Article:
Alsoghayer, R and Djemame, K (2014) Resource failures risk assessment modelling in distributed environments. Journal of Systems and Software, 88. 42 - 53. ISSN 0164-1212

https://doi.org/10.1016/j.jss.2013.09.017

Resource Failures Risk Assessment Modelling in Distributed Environments

Raid Alsoghayer[a], Karim Djemame[b,*]

[a]*Computer Science Department, King Saud University, Riyadh 11543, Saudi Arabia*
[b]*School of Computing, University of Leeds, Leeds LS2 9JT, UK*

Abstract

Service providers offer access to resources and services in distributed environments such as Grids and Clouds through formal Service level Agreements (SLA), and need well-balanced infrastructures so that they can maximise the Quality of Service (QoS) they offer and minimise the number of SLA violations. We propose a mathematical model to predict the risk of failure of resources in such environments using a discrete-time analytical model driven by reliability functions fitted to observed data. The model relies on the resource historical data so as to predict the risk of failure for a given time interval. The model is evaluated by comparing the predicted risk of failure with the observed risk of failure, and is shown to accurately predict the resources risk of failure, allowing a service provider to selectively choose which SLA request to accept.

Keywords:
Grid Computing, Cloud Computing, Risk Assessment, Quality of Service, Resource Failure, Markov Chains

*Corresponding author
Email addresses: ralsoghayer@ksu.edu.sa (Raid Alsoghayer), K.Djemame@leeds.ac.uk (Karim Djemame)

1. Introduction

Advances in Grid/Cloud computing research have in recent years resulted in considerable commercial interest in utilising infrastructures such distributed environments provide to support commercial applications and services [1]. However, significant developments in the areas of risk and dependability are necessary before widespread commercial adoption can become a reality. Specifically, risk management mechanisms need to be incorporated into Grid/Cloud infrastructures, in order to move beyond the best-effort approach to service provision that current Grid infrastructures follow [2] .

Risk management is a discipline that addresses the possibility that future events may cause adverse effects and is defined in [3] as the process whereby organisations methodically address the risks attaching to their activities with the goal of achieving sustained benefit within each activity and across the portfolio of all activities.

The importance of risk management in Grid/Cloud computing is a consequence of the need to support various parties involved in making informed decisions regarding contractual agreements. Consider a provider that wishes to offer use of its resources as a pay-per-use service. Interactions between a provider and an end-user (a service consumer or a broker acting on their behalf) can then be governed through a Service Level Agreement (SLA), contractually defining the resource provider's obligations, the price the end-user must pay and the penalty the provider needs to pay in the event that it fails to fulfil its obligations. The use of SLAs to govern such interactions in Grid computing is gaining momentum [4, 5, 6]. However, such agreements represent a business risk to the parties involved. An SLA violation could be caused by various events such as a node outage or network failure. Consequently a provider may be unwilling to implement such an approach without effective risk assessment.

This paper focuses on a specific aspect of risk management as applied to Grid/Cloud computing: techniques that can be used by a resource provider to assess the risk of failure of resources within its infrastructure. This will enable a provider to identify infrastructure bottlenecks, evaluate the likelihood of an SLA violation and, where appropriate, mitigate potential risk, in some cases by identifying fault-tolerance mechanisms such as job migration to prevent SLA violations. A resource provider's reputation is closely related to the reliability of its product (here risk assessment). The more reliable the provider's risk assessment is, the more likely the provider is to have a favourable reputation.

A mathematical model for the prediction of the resources risk of failure is proposed with the use of a discrete-time analytical model driven by availability functions fitted to observed historical data.

This research has considered resource failures in Grid computing and the proposed mathematical model can equally be applied in a cloud environment. The main contributions of this paper are:

- A detailed analysis of Grid resource failures using failure data collected from different Grid resources and spanning for three years. The analysis

focuses on the statistical properties of the failure data, including the root cause of failures, the mean time between failures, and the mean time to repair.

- A model to describe the time between failures in Grid resources, as well as a model for the time to repair a resource. Modelling failures and repairs are crucial in the design of reliable systems and also when creating realistic benchmarks and test-beds for reliability testing.
- A model to predict the Grid resources risk of failure, which can also be used to rank Grid resources.

The remainder of the paper is organised as follows. Section 2 introduces the risk management discipline. Section 3 explains the vision of risk in Grid computing. Section 4 provides an overview of Grid resources failures data along with the data-collection process and presents an analysis of such data. Section 5 presents the proposed model to predict the risk of failure of a Grid resource using a discrete time analytical approach driven by reliability functions fitted to observed failures data. Section 6 presents some related work and section 7 ways to further extend this research. In conclusion, section 8 provides a summary of the research.

2. Risk Management

Risk management plays an important role in a wide range of fields, including statistics, economics, systems analysis, biology and operations research. The most central concepts in risk mamagement are the following: an *asset* is something to which a party assigns value and hence for which the party requires protection. An *unwanted incident* is an event that harms or reduces the value of an asset. A *threat* is a potential cause of an unwanted incident whereas a *vulnerability* is a weakness, flaw or deficiency that opens for, or may be exploited by, a threat to cause harm to or reduce the value of an asset. Finally, *risk* is the likelihood of an unwanted incident and its consequence for a specific asset, and *risk level* is the level or value of a risk derived from its likelihood and consequence. For example, a server is an asset, a threat may be a computer virus, the vulnerability a virus protection not up to date, which leads to an unwanted incident: a hacker getting access to this server. The likelihood of the virus creating a back door to the server may be medium, but the integrity of the server (consequence in term sof harm) may be high.

As explained earlier, this paper focuses on a specific aspect of risk management as applied to Grid computing: methods that can be used by a resource provider to evaluate the risk of failure of Grid resources. In this context, a Grid resources is an asset, a threat may be a loss of its connectivity, the vulnerability a faulty hardware, which leads to an unwanted

incident: the failure of the resource. The paper only focuses on the *likelihood* (probability) of Grid resource failures, and therefore uses the terms Probability of Failure and Risk of Failure interchangeably.

3. Risk Aware Grid Computing - The Vision

The overall vision is the production of a risk aware decision support system allowing individuals to negotiate and consume Grid resources using Service Level Agreements (SLA). This embraces an extended approach to the utility computing business model, which fits in an open market business model (for example for access to compute power) as used in sectors such as finance, automotive, and energy. This section presents the main actors (end-user and resource provider), an example scenario in which they participate, and the resource provider architectural components for risk assessment.

3.1. Actors

An end-user is a participant from a broad public approaching the Grid in order to perform a task comprising of one or more services. The user must indicate the task and associated requirements formally within an SLA template. Based on this information, the end-user wishes to negotiate access with providers offering these services, in order that the task is completed. The end-user must make informed, risk-aware decisions on the SLA quotes it receives so that the decision is acceptable and balances cost, time and risk.

A provider offers access to resources and services through formal SLAs specifying risk, price and penalty. Providers need well-balanced infrastructures, so they can maximise the Quality of Service (QoS) and minimise the number of SLA violations. Such an approach increases the economic benefit and motivation of end-users to outsource their IT tasks. A prerequisite to this is a provider's trustworthiness and their ability to successfully deliver an agreed SLA. The assessment of risk allows the provider to selectively choose which SLA requests to accept.

Note the possible consideration of a broker in such context, which acts as a matchmaker between end-users and providers, furnishing a risk optimised assignment of SLA requests to SLA quotes [7]. It is responsible for matching SLA requests to resources and services, which may be operated by an arbitrary number of providers. The broker's goal is to drive this matchmaking process to a conclusion, when the provider makes an SLA offer.

