

## Availability-based Resource Selection Risk Analysis in the Grid

*Farrukh Nadeem , Radu Prodan, Thomas Fahringer*  
*Institute of Computer Science University of Innsbruck*  
*{farrukh, radu, tf}@dps.uibk.ac.at*

*Vincent Keller*  
*Ecole Polytechnique Federale de Lausanne, LIN-STI, Switzerland*  
*Vincent.Keller@epfl.ch*



---

CoreGRID Technical Report  
Number TR-0169  
August 19, 2008

Institute on Grid Information, Resource and Workflow  
Monitoring Services

CoreGRID - Network of Excellence  
URL: <http://www.coregrid.net>

CoreGRID is a Network of Excellence funded by the European Commission under the Sixth Framework Programme

Project no. FP6-004265

# Availability-based Resource Selection Risk Analysis in the Grid

Farrukh Nadeem , Radu Prodan, Thomas Fahringer  
Institute of Computer Science University of Innsbruck  
{farrukh, radu, tf}@dps.uibk.ac.at

Vincent Keller  
Ecole Polytechnique Federale de Lausanne, LIN-STI, Switzerland  
Vincent.Keller@epfl.ch

*CoreGRID TR-0169*

August 19, 2008

## Abstract

Resources in the Grid exhibit different availability properties and patterns over time, mainly due to their administrators' policies for the Grid, and the different domains to which they belong, e.g. non-dedicated desktop Grids, on-demand systems, P2P systems etc. This diversification in availability properties makes availability-aware resource selection, for applications with different fault tolerance capabilities, a challenging problem. To address this problem, we introduce new availability metrics for resource availability comparison. We further predict resource availability considering their availability policies. We introduce a new resource availability predictor based on pattern matching through availability *pattern recognition and classification* for resource *instance* and *duration* availability, and compare it with other methods. Notably we are able to achieve an average accuracy of more than 80% in our predictions.

## 1 Introduction

With the growing maturity of the Grid technology, the composition, functionality, utilization and scale of computational Grids continue to evolve. The large scale computational Grid environments and testbeds under centralized administrative controls, such as, Cern LCG, Tera Grid, Grid'5000, Austrian Grid etc. amass hundreds/thousands of resources, which vary considerably in terms of their computation power and their availability patterns. A large number of these resources may be unavailable at any time, mainly due to wide range of policies for *when* and *how* to make their resources available to the Grid. These policies are usually based on resources' usage patterns and owners' other preferences, like shutting down the resources during night. Resources may also be made available to the Grid on demand [1] and may even be temporarily removed partially or completely from the Grid to accommodate other tasks or projects.

The Grid schedulers and resource brokers need information about resource availability properties and predictions about their future availability, besides application execution time predictions on them, to compare and select the most suitable resources. Knowing the resource availability properties, *how* often and *when* the resources become unavailable, they can be more cogent, especially if this knowledge is coupled with information about application characteristics. For example, the applications that do not implement checkpointing mechanism and have long run times require resources with more steady and reliable availability, whereas, the applications which require heavy checkpointing may be scheduled to the resources with a longer availability durations and applications with light checkpointing and easily replicable processes might even be scheduled on less powerful and more intermittently available resources. Similarly, considering resource availability patterns, heavy weight checkpointing could be created on demand rather than periodically, thereby circumventing unnecessary overheads.

---

This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

Predicting the resources' availability and comparing them on this basis is a hard problem due to its dependency on multiple factors like resource's components stability, Grid middleware maturity, varying resource maintenance and manageability. Particularly, the wide range of policies for resource availability in the Grid makes the problem even more challenging. To address this problem, we present in this paper resource availability comparison framework and a new prediction method *Pattern Matching* from *pattern recognition and classifications*, for resource *instance* and *lifetime* availability predictions. We compare resource availability using different metrics, in particular with *resource stability*, which better compares resources for their availability as a function of time. Availability trace from Austrian Grid is undertaken for our present study. Please note that in this paper we use the terms *machine* and *resource* interchangeably.

We compare the effectiveness of our prediction technique by designing and building different predictors that take advantage of different availability properties. Results from the prototype implementation of our system in ASKALON [15] show that more specific information is critical for better predictions. On average, we are able to get more than 95% accuracy in our *instance* availability predictions and more than 75% accuracy in *duration* predictions.

## 2 Austrian Grid Availability Trace

Here we describe the availability trace of resources in a large-scale Grid: Austrian Grid. Austrian Grid is a nationwide, multi-institutional, -administrative and -VO Grid platform, consisting of 28 Grid-sites geographically distributed in Austria, and collectively there are over 1.5 thousand processors. To better expose resource availability properties, we include (external) knowledge of high level resource availability policies for the Grid. Based on these policies, we identify three main classes of resources in Austrian Grid: the *dedicated resources*, which are meant to be always available to the Grid users; the *temporal resources*, the resource from the university labs, are available in the Grid as long as they are turned on; and the *on-demand resources*, which are made available to the Grid only on demand from the Grid users for large scale jobs or experiments. The names of the resource classes reflect their respective availability policies.

The complete resource availability trace, in which each event (for every individual Grid-sites) separated by 5 minutes time represents resource state (available or unavailable), was analyzed for a duration of approx. one year from mid-June 2006 to mid-April 2007. Altogether this trace is comprised of more than 23 million events that occurred on the whole Austrian Grid. In the presented work, a resource is considered available if it is turned on and accessible remotely (e.g. through GRAM), otherwise it is considered unavailable.

