for learning two proposed prediction systems. On each desktop grid, leaning samples have been chosen separately. In the CA-based prediction system, CA has been trained with a sample from each real desktop grid platform and extracted separate update rule for each one. In fact, for each configuration of cell state and state of neighbours in the samples, the next state of cell is counted and the following state for that configuration will be based on the majority voting. If in the most of situation, the next sate is unavailable, zero is selected for that configuration otherwise one is selected. Afterward, CA has been equipped with this extracted update rule and has been run some time-step for availability prediction.

In the Bayesian Network based prediction system, in order to compute "2" for each feature vector x, the values of $P(x/w_i), P(w_i), p(x)$ for i=1, 2 should be computed. These probabilities have been computed during the learning phase for the feature vector extracted from the samples.

## 4. Empirical Evaluations

The prediction systems have been evaluated for their prediction accuracy. This accuracy is computed as

$$Accuracy = \frac{(No.\ of\ correct\ predictions)}{(total\ number\ of\ prediction\ queries)} \quad (4)$$

Accuracy has been represented as percentage. Evaluation has been conducted on the four trace data set that has been described in the section 3.3. Prediction accuracy evaluation of two prediction systems has been done in the terms of instance availability and duration availability predictions. In the case of instance availability prediction, a prediction is true if the immediate next state of resource is the same as predicted, otherwise, it is evaluated false. In the case of duration availability prediction, prediction is true if the next state of resource along some duration spends is the same as predicted otherwise it is considered false. Computation of instance availability and duration availability is described in the coming sections.

### 4.1 Instance Availability Prediction

The evaluation of instance availability prediction is done for all of resources in each desktop grid. For one resource, the prediction system is evaluated on all days which are traces were collected on it. One thousands points are selected randomly and the next immediate state on these points has been computed over all of resources. This computation is repeated on all days that traces have been collected. Then, the average of prediction accuracy according to "4" on all days for each resource is reported in following figures. Fig. 4, reports instance availability prediction accuracy in LRI trace dataset. The behaviour of CA and BN almost is the same, but the CA performs a bit better. However, the average of instance prediction accuracy on this dataset is %99.99 for CA and %99.97 for BN.

Fig. 5 also shows instance prediction accuracy for DEUG trace dataset. The average of prediction accuracy for CA is %99.96 and for BN is %99.89, therefore, CA performs a little better than BN.

Fig. 6 represents the behaviour of two prediction systems on SDSC trace dataset. The average of prediction accuracy for CA is %91.66; meanwhile, for BN is %84.77. As it is shown in Fig. 6, the prediction accuracy of CA is better than BN on the most of resources and on the some of resources the behaviour of two systems are the same. Fig. 7 shows the instance prediction accuracy on UCB trace dataset. In this dataset, BN performs a bit better than CA. the average of prediction accuracy on CA is %99.63; just the same on BN is %99.74.

Each point in all of figures shows prediction accuracy on one resource in the corresponding desktop grid. Since resources are managed by volunteers on the desktop grid and behaviour of them is random, it may be some of resources do not obey the general trend of others such as host number 175 on Fig. 4. Although the prediction accuracy on this host is almost %99.73 and very near to the majority of hosts with prediction accuracy %99.99. This is the same for host numbers 10, 42, 69 on Fig. 7.

the similar prediction system on the multi purpose grid [17] is gained instance prediction accuracy more than %90 on average; meanwhile ,the proposed prediction systems on average is gained more than %95 instance availability accuracy. Although , that system is done on the grid with dedicated resources not on the desktop grid with volatile resources. On the desktop grid, such computation for instance and duration prediction accuracy has not been done yet.

