European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies

A Network of Excellence funded by the European Commission

A Framework for Resource Availability Characterization and On-Line Prediction in Large Scale Computational Grids

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

Alexandru Iosup
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology, The Netherlands.
A.Iosup@tudelft.nl

CoreGRID Technical Report
Number TR-0130
April 9, 2008

Institute on Resource, Management and Scheduling

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

A Framework for Resource Availability Characterization and On-Line Prediction in Large Scale Computational Grids

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

Alexandru Iosup
Faculty of Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology, The Netherlands.
A.Iosup@tudelft.nl

CoreGRID TR-0130
April 9, 2008

Abstract

Production Grids integrate today thousands of resources into e-Science platforms. However, the current practice of running yearly tens of millions of single-resource, long-running Grid jobs with few fault tolerance capabilities is hampered by the highly dynamic Grid resource availability. In additional to resource failures, Grids introduce a new vector of resource availability dynamics: the resource sharing policy established by the resource owners. As a result, the availability-aware Grid resource management is a challenging problem for today's researchers. To address this problem, we present in this work GriS-Prophet, an integrated system for resource availability monitoring, analysis, and prediction. Using GriS-Prophet's analysis tools on a long-term availability trace from the Austrian Grid, we characterize the Grid resource availability for three resource availability policies. Notably, we show that the three policies lead to very different capabilities for running the typical Grid workloads efficiently. We introduce a new resource availability predictor based on Bayesian inference. Last but not least, using GriS-Prophet's prediction tools we achieve an accuracy of more than 90% and 75% in our instance and duration availability predictions respectively.

1 Introduction

Today's Grids, e.g., CERN's LCG [4], the TeraGrid [14], the Austrian Grid [13], integrate into a single e-Science platform (tens of) thousands of resources provided by various owners. On these platforms, the typical Grid workload comprises yearly millions of single-resource, long-running jobs, with few or even no fault tolerance capabilities [4]. However, the highly dynamic resource availability of the current Grids [6, 2] increases for the typical workload the difficulty of resource capacity planning and of scheduling. In traditional computing centers, a human administrator can respond to unavailability events usually after they have occurred; often, on-line tools are employed to predict errors before they occur [3, 9]. In Grids, human administration of the system would be required for each participating resource owner, leading to unmanageable costs of ownership. To build predictive tools, more knowledge is needed about the characteristics of the Grid resource avilability, and about the accuracy and the overhead of on-line predictors. To this end, in this work we present **Gri**d re**S**ource Prophet (GriS-Prophet), a system for resource availability characterization and prediction.

Similarly to other computing resources, Grid resources may become unavailable due to failures in hardware, the operating system, or the middleware. In addition, the Grid resource owners may set different resource availability

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

policies, from resources that are dedicated to Grid use [6] to desktop Grids [2] and to resources that are available on-demand from computing centers.

Currently, there is no tool to characterize the resource availability in a Grid system in which the different resource availability policies coexist. To address this situation, we design GriS-Prophet to characterize the Grid resource availability per resource availability policy.

The large number of resources present in Grids increases the difficulty of accurate on-line resource availability predictions. While several approaches exist in the context of clusters [9], it is unclear if they can be applied directly in a Grid context. Similar work in Grids has recently begun [5], but more research is needed before such approaches can be widely deployed. To this end, we propose and implement in GriS-Prophet two methods from pattern recognition and classification: *Bayesian Inference* [1] and the *Nearest Neighbor Predictor*. Our contribution in this work is threefold: 1) We introduce the design of the GriS-Prophet, a system for resource availability characterization and prediction (Section 3); 2) Using GriS-Prophet on a long-term availability trace from the Austrian Grid, we characterize the Grid resource availability for three resource availability policies (Section 4); 3) We evaluate the use of two methods from pattern recognition and classification, and in particular of one based on Bayesian Inference, for Grid resource availability predictions (Section 5).

We have implemented a prototype of our system ASKALON [15] project as Grid service using Globus Toolkit 4.

2 Related Work

Much research has been devoted to analyzing or modeling the resource availability in multi-computers and supercomputers [12], clusters of computers [12], pools of desktop computers [8, 11], cluster-based Grids [6], desktop Grids [2], and even peer-to-peer systems [8]. However, none of these efforts presents data for a system integrating supercomputers, clusters of computers, and pools of desktop computers at the same time. Futhermore, much of the related work uses few traces or traces from early systems, and needs further research to confirm their findings. Arguably the largest study of a large computing environment to date, the work of Schroeder and Gibson [12] characterizes 22 clusters and supercomputers from the LANL environment using data spanning 9 years. Closest to our work, Kondo et al. [2] and Iosup et al. [6] analyze long-term traces from an enterprise desktop Grids and from cluster-based, respectively. The former uses two-minutes sampling coupled with the execution of probe applications to collect availability information from the system. The latter introduces cross-cluster resource availability properties.

