# Probabilistic Risk Assessment for Resource Provision in Grid

Raid Alsoghayer*, Karim Djemame*

**Abstract**

Service Level Agreements (SLAs) are introduced to overcome the shortages of best-effort approach in Grid computing and make Grid computing more attractive for commercial uses. Yet commercial Grid providers are not keen to adopt SLAs, since there is a risk of SLA violation, which will result in a penalty fee. This paper analyses failure data collected from three different Grid sites. We study the statistics of the data including the root cause, the mean time to repair and time between failures. We find that software and hardware failures are the largest contributors, and that the time to repair varies, depending on the root cause, from 13 hours in network errors to around 46 hours in unknown errors. We also find that the repair time is well modelled by a Weibull distribution. From the analysis of the historical data we find that the distribution between failures in a Grid system is well modelled by a Weibull distribution with decreasing hazard rate, and this can be used by a resource provider to predicate the risk of failure.

# 1. Introduction

Grid computing is the coordinated sharing of resources and solving problems in dynamic, multi-institutional virtual organizations. This sharing must be controlled with clear boundaries on what will be shared, who are allowed to share, and the conditions under which sharing occurs, whether the resources are hardware, software, or users [1, 2]. The sharing should be done using standard, open, and general-purpose protocols and interfaces, and should deliver nontrivial quality of services (QoS) [3, 4].

The sharing and coordination of resources on the Grid is complicated, since both the user and the Grid resource provider are geographically distributed in different time zones, and have competing needs. The user application needs to understand the resource status and affect it. It often needs assurance as to the type and level of services provided by the resources. On the other hand, the resource provider wants to control usage policy, and often restricts the service information exposed to users. A solution exists for these competing needs, in which a contract (Service Level Agreement, SLA) between the Grid resource provider and the Grid user is negotiated. The SLA either provides some measurable capability or performs a specific task. An SLA allows the user to know what is expected from a service without requiring detailed knowledge of the provider's policies [5, 6].

Even with the introduction of SLAs commercial Grids are not attracting users and providers. Current Grid middleware (e.g. Globus Toolkit [7]) still follows the best-effort approach, there is a risk that users do not get any guarantees that his/her SLA will not be violated. Also commercial resource providers are not attracted either. For a

* School of Computing, University of Leeds, {raid, karim}@comp.leeds.ac.uk

resource provider agreeing on an SLA without enough information about the state of resources and the availability of devices introduces a chance of violating the SLA, which results in a penalty fee. There is a risk attached to system failure, service unavailability, insufficient resources etc, which might lead to SLA violation. Without a method for assessing the risk of accepting an SLA providers are only able to make uncertain decision regarding suitable SLA offers. Also end-users would like to know the risk of violating an SLA so that they can make decisions on what Grid resource provider to select and the acceptable cost/penalty fee.

The word risk is used in variety of discipline/context and has a different meaning in each different discipline/context. Even in a single corporation, different departments have different definitions for risk. In the Oxford English Dictionary Risk is defined as *"Hazard, danger, exposure to mischance or peril"* [8]. As different entities of a single corporation have different definitions for risk, they also have different views too. For example, Health and Safety department personnel view risk as a bad thing or a negative force. Thus, any risk to the health and safety of company employees or the public is to be avoided, or the probability and consequence of that risk are to be reduced as much as possible. On the other hand, people working in the finance sector might have different views, because part of their job is making risk/reward evaluation. Generally greater risk yields greater returns, therefore they view risk as a positive force [9].

This study analyses failure data collected from three different Grid sites, and presents statistics in relation to data including the root cause, the mean time to repair, and time between failures. It gives a description of the statistical properties of the data and use the historical data to assess risk using a frequentist approach. This paper is organized as follows. A motivation scenario is presented in Section 2. Section 3 describes the data collection process and the structure of the data records. The methodology used to analysis the data is presented in Section 4. In Section 5 we analyse data with respect to three important proprieties of system failures: root cause, time to repair and time between failures. Section 6 presents some related work. Then we conclude in section 7.