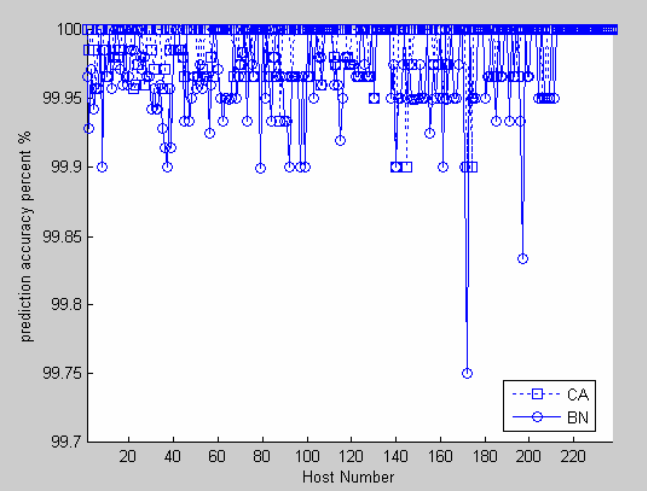


Figure 4. Instance Prediction Accuracy on LRI over All of its Resources

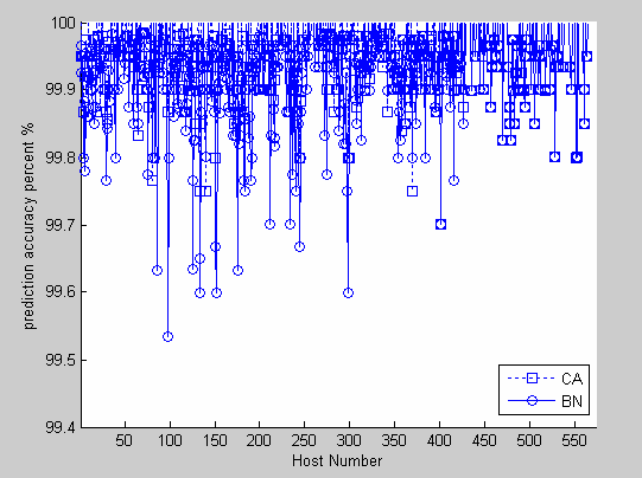


Figure 5. Instance Prediction Accuracy on DEUG over All of its Resources

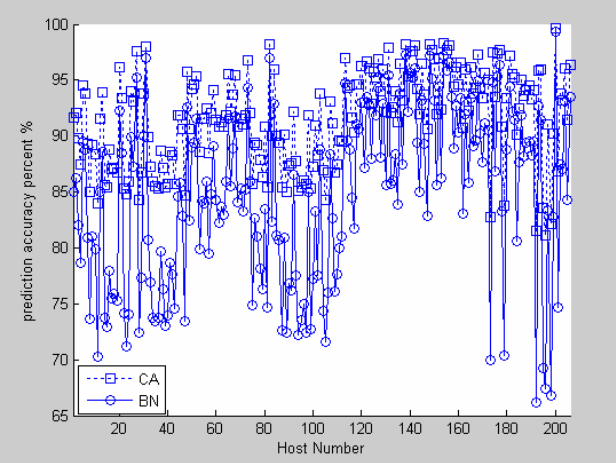


Figure 6. Instance Prediction Accuracy on SDSC over All of its Resources

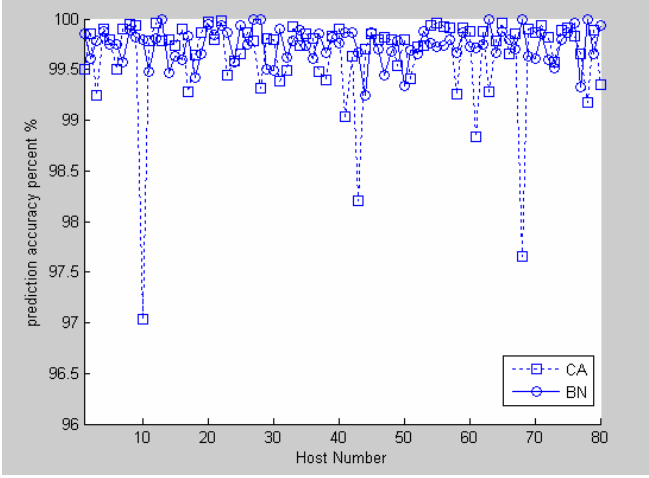


Figure 7. Instance Prediction Accuracy on UCB over All of its Resources

### 4.2 Duration Availability Prediction

The evaluation of duration availability prediction has been done on all resources. On each resource, prediction accuracy has been evaluated for every time slot from 10 min. to 60 min. with the increment of 5 min. For every time slot, one hundred points have been selected randomly as a start point of duration accuracy measurement and the average of prediction accuracy on these points is reported for that time slot and on that particular day. The above process is repeated over all of days on which trace dataset has been collected. Finally, the average of prediction accuracy on all resources over all days has been reported for that time slot. Fig. 8. represents duration prediction accuracy for CA and BN on LRI trace dataset. As it is shown, the behaviour of CA is better than Bayesian network for the majority of time slots. The average of prediction accuracy over all time slots is %99.77, %99.74 for CA and BN respectively. Fig. 9 shows duration prediction accuracy on DEUG trace dataset. In this dataset, CA performs better than BN on all of time slots. The average of prediction accuracy is %97.60 for CA and %97.42 for BN. Fig. 10 represents duration prediction accuracy on SDSC trace dataset. The behaviour of CA based system is very close to BN, but BN perform better than CA. the average of prediction accuracy is %87.66 for CA and %90.27 for BN. Fig. 11 shows duration prediction accuracy on UCB trace dataset. The average of prediction accuracy is %87.66, %87.72 for CA and BN respectively. In this case CA and BN almost have similar behaviour, but in some few cases BN has gained better results than CA. Although, in some cases like UCB and SDSC trace data set, Bayesian network shows a little better result than cellular automata, but the speed of CA is much more than Bayesian network in all of situations.