For resource availability prediction in computing systems, the many research efforts have to date employed a wide variety of data mining techniques: time-series analysis [9, 8, 10], rule-based classification [9], Bayesian statistics [9], signal processing [7], and hybrid models [5]. However, different methods have been found to give better predictions for various data sets; in particular, the Nearest Neighbor predictor (see Section refsec:predictions:nn) gives surprisingly accurate availability state estimates [9, 7, 10, 5]. Sahoo et al. [9] compare time-series analysis, rule-based classification, and Bayesian networks to make accurate on-line predictions about system reliability parameters and availability events. Mickens and Noble [7] and Ren et al. [10] predict the resource availability over the next period of time, under the assumption of the independence of failures across computing nodes.

3 The GriS-Prophet Architecture

In this section we present GriS-Prophet, an integrated system for resource availability monitoring, characterization, and prediction. With GriS-Prophet, we address two main goals: building a framework for resource availability characterization in Grids, and building a framework for resource availability predictions in Grids. For both goals, we focus on the concept of resource availability policy, through which the resource owner can specify a dynamic participation of the resource in the Grid. Our framework can analyze the characteristics for each resource availability policy, or for a class of related resource availability policies. We also provide a generic prediction framework that takes into account availability policies.

Figure 1 depicts the GriS-Prophet architecture. The GriS-Prophet aggregates resource availability characteristics for each of the resource classes defined through the High Level Resource Availability Policies. The main input for the system is the monitoring data concerning resource availability that is provided by the Resource Monitoring System. The Resource Trace Characterization module analyzes these data based on the resource class, and extracts the resource availability properties over time. The Resource Availability Modeling module tries to model the main input data by fitting them against well-known distributions, e.g., fitting the inter-arrival time between consecutive failures using the

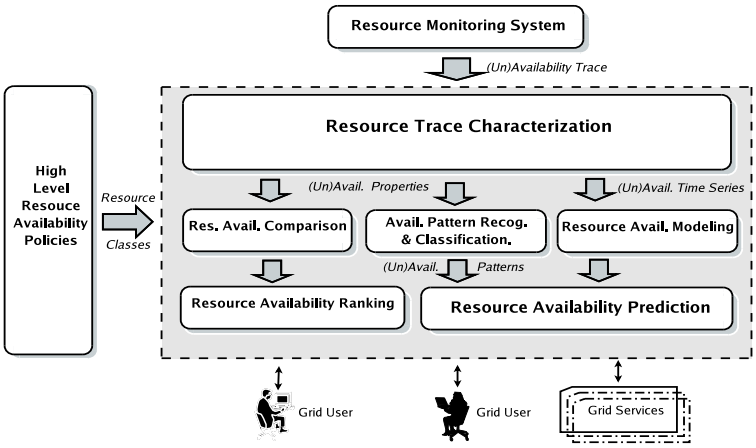


Figure 1: Gris-Prophet System Architecture

Weibull distribution [12, 6]. The Availability Pattern Recognition and Classification module is a general toolbox of data mining tools that uses expert knowledge about Grid systems to provide fast and accurate predictions for Grid resource availability. Finally, the Resource Availability Comparison and Ranking modules rank resources in the same class based on their suitability to run jobs of different durations and fault-tolerance capabilities.

4 Resource Availability Characterization

In this section we characterize the availability of the Austrian Grid resources. First, we present the Austrian Grid and the general characteristics of a long-term availability trace taken from this environment. Then, we extract from this trace the detailed characteristics for each of the resource availability policies used in the Austrian Grid: the dedicated resources, the temporary resources, and the on-demand resources. Finally, we propose a resource availability classification, and a novel reliability metric specific to Grids, the resource stability.

4.1 The Austrian Grid Resource Availability Trace

The Austrian Grid is a nation-wide, multi-institutional, -administrative and -VO Grid platform, consisting of 28 Grid sites geographically distributed in Austria. Eeach Grid site comprises multiple computational nodes; in total, there are over 1500 processors in the Austrian Grid.