# 2. Motivation Scenario

Most resource providers use Network batch queuing systems as Resource Managers (RM). These systems handle jobs by allocating resources from a networked pool of computers. Some examples of these systems are Sun Grid Engine (SGE) [10], Load Sharing Facility (LSF) [11], Portable Batch System (PBS) [12], and LoadLeveler [13]. Computing Risk depends significantly on the provider infrastructure, since the risk of failure depends especially on the resource's stability which varies between times. Therefore Risk Assessment is difficult to support in queuing based systems. Thus making resource reservations is the initial step for integrating risk assessment into the Grid [14]. Some RM have the capacity to perform advance reservation, e.g. PBS-Pro [12]. Another important requirement for risk assessment is the availability of historical failure data. The data is composed of scheduled system failures (e.g. system upgrades, backups, etc) and unscheduled system failure (e.g. power cut, hardware failure, etc). Scheduled failures, unlike unscheduled failure, are known in advance and stored in the dataset as references. Using advance reservations overcome the affect of scheduled system failures. A resource provider cannot reserve resources at times with scheduled failures. Unscheduled failure can not be ignored and the resource provider should have a way to compute the possibility of such events occurring.

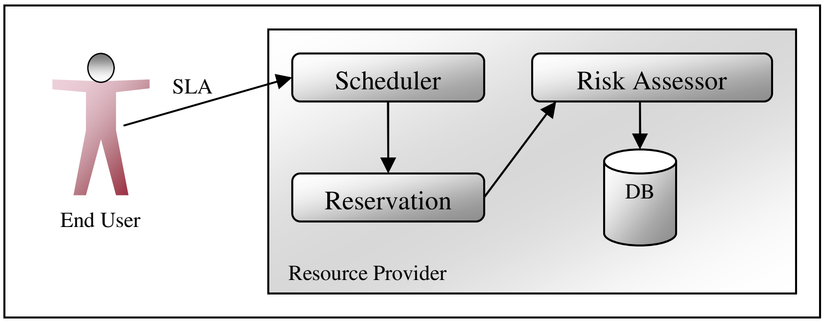



**Figure 1:** Example Scenario.

Figure 1 presents an example scenario. The end user submits an SLA request to the resource provider. The SLA includes the user requirements (e.g. number of CPU, memory size, deadline, etc). When the resource provider receives the SLA it contacts the scheduler to schedule the user job. The scheduler requests the resource reservation component to reserve the end user required resources within the deadline requested. If resources are not available the SLA is rejected, otherwise the time t in witch the reservation starts is sent to the risk assessor. The risk assessor role is to compute the probability of system failure (Risk) at time t by looking at the data stored in the historical database. The failure rate will help the provider to decide whether to accept or reject the SLA. It will significantly influence charge and penalty fee. Also it will inform the end user of the rate of not getting the desired QoS and give the possibility to select the desired level of realistic guarantees.

## 3. Data Collection

Research in the area of dependable computing depends on understanding how failures in the real world look like, e.g. knowledge of failure characteristics can be used in resource management to improve cluster availability [15]. Also creating realistic benchmarks and test-beds for reliability testing requires the knowledge of failure characteristics [16]. Therefore access to failure data is very important.

The Grid Operations Centre Data Base (GOCDB)[17] is a database of all sites within the Enabling Grids for E-science (EGEE)[18], the National Grid Service (NGS)[19] and Worldwide LHC Computing Grid (WLCG)[20] that contains information on sites, nodes, services, and downtime. GOCDB is publicly available and accessed following registration.

A downtime is a period of time for which a grid resource is declared to be inoperable. A downtime record contains unique downtime ID, downtime classification (scheduled or unscheduled), the severity of the downtime, the user who recorded the downtime, the date at which the downtime was added to GOCDB, 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, usually triggered by an unexpected failure. EGEE define specific rules[21] about what should be classified as scheduled downtime and what should be classified as unscheduled downtime. The rules are based on the length of the intervention, the impact severity, and how long in advance the downtime is declared. Yet currently it is up to the person who declares the downtime to decide if it is scheduled or not.

The severity of the downtime is either “At Risk” (Resource will probably be working as normal, but may experience problems) or “Outage” (Resource will be completely unavailable).

The Grid administrator who has permission to make downtime updates can add, edit, or delete downtime information. This is done manually and there are no rules or protocols to make such updates, thus it might be possible that the system had a failure and there is no record on the GOCDB for that failure.