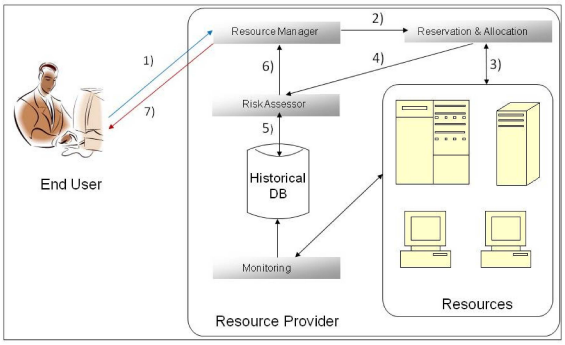

Figure 1: Resource Reservation and Risk Assessment - Resource Provider Components Interaction

3.2. Motivating Scenario

Considering the situation where a provider wishes to offer use of its resources as a pay-per-use service to potential end-users, and where the use of SLAs govern the interaction between them, a provider may need to implement an effective risk assessment prior to making an SLA offer. In this case, the provider computes the risk of failure for each resource and subsequently allocates the resources to the end-user's job. If the resulted allocation fails to satisfy the end-user's requirements, the resource reservation is revisited; if it does satisfy the end-user's requirements, the resource provider then sends back the SLA offer, updated with cost/penalty fee and pre-commits the resources. The end-user either commits to the SLA or rejects it.

Figure 1 provides an overview of the interaction of the provider infrastructure components. The user sends an SLA request to the provider specifying the job requirements (1). The provider's *Resource Manager* requests the *Reservation and Allocation* component to reserve the required resources (2). The Reservation and Allocation component reserves the physical resources (3) and passes to the *Risk Assessor* for each reserved resource the time and duration of the reservation (4). The Risk Assessor computes for each resource the risk of failure based on the resource historical information stored in the *Historical Database* (5). The *Monitoring* component is responsible for gathering all necessary runtime information that is collectable by sensors in the infrastructure. The Risk Assessor returns the risk of failure information to the Resource Manager (6). Finally, the Resource Manager sends a response back to the user (7), either in the form of an SLA offer or reject.

4. Analysis of Failures in Grid Environments

As explained in section 2 the resource provider's assets are the Grid resources in the context of this paper, the Risk of Failure (ROF) of which is of great concern. Therefore, the probability of failure of a resource as well as the impact of the failure need to be identified. In order to compute such probability, the events that cause a resource to fail first need to be specified. Grid resources can fail as a result of a failure of one or more of the resource components, such as CPU or memory; this is known as *hardware* failure. Another event which can result in a resource failure is the failure of the operating system or programs installed on the resource; this type is known as *software* failure. The third event is the failure of communication with the resource; this is referred to as *network* failure. Finally, another event is the disturbance to the building hosting the resource, such as a power cut or an air conditioning failure; this type is event is known as *environment* failure. Sometimes, it is difficult to pinpoint the exact cause of the failure, i.e. whether it is hardware, software, network, or environment failure; this is therefore referred to as *unknown* failure.

A set $E = (E_H \cup E_S \cup E_N \cup E_E \cup E_U)$ denotes a full set of events which include E_H the events causing hardware failures, E_S causing software failures, E_N causing network failures, E_E causing environment failures, and E_U denoting events which cause unknown failures.

An assumption the paper makes is that the sets E_H, E_S, E_N, E_E, and E_U are disjoint (or mutually exclusive), i.e. if a resource fails at a given time t, then only one event from the sets could have caused this failure [1]. Of course, it is possible that two events or more from different sets might take place at once, yet the person responsible for the resource maintenance will only identify a single event (see section 4.1).

4.1. Failures Data Gathering

Monitored data is essential in scheduling, performance analysis, performance tuning, performance prediction, optimisation of Grid systems etc. Therefore, gathering data relating to past and current status of Grid resources is an essential activity. Monitoring resource failures is crucial in the design of reliable systems, e.g. the knowledge of failure characteristics can be used in resource management to improve resource availability [8]. Furthermore, calculating the risk of failure of a resource depends on past failures as well.

[1] Note that causality is the relationship between an event (the cause) and a second event (the effect), where the second event is understood as a consequence of the first. In risk management, a threat scenario is a chain of events that is initiated by a threat and that may lead to an unwanted incident.

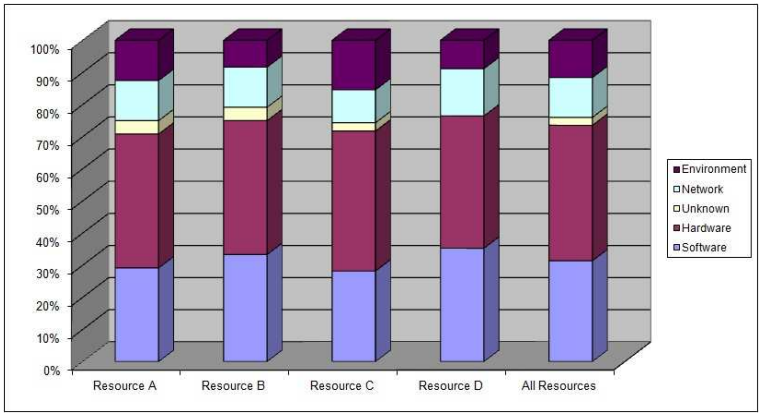

Figure 2: Breakdown of Resource Failures into Root Causes (Site 1)

Failures data was collected from the publicly available Grid Operations Centre Data Base (GOCDB) [9], the official repository for storing and presenting European Grid Infrastructure (EGI) [10] topology and resources information, which includes monitored data from the UK National Grid Service (NGS) [11] and Worldwide LHC Computing Grid (WLCG) [12]. In GOCDB, a Grid resource provider is represented as a *site*, e.g. the European Organisation for Nuclear Research (CERN) [13]. A Grid resource is represented as a *node*, a computer/cluster providing Grid services. A *downtime* is a period of time in which a Grid node is declared to be inoperable. A downtime record contains a unique downtime ID, downtime classification (scheduled or unscheduled), the severity of the downtime, the contact person who recorded the downtime, the record date, the start and end of the downtime period, the description of the downtime, and the entity affected by the downtime. *Scheduled* downtimes are planned and agreed in advance, while *unscheduled* downtimes are unplanned and are usually triggered by an unexpected failure. The status of the resource is either *at risk* (where the resource is working as normal, but may experience problems) or *outage* (where the resource is completely unavailable). This research has considered downtime data of seven Grid resources from two different Grid sites: four from Site 1, and three from Site 2, in order to generalise the findings. Downtime data for all resources span over three years (2008-2010). Downtime data include scheduled and unscheduled events, but only unscheduled downtimes in relation to resource failures are considered. The reason for this is that the use of advance reservation in Grid systems takes into account scheduled downtimes. Grid schedulers are expected to have access to information regarding scheduled downtimes so that a job will not be scheduled on a resource with a planned downtime.

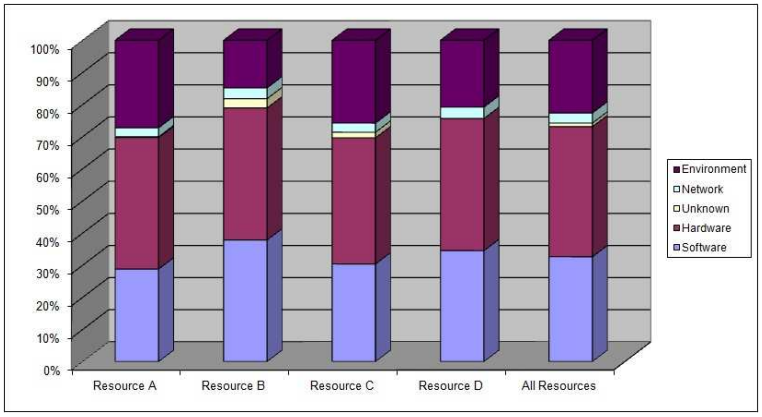

Figure 3: Breakdown of Resource Downtime into Root Causes (Site 1)

4.2. Failures Data Analysis

Resource failure data is analysed with respect to three important properties of system failures: root cause breakdown, repair time, and time between failures. The sequence of failure events are studied using stochastic process [14] and the distribution of its time between failures is also considered. Notably, repair times are characterised for each resource using the mean, median and standard deviation. We also consider the empirical Cumulative Distribution Function (CDF) of repair time for each resource, as well as how well it fits the probability distributions commonly used in reliability theory: Exponential, Weibull, Gamma and Lognormal distributions. These distributions fit the data well, and so there are no reasons for using other distributions or more degree of freedom e.g. a phase-type distribution. Notably, the Maximum Likelihood Estimation (MLE) is used to parameterise the distributions and thereby evaluate the goodness of fit by visual inspection, and the negative log-likelihood test. The MLE, unlike moment estimation, is consistent, unbiased and efficient. The CDF for the time between failures for each resource is analysed also using MLE and the negative log-likelihood test [14].