In comparing to the similar prediction system on the grid [17] which is computed on time duration between 10min. to 24hrs. and duration prediction accuracy more than %70 on average, the proposed prediction system has been

achieved prediction accuracy more than %75 for the same time duration.

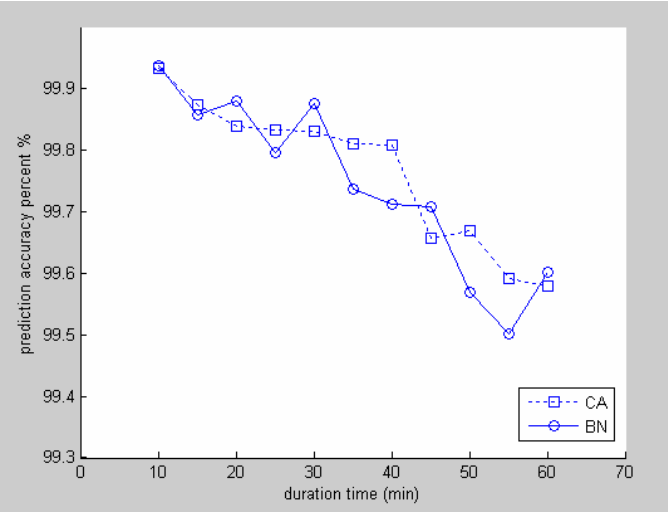


Figure 8. Duration Prediction Accuracy on LRI over 60 min.

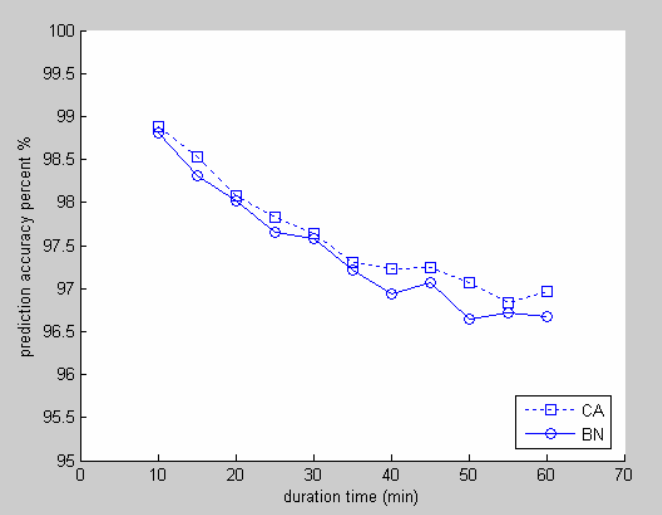


Figure 9 . Duration Prediction Accuracy on DEUG over 60 min.

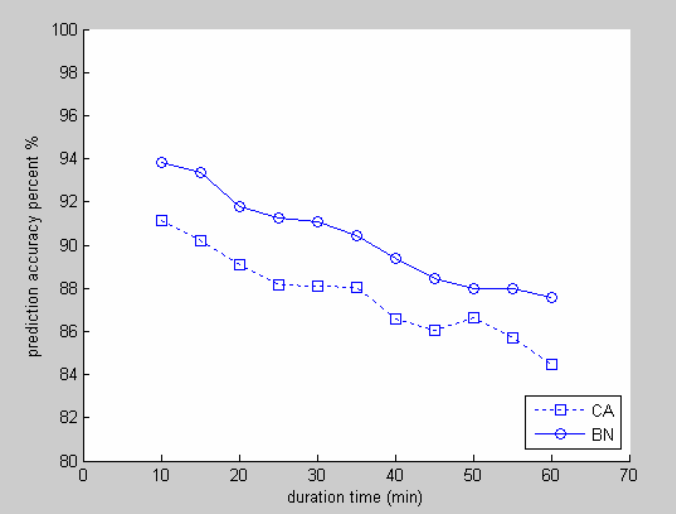


Figure 10. Duration Prediction Accuracy on SDSC over 60 min.