The description of the downtime is left to the Grid administrator. It is a short description of the cause the downtime. There are no rules or protocols to follow when writing the description, thus descriptions are mostly incomplete and are very short. Some may have only one very brief word description (e.g. Test).

The data collected in GOCDB is different compared to the data in error-logs. Error-logs are generated automatically and treat every unexpected event the same whether it resulted in a system failure or not. Also error-logs might contain multiple entries for the same event. On the other hand data in the GOCDB are created manually by system administrators. Human created failure data have two potential problems underreporting of failure events and misdiagnosing the cause of the downtime. While it is possible for a failure to be not reported at all, in this study we are assuming that this is not the case. Misdiagnosing the cause of the downtime is feasible. GOCDB dose not have classification of the root cause (e.g. Hardware, Software, etc) it has only a description of what might cause the downtime. The diagnosis and description depend hugely on the administrators’ skills.

In this study we take into account the downtime data for three Grid Sites A, B and C from GOCDB. The downtime data are for the whole site and show only the time when the site was down. The data for site A span form July 2007 till mid November 2008. Site B data span from mid October 2007 till mid October 2008. Site C data span from mid June 2007 till mid November 2008. All data have scheduled and unscheduled downtime. The reason that each site has different time span compared to the other two is because sites’ administrators start recording failures at different times. Here we only consider unscheduled failures. The reason is that a resource provider uses advance reservation, which takes into account scheduled downtimes.

# 4. Methodology

Risk theory has been developed in the framework of probability theory. Probability theory models only those events which are produced with a large frequency. When this condition is not fulfilled, the possibility theory initiated by Zadeh in [22] can be used to treat risk. Statistical inference is a well known problem within probability theory. There is no agreement on the ways in which statistical inference should be carried out, and no agreement on the principles that should be used to judge the quality of estimation techniques [23].

In this study we use a probabilistic risk assessment method, since sufficient failure data exist. We use a frequentist approach to analyze the failure data and the Maximum Likelihood Method will be used to estimate the parameters of the distribution. We will evaluate the goodness of fit by visual inspection and the negative log-likelihood test. We view the sequence of failure events as a stochastic process and study the distribution of its time between failures. We take into account the failures that affect the whole system. We characterise repair times using the mean, median and standard deviation. We also consider the empirical cumulative distribution function (CDF) of repair time and how well it fits four probability distributions commonly used in reliability theory: the exponential, the Weibull, the gamma and the lognormal distributions. These distributions fit the data well and there are no reasons for using more degree of freedom e.g. a phase-type distribution. We use the maximum likelihood estimation to parameterize the distributions and evaluate the goodness of fit by visual inspection and the negative log-likelihood test.

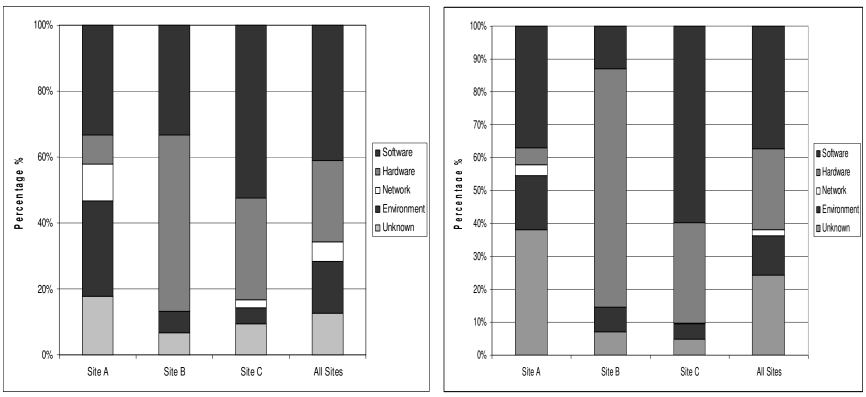
# 5. Data Analysis

In this section we will analyse data with respect to three important properties of system failures: root cause, time to repair, and time between failures.