Considering the description of the cause of failures information available in GOCDB data, the description of the failures was mapped into five different categories: *Environment, Network, Software, Hardware and Unknown.* Figure 2 shows the percentage of failure for each category in Site 1, software and hardware failures being the largest contributors. The actual percentage for software failures ranges between 28% to 35%, and between 41% and 43% for hardware failures. Figure 3 shows the percentage of downtime for each category in Site 1. Software and hardware failures also contribute hugely to the downtime: between 28% and 37% for software and between 39% and 41% for hardware. Note that the downtime due to

Table 1: Repair Time (minutes) - Mean, Median, and Standard Deviation - Site 1

Resource	A	B	C	D
Mean	1922	1611	1658	1829
Median	945	433	1116	865
Stand.Dev.	2496	2341	2089	2346

environment failures is high, ranging between 14% and 27%. The reason for this is that the site often had air conditioning failure, which required a long maintenance work.

The *repair time* metric is investigated by considering first how the repair time varies among resources, the statistical proprieties of the repair time for each resource including their distributions, and finally how the root cause affects the repair time. Table 1 shows the mean, median and standard deviation for the repair time in Site 1. The mean repair time of all resources is high because the repair time depends mainly on the availability of the Grid administrator, and this site does not have 24-hour user support. Thus, any resource failure occurring after normal working hours is not dealt with before the next working day; this also holds for weekends and public holidays. Another reason is that there is no automatic monitoring in place that will report a resource failure when it occurs. The standard deviation shows a large spread of the data.

Another observation is that resource repair time is highly variable, which indicates that the Exponential distribution is not conventional to express it. With this in mind, it should be noted that an Exponential distribution with failure rate λ the mean is $1/\lambda$ and the median $\ln(2)/\lambda$, which is $0.6931/\lambda$ [15]; thus, the mean and median should not have a huge difference. To confirm this observation, the empirical Cumulative Distribution Function (CDF) for repair time in each resource is fitted with four standard distributions: Exponential, Weibull, Gamma and Lognormal. The CDF - referred to as $F(x)$ - describes the probability distribution of a real-valued random variable X to be less than x:

$F(x) = P\{X < x\}$

That is, for a given value x, $F(x)$ is the probability that the observed value of X will be at most x.

Figure 4 (left) shows the CDF of repair time for Resource A, Site 1. Visual inspection indicates both Lognormal and Weibull distributions have a good fit, but Lognormal fit the data slightly better when tested using the negative log-likelihood. The Exponential distribution is the worst fit, as expected, and it is not accurate for the purpose of modelling the repair time of this resource. Similar findings are recorded for resources B and C with Weibull and Lognormal having the best fit respectively, wheras for

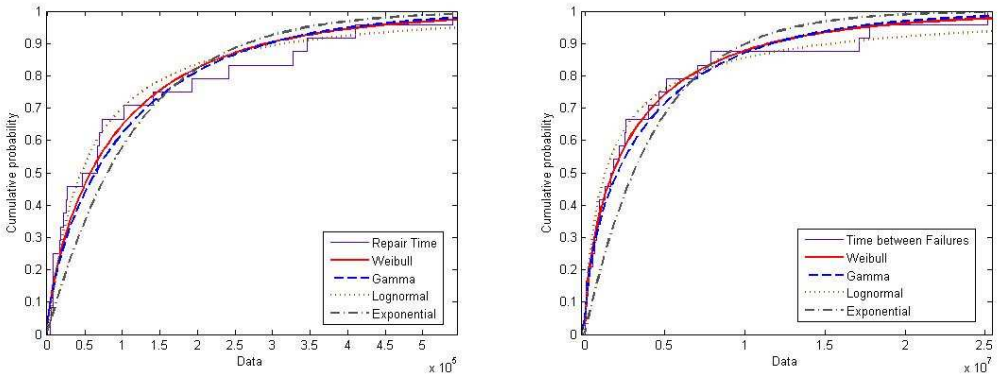

Figure 4: (a) Resource A Site 1 - Repair Time; (b) Time between Failures

Table 2: Resource A Site 1 - Repair Time (minutes) according to Failure Categories

Category	Software	Hardware	Network	Environment	Unknown
Mean	1900	1887	432	4185	1120
Median	1120	961	120	5444	1120
Stand.Dev.	2136	2710	593	3451	-

resource D Weibull and Lognormal distributions create an equally good visual fit. In the following, due to space limitations the results will be shown for resource A in site 1 only.

Table 2 shows the mean, median and standard deviation for resource A site 1 repair time according to failure categories. The repair time average is 7 hours for network failures and up to 3 days for environment errors. Overall, the repair time is highly variable for all resources in site 1. Another observation is that software and hardware failure affect individual machines, whilst a network or an environment failure, e.g. a power cut may affect a cluster or even the entire Grid site.

The *time between (unscheduled) failures* is also analysed for each resource. Figure 4 (right) shows the CDF of the time between failures for Resource A, Site 1. In this case, the distribution between failures is well modelled by a Weibull distribution, which creates a good visual fit and the best fit when tested using the negative log-likelihood. The Gamma distribution is the second best fit. Similar findings are recorded for resources B and C, wheras for resource D Weibull and Gamma distributions create an equally good visual fit. The Weibull distribution is the most popular and widely used method of analysing and predicting failures and malfunctions of all types, and offers flexibility in modelling failure rates [16].

The Weibull distribution is used to mathematically characterize the probability of system failures as a function of time. How the time since the last failure influences the expected time until the next failure is captured by a distribution's hazard rate function. An increasing hazard rate function predicts that the probability of failure increases with time. A decreasing hazard rate function predicts the reverse. The maximum likelihood estimation is used to predict the shape parameter, which is found to be 0.63 for resource A. This shape parameter of less than 1 indicates that the hazard rate function is decreasing, i.e. not seeing a failure for a long time decreases the chance of seeing one in the near future.

Random processes [17] are tested as probabilistic models for Grid resource failures [18]. The results show that random processes are not suitable for modelling Grid resources failure. The *Homogeneous Poisson Process* (HPP) assumes that the time between failures follows the exponential distribution, yet the time between failures in Grid environments follows a Weibull distribution. The *renewal process* assumes that the repair of failed component return it to as good as new state, yet in Grid environments repairs do not return the resources to as good as new state. The *modified renewal process* assumes that the distribution of the first failure differs from the distribution of the time of the second, third or subsequent failures. This assumption is not valid in Grid environments since the distribution of the time between failures follows the Weibull distribution and does not change between subsequent failures. The *alternating renewal process* assumes that the distribution of the time between failures is identical and independent. In Grid environments assuming an identical distribution is inadequate. Finally the *Non-Homogeneous Process (NHPP)*, which is widely assumed in modelling computer systems, is not fit for modelling Grid resources failure. Results in [18] show that it is highly unlikely that Grid resource failures are modelled by a NHPP following a power or exponential low.

So far, the behaviour of Grid resources has been described in statistical terms. Next, a new mathematical model to assess the risk of Grid resource failures is introduced.

5. Risk of Failure Models

5.1. Availability Model

Recall that a Grid resource ROF at time t is the probability of the resource not functioning at t. This can be defined as one minus the probability of the resource functioning at t. By computing the probability of the resource functioning at t, known as availability $A(t)$, the resource ROF is expressed as $(1 - A(t))$. The proposed availability model is based on Markov Models [19] where the concepts of *state* and *state transition* are used to model the

Grid resource states as shown in data collected from GOCDB (see section 4.1): up (state 0), at risk (state 1), and outage (state 2).

The memoryless property or constant failure rate assumption is a crucial assumption in Markov modelling, which is a popular technique for reliability analysis. In other words, for Grid resources the transition probabilities between states are determined by the present state only, and not by history. For *continuous-time* models, the length of time already spent in a state does not influence either the transition rate of the next state or the remaining time in the same state before the next transition. This general assumption implies that the waiting time spent in any state is exponentially distributed in the continuous-time case or geometrically distributed in the *discrete-time* models.