There are three availability policies in the Austrian Grids: the *dedicated resources*, which are meant to be always available to the Grid users for production and experimental work, the *temporary resources*, which are resources belonging to university laboratories that join the Grid when powered and left idle by their users, and the *on-demand resources*, which are made available to the Grid only on user demand and only for large scale jobs or experiments. We define a resource class for each availability policy; for each resource class the class name is identical to that of its associated availability policy. We argue that these classes are most likely to be found in other Grids; in particular, the dedicated resources and the temporary resources correspond to traditional cluster-based Grids [6] and to desktop Grids [2], respectively.

We accessed, organized, and analyzed a long-term Austrian Grid resource availability trace, from mid-June 2006 to mid-April 2007, for a total of 274 days. The trace records availability information at intervals of 5 minutes; altogether this trace comprises of more than 23 million events occurring in the whole Austrian Grid. Each trace event represents the availability status of a Grid resource (i.e., available or unavailable) with the time stamp at which the status of the resource is recorded. A resource is considered available if it is accessible remotely, otherwise unavailable.

4.2 Time Patterns in Resource Availability

We first investigate the evolution of resource availability over time. Figure 2 shows the evolution over time of the daily resource availability for each resource class from the Austrian Grid. On average, the resource availability in Austrian

Grid is 33%, with a minimum (maximum) of 4% (71%) per day. The dedicated resources have the highest availability of 92% with a minimum (maximum) of 48% (100%) per day. The temporary resources collectively show a relatively lower average daily availability of 47% with a minimum (maximum) of 0% (92%) per day. The on-demand resources are the least available: on average 9% daily, with a minimum (maximum) of 0% (88%) per day.

We further analyze the patterns that occur in the resource availability time series. We report the average availability of the three resource classes in the Austrian Grid as a function of *hour of the day* (daily patterns) and *day of the week* (weekly patterns). We have also investigates individual resources, as opposed to resource classes, and found more patterns, sometimes even inverse from the resource class patterns; the individual resource patterns are used by the Resource Availability Comparison and Ranking module (see Section 3).

Figure 3 shows the daily resource availability patterns for the three resource classes in the Austrian Grid. The resource availability peaks for all classes during 10AM to 8PM. The dedicated resources show an additional peak between 2AM and 3AM, while the temporary resources show a high level of availability during the night hours. The resource availability is minimal between 4AM and 8AM, the time when most of the resources are turned off or automatically restart.

Figure 4 depicts the weekly resource availability patterns for the three resource classes in the Austrian Grid. The peak is observed for all resources from the middle to the end of the work days (i.e., Wednesday for on-demand resources, Thursday for the other resource classes). This corresponds to the typical Grid users behavior patterns, where most of the work (job submission) is done in the last half of the week.

4.3 Resource Availability Duration

We now look at the duration of resource availability. We define the *life time* of a resource as the time period between two consecutive failures, and the *life duration* as the length of the life time. The distribution of life durations for a resource class is critical for understanding the types of applications that class can reliably service. Figure 5 shows the distribution of the life duration for the three resource classes in the Austrian Grid. The dedicated, the temporary, and the on-demand resources have on average a maximum life duration of 93495 minutes (65 days), 5275 minutes (4 days), and 1000 minutes (less than one day), respectively. Over 50% of the dedicated resources have a life duration over a day; by comparison, over 50% of the temporary (on-demand) resources have a life duration below 6 hours.

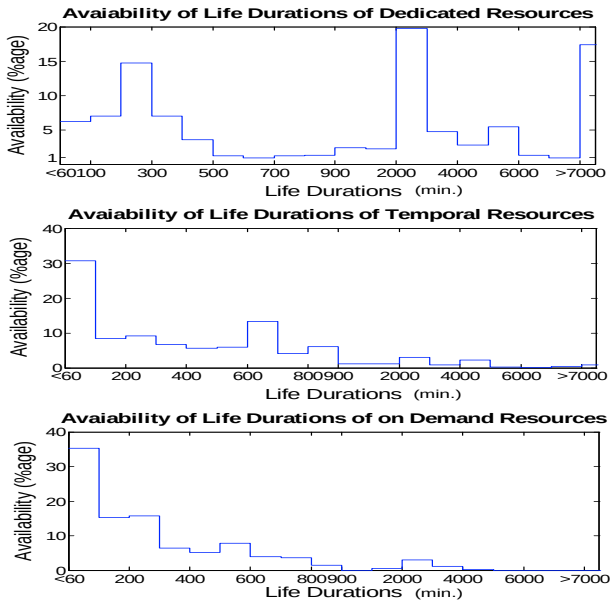


Figure 5: Lifetime distributions in three resource classes

4.4 Resource Availability Classification

Based on the results obtained for the time patterns in resource availabilitys, we propose four classes of resource