## 5.1 Root Cause Breakdown

The first question to answer when studying failures in computer systems is what caused them? In GOCDB data there is a description of the cause of failure yet there is no classification of these causes. We therefore had to map the description of the failure to five different categories, Environment (Air Condition failure, power outage), Network, Software, Hardware and Unknown. Figure 2(a) shows the percentage of failure in each category. The right-most bar shows the breakdown of all the failure recorded in the three sites, while the first three bars are for Grids sites A, B, and C respectively.

We can see that software and hardware failures are the largest contributors to failures with the actual percentage for software ranges from 33.33% to 51.16%. The actual percentage for hardware ranges from 8.89% to 53.33%. Over all, the two categories are responsible for 65.69% of all the failures recorded in the database.




(a) (b)

**Figure 2:** Breakdown of failure (a), and downtime (b) into root cause.

We studied the total downtime for each category. Figure 2(b) shows the percentage of downtime for each category. The right-most bar shows the breakdown of all the downtime recorded in the three sites, while the first three bars are for Grids sites A, B, and C respectively.

We can see that software and hardware failures are contributing hugely to the downtime, but we can't ignore the unknown failures contribution, since unknown errors contribute 24.23% of the total downtime of all failures recorded. Software failures contribute 37.19% of the total downtime and hardware failures contribute 24.73%.

### 5.2 Repair Time Analysis

The second important metric in studying failures is the time to repair the system. We start by looking at how the repair time varies between the three sites. Then we study the statistical proprieties of repair time, including their distribution. Finally we look at how the root cause affect the repair time.

Table 1 shows the mean, median and standard Deviation for time to repair in each site. The mean time to repair in all three sites is very high. The first reason is that the repair time depends hugely on the availability of the Grid administrator, and the three sites do not have 24 hour support. Thus any failure occurring after normal working hours is not resolved until the next working day. This is also true for weekends and public holidays. The second reason is that there are no automatic monitoring that will report a failure when it occurs. Finally the Grid sites are mainly used for research, not commercial use. In order to improve the mean repair time the sites should increase the availability of administrators and deploy automatic monitoring agents.

| | *Site A* | *Site B* | *Site C* |
|---|---|---|---|
| Mean | 2119 | 2043 | 821 |
| Median | 821 | 1770 | 310 |
| Standard Deviation | 3227 | 2001 | 1174 |

**Table 1:** Mean median and standard Deviation for each site (in minutes).

Figure 3 shows the empirical Cumulative Distribution Function (CDF) for all repair times recorded, and four standard distributions fitted to it. Exponential distribution is a poor fit because of the high variability in the repair times. The other distributions have good visual fit with Weibull having the best fit when measured by the negative log-likelihood.

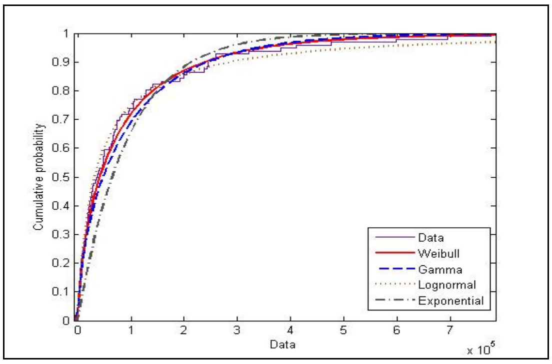



**Figure 3:** CDF for all repair times recorded.

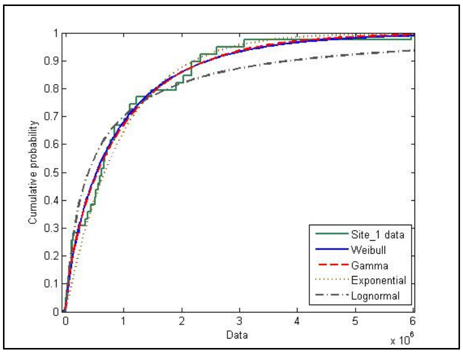
Now we look at how the root cause of failure affects the repair time. Table 2 shows the mean, median and standard Deviation of time to repair as a function of root cause. The mean repair time ranges from around 13 hours in network errors to around 46 hours in unknown errors. The mean repair time for all the errors is around 26 hours and the median is around 10 hours. This is close to the mean and median repair time of hardware and software errors. The second observation is that the time to repair is highly variable. For example, the median of hardware repair times is about 4 times lower than the mean.