Thus, Markov models assume that failure rates are constant, thereby leading to exponentially distributed inter-arrival time of failures and Poisson arrival of failures. A useful generalisation of Markov Models is the Time-Varying Markov Models, which allow state transition probability to change over time; thus, the failure rate is no longer assumed as constant [19]. With this relaxed assumption, the Grid resources can be modelled with the use of the time-varying Markov model. Since Grid resources failures and repairs occur at varying intervals, a continuous time-varying Markov model is used for Grid resource availability (see Figure 5). The transition matrix for the continuous time-varying Markov model is:

$$P(t) = \begin{vmatrix} 0 & Z_W(t) & Z_F(t) \\ Z_R(t) & 0 & z_F(t) \\ Z_G(t) & 0 & 0 \end{vmatrix}$$

The resource will start in State 0 (UP) and operates until either: (1) the performance degrades and the resource transits to State 1 (AT RISK); or (2) the resource stops working and transits to State 2 (DOWN). The rate of events causing transition from state 0 to 1 is $Z_W(t)$ whilst $Z_R(t)$ is the rate of recovery events that result in the resource returning to State 0. Moreover, $Z_F(t)$ is the rate of events that leads to resource failure, whereas $Z_G(t)$ is the rate of repair resulting in the resource returning to State 0.

In order to predict the Grid resource availability, the continuous time-varying Markov model is developed by applying transition functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$ derived from the distributions fitted to failure data. Therefore, Section 5.2 deals with establishing distributions for the transition functions, whilst Section 5.3 presents the analysis of the model.

5.2. Failure Data Fitting Distributions

In order to determine the time-varying functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$ for the continuous time-varying Markov model shown in Figure

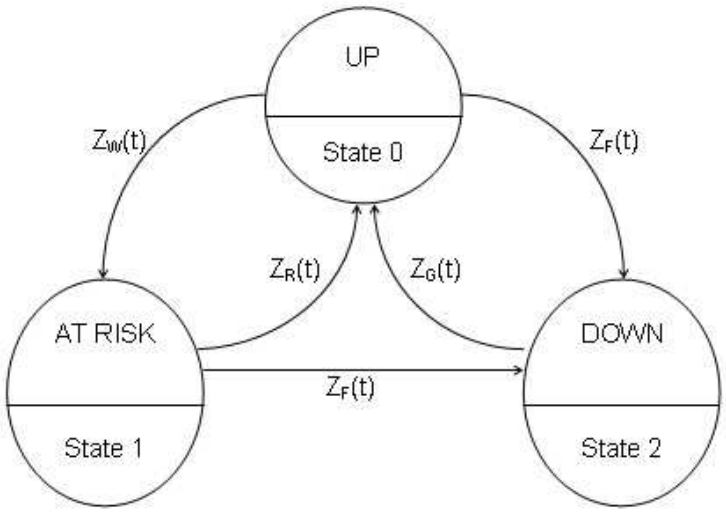

Figure 5: Continuous Time-Varying Markov Model for Resource Availability

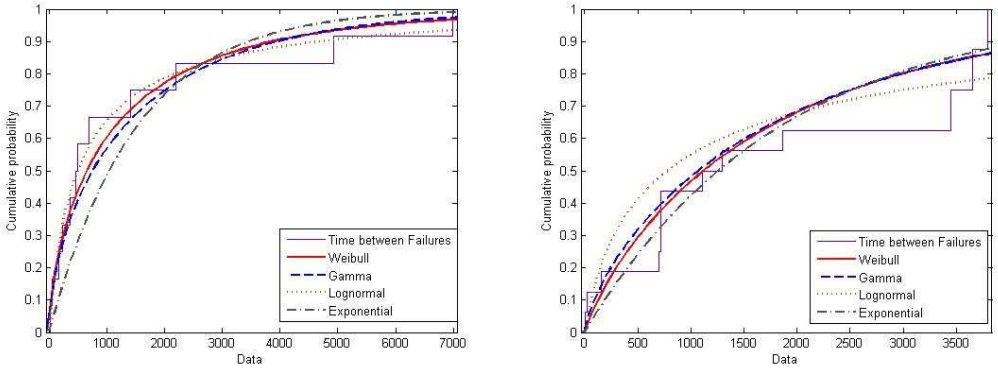

Figure 6: Time-Varying Functions $Z_W(t)$ and $Z_R(t)$ (Resource A Site 1)

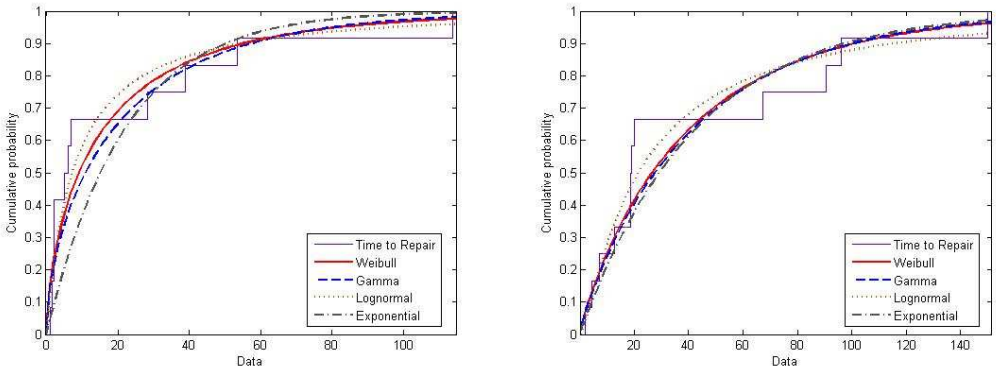

Figure 7: Time-Varying Functions $Z_F(t)$ and $Z_G(t)$ (Resource A Site 1)

5, the sequence of unscheduled events as well as the downtime data are analysed for each resource. There are two types of events: the first is At Risk, which represents a transition from State 0 to State 1; the second is complete failure, which represents the transition from State 0 to State 2. For each event, the time to repair the resource is recorded and represents the time to return the resource to State 0 from State 1 or 2.

The CDF of the functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$ for each resource is fitted with four standard distributions: Exponential, Weibull, Gamma and Lognormal; this helps to determine the best fit for each function. The MLE is used to parameterise the distributions, and the goodness of fit is evaluated using the negative log-likelihood test.

Figures 6 and 7 show that for resource A in site 1 the time between transitions from State 0 to 1, $Z_W(t)$, is well modelled by Weibull or Lognormal distribution, yet the Weibull is a better fit when tested with the use of a negative log-likelihood. The time between transitions from State 0 to 2, $Z_F(t)$, is well modelled by Weibull or Gamma; both distributions create an equally good visual fit and the same negative log-likelihood. The repair time is the time to return the resource to State 0 from State 1 or State 2. Moreover, the time between the transitions from State 1 to State 0, $Z_R(t)$, is well modelled by Weibull or Lognormal distribution, yet the Lognormal is a better fit when tested with the use of a negative log-likelihood. Finally, the time between transitions from State 2 to State 0, $Z_G(t)$, is well modelled by Weibull or Lognormal distribution, yet the Weibull is a better fit when tested using the negative log-likelihood. Table 3 shows the individual resources along with the best distribution fit for the four transition functions.

Table 3: Best Fit Distribution for the Transition Functions (All Resources)

Site	Resource	$Z_W(t)$	$Z_F(t)$	$Z_R(t)$	$Z_G(t)$
1	A	Weibull	Weibull	Lognormal	Weibull
	B	Weibull	Weibull	Lognormal	Weibull
	C	Weibull	Weibull	Weibull	Lognormal
	D	Weibull	Gamma	Lognormal	Weibull
2	A	Weibull	Weibull	Weibull	Lognormal
	B	Weibull	Gamma	Weibull	Lognormal
	C	Weibull	Gamma	Weibull	Lognormal

5.3. Risk of Failure Model

Following the results in Table 3 we make the assumption that the time-varying functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$ are based on a Weibull probability density function with unique shape α and scale λ values for each function:

$$Z_W(t) = \alpha_W \ \lambda_W(\lambda_W \mathrm{t})\alpha_W \text{ - } 1\mathrm{e}^{-(}\lambda_W \mathrm{t})\alpha_W$$
$$Z_R(t) = \alpha_R \ \lambda_R(\lambda_R \mathrm{t})\alpha_R \text{ - } 1\mathrm{e}^{-(}\lambda_R \mathrm{t})\alpha_R$$
$$Z_F(t) = \alpha_F \ \lambda_F(\lambda_F \mathrm{t})\alpha_F \text{ - } 1\mathrm{e}^{-(}\lambda_F \mathrm{t})\alpha_F$$
$$Z_G(t) = \alpha_G \ \lambda_G(\lambda_G \mathrm{t})\alpha_G \text{ - } 1\mathrm{e}^{-(}\lambda_G \mathrm{t})\alpha_G$$

To solve the continuous time-varying Markov model the method that approximates the continuous-time process with discrete-time equivalent [19] is used. Figure 8 shows the resulting discrete-time Markov model for time step δt. Since more than one transition may occur during a time step, the model must take into account the joint probability of state transition. As the state transition probabilities for the discrete-time Markov model change over time, we need to drive an expression for A(t), B(t), C(t), D(t), and E(t). This is achieved thanks to the models developed by Siewiorek and Swarz [20].