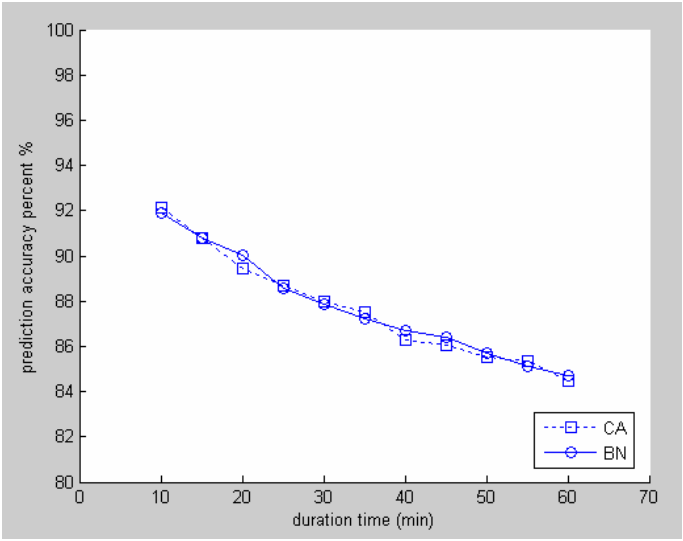


Figure 11. Duration Prediction Accuracy on UCB over 60 min.

## 5. Conclusion

In this paper, two prediction systems for task execution availability in desktop grid have been proposed. The prediction accuracy of two systems based on cellular automata and Bayesian network have been computed over 4 real desktop grid platforms. Behavior of two systems is very close to each other, but in the most cases CA has better results. The average of prediction accuracy for cellular automata and Bayesian network are %95.5, %94.94 respectively. Theses prediction systems can help grid scheduler and resource broker since they need information about resource availability properties and future availability prediction of them, in order to compare and select the most suitable ones. For example scheduler can choose resources with more steady availability for long run application, whereas; more intermittently available resources can be chosen for replicable process. However, future availability prediction of resources can increase efficiency of scheduling system in the desktop grid.

## References

[1] D. Kondo, G. Fedak, F. Cappello et al. , Characterizing Resource Availability in Enterprise Desktop Grids ,*Journal of Future Generation Computer Systems, 23*(7), 2007 , 888-903.

[2] R. Bhagwan, S. Savage, G. Voelker. Understanding Availability. *Proc. 17th International Parallel and Distributed Processing Symp.*,Nice ,FR, 2003,256-267.

[3] A. Acharya, G. Edjlali, J. Saltz,,The Utility of Exploiting Idle Workstations for Parallel Computation ,In *Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, Seattle, Washington , USA, 1997, 225–234.

[4] R.H. Arpaci, A.C. Dusseau, A.M. Vahdat et al. , The Interaction of Parallel and Sequential Workloads on a