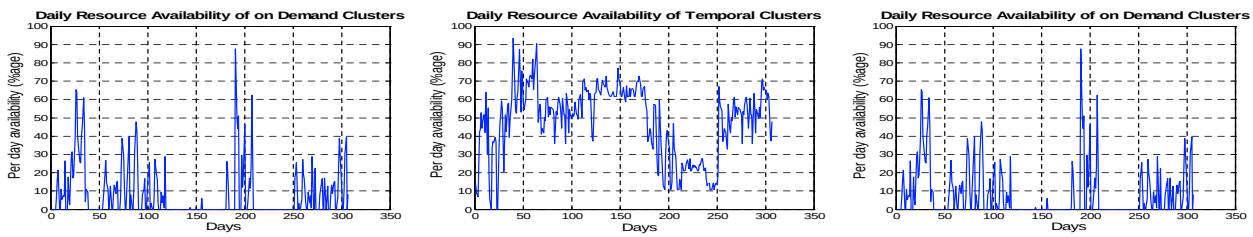


Figure 2: The daily resource availability for the three resource classes in the Austrian Grid: (left) dedicated resources; (middle) temporary resources; (right) on-demand resources.

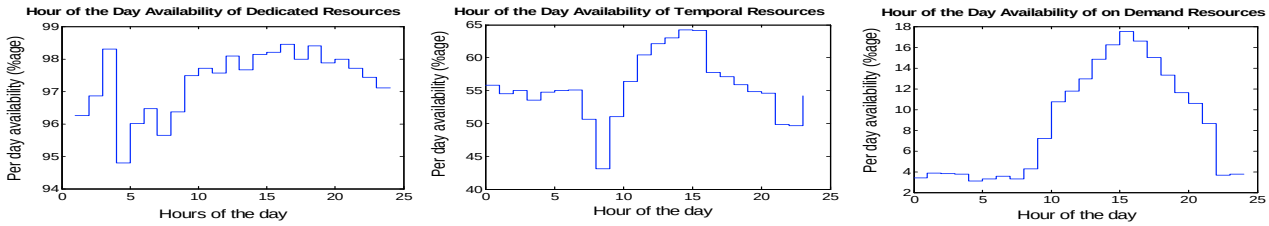


Figure 3: Daily resource availability patterns for the three resource classes in the Austrian Grid: (left) dedicated resources; (middle) temporary resources; (right) on-demand resources.

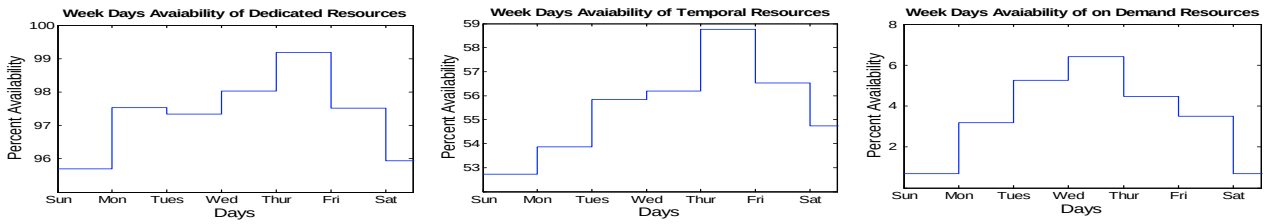


Figure 4: Weekly resource availability patterns for the three resource classes in the Austrian Grid: (left) dedicated resources; (middle) temporary resources; (right) on-demand resources.

Table 1: Resource classification based on their average availability duration within the resource classes in Austrian Grid.

		Percentage of resources		
	Res. classes	Ded. Res.	Temp. Res.	on-Dem. Res.
High	daily avail.$(75\%, 100\%]$.	78.57	0	0
	avg. dur. $(72, \infty)$hrs.	14.28	10	0
Med. Upper	daily avail.$(50\%,75\%]$.	14.28	10	0
	avg. dur. $(36, 72]$hrs.	35.71	0	0
Med. Lower	daily avail.$(25\%,50\%]$.	7.14	0	0
	avg. dur. $(24, 36]$hrs.	50	0	0
Low	daily avail.$[0\%, 25\%]$.	0	90	100
	avg. dur. $[0, 24]$hrs.	0	90	100

availability: high, medium upper, medium lower, and low. The four resource availability classes correspond respectively to the daily availability percentage ranges $(75\%, 100\%]$, $(50\%,75\%]$, $(25\%,50\%]$, and $[0\%, 25\%]$, and to the availability duration ranges (in hours) $(72, \infty)$, $(36, 72]$, $(24, 36]$, and $[0, 24]$.