The probability transition equations are derived, in which $q_{ij}(s,t)$ is the probability that the system is in state j at time t given that it was in state i at time s ($s \leq t$). With this notation, in matrix form the Chapman-Kolmogorov equation [21] is:

$$\mathrm{Q(s,\ t) = Q(s,\ k)\ Q(k,\ t)\ s \leq k \leq t}$$

Letting k = t - 1,

$$\mathrm{Q(s,\ t) = Q(s,\ t\ -\ 1)\ Q(t\ -\ 1,\ t)}$$

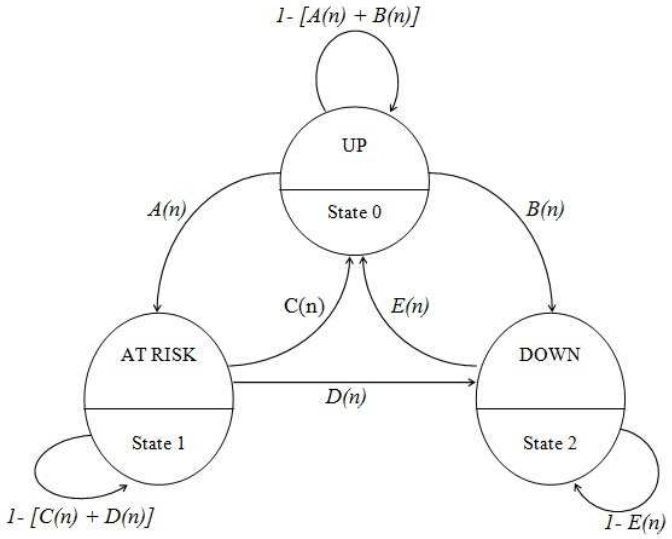

Figure 8: Discrete Time Markov Model for Resource Availability

Defining P(t) = Q(t, t + 1),

$$Q(s, t) = Q(s, t - 1)P(t - 1)$$

Expanding the equation recursively

$$Q(s,t) = Q(s,t-2)P(t-2)P(t-1) = Q(s,t-3)P(t-3)P(t-2)P(t-1) \tag{1}$$

Yielding to:

$$Q(s,t) = \prod_{i=s}^{t-1} P(i) \tag{2}$$

In order to translate the continuous-time probability functions into discrete-time probability functions, a discrete-time probability distribution is established that corresponds to the continuous-time distribution. The corresponding parameters can then be calculated for the desired time-step δt. Furthermore, a discrete-time approximation has to consider the probability of two failures during the same interval. Recall that the time-varying reliability functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$ are based on a Weibull probability density function (pdf).

pdf = f(t) = $\alpha\lambda(\lambda t)^{(\alpha-1)}e^{-(\lambda t)^{\alpha}}$

The corresponding discrete Weibull function, probability mass function, is:

$$\text{pmf} = \text{f(k)} = q^{k^{\alpha}} \text{ - } q^{(k+1)^{\alpha}}$$

Given that $f(k)$ is defined as the probability of an event occurring between time Δt and time (k + 1)Δt for some chosen interval size Δt, the probability mass function can be expressed as:

$$\text{f(k)} = \text{P[no event by k}\Delta t\text{] - P[no event by (k + 1)}\Delta t\text{]}$$
$$\text{f(k)} = \text{R(k) - R(k+1)}$$

R(k) is the reliability function. By substituting the continuous-time equivalents yields:

$$\text{f(k)} = \text{R(k}\Delta t\text{) - R[(k + 1)}\Delta t\text{]}$$
$$\text{f(k)} = e^{-(\lambda k\Delta t)^{\alpha}} \text{ - } e^{-(\lambda(k+1)\Delta t)^{\alpha}}$$

By rearranging terms, we can find that:

$$\text{q} = e^{-(\lambda\Delta t)^{\alpha}}$$

The probability mass functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$ provide the reliability for a discrete time step $n = t_n/\Delta$t. The time-varying functions are:

$$q_W = e^{-(\lambda_W\Delta t)^{\alpha_W}}$$
$$Z_W(t) = 1 \text{ - } q_W^{(t+1)\alpha_W - t\alpha_W}$$
$$q_R = e^{-(\lambda_R\Delta t)^{\alpha_R}}$$
$$Z_R(t) = 1 \text{ - } q_R^{(t+1)\alpha_R - t\alpha_R}$$
$$q_F = e^{-(\lambda_F\Delta t)^{\alpha_F}}$$
$$Z_F(t) = 1 \text{ - } q_F^{(t+1)\alpha_F - t\alpha_F}$$
$$q_G = e^{-(\lambda_G\Delta t)^{\alpha_G}}$$
$$Z_G(t) = 1 \text{ - } q_G^{(t+1)\alpha_G - t\alpha_G}$$

The transition probability functions in Figure 8, which represent the probability of transition from one state to another state, are:

$$A(t) = [1 \;\; Z_F(t)]\; Z_W(t)$$
$$B(t) = [1 \;\; Z_W(t)]\; Z_F(t)$$
$$C(t) = [1 \;\; Z_F(t)]\; Z_R(t)$$
$$D(t) = [1 \;\; Z_R(t)]\; Z_F(t)$$
$$E(t) = Z_G(t)$$

The transition probability matrix

$$P(t) = \begin{vmatrix} 1 - (A(t) + B(t)) & A(t) & B(t) \\ C(t) & 1 - (C(t) + D(t)) & D(t) \\ E(t) & 0 & 1 - E(t) \end{vmatrix}$$

- A(t) is the probability of transiting from *Up* to *At Risk*;
- B(t) is the probability of transiting from *Up* to *Down*;
- C(t) is the probability of transiting from *At Risk* to *Up*;
- D(t) is the probability of transiting from *At Risk* to *Down*;
- E(t) is the probability of transiting from *Down* to *Up*.

Taking into account that $P_{i,j}$ is the probability of a transition from state i to state j, it can then be stated that the probability of transition $P_{0,0}$ is the probability of remaining in State 0, which is 1 minus the probability of leaving State 0, hence 1 - (A(t) + B(t)). The same can then be applied for the probability of transition $P_{1,1}$ and $P_{2,2}$.

$P(t)$ can be used to compute instantaneous risk of failure, which is the probability that the system will not be operational at any random time t. Another important aspect is the risk of failure duration, which refers to the probability that the system will not be operational for a certain duration (e.g. job execution time). Computing the risk of failure duration is an iterative process. Accordingly, applying the appropriate values for α and λ, starting at S = start time, $P(t)$ is computed forward for successive values of t until the desired finish time T = t ΔT is reached.

5.4. Evaluation

Adopting the technique described in the previous section, the transition matrix $P(t)$ is computed for each resource using the data from GOCDB with Δt = 1 hour. Since Grid jobs usually require long execution times, ΔT should be selected accordingly. However, long ΔT lowers the accuracy of the model, since a state transition is not promptly recorded. On the other hand, short ΔT has the overhead of calculating $P(t)$ multiple times, despite the probability of transition not changing. Therefore, ΔT was selected to be 1 hour.