Network of Workstations,*Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, Ottawa, Canada,1995, 267–278.
[5] W. Bolosky, J. Douceur, D. Ely et al. , Feasibility of a Serverless Distributed file System Deployed on an Existing Set of Desktop PCs, *Proc. ACM SIGMETRICS International Conf. Measurements and Modeling of Computer Systems*, Santa Clara, CA, USA, 2000,34-43.
[6] H. Casanova, A. Legrand, D. Zagorodnov et al. , Heuristics for Scheduling Parameter Sweep Applications in Grid Environments, *Proc. 9th Heterogeneous Computing Workshop*, Cancun, Mexico, 2000, 349–363.
[7] J. Chu, K. Labonte, B. Levine, Availability and locality measurements of peer-to-peer file systems , *Proc. Of ITCom: Scalability and Traffic Control in IP Networks* , Boston, MA, USA ,2002 , 310-321.
[8] P. Dinda, The Statistical Properties of Host Load, *Scientific Programming*, *7*(3-4), 1999,211-229.
[9] S. Saroiu, P.K. Gummadi, S.D. Gribble, A measurement study of peer-to-peer file sharing systems, *Proc. Multimedia Computing and Networking* , San Jose, CA, USA, 2002 ,259-270.
[10] S. Smallen, H. Casanova, F. Berman, Applying Scheduling and Tuning to On-line Parallel Tomography, *Scientific Programming*,*10*(4), 2002 ,271-289.
[11] R. Wolski, N. Spring, J. Hayes, Predicting the CPU Availability of Time-shared Unix Systems, *Proc. 8th IEEE International Symp. on High Performance Distributed Computing*, California, USA,1999,105-112.
[12] X. Ren, S. Lee, R. Eigenmann, S. Bagchi , Prediction of Resource Availability in Fine-Grained Cycle Sharing Systems and Empirical Evaluation , *Journal of Grid Computing , 5*(2) , 2007,173–195.
[13] J. Wingstrom, H. Casanova , Statistical Modeling of Resource Availability in Desktop Grids, *Technical Report ICS2007-11-01*, Information and Computer Sciences Dept.,University of Hawaii, USA, 2007.
[14] The BOINC Wikihttp://boinc.berkeley.edu/trac/wiki/.
[15] I. H. Witten , F. Eibe, *Data mining: practical machine learning tools and techniques with Java implementations*( Morgan Kaufmann Publishers, 2000).
[16] P. Dinda , Design, implementation, and performance of an extensible toolkit for resource prediction in distributed systems, *IEEE Transactions on Parallel and Distributed Systems, 17*( 2),2006, 160- 173.
[17] F. Nadeem ,R. Prodan , T. Fahringer , Characterizing, Modeling and Predicting Dynamic Resource Availability in a Large Scale Multi-Purpose Grid , *8th IEEE International Symp. On Cluster Computing and the Grid*, Lyon, France,2008,348-357.
[18] A. Andrzejak , P. Domingues , L. Silva , Predicting Machine Availabilities in Desktop Pools , *Proc. 10th IEEE/IFIP Network Operations and Management Symposium*, Vancouver, BC, 2006,1-4.
[19] J. Brevik , D. Nurmi , R. Wolski , Automatic Methods for Predicting Machine Availability in Desktop Grid and Peer to peer Systems ,*Proc. 4th IEEE/ACM International Symp. on Cluster Computing and the Grid*, Chicago, Illinois, USA,2004, 190- 199.
[20] R. H. Arpaci, A. C. Dusseau, A. M. Vahdat et al. , The Interaction of Parallel and Sequential Workloads on a Network of Workstations, *Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, Ottawa, Canada ,1995,267–278.
[21] D. Kondo, B. Javadi, P. Malecot et al. , Cost-Benefit Analysis of Cloud Computing versus Desktop Grids, *Proc. IEEE International Symposium on Parallel&Distributed Processing ,* Atlanta,USA, 2009, 1-12.
[22] P. Barham, B. Dragovic, K. Fraser et al. , Xen and the art of virtualization, *Proc. 19th ACM Symposium on Operating Systems Principles*, New York, USA, 2003, 164–177.
[23] A. Csaba Marosi, P. Kacsuk, G. Fedak et al. , Using virtual machines in desktop grid clients for application sandboxing , *Technical Report TR-0140,* Institute on architectural Issues: Scalability, Dependability, Adaptability, CoreGRID - Network of Excellence, 2008.
[24] A. Ganguly, A. Agrawal, P. Oscar Boykin et al. , Wow: Self-organizing wide area overlay networks of virtual workstations, *journal of Grid Computing ,5*(2) , 2007 ,151–172.
[25] T. To_oli, N. Margolus , *Cellular Automata Machines: a New Environment for Modeling(*Cambridge: MIT Press, 1987).
[26] S. Wolfram , a New Kind of Science(Wolfram Media , 2002).
[27] M. Sipper, the Emergence of Cellular Computing , *IEEE Computer journal, 32*(7), 1999,18-26.
[28] http://mescal.imag.fr/membres/derrick.kondo
[29] D. Kondo, Scheduling Task Parallel Applications on Enterprise Desktop Grids , *PhD thesis*, Dept. of Computer Science and Engineering, University of California ,San Diego, 2005.
[30] G. Ghare ,L. Leutenegger, Improving Speedup and Response Times by Replicating Parallel Programs on a SNOW, *Proc. 10th Workshop on Job Scheduling Strategies for Parallel Processing* , New York,USA, 2004,264-287.
[31] F. Berman, G. Fox, T. Hey, *Grid Computing: Making the Global Infrastructure a Reality*( Wiley and Sons, 2003).
[32] I. Foster , C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*( Morgan Kaufmann Publishers, 1998).
[33] T. M. Mitchell , *Machine Learning*( McGraw-Hill Science/Engineering/Math , 1997).
[34] W. Enders, *Applied Econometric Time Series*, *2nd ed(* Wiley Canada, 2003).