## 3 Resource Availability Comparison

The main purpose of resource comparison based on the availability properties is to provide the resource broker and meta-scheduler a ranking of Grid resources to assist the optimized resource selection process for advance reservations and execution planning respectively. We compare the Grid resources for their availability in two dimensions: the static availability comparison through MTBF (mean time before failure) and MTR (mean time to reboot); the dynamic availability comparison through resource stability (for different job durations), and the resource dependability (as a function of time). We argue that the later two metrics are more suitable for resource comparisons as they include better probabilities approximations about resources' availability. These are described in the following sections.

### 3.1 Resource Stability

Stability of a Grid resource for a time duration reflects its suitability for executing jobs of that duration. We define stability (of availability) of a Grid resource for a duration  $t_k$ , having a set of  $n$  durations  $t = \{t_1, t_2, t_3, \dots, t_n\}$ , as:

$$\zeta(t_k) = \sum_{\forall t_i \geq t_k} \left\lfloor \frac{t_i}{t_k} \right\rfloor \times P(t_i)$$

where  $P(t_i)$  denotes the probability of duration  $t_i$  for the resource.

We present the resource stability comparison at resource class level in Figure 1. The on-demand resources have the lowest stability while the dedicated resources have the highest. For job duration of 60 minutes the dedicated resources

have a stability of 44 (more than 8 times that of on-demand resources), and temporal resources exhibit that of 11 (more than 2 times that of on-demand resources). The space limit does not allow us to present individual resource comparisons here.

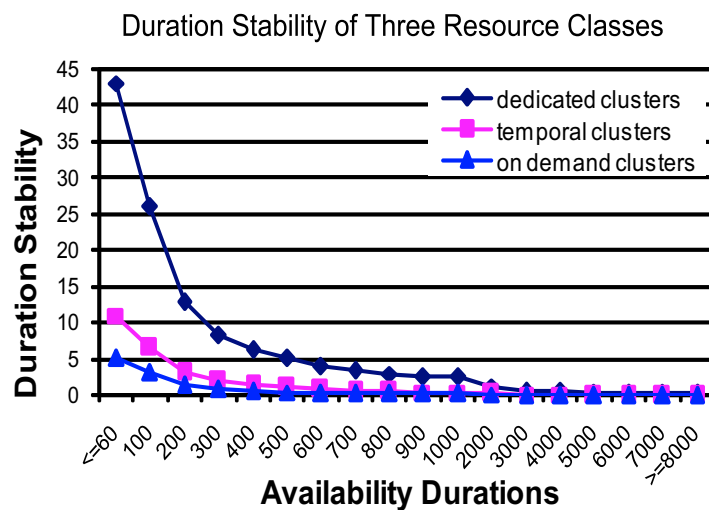


Figure 1: Resource stabilities for jobs of different durations at resource class level

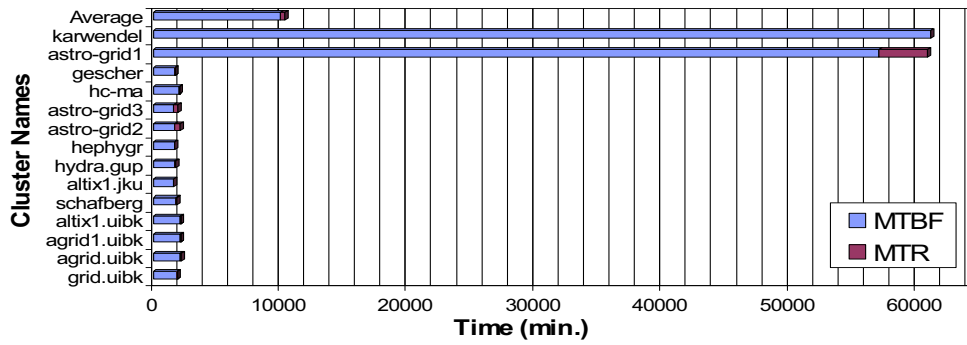
### 3.2 Resource Consistent Availability over Time

In contrast to existing works [9, 4] which use only simple availability properties like daily and hourly availability to compare resources, we consider mean duration of availability as MTBF and mean duration of unavailability as MTR for the Grid resources too. Duration of availability of Grid resources is of critical importance for execution of long jobs. Likewise, duration of unavailability is also an important measure for ahead of time planning for job executions as in advance reservations. We define the duration of availability as the time elapsed between the occurrence of resource turned *on* or recovered after the failure and again turn *off/failed*. Likewise, we formulate our definition for duration of unavailability. It is noteworthy that our definitions coincide with that of time to repair by [14, 13] and of recovery time used in [6]. Figure 2 shows average availability duration(MTBF), and average unavailability durations(MTR), for each of the individual resource classes. The MTBF(MTR) of resources at Grid level, is observed about 4 days(1008 min). At resource classes level, these estimates clearly give a better view of resource duration availability. Intuitively, the class of dedicated resources show the highest (of three classes) MTBF of 7 days on average, with max.(min.) of 42 days (1609 min). These also exhibited the lowest (of three classes) MTR of 45 min., with max.(min.) of 89 (26) min. Temporal resources, at class level showed almost equal MTBF and MTR and highest MTR of the three classes. They showed an average MTBF and MTR of 18 hrs. Max.(min.) of MTBF were 77(6) hrs., whereas those of MTR are 5149(381) min. On demand resources on average showed the lowest (of three classes) MTBF of 269 min. and an MTR of 2944 min. We found on-demand resources with higher MTR than temporal resources. The SD was quite low for both MTBF and MTR in case of dedicated resources, whereas these were quite high for temporal and on-demand resource classes. It is also important to note that the failure durations may include the night hours during which the system administrators of the Grid sites are not available, and the durations for which the machines are automatically turned off. Furthermore, it may also include the durations for Grid-sites' maintenance/repair etc.