The observed risk of failure is calculated considering the data generated over the last 6 months of 2010. The Weibull shape parameter for resource

Table 4: The Shape α and Scale λ Parameters for Functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$

Site	Resource	$Z_W(t)$		$Z_F(t)$		$Z_R(t)$		$Z_G(t)$	
		α	λ	α	λ	α	λ	α	λ
1	A	0.6741	1124.29	0.6002	1818	0.665	15.784	0.899	40.08
	B	0.8616	376.63	0.6409	1385.26	0.7385	10.454	0.5779	47.05
	C	0.7154	691.27	0.6384	1113.28	0.8022	17.387	0.8708	32.37
	D	0.8326	1138.13	0.6236	974.053	0.7565	12.936	0.8610	37.80
2	A	0.5930	4160.27	0.8959	866.254	0.8715	11.014	0.7814	6.676
	B	1.0563	398.589	0.6806	613.096	0.7679	7.8319	0.6767	9.946
	C	0.8937	321.602	0.6930	657.811	0.9098	10.984	0.7593	7.392

failures is less than 1, which means that, following a failure, the risk of another failure occuring *soon* increases. Therefore, a short time-span does not reflect the true behaviour of the resource failures.

Table 4 shows, for the resources considered, the values of the Weibull shape α and scale λ parameters for the functions $Z_W(t)$, $Z_R(t)$, $Z_F(t)$, and $Z_G(t)$. The MLE was used to estimate these parameters. The risk of failure is calculated as the sum of the probability of transition from *Up* to *At Risk* and the probability of transitioning from *Up* to *Down*.

The data from GOCDB is used to validate the predicted risk of failure. Let T_{Down} denote the time the resource is down, and T_{Up} the time the resource is up, the observed accumulated risk of failure is defined as:

$$ROF = \frac{T_{Down}}{T_{Down}+T_{Up}}$$

Figure 9 shows the predicted one-day duration risk of failure over a number of days, as well as the observed risk of failure for resource A (site 1). The resource observed and predicted risks of failure are clearly comparable.

In order to validate the predicted risk of failure, i.e. is a true projection of the observed risk of failure, the two-sample `t test` is used to compare the means of the two groups (observed and predicted risk of failure).

From the above figures and the results of the `t test`, the conclusion is that the risk assessment model predicts accurately the resources risk of failure. Therefore, the Grid resource provider can integrate the risk assessment model in order to compute the risk of resources failure.

Grid resource failures are unavoidable and, as such, ranking the resources with respect to their ROF is an important outcome of the risk assessment process. Figure 10 shows the predicted ROF of resources over time. The ROF was computed, assuming all resources are available at time t = 0. It is observed that Site 1 resources ROF is higher than Site 2 resources ROF.

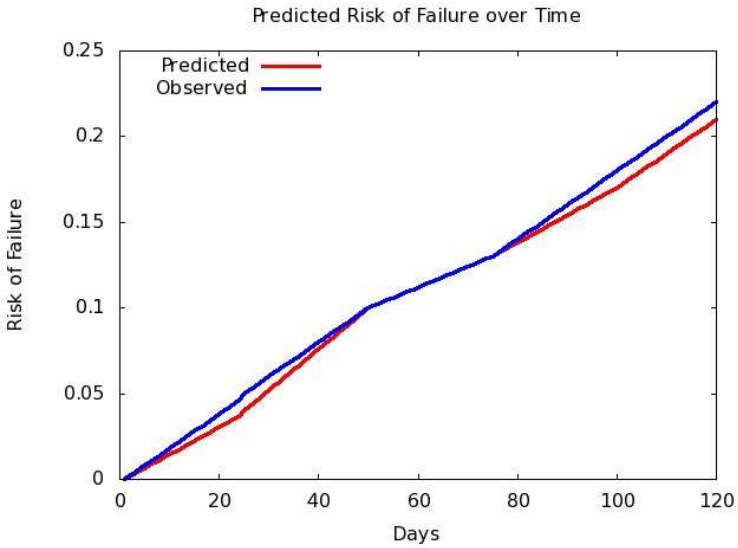

Figure 9: Predicted and Observed Risk of Failure (Resource A Site 1)

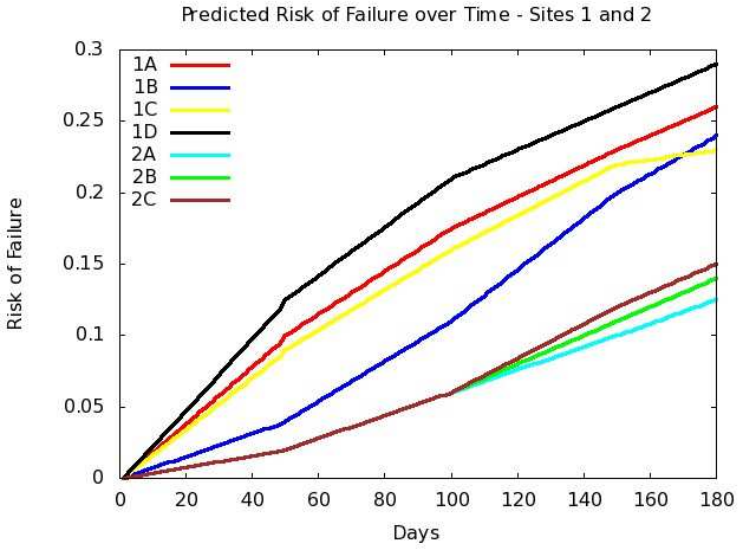

Figure 10: Risk of Failure over time (All Resources)

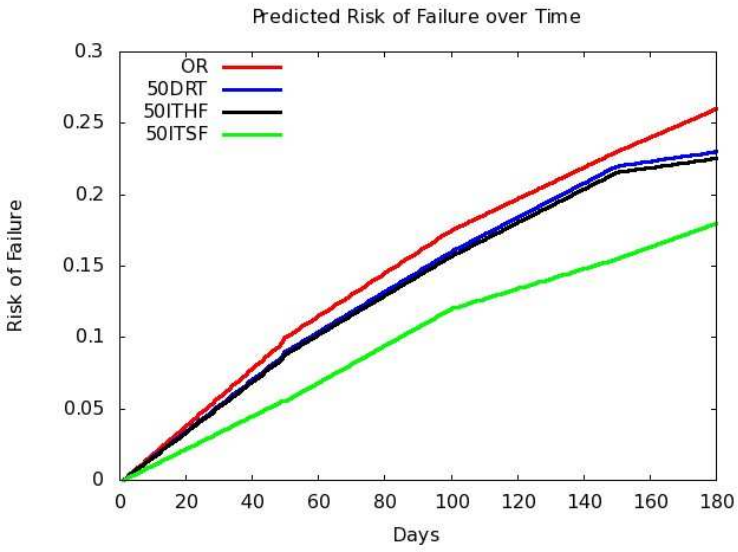

Figure 11: Resource A Site 1 Risk of Failure - with Parameters Variation (OR: Original Resource, 50DRT: 50% Decrease in Repair Time, 50ITHF: 50% Increase in Time between Hardware Failures, 50ITSF: 50% Increase between Software Failures)

In addition to ranking resources, the ROF model can be used to measure the significance of the effect of changes in the Grid environment, which may include the introduction of new hardware and software, or an upgrade to the current infrastructure in order to lower resources repair time. There are various techniques for measuring this significance, the most commonly used of which is the one-at-a-time method [17]. In this case, the assumptions and parameters are changed individually so as to measure the change in output. The one-at-a-time method, along with the ROF model, are powerful tools for Grid providers to understand the limitations of current infrastructures and plan future investments. These tools are explained next.

Assume a Grid resources provider would like to make an investment to minimise the resources ROF. This investment may be on hardware and software as they are the largest contributors to failures (See section 4.2 Root Cause Breakdown), or even on experienced system administrators, expecting a decrease in the resources repair time. Therefore, an experiment is setup in order to investigate the effect of:

1. decreasing the time to repair the resource by 50%;
2. increasing the time between hardware failures by 50%;
3. increasing the time between software failures by 50%.

Figure 11 shows the original ROF, the ROF if the repair time is decreased by 50%, the ROF if the time between hardware failures is increased by 50% and the ROF if the time between hardware failures is increased by 50% over a number of days for Site 1, resource A. Day 0 is the time when the resource became available either after a scheduled maintenance or unscheduled failure. It can be observed that the investment in lowering the repair time is the most rewarding; this is because the repair timein the case of all resourcesis very high, even after 50% decrease. Investment in hardware or software, at this stage, is not much rewarding as the benefit on lowering the ROF is limited.