| | *Software* | *Hardware* | *Network* | *Environment* | *Unknown* | *All* |
|---|---|---|---|---|---|---|
| Mean | 1422 | 1603 | 785 | 1183 | 2777 | 1553 |
| Median | 610 | 360 | 334 | 450 | 1367 | 599 |
| Standard Deviation | 2235 | 2093 | 498 | 1687 | 4075 | 2418 |

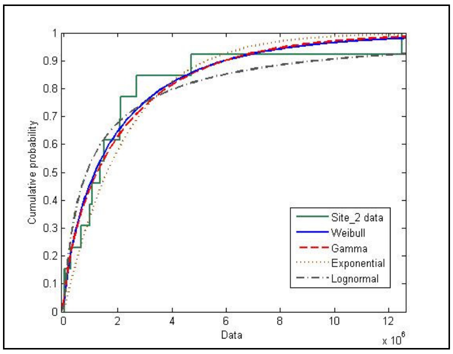
**Table 2:** Mean median and standard Deviation of time to repair (in minutes).

### 5.3 Time Between Failures Analysis

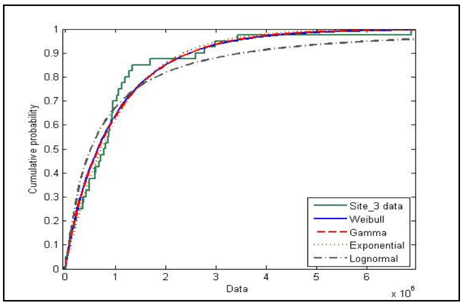
In this section we view the sequence of failure events as a stochastic process and we study the time between unscheduled failures, inter-arrival times, as seen by the whole system. Figure 4(a) shows the empirical CDF at the first site fitted by four standard distributions. We see that the distribution between failures is well modelled by a Weibull or Gamma distribution. Both distributions create an equally good visual fit and the same negative log-likelihood.




(a)




(b)




**(c)**

**Figure 4:** Empirical CDF for inter-arrival times of failures at sites 1(a), 2 (b) and 3(c).

Figure 4(b) shows the empirical distribution at the second site fitted by four standard distributions. We also see that the distribution between failures is well modelled by a Weibull or Gamma distribution. Both distributions create an equally good visual fit and the same negative log-likelihood.

Figure 4(c) shows the empirical distribution at the third site fitted by four standard distributions. We also see that the distribution between failures is well modelled by a Weibull or Gamma distribution. Both distributions create an equally good visual fit and the same negative log-likelihood.

From the above we can say that the Weibull and Gamma distributions are the best distributions to model distribution between failures in Grid system. We select the Weibull distribution since it is the most popular and widely used method of analyzing and predicting failures and malfunctions of all types, offers flexibility in modelling failure rates, and is easy to calculate [23-26].

The Weibull distribution mathematically characterizes the probability of system failures as a function of time. The two parameters Weibull function is used in this research and the probability density function *pdf* is defined as:

$$f(t)= \beta/\eta\ (t/\eta)^{\beta-1}\ e^{-(t/\eta)^{\beta}}$$

The cumulative density function *cdf* is defined as:

$$F(t)=1 - e^{-(t/\eta)^{\beta}}$$

Where $\beta$ is the shape parameter, $\eta$ is the scale parameter, and t is time. Recalling that the reliability function of a distribution is simply one minus the *cdf*, the reliability function for the Weibull distribution is given by:

$$R(t)= 1- cdf$$

From the above we can calculate the Weibull failure rate (or hazard rate) function:

$$\lambda(t)= f(t)/R(t) = \beta t^{\beta-1}\ \eta^{-\beta} \qquad (1)$$

The shape parameter $\beta$ directly influences the hazard function as follows:

If $\beta < 1$, the hazard function is decreasing with time;

If $\beta = 1$, the hazard function is constant with time, i.e., the exponential distribution;

If $\beta > 1$, the hazard function is increasing with time.

It is useful to know how the time since the last failure influences the expected time until the next failure. This notion is captured by a distribution's hazard rate function. An increasing hazard rate function predicts that if the time since a failure is long then the next failure is coming soon. And a decreasing hazard rate function predicts the reverse. The 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.

In this research we use the maximum likelihood estimation to estimate the parameters and we find decreasing hazard rates a Weibull shape parameter of 0.7–0.8. This means not seeing a failure for long time decreases the risk of seeing one soon.

From the above analysis we found that distribution between failures in Grid system is well modelled by a Weibull distribution with decreasing hazard rate. The scenario now is: When the resource provider receives an SLA it contacts the scheduler to schedule the user job. The scheduler requests the resource reservation to reserve the end user required resources within the deadline requested. If resources are not available the SLA is rejected, otherwise the time *t'* in witch the reservation starts is sent to the risk assessor. The risk assessor estimates the Weibull Parameters using the historical data and computes the risk of failure from (1). *t* in (1) indicates the time from the last failure till the time of resource reservation *t'*.

# 6. Related Work

There are a number of projects such as Akogrimo [27], GRIA [28], TrustCom [29], GridTrust [30], NextGrid [31], GridEcon [32], and AssessGrid [33, 34] which have considered specific aspects of SLAs, and therefore have developed basic Grid components and architectural support for building economic-aware components. However all the above projects, except AssessGrid, do not address the key issue of risk by developing a framework to support risk assessment for resource providers.

The main objective of the AssessGrid project [33, 34] is to address obstacles of a wide adoption of Grid computing by bringing risk management and assessment to this field, enabling use of Grid technologies in business and society. In this scope, AssessGrid delivers generic, customisable, trustworthy, and interoperable open-source software for risk assessment, risk management, and decision-support in Grids. The approach used to develop the risk assessment model in AssessGrid is different than the approach used in this research. The approach used in AssessGrid is based on the Possibility theory initiated by Zadeh in [22]. It assumed that Grid failure data are hardly available and are not frequent; therefore probability theory models can not be used. Possibility theory is based on new concepts such as possibility measure, necessity measure, possibilistic distributions, etc.

A large number of studies that look at systems failure is found in the literature and includes [15, 16, 35-37]. The study in [37] looks at the availability of CPUs in a Grid environment and analyzes availability traces recorded from all the clusters. The finding is that the best fit distribution is Weibull with a shape parameter > 1. The reason is that many of today's Grids comprise computing resources grouped in clusters, whose owners may share them only for limited periods of time. Often, many of a grid's resources are removed by their owner from the system, either individually or as complete clusters, to serve other tasks and projects. Thus the unavailability of CPUs is not due to a system failure but CPUs made unavailable by their owner. Other studies [15, 16, 35, 36] analyze the time between failures and find the Weibull distribution to be a good fit with decreasing hazard rates (Weibull shape parameter < 1). Recall that in this study we find decreasing hazard rates with Weibull shape parameter of 0.7–0.8.

# 7. Conclusion

In this paper we study failure data collected from three different grid sites spanning for more than a year. We find that software and hardware failures are responsible for 65.69% of all the failures recorded in the database. We also find that 37.19% of downtime is due to software failures, 24.73% to hardware failures, and 24.23% to unknown failures. The mean time to repair in all the three sites is very high, and repair time is will fitted by a Weibull distribution. Mean repair times vary widely depending on the root cause and is extremely variable. We use a probabilistic risk assessment method to study time between failures and find that Weibull distribution with decreasing hazard rate is the best fit.

Schroeder and Gibson [16] study The data, collected from 1996 till 2006, at Los Alamos National Laboratory that includes 23000 failures recorded on more than 20 different systems, mostly large clusters of SMP and NUMA nodes. They find that the time between failure at individual nodes, as well as at an entire system, is fit well by a gamma or Weibull distribution with decreasing hazard rate (Weibull shape parameter of 0.7–0.8). This is the same finding as our study.

The prior risk analysis was focused on determining the probability of failure at a given time *t*. The formulations provided us with the probability of failure of the entire system, up to a point in time, without looking at what happens if the system fails during that time and is then fixed. To address this, in the future work, we need to look at the Grid as a repairable system. Repairable systems receive maintenance actions when they fail. These actions change the overall makeup of the system and must be taken into consideration when assessing the Risk of Failure of the system because the age of the system components is no longer identical and the time of operation of the system is not continuous. Additional information and models are now needed. The previous model only described how the system failed (its failure probability distribution). When dealing with repairable system we need to know also how long it takes the system to be restored, at least a model that describes how the component is restored.

# References


1. Foster, I. and C. Kesselman, *The Grid in a Nutshell*, in *Grid resource management : state of the art and future trends*, J. Nabrzyski, J.M. Schopf, and J. Weglarz, Editors. 2004, Kluwer Academic Publishers: Boston. p. 3-13.
2. Foster, I., C. Kesselman, and S. Tuecke, *The anatomy of the Grid*, in *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, G. Fox, and A.J.G. Hey, Editors. 2003, J. Wiley: New York. p. 169-197.
3. Roure, D.D., et al., *The evolution of the Grid*, in *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, G. Fox, and A.J.G. Hey, Editors. 2003, J. Wiley: New York. p. 65-100.
4. Foster, I. and C. Kesselman, *Concepts and Architecture*, in *The grid: blueprint for a new computing infrastructure*, I. Foster and C. Kesselman, Editors. 2004, Morgan Kaufmann: Amsterdam; Boston. p. 37-64.
5. Czajkowski, K., et al., *Grid Service Level Agreements: Grid resource management with intermediaries* in *Grid resource management: state of the art and future trends*, J. Nabrzyski, J.M. Schopf, and J. Weglarz, Editors. 2004, Kluwer Academic Publishers: Boston. p. 119-134.
6. Czajkowski, K., et al., *SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems*. Lecture Notes in Computer Science. 2002. 153-183.
7. Foster, I. *Globus Toolkit Version 4: Software for Service-Oriented Systems*. in *Proceedings of the FIP International Conference on Network and Parallel Computing*. 2005: Springer-Verlag LNCS 3779.
8. *Oxford English Dictionary*. Available from: http://dictionary.oed.com.

9. Koller, G.R., *Risk assessment and decision making in business and industry: a practical guide*. 2nd ed. 2005, Boca Raton, FL: Chapman & Hall/CRC.
10. Sun Microsystems, I. *N1 Grid Engine 6 Administration Guide*. Available from: http://docs.sun.com/app/docs/doc/817-5677.
11. Lumb, I. and C. Smith, *Scheduling Attributes and Platform LSF*, in *Grid resource management: state of the art and future trends*, J. Nabrzyski, J.M. Schopf, and J. Weglarz, Editors. 2004, Kluwer Academic Publishers: Boston. p. 171-182.
12. Nitzberg, B., J.M. Schopf, and J.P. Jones, *PBS PRO: Grid Computing and Scheduling Attributes*, in *Grid resource management: state of the art and future trends*, J. Nabrzyski, J.M. Schopf, and J. Weglarz, Editors. 2004, Kluwer Academic Publishers: Boston. p. 183-190.
13. Kannan, S., et al., *Workload Management with LoadLeveler*. IBM RedBookes. 2001.
14. Matthias, H., K. Odej, and V. Kerstin. *The First Step of Introducing Risk Management for Prepossessing SLAs*. in *Proceedings of the IEEE International Conference on Services Computing, SCC '06*. 2006.
15. Taliver, H., P.M. Richard, and D.N. Thu, *Improving cluster availability using workstation validation.* SIGMETRICS Perform. Eval. Rev., 2002. **30**(1): p. 217-227.
16. Schroeder, B. and G.A. Gibson. *A large-scale study of failures in high-performance computing systems*. in *Proceedings of the International Conference on Dependable Systems and Network, DSN 2006*.
17. *Grid Operations Centre DataBase*. Available from: http://www.grid-support.ac.uk/content/view/406/293/.
18. *Enabling Grids for E-sciencE.* Available from: http://www.eu-egee.org/.
19. *National Grid Service.* Available from: http://www.grid-support.ac.uk/.
20. *Worldwide LHC Computing Grid.* Available from: http://lcg.web.cern.ch/LCG/.
21. *EGEE Production Infrastructure: Intervention Procedures*. Available from: https://edms.cern.ch/document/829986.
22. Zadeh, L.A., *Fuzzy sets as a basis for a theory of possibility.* Fuzzy Sets and Systems, 1999. **100**(Supplement 1): p. 9-34.
23. Bedford, T. and R