By mapping the resources from each resource class are mapped to the resource availability classes, we are able to better understand how each resource class can reliably service jobs of various durations. Table 1 describes the resource availability classes, and maps to them the three resource classes in the Austrian Grid. Considering the daily availability, none of the dedicated resources fall in the *low* category; in contrast, all on-demand resources fall in this category. Most of the temporary resources (90%) have low availability. Roughly 78% of dedicated resources are highly available during the day and only 14% of their total are suitable for jobs longer than 72 hours. Most of temporary and all of the on-demand resources are suitable for jobs smaller than 24 hours of duration.

4.5 Resource Stability

We introduce in this section the *resource stability*, a new reliability metric that defines the ability of a resource (or group of resources) to run multiple jobs of a given duration. The motivation for this new metric is that while Grid workloads are dominated both numerically and in terms of resource consumption by groups of single-resource jobs, the core Grid resource management middleware (e.g., Globus GRAM) deals with each individual job in turn, incurring high overhead

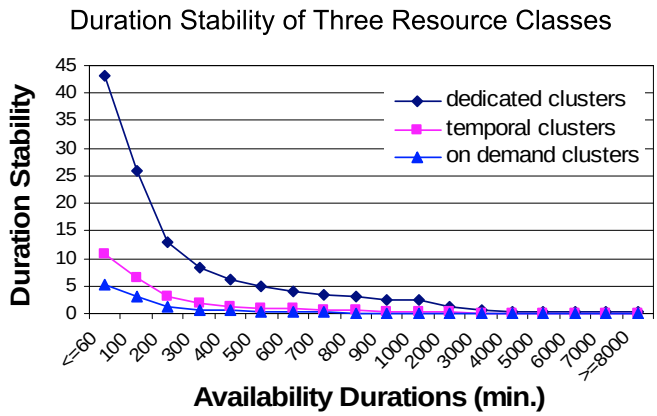


Figure 6: Resource stability for the three resource classes in the Austrian Grid.

(and, in turn, higher job wait time than expected for the remaining jobs). Selecting one resource with high availability and running on it several jobs of the same runtime characteristics is critical for achieving low overhead.

We define a job duration as the job's uninterrupted run time, i.e., the run time of the job if the job cannot use checkpointing, and the time between checkpoints otherwise. Then, we define the stability of a Grid resource r for a job duration ΔJ as

$$S_r(\Delta J) = \sum_{i=1,n} \lfloor \frac{\Delta t_i}{\Delta J} \rfloor \times P(\Delta t_i)$$

where n is the number of life time periods for the resource r, the set of Δt_i are the unique life durations for the resource r, and $P(\Delta t_i)$ denotes the probability of life duration Δt_i for the resource r. The integral term $\lfloor \frac{\Delta t_i}{J} \rfloor$ counts the number of jobs of duration Δt_k that can be run on resource r consecutively; for $\Delta t_i < \Delta J$ the count is 0.

Figure 6 depicts the resource stability for the three resource classes in the Austrian Grid. The dedicated resources have the highest stability: on average more than 8 times that of the the class with the lowest stability, the on-demand resources. For job durations of 1 hour, the average dedicated resource can run over 40 consecutive jobs; by comparison, the average temporary (on-demand) resources can run 10(5) consecutive jobs.

5 Resource Availability Prediction

In this section, we present and evaluate parts of our resource prediction framework.

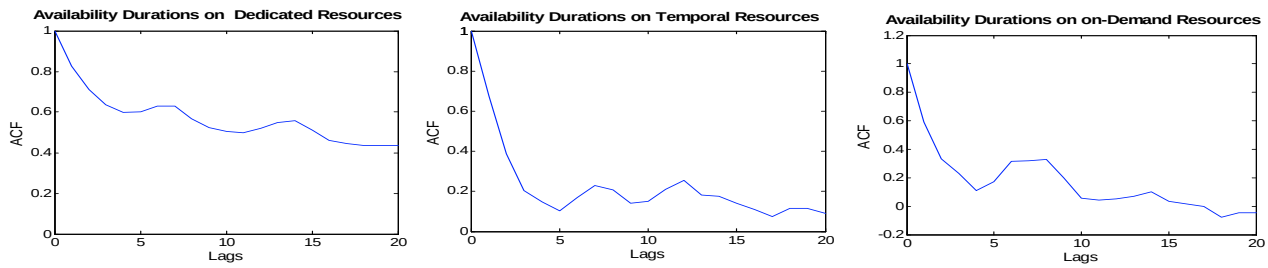


Figure 7: The auto-correlation function for the three resource classes in the Austrian Grid: (left) dedicated resources; (middle) temporary resources; (right) on-demand resources.