6. Related Work

The approach in this paper focuses on the risk of resource failure in Grids, which leads to three different fields of research: risk assessment/management, resource failures in distributed systems, and the use of Markov chains for problem solving. Some of the related work is reviewed next.

Risk assessment has been addressed by various projects. The objective of the Consequence project [22] is to provide an information protection framework and to thereby identify the security risk in sharing data in a distributed environment. The risk items are used as a checklist of items to be addressed in the Consequence architecture, without any assessment of the probability and the negative impact of a risk item.

The SLA@SOI project [23] does not explicitly address risk assessment, although it does propose the utilisation of a prediction service for estimating the probability of software and network failures, as well as hardware availability in an attempt to evaluate QoS.

The AssessGrid project [2] proposes a model to estimate the probability of SLA failures in Grid environments, and considers the probability of `n` resources failing for the scheduled duration of a task as well as the probability that `m` reserved resources are available for that duration. The probability of node failure is calculated by assuming that the node failures represent a Poisson process , which is non-homogenous in time. The resource provider risk assessment techniques enable the identification of infrastructure bottlenecks, evaluate the likelihood of an SLA violation and, where appropriate, mitigate potential risk, in some cases by identifying fault-tolerance mechanisms such as job migration to prevent SLA violations [24, 25]. The AssessGrid broker acts as a matchmaker between end-users and providers, furnishing a risk optimised assignment of SLA requests to SLA quotes [7] by evaluating the provider reliability with respect to systematic errors. Here, systematic errors refer to provider errors whereby their risk assessments exhibit a typical trend in the sense that they tend to overestimate/underestimate the risk of failure.

The OPTIMIS project aims towards optimized service construction, deployment, and execution for Cloud Infrastructures by offering tools to efficiently manage the full life cycle of services [26]. The risk factor is considered during all phases of the service lifecycle for the two stakeholders: Service Providers (SP) during service construction, deployment, and operation, and Infrastructure Providers (IP) during admission control and internal operations [27].

A number of studies have looked at resource failures in distributed environments [28, 8, 29, 30, 31, 32, 33, 34, 35]. Schroeder and Gibson [28] analyse failure data collected over 9 years at Los Alamos National Laboratory (LANL), and includes 23,000 failures recorded on more than 20 different systemsmostly large clusters of Symmetric-Multi-Processing (SMP) and Non-Uniform-Memory-Access (NUMA) nodes. The source of a failure falls in one of the following: human errors and environments, such as power outages, hardware failure, software failure, network failure and unknown failures. They find that the time between failure at individual nodes as well as at an entire system fit well by a Gamma or Weibull distribution with decreasing hazard rate (Weibull shape parameter of 0.70.8). The observation that the time between failures is best fitted by a Weibull distribution with decreasing hazard rate is also evidence in [8, 29, 30]. Iosup et al. [35] consider the availability of CPUs in a Grid environment and analyse availability traces recorded from all the clusters. The finding is that the best fit distribution is Weibull with a shape parameter large than 1. The reason for that is that many of todays Grids comprise computing resources grouped in clusters, the owners of which may share them only for limited periods of time. Often, many of a Grids resources are removed by their owner from the systemeither individually or as complete clusters in order to serve other tasks and projects; thus, the unavailability of CPUs is not owing to a system failure but rather their unavailability by their owner. Most of the previous studies considered only short-term availability data [29]. Other studies used statistical modelling to predict failure at Grid level not resources level [30]. More importantly, these studies only consider distribution fitting to failure data.

Nadeem, Prodan Fahringer [30] propose a model to predict the availability of three different Grid resources: dedicated resources which are always available to Grid users, temporal resources which are available to Grid users as long as they are switched on, and on-demand resources which are only available to Grid users by demand. The models proposed are building on Bayes Theorem, and predict the availability as a function of day-of-the-week and hour-of-the-day. This approach has a number of limitations: for example, it does not differentiate between the unavailability as a result of node failure and the unavailability as a result of scheduled maintenance or repair; secondly, the models only consider the hour-of-the-day, and so a 1-minute unavailability and 1-hour unavailability are treated the same; even worse if the unavailability falls at the end of an hour and into the beginning

of the next, and the unavailability subsequently becomes 2 hours.

Another approach to model system availability and reliability in computing is through the use of Markov models. Hacker, Romero and Carothers [31] investigated the use of Semi-Markov models to model node reliability in relation to large supercomputing systems. Platis et al. [32] adopt a two-phase cyclic non-homogeneous Markov chain with the objective to evaluate the performance of a replicated database. Koutras, Platis Gravvanis [33] explored the use of homogeneous continuous time Markov chain with the amount of free memory to model the resource degradation of a computer system. Furthermore, the use of a cyclic non-homogeneous continuous time Markov chain in terms of driving an optimal software rejuvenation model is studied [34]. Dai, Levitin and Wang investigate maximizing the expected profit in Grid systems by partitioning the service task into subtasks and by distributing them among the available resources [36]. A genetic algorithm is presented to solve this type of optimization problem where the Grid service reliability is a critical component as the basis of the profit function. An analysis of different types of failures in Grid system and their influence on its reliability and performance is found in [37]. Models for star-topology Grid considering data dependence and tree-structure Grid considering failure correlation are presented. The universal generating function, graph theory, and the Bayesian approach are used for the development of evaluation tools and algorithms. In [38] Doguc and Ramirez-Marquez discuss an automated method for estimating service reliability in Grids without relying on any assumptions about the component and link failures. The proposed method is based on a popular data mining algorithm, K2, and finds the associations between the Grid components automatically.

Other work has addressed the closely related issue of failure rates for resource components [39] such as disk failures [40]. The approach towards risk assessment is aimed at a granularity level of individual components as compared to resource level.

7. Extensions

There are many ways to further extend the work presented in this paper.

The risk assessment model presented in this work only considered the resources historical data. An extension to this model is to consider dynamic data, such as the current resource load or the availability of administrators to enhance the model, since the mean time to repair a resource is hugely influenced by the availability of administrators.

A number of studies in section 6 have looked at resource failures in distributed environments and concluded and there is clear evidence that metrics such as the time between failures is best fitted by a Weibull distribution. There is scope to re-develop the risk of failure model to automatically

select the *best* distribution for the time varying functions, e.g. Gamma, Lognormal.

Another extension to the risk model is to consider the internal components of a resource rather than considering a resource as a black-box. This extension model has different component failures, such as CPU, memory, hard drive, etc, and drives the resource risk of failure through campaigning all the component models.

The risk assessment model did not consider the type and intensity of the workload running on a resource. However, there is evidence of a correlation between the type and intensity of the workload and the failure rate of the resource [28]. More importantly, extending the model to cater for this information will provide a more accurate risk estimation.

The data used to develop the model has been provided by a research institution. The resources mean time to repair is quite due to 1) the lack of 24-hour support service, and 2) the absence of an automatic monitoring service which to report resource failures when they occur. It would be ideal to use data from commercial Grid providers, if available, to further validate the risk assessment model.

The risk assessment model was developed and evaluated analytically. Therefore, it would be beneficial to implement the model on a production Grid in order to evaluate its overall performance.

This research has considered resource failures in Grid computing and can equally be applied in cloud computing. The proposed risk assessment model, its implementation, testing, and evaluation in a cloud environment are considered in the OPTIMIS project [27, 26].

8. Conclusions

This paper has presented the steps towards the development of a mathematical model to predict Grid resources risk of failure.

The motivation scenario for the Grid resources risk of failure model is first presented. The events causing resource failures are identified, and the method for measuring the risk of these events is presented. The need for historical failure data is showcased, along with the data collection process.

The mathematical model was developed after a detailed analysis of Grid resource failures using failure data collected from different Grid resources and spanning for three years. The analysis focused on the statistical properties of the failure data, including the root cause of failures, the mean time between failures, and the mean time to repair. The best model for the time between failures is the Weibull distribution, with decreasing hazard function rate. Repair times are much better modelled by a lognormal distribution than an exponential distribution.