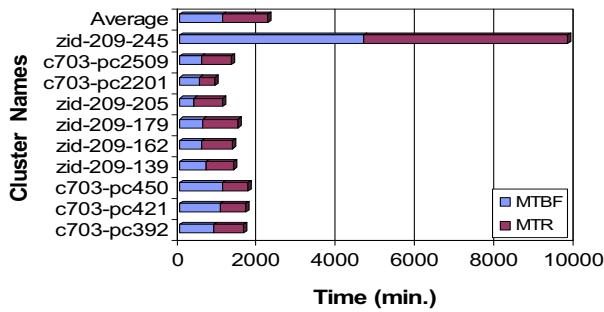
### 3.3 Resource Dependability

The resource dependability describes the extent to which a resource can be depended for being available. We define resource dependability as a function of time duration, as its probability of availability for a given time duration or

### MTBF-MTR Comparison of Dedicated Resources



### MTBF-MTR Comparison of Temporal Res.



### MTBF-MTR Comparison of on-Demand Res.

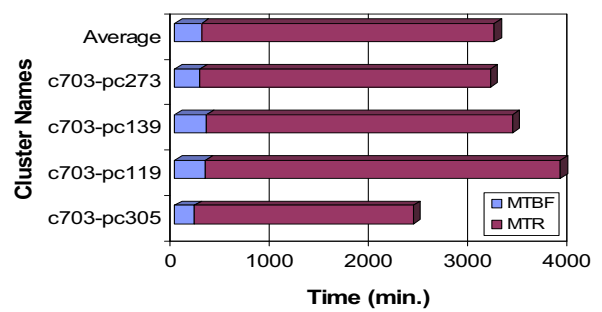


Figure 2: MTBF and MTR comparison of resources in three resource classes

higher. For a resource exhibiting a set of different consistent life times  $l = l_1, l_2, l_3 \dots l_m$

$$\eta(t) = \sum_i^m P(l_i) | l_i > t$$

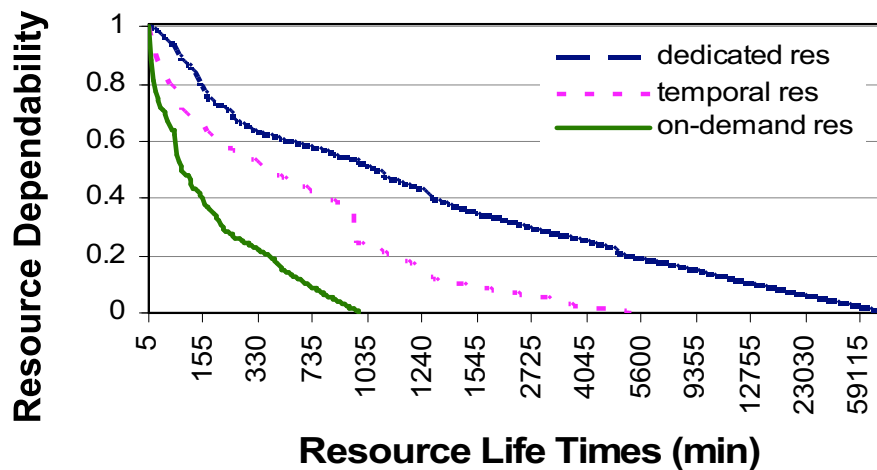


Figure 3: Dependability comparison of three resource classes

A comparison of resource dependability at the level of resource classes is shown in Figure 3 as a sum of probabilities of the different lifetimes( $t$ ) and their greater lifetimes  $P(x \geq t)$ . On average, classes of dedicated, temporal and on-demand resources have a maximum dependability of approx. 65 days, 4 days and 1000 min. respectively. It is interesting to note that dedicated resources on average have 35% dependability for a job lasting a little more than a day. The temporal resources are 50% dependable for the jobs of 6 hours or more, and on-demand resources have 50% dependability for the jobs lasting more than 90 min.

## 4 Resource Availability Prediction

In this section we present our resource availability prediction methods. We employ three methods from *pattern recognition and classification* for resource availability predictions, to serve resource *instance* or *point* availability and *duration* availability predictions: the *Bayes' Rule*, *Nearest Neighbor Rule* and *Pattern Matching*. The motivation behind employing these methods is the shape of the auto-correlation function of availability of the three resource classes (figure not shown), which indicates that there are strong patterns in the availability of the resources. Instance or point availability describes resource availability at a certain point of time; in our presented work it refers to the next monitoring instance. Whereas, the duration availability describes resource availability for a certain time duration, in our presented work it refers to the immediate next duration of a certain time span. These methods exploit different patterns of resource availability to serve its future availability predictions. Below we describe some patterns in resource availability at resource class level.

### 4.1 Patterns in Resource Availability

We observed different resource availability patterns over time— in a day and over different days of week. Figure 4 depicts diurnal resource availability patterns in a day; availability peaks during 8a.m. to 8p.m. and recesses during the remaining hours. The maximum availability was found at 4p.m., 2p.m., 3p.m. and the minimum at 4a.m., 8a.m., 4a.m. respectively for the three classes. We observed interesting patterns at resource class level as well at Grid level. The availability remains comparatively in a small range during the night hours with the minimum value between 4a.m. and 8a.m. This is the time when most of the resources are turned off or given a restart. After the minimum value it starts

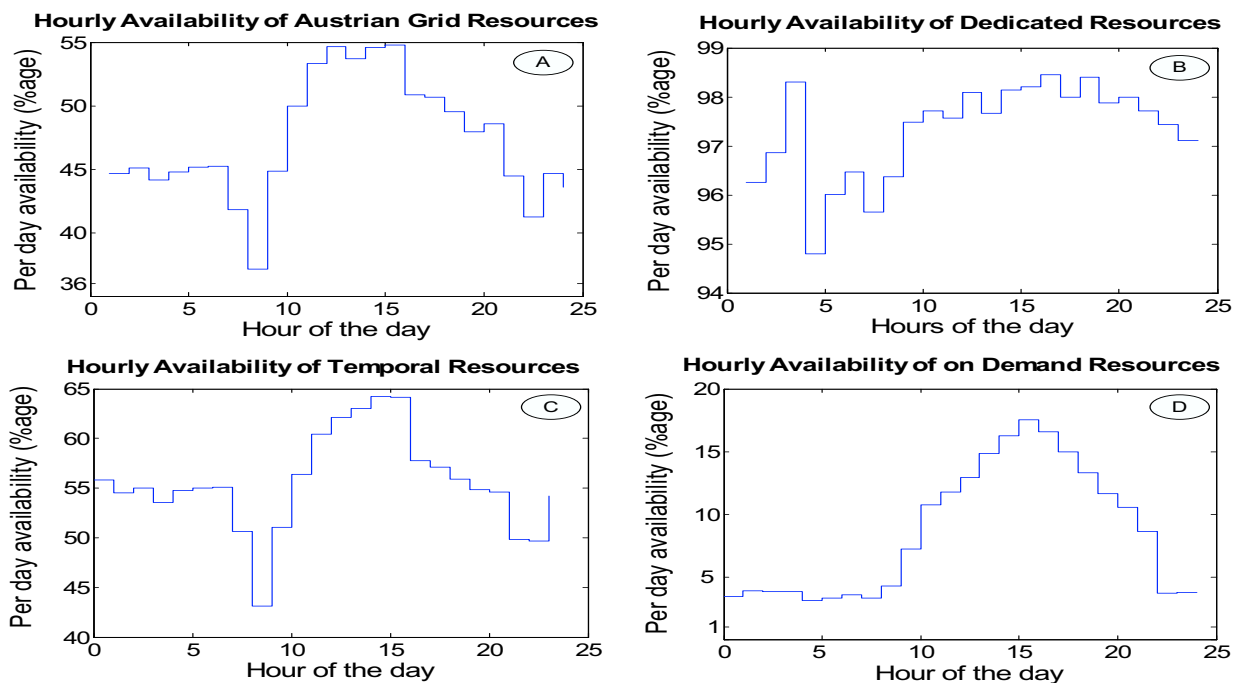


Figure 4: Resource hour-of-the-day availability at resource class and Grid level

increasing towards the peak value (except one decrease in dedicated resources), and from the maximum it decreases towards a certain level to be maintained till the minimum occurs.

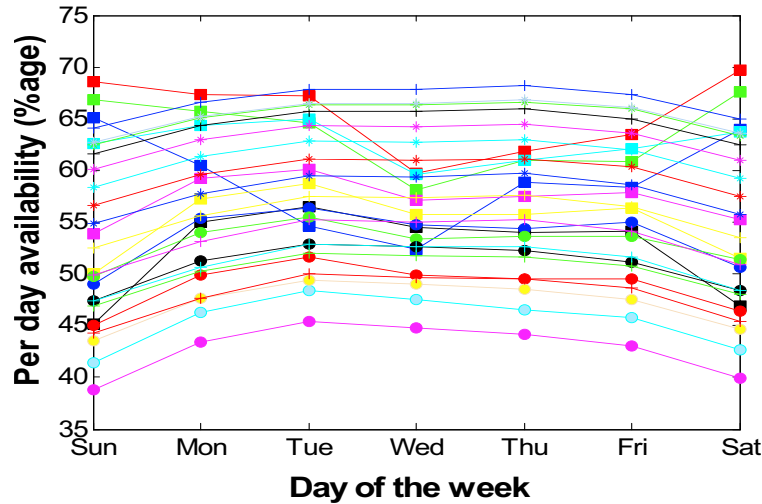


Figure 5: Resource availability patterns over different days of week

The availability patterns on 28 machines (machine names are irrelevant) over different week days are shown in Figure 5. Two classes of patterns can be visualized from this figure; first, with lower availability at week ends, and higher during the working week days. This class has an increasing availability till Tuesday or Wednesday and then decreasing till weekend. Second, with higher availability on weekends and lower during the working weekdays. This class has a decreasing availability till Wednesday and then again increasing availability till Saturday.

The average SD over different hours of the day was 4.16 as compared with that of 1.37 on a specific hour of the day. Moreover, at Grid level, an average SD over different week days was 12.3 and that of 6.6 was observed in availabilities on specific week days. This remarkable lower SD in availabilities on a specific day of the week than that on all week days, indicates strong availability pattern on specific week days. In sequel, lower SD in availabilities on specific hour of the day than that on all hours, shows similar availability patterns over specific hours of the day. In the following sections we present different prediction methods we employed.

## 4.2 Prediction Methods

In this paper we present a new prediction method from *pattern recognition and classification: pattern matching* and compare the results with our previous work in [8]. The pattern matching is a well recognized technique from pattern recognition and classification [5]. It considers resource's recent patterns of availability of certain duration and searches for the same from the past trace. It predicts the resource behavior as its most likely behavior subsequently after the same patterns in the past. For *instance* availability prediction the first subsequent resource status is considered, whereas for *duration* prediction the subsequent duration of interest is taken into account. We employ *Boyer-Moore String Matching* algorithm with *bad-character heuristic* and *good-suffix heuristic* [5] for pattern matching. The main advantage of this algorithm is that it has a complexity just of  $O(n)$ , and is much faster than similar others.

## 5 Performance Evaluation

This section describes the evaluation results for the prediction methods presented in the last section. We evaluated these methods for their prediction accuracy, calculated as  $Accuracy = (no. \text{ of true predictions}) / (total \text{ number of prediction queries})$  represented as percentage. In case of instance availability prediction, a prediction is treated true if the resource immediate next status is the same as predicted, otherwise it is considered false. In case of duration availability prediction, prediction is treated as true if a resource is predicted to be available for a certain immediate duration and resource is available for that duration or longer, otherwise it is considered false. In case when resource is

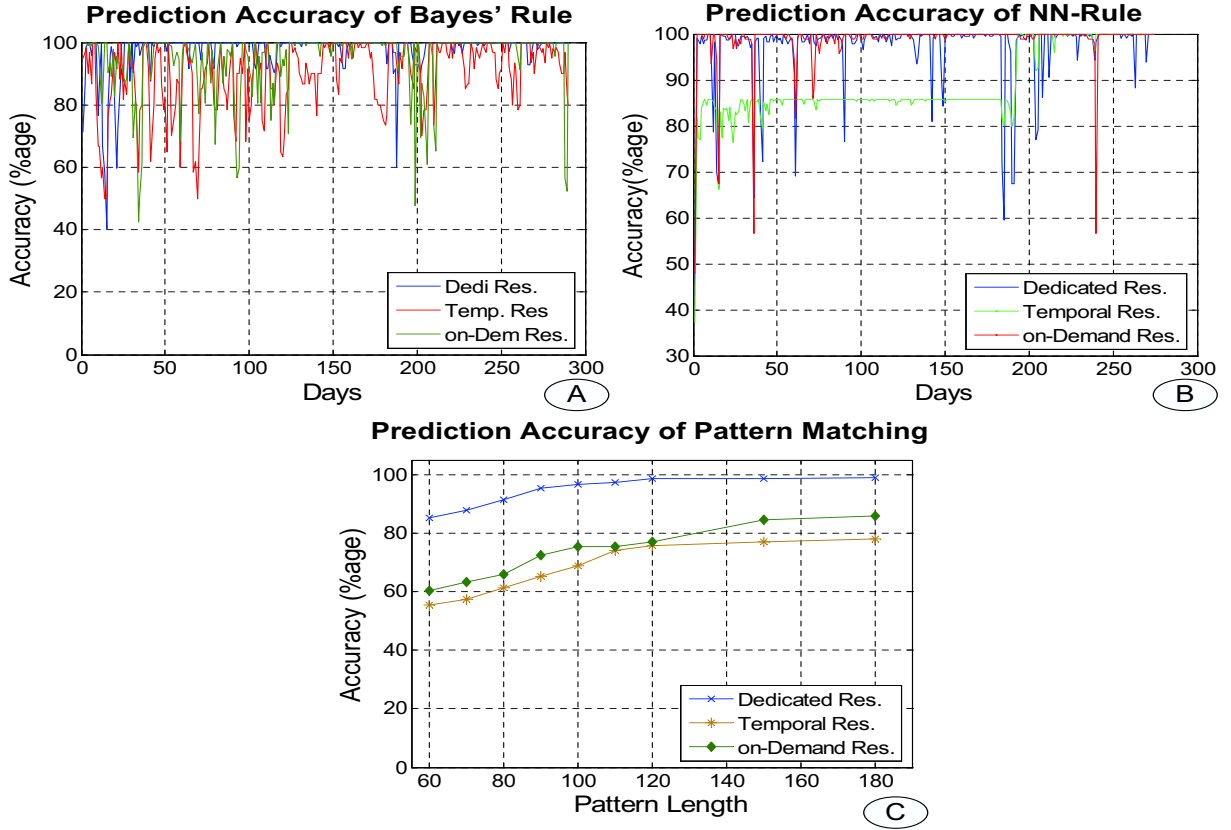


Figure 6: Resource instance availability prediction through Bayes' rule, NN-rule and pattern matching: Prediction accuracy comparisons of the three resource classes

available for a duration lesser than the predicted, the accuracy of prediction is calculated as the ratio of actual available duration to the predicted. Furthermore, the accuracy is recorded for different predictors which use different features from the feature vector described in [8].

## 5.1 Instance Availability Prediction

We made a greedy evaluation of accuracy of predictions made through Bayes' rule [8]. The accuracy was evaluated for all the resources in every resource class and accuracy at class level is presented by taking their average. For one resource, the prediction was evaluated on all the days of monitoring period (more than 300) and the accuracy was recorded on daily basis, where prediction query time in a day was selected at random. Thus, every resource was evaluated for more than 300 times, while the class-level daily accuracy was averaged from all the resources in the resource class. The daily accuracy of predictions for three resource classes using predictor `hourOfDay_cur` [8] is presented in Figure 6(A). For dedicated resources, the average accuracy was 97%. Accuracy for temporal resources averaged to 90% and on-demand resources exhibited that of 94%. We found for the all three classes that prediction accuracy decreases as more historical data distant from the prediction query time is included in calculating the priories.

A similar greedy setup was made for evaluation of instance availability predictions through NN-rule [8], as the same for Bayes' rule. Prediction accuracy results for the three resource classes using NN-rule are depicted in Figure 6(B). The average accuracies were recorded as 99.98%, 82.4% and 98.4% respectively for dedicated, temporal and on-demand resources at class level. The very high prediction accuracy for dedicated and on-demand resources is due to their high MTBF and high MTR, respectively. NN-rule gives higher accuracies for instance availability predictions for all three resource classes as compared with those through Bayes' rule.

Accuracy of instance availability predictions through pattern matching was also evaluated through a similar test phase. The accuracy results for the three resource classes are shown in Figure 6(C). Pattern matching showed 94.41%

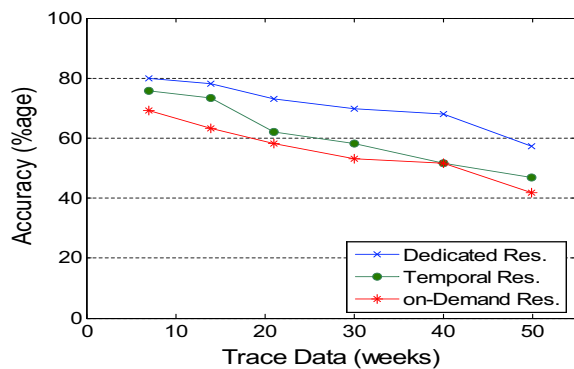


accuracy for dedicated resources, while the same for temporal and on-demand resources was 68.13% and 73.34% respectively. The lower accuracy of temporal resources is due to a higher variation in their patterns of (un)availability over time. The pattern matching yielded the least accuracy of 78.63% at Grid level, as compared with the three methods.

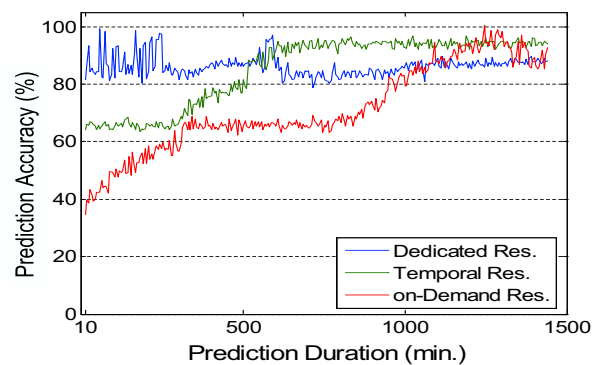
## 5.2 Duration Availability Prediction

Prediction accuracy results of the the three methods, for duration availability, are evaluated extensively through a series of experiments. For every resource, predictions were evaluated for the time durations starting from 10 min. to 24 hrs. with the increments of 5 min. Prediction for every duration was repeated 100 times, where date and time were randomly selected from the resource monitoring duration. These 100 repetitions were later averaged to record accuracy for that duration for the selected resource. Average accuracy for each time duration on all resources in a resource class, was recorded as accuracy of the class for that duration. The prediction accuracy of Bayes' rule with the predictor dayOfWeek using different amount of trace data, for the three resource classes is shown in Figure 7(a). We have noted that the prediction accuracy decreases as amount of historical data more distant from the prediction query time is included while calculating priories. This was also conformed by ACF (auto-correlation function) of availability durations(not shown here). The highest accuracy of duration availability predictions was observed in class of dedicated resources, which is due to their better stability for jobs of different durations, as shown in Figure 1. Bayes' rule exhibited a better accuracy of duration availability predictions for temporal resources than on-demand resources, which shows that on-demand resources were made available for quite different time durations.

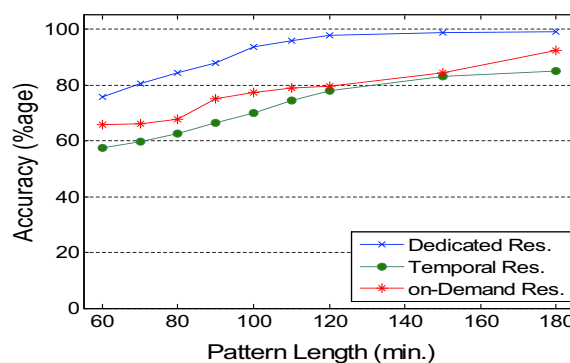
We present the accuracy of predictions for duration availability through NN-rule for the time durations of 10min-



(a) Prediction accuracy through Bayes' rule using different amount of trace data

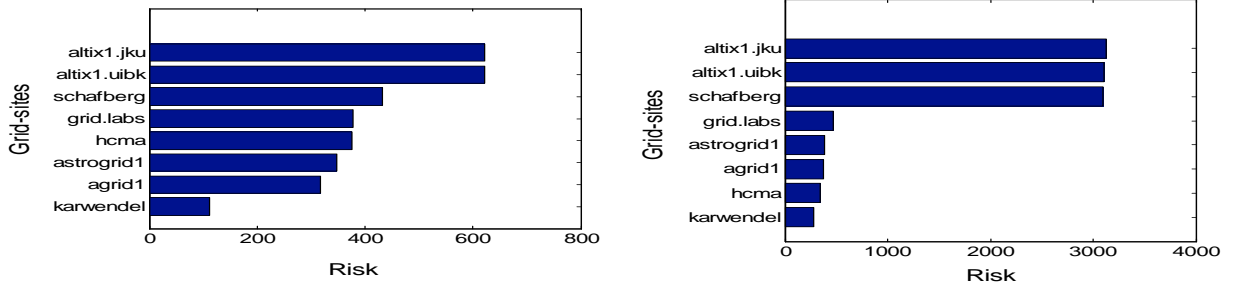


(b) Prediction accuracy through NN-rule for different durations



(c) Prediction accuracy through Pattern Matching for different pattern lengths

Figure 7: Accuracies comparison of duration availability predictions of the three resource classes through different methods.



(a) Resource availability-aware selection-risk comparison for application Wien2K (b) Resource availability-aware selection-risk comparison for application Invmod

Figure 8: Resource selection risk analysis

24hrs. The Figure 7(b) presents the prediction accuracy results at resource class level. Dedicated resources, as in all other evaluations showed the highest accuracy, within the range of 79% – 99%. NN-rule prediction accuracy increases from a minimum of 64% to a maximum of 93% for temporal resources, whereas, in case of on-demand resources, it increases from a minimum of 36% to a maximum of 100%. The increase in prediction accuracy by NN-rule, for temporal and on-demand resources is as expected– because resources in these two classes were available for relatively shorter time durations and NN-rule made availability predictions with higher accuracy for longer durations than for smaller durations.

The average accuracy of predictions for different durations, through pattern matching, using patterns of different lengths, are shown in the Figure 7(c). We observed accuracies of 93.18%, 69.39% and 75.56% for dedicated, temporal and on-demand resources respectively. We observed an increase in the accuracy when the pattern length was increased. This shows that resources exhibit more consistent behavior over longer time durations than shorter durations. Pattern matching resulted in better accuracy than NN-rule for duration predictions.

### 5.3 Resource Selection Risk Analysis

Let  $\{\alpha_1, \dots, \alpha_m\}$  be a set of  $m$  candidate resources. The *loss* function  $\xi(\alpha_i|\tau_j)$  describes the loss incurred for selecting resource  $\alpha_i$  when resource state is  $\tau_j$ . In this study, we compute the *loss* as:  $loss = l_{\alpha_i} + T_{\alpha_k} - g(\alpha_i, \alpha_k)$ . Here  $l_{\alpha_i}$  represents time lost on resource  $\alpha_i$  in case of its failure,  $T_{\alpha_k}$  represents total time (including overheads) taken to execute job on next potential resource  $\alpha_k$ , and  $g(\alpha_i, \alpha_k)$  represents expected time gain in selecting  $\alpha_i$  over  $\alpha_k$ . Suppose that we select one property  $y_i$  from the feature space  $y$  and that we contemplate to select resource  $\alpha_i$ . If the true state of the resource is  $\tau_j$  and it is predicted to be  $\tau_i$  then we will incur a loss of  $\xi(\alpha_i|\tau_j)$ . Since  $P(\tau_j|y_i)$  is the probability that the true state of the resource is  $\tau_j$ , the expected loss with selecting resource  $\alpha_i$  is merely:  $R(\alpha_i|y_i) = \sum_{j=1}^2 \xi(\alpha_i|\tau_j)P(\tau_j|y_i)$ . In decision-theoretic terminology, an expected loss is called a risk [5], and  $R(\alpha_i|y_i)$  is called the *conditional risk* for selecting a resource property  $y_i$  from feature vector  $x$ . Our objective is to minimize the overall risk that is given by  $\int R(\alpha(y_i)|y_i)p(y_i)dx$ , where  $\alpha(y_i)$  represents selection of resource  $\alpha$  when feature  $y_i$  is selected,  $d$  is our notation for  $d$ -space volume element and integral intends over the entire feature space. To minimize the overall risk, we compute the conditional risk for all the  $m$  resources and select resource  $\alpha_i$  for which  $R(\alpha_i|y_i)$  is minimum. We evaluated the selection-risk for one activity of each of the two real world applications Wien2K [7] and Invmod [7] for 8 Grid-sites. The comparison of these Grid-sites based on this risk is shown in the Figures 8(a) and 8(b) respectively.

## 6 Related Work

Several other studies characterize or model availability in different environments like cluster of computers [3], multi-computers [14], meta-computers (also called desktop Grids) [10], Grid [12], [2], super-computers [14], and peer-to-peer systems [9]. Most of these studies are about early systems considering only short term availability data, and ignoring machine availability policies. The two most closely related studies to ours are [2] and [12]. These works consider the availability characteristics at Grid level only, whereas we identify different classes of resources based on

their policies of availability in the Grid, and analyze their properties on the class level, and achieve better representative aggregates. Another closely related study to ours is [17] that analyzes resource availability through CPU failures, and finds its implications on large-scale clusters. We further compare resources based on their daily availability, hourly availability, MTBF, MTR, their dependability for different jobs and are the first to compare resources based on their stability for jobs of different durations.

Predictions from availability models give probability of resource in available state and are most of the times insufficient to decide whether the resource will be available or not [8]. Works in [11], [16] and [12] have made predictions based on previous weekday or weekend only, and have got moderate accuracy in their predictions. In contrast we employ methods from pattern recognition and classification, which exploit knowledge from our resource availability characterization phase, and takes into account patterns from resource past behavior as well as its current behavior, and yield very promising results.

## 7 Conclusion

Besides that the Grid technology is getting more matured day by day, different policies for resource availability in the Grid, and resources' working *stability* raise serious issues about their suitability for different jobs. In this work we compare resources based traditional and new metrics- their MTBF, MTR in general and their *stability* and *dependability* for jobs of different durations in particular. Different comparison criteria suit to different needs. We argue the later two metrics better compare the resources considering the job durations. Dedicated resources revealed the highest *stability* for all jobs of different durations and showed the highest MTBF and the lowest MTR, the converse it true for on-demand resources. Stability of temporal resources falls in the mid range of other two resource classes, and they showed an almost equal MTBF and MTR. Next, we find patterns in resource availability and make resource instance and duration availability predictions by using three methods from *pattern recognition and classification*; Bayes' Rule, Nearest Neighbor Rule and Pattern Matching. We extensively evaluated the accuracy for these methods using different predictors and different amount of historical data. On average, more than 90% and 70% accuracy is found for instance and duration predictions respectively, when minimum of historical data was used.

As future work, we will try to integrate resource computational metrics (MFlops and others) with availability metrics to come up with a unified metric to compare the resources in a generic way as well. We would also like to validate our resource availability prediction methods using availability traces from other Grids. We also plan to evaluate improvements in job execution performance through availability aware resource selection, and availability aware data storage.

## References

- [1] The Austrian Grid Consortium. <http://www.austriangrid.at>.
- [2] Alexandru Iosup et al. On the dynamic resources availability in grids. In *CCGrid*, Rio de Janeiro, Brazil, May 14-17, 2007.
- [3] Anurag Acharya et al. The utility of exploiting idle workstations for parallel computation. In *international conference on measurement and modeling of computer systems*, Seattle, Washington, USA, 1997.
- [4] D. Kondo et al. Characterizing and evaluating desktop grids: An empirical study, 2004.
- [5] Duda, Richard O. et al. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.
- [6] C. Ebeling. *An Introduction to Reliability and Maintainability Engineering*. McGraw-Hill, Boston, MA, 1997.
- [7] Farrukh Nadeem et al. Soft benchmarks-based application performance prediction using a minimum training set. In *e-Science 2006*, Amsterdam, Netherlands.
- [8] Farrukh Nadeem, Radu Prodan, and Thomas Fahringer. Characterizing, Modeling and Predicting Dynamic Resource Availability in a Large Scale Multi-Purpose Grid. In *Proc. of CCGrid 2008*, Lyon, France, May 19-22 2008.

- [9] R. Bhagwan et al. Understanding availability. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems*, Feb 2003.
- [10] R. Wolski et al. Automatic methods for predicting machine availability in desktop grid and peer-to-peer systems. In *CCGrid2004*, Chicago, Illinois, USA.
- [11] Xiaojuan Ren and Rudolf Eigenmann. Empirical studies on the behavior of resource availability in fine-grained cycle sharing systems. In *International Conference on Parallel Processing*, Washington, DC, USA, 2006.
- [12] Brent Rood and Michael J. Lewis. Multi-state grid resource availability characterization. In *International Conference on Grid Computing*, Austin, TX, September 17-19, 2007.
- [13] Bianca Schroeder and Garth A. Gibson. A large-scale study of failures in high-performance computing systems. In *International Conference on Dependable Systems and Networks*, Washington, DC, USA, 2006.
- [14] D. Tang and R. K. Iyer. Dependability measurement and modeling of a multicomputer system. *IEEE Trans. Comput.*, 42(1), 1993.
- [15] Thomas Fahringer et al. Askalon: a grid application development and computing environment. In *GRID05*.
- [16] Xiaojuan Ren et al. Resource availability prediction in fine-grained cycle sharing systems. In *International Performance and Distributed Computation*, 2006.
- [17] Yanyong Zhang et al. Performance implications of failures in large-scale cluster scheduling. 2004.