5.1 The Predictors

The autocorrelation functions of availability for the three resource classes are shown in Figure 7. The wave like shapes of the curves indicate similar patterns over time, and high value of autocorrelation for smaller lags (1-3) indicate that the most recent values will yeild better predictions. To exploit the availability features and patterns over time, we employ two methods from *Pattern Recognition and Classification*, the *Bayesian Inference* [1] and *Nearest Neighbor* predictor to serve resource *instance* or *point* availability and *duration* availability predictions. Instance availability predictions describes resource availability at the next monitoring instance, and the duration availability prediction describes resource availability for the immediate next duration of a certain time span.

The bayesian inference (BI) enables estimating the possibility of an event from its *likelihood* and *prior probability* as its probability conditional to its characteristics. We first model resource availability with as either available or unavailable. Then, we predict the resource availability using BI as follows. Let σ_a and σ_u represent the two classes to which our resource state may belong. The *prior probabilities* $P(\sigma_a)$,$P(\sigma_u)$ are known from the characterization phase, and can be calculated as . These can be calculated as $P(\sigma_i) = N_i/N$ for $i = a, u$, where N is the total number of events and N_i the number of number of events for event that leads to state σ_i. We will use as specific BI features the availability features found in Section 4: x=[*day_of_week*,*hour_of_day*,*hour_of_day* & *day_of_week*]. Then, $p(x|\sigma_i)$ for $i = a, u$ represents the class-conditional probability density functions (PDFs), describing the distributions of a feature vector x for each BI feature.

The class-conditional PDFs are also calculated from the trace data during the characterization phase. Then according to BI:

$$P(\sigma_i|x) = \frac{p(x|\sigma_i)P(\sigma_i)}{p(x)} \tag{1}$$

where $p(x)$ is the PDF of x and for our resource availability model
$p(x) = \sum_{i=a,u} p(x|\sigma_i)P(\sigma_i)$. We consider one feature from the feature vector x at a time, which can have any value from its feature space, e.g., $\{0, 1, 2, ..., 23\}$ for the feature "hour_of_day". In our case feature vector only takes discrete values, thus the density functions $p(x|\sigma_i)$ are equivalent to the probabilities $P(x|\sigma_i)$.

The Nearest Neighbor (NN) predictor is a well known pattern classification technique, which selects the the nearest neighbor as a prediction for the current location. To predict resource availability, the last monitored status is used as the nearest neighbor [9, 7, 10, 5]. This method typically suits to the machines with high MTBF(Mean Time Between Failure) and MTR(Mean Time to Reboot).

5.2 Resource Ranking

We now focus on the problem of resource ranking based on the predicted life time duration. Let $\{\gamma_1, ..., \gamma_m\}$ be a set of m resources that can execute a job. The *loss* function $\eta(\gamma_i|\sigma_j)$ describes the loss incurred for selecting resource γ_i when resource state is σ_j, computed as $\eta(\gamma_i|\sigma_j) = Tl_{\gamma_i} + T_{\gamma_k} - Tg(\gamma_i, \gamma_k)$, where Tl_{γ_i} represents the time lost on resource γ_i in case the resource fails after selection, T_{γ_k} represents total time (including overheads) taken to execute job on next potential resource γ_k, and $Tg(\gamma_i, \gamma_k)$ represents expected time gain in selecting γ_i over γ_k.

Suppose that we select one property from the feature space x and the resource γ_i. If the true state of the resource is σ_j but the predicted state is $\sigma_i \neq \sigma j$ then we will incur a loss of $\eta(\gamma_i|\sigma_j)$. Since $P(\sigma_j|x_i)$ is the probability that the true state of the resource is σ_j, the expected loss with selecting resource γ_i is $R(\gamma_i|x) = \sum_{j=a,b} \eta(\gamma_i|\sigma_j)P(\sigma_j|x_i)$. In decision-theory terminology, an expected loss is called a risk; thus, we call $R(\gamma_i|x_i)$ the *conditional risk* for selecting a resource property x_i from feature vector x. Our objective is to minimize the overall risk that is given by $\int R(\gamma(x_i)|x_i)p(x_i)dx$, where $\gamma(x_i)$ represents selection of resource γ when feature x_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 γ_i for which $R(\gamma_i|x_i)$ is minimum.

5.3 Experimental Results

We now present initial results of our work on resource availability predictions. The work on resource selection risk analysis is in progress.

The first experiment evaluates the accuracy of the predictions for availability instances, that is, for predicting the resource availability state at arbitrary moments in time. We simulate the Austrian Grid environment based on the availability traces taken from the Austrian Grid (see Section 4.1). For each of the 274 days present in the Austrian Grid

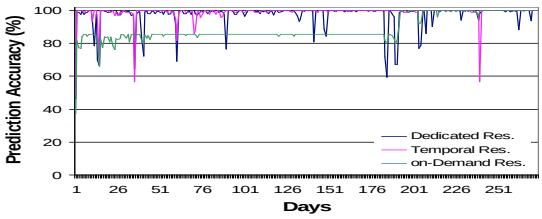


Figure 8: The daily prediction accuracy for availability instances using the NN predictor.

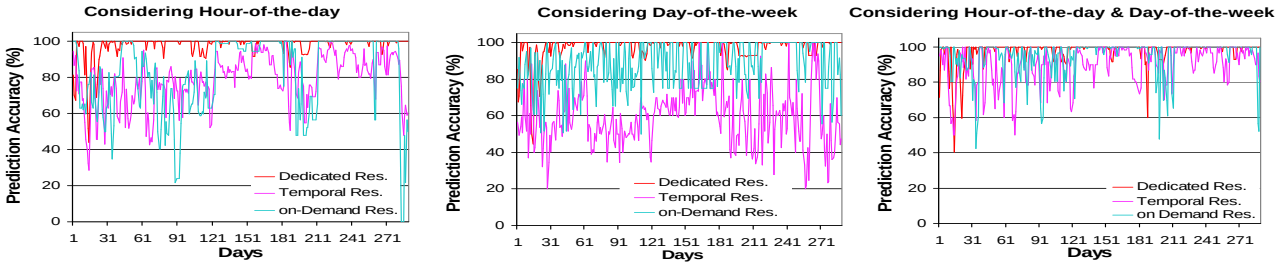
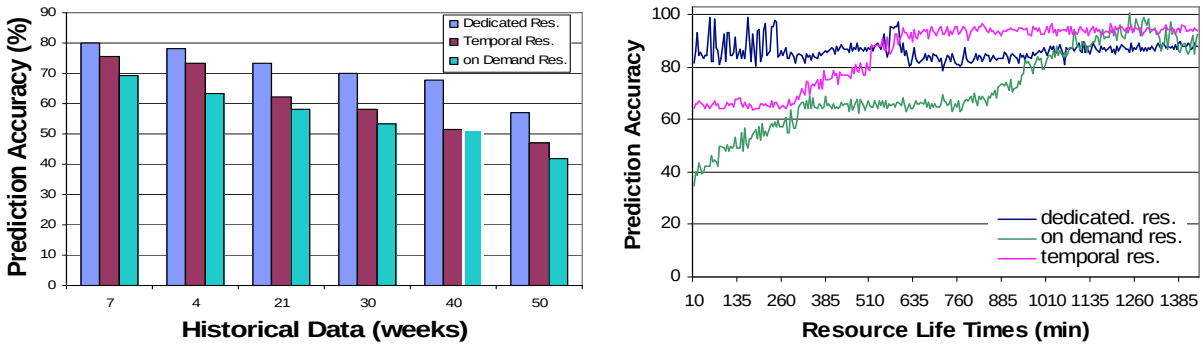


Figure 9: The daily prediction accuracy for availability instances using the BI predictor for three availability properties: (left) hour of the day; (middle) day of the week; (right) combined properties.



(a) Average accuracy, BI predictor.

(b) Accuracy over time, NN predictor.

Figure 10: The accuracy of duration predictions using the BI and the NN predictors.

traces, and for every resource, we make predictions for 24 moments of time, one for each hour where the minuts and the seconds are selected at random. We define the *daily prediction accuracy* as the percentage of correct vs. incorrect predictions for one day, using the traces as ground truth.

Figure 9 depicts the daily prediction accuracy using BI for three resource availability properties: *hour-of-the-day*, *day-of-the-week* and *hour-of-the-day:day-of-the-week*. The last predictor exhibits the best average accuracy: 97% for dedicated resources, and above 90% for temporal and on-demand resources. The *hour-of-the-day* predictor showed the second best accuracy for the three classes, with the *day-of-the-week* predictor proving to be the least accurate (59% average accuracy for temporary resources). In comparison, the NN predictor achieves a higher accuracy for availability instance predictions, of over 97% for all classes of resources, as shown by Figure 8. In addition, the NN predictor never reacher a prediction accuracy below 55%, while the BI predictor can lead to a prediction accuracy as low as 20%.

The second experiment evaluates the accuracy of the predictions for availability duration. We use the same setup as for the first experiment. For each of the 274 days present in the Austrian Grid traces, and for every resource, we evaluate the predictions for all the time durations between 10 minutes and 24 hours in increments of 5 min. The accuracy for each prediction duration was evaluated over 100 trials where the date and time of the prediction were selected randomly. For the BI predictor, the likelihood and priori probabilities were computed from the data in the

time window before the prediction moment. Figure 10(a)(a) depicts the average accuracy of the availability duration predictions using the BI predictor, for windows of size 7 to 50 weeks. The prediction accuracy decreases as amount of historical data more distant from the prediction moment is included while calculating prior probabilities, this was also confirmed by investigating the auto-correlation function (ACF) of the availability durations in the Austrian Grid traces. The best accuracy using the BI predictor is achieved for windows of 7 weeks: 80%, 75%, and 69% accuracy for dedicated, temporary, and on-demand resources, respectively. Figure 10(a)(b) depicts the accuracy of availability duration predictions using the NN predictor. The NN predictor yields better or similar results to the BI predictor for dedicated and temporary resources, but exhibits lower accuracy and especially lower minimum accuracy for the on-demand resources.

6 Conclusion and Future Work

The highly dynamic Grid resource availability is due not only to resource failure, but also to the sharing policy enforced by the resource owners: resources may be dedicated to Grid use, or become temporary part of the Grid. As a result, the typical Grid workloads are difficult to manage efficiently. To address this problem, we have introduced in this work GriS-Prophet, an integrated system for resource availability monitoring, analysis, and prediction. The GriS-Prophet receives resource availability information, and transforms it into useful predictions for the Grid resource management systems. For this work in progress, we have first used the analysis tools on a long-term availability trace from the Austrian Grid, and characterized the Grid resource availability for three resource availability policies. Notably, we have shown that the three policies lead to very different capabilities for running the typical Grid workloads efficiently. We have also introduced a new resource availability metric, the resource stability, which characterizes the ability of a resource to execute groups of jobs of a given duration; we argued that selecting resources based on this metric will greatly increase the efficiency of the Grid resource selection process. For the GriS-Prophet prediction component, we have introduced a new resource availability predictor based on Bayesian inference, and the notion of resource selection risk. Compared with a predictor used often in resource availability predictions, the Nearest Neighbor, we have shown that our new predictor can deliver better accuracy for specific cases.

For the future, we plan to extend the predictors with traditional data mining algorithms adapted to the Grid resource availability data, and to follow our investigation of novel metrics for Grid resource availability. Last but not least, we plan to research the optimization problem related to the notion of resource selection risk.

References

[1] William M. Bolstad. *Introduction to Bayesian Statistics*. August 2007.

[2] Derrick Kondo et al. Characterizing resource availability in enterprise desktop grids. *Future Generation Comp. Syst.*, 23(7):888–903, 2007.

[3] Peter A. Dinda. A prediction-based real-time scheduling advisor. In *IPDPS*. IEEE Computer Society, 2002.

[4] EGEE Team, LCG. [Online] http://lcg.web.cern.ch/, 2007.

[5] Song Fu and Cheng-Zhong Xu. Exploring event correlation for failure prediction in coalitions of clusters. In *SC*. ACM, 2007.

[6] Alexandru Iosup, Mathieu Jan, Ozan Sonmez, and Dick Epema. On the dynamic resources availability in grids. In *Grid 2007*, Austin, TX, USA, September 19-21.

[7] James W. Mickens and Brian D. Noble. Exploiting availability prediction in distributed systems. In *NSDI*. USENIX, 2006.

[8] R. Wolski et al. Automatic methods for predicting machine availability in desktop grid and peer-to-peer systems. In *CCGRID '04*.

[9] Ramendra et al. Critical event prediction for proactive management in large-scale computer clusters. In *KDD*, pages 426–435, 2003.

[10] Xiaojuan Ren, Seyong Lee, Rudolf Eigenmann, and Saurab Baghci. Resource availability prediction in fine-grained cycle sharing systems. In *HPDC*, 2006.

[11] Brent Rood and Michael J. Lewis. Multi-state grid resource availability characterization. In *Grid 2007*, Austin, TX, September 17-19,.

[12] Bianca Schroeder and Garth A. Gibson. A large-scale study of failures in high-performance computing systems. In *DSN*, pages 249–258. IEEE Computer Society, 2006.

[13] The Austrian Grid Consortium. [Online] http://www.austriangrid.at, 2007.

[14] The TeraGrid Project. [Online] http://www.teragrid.org/, 2007.

[15] Thomas Fahringer et al. Askalon: a grid application development and computing environment. In *GRID*, 2005.