The probability of resource failures plays a central role in the risk assessment process. The reviewed models found in the literature to compute this probability have clear limitations: the unrealistic assumption that the resource failures represent a Poisson process, the subjective prior distribution selection in the Bayesian model or ignoring resource unavailability due to scheduled maintenance. Therefore, this paper has shown that the resource failures do not represent a Poisson process, fit distributions to observed resource failures data, and use a Markov model to represent all the resource states. A continuous time-varying Markov Model described the Grid resource availability. In order to solve the Markov model, there is the need to approximate the continuous-time process with discrete-time equivalents. The resulting discrete time-varying Markov Model is used to estimate the resources risk of failure.

Such model can be integrated in the resource provider risk assessment and is a viable contender to enable the provider to identify infrastructure bottlenecks and mitigate potential risk, in some cases by identifying fault-tolerance mechanisms to prevent SLA violations.

This research can be extended in many ways: 1) consideration of dynamic data, such as the current resource load or the availability of administrators to enhance the model; 2) risk assessment at the level of the resource's components (CPU, memory, hard drive, network interface card etc); 3) consideration of the type and intensity of the workload running on a resource; 4) consideration of data from commercial Grid/Cloud providers, and 5) the implementation of the risk model on a production Grid/Cloud in order to evaluate its performance.

References

[1] F. Berman, G. Cox, A. Hey (Eds.), Grid Computing: Making the Global Infrastructure a Reality, Wiley, 2003.

[2] K. Djemame, I. Gourlay, J. Padgett, G. Birkenheuer, M. Hovestadt, O. Kao, K. Vo, Introducing Risk Management into the Grid, in: Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing (eScience2006), IEEE Computer Society, Amsterdam, Netherlands, 2006.

[3] The risk management standard, Institute of Risk Management. The Association of Insurance and Risk Managers,National Forum for Risk Management in the Public Sector, http://www.theirm.org/publications/PUstandard.html (2009).

[4] W. Z. P. Wieder, R. Yahyapour (Ed.), Grids and Service-Oriented Architectures for Service Level Agreements, Springer, 2010.

[5] D. Battré, O. Kao, K. Vo, Implementing ws-agreement in a globus toolkit 4.0 environment, in: Usage of Service Level Agreements in Grids Workshop in conjunction with The 8th IEEE International Conference on Grid Computing (GRID2007), Austin, Texas, 2007.

[6] O. Waeldrich, W. Ziegler, WS-Agreement Framework for Java (WSAG4J), http://packcs-e0.scai.fraunhofer.de/mss-project/index.html (2010).

[7] K. Djemame, J. Padgett, I. Gourlay, D. Armstrong, Brokering of risk-aware service level agreements in grids, Concurrency and Computation: Practice and Experience 23 (7).

[8] T. Heath, R. P. Martin, T. D. Nguyen, Improving cluster availability using workstation validation, in: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, SIGMETRICS '02, ACM, New York, NY, USA, 2002, pp. 217–227.

[9] Grid operations centre database, https://goc.egi.eu (2010).

[10] European grid infrastructure - towards a sustainable grid infrastructure, http://www.egi.eu (2010).

[11] National grid service - connecting infrastructure, connecting research, http://www.ngs.ac.uk (2010).

[12] Worldwide lhc computing grid, http://lcg.web.cern.ch/LCG/ (2010).

[13] Cern - the european organization for nuclear research, http://www.cern.ch (2010).

[14] M. Rausand, A. Høyland, System Reliability Theory: Models, Statistical Methods, and Applications, 2nd Edition, Wiley, 2008.

[15] R. Blischke, D. Murthy, Reliability: Modeling, Prediction, and Optimization, Wiley Series in Probability and Statistics, John Wiley and Sons, 2000.

[16] D. Murthy, M. Xie, R. Jiang, Weibull Models, Wiley-Interscience, John Wiley and Sons, 2004.

[17] M. Modarres, Risk Analysis In Engineering: Techniques, Tools, and Trends, Taylor & Francis, Boca Raton, 2006.

[18] R. Alsoghayer, Risk assessment models for resource failure in grid computing, Ph.D. thesis, School of Computing, University of Leeds, UK (2011).

[19] J. Pukite, P. Pukite, Modeling for Reliability Analysis: Markov Modeling for Reliability, Maintainability, Safety, and Supportability Analyses of Complex Systems, Wiley-IEEE Press, 1998.

[20] D. Siewiorek, R. Swarz, Reliable Computer Systems: Design and Evaluation, Digital Press, 1998, 3rd edition.

[21] A. Papoulis, S. U. Pillai, Probability, random variables, and stochastic processes, 4th Edition, McGraw Hill, 2002.

[22] Consequence: the context-aware data-centric information sharing, http://www.consequence-project.eu (2011).

[23] Sla@soi: Slas empowering a dependable service economy, http://sla-at-soi.eu (2011).

[24] C. Carlsson, Risk Assessment for Grid Computing with Predictive Probabilities and Possibilistic Models, in: Proceedings of the 5th International Workshop on Preferences and Decisions, Trento, Italy, 2009.

[25] D. Battré, G. Birkenheuer, M. Hovestadt, O. Kao, K. Voss, Applying Risk Management to Support SLA Provisioning, in: The 8th Cracow Grid Workshop, Academic Computer Center CYFRONET AGH, 2008.

[26] A. J. Ferrer, F. Hernandez, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S. K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgo, T. Sharif, C. Sheridan, Optimis: A holistic approach to cloud service provisioning, Future Generation Computer Systems 28 (1) (2012) 66 – 77.

[27] K. Djemame, D. Armstrong, M. Kiran, M. Jiang, A risk assessment framework and software toolkit for cloud service ecosystems, in: Proceedings of the Second International Conference on Cloud Computing, GRIDs, and Virtualization, Rome, Italy, 2011.

[28] B. Schroeder, G. A. Gibson, A large-scale study of failures in high-performance computing systems, in: Proceedings of the International Conference on Dependable Systems and Networks, DSN '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 249–258.

[29] D. Nurmi, J. Brevik, R. Wolski, Modeling machine availability in enterprise and wide-area distributed computing environments, in: Proceedings of Euro-Par2005, Springer, 2005, pp. 432–441.

[30] F. Nadeem, R. Prodan, T. Fahringer, Characterizing, modeling and predicting dynamic resource availability in a large scale multi-purpose grid, in: Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid, CCGRID '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 348–357.

[31] T. J. Hacker, F. Romero, C. D. Carothers, An analysis of clustered failures on large supercomputing systems, Journal of Parallel and Distributed Computing 69 (7) (2009) 652–665.

[32] A. Platis, G. Gravvanis, K. Giannoutakis, E. Lipitakis, A two-phase cyclic nonhomogeneous markov chain performability evaluation by explicit approximate inverses applied to a replicated database system, Journal of Mathematical Modelling and Algorithms 2 (2003) 235–249.

[33] V. P. Koutras, A. N. Platis, G. A. Gravvanis, Software rejuvenation for resource optimization based on explicit approximate inverse preconditioning, Applied Mathematics and Computation 189 (1) (2007) 163 – 177.

[34] V. P. Koutras, A. N. Platis, G. A. Gravvanis, On the optimization of free resources using non-homogeneous markov chain software rejuvenation model, Reliability Engineering amp; System Safety 92 (12) (2007) 1724 – 1732.

[35] A. Iosup, M. Jan, O. Sonmez, D. Epema, On the dynamic resource availability in grids, in: Grid Computing, 2007 8th IEEE/ACM International Conference on, 2007, pp. 26 –33.

[36] Y.-S. Dai, G. Levitin, X. Wang, Optimal task partition and distribution in grid service system with common cause failures, Future Generation Computer Systems 23 (2) (2007) 209 – 218.

[37] Y. Dai, G. Levitin, Performability Modeling and Analysis of Grid Computing, Handbook of Performability Engineering, K.B. Misra (Ed.), Springer-Verlag, 2008, Ch. 65.

[38] O. Doguc, J. E. Ramirez-Marquez, An automated method for estimating reliability of grid systems using bayesian networks, Reliability Engineering System Safety 104 (0) (2012) 96 – 105.

[39] A. Sangrasi, K. Djemame, Component level risk assessment in grids: A probablistic risk model and experimentation, in: Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on, 2011, pp. 68 –75.

[40] E. Pinheiro, W.-D. Weber, L. A. Barroso, Failure trends in a large disk drive population, in: Proceedings of the 5th USENIX conference on File and Storage Technologies, USENIX Association, Berkeley, CA, USA, 2007, pp. 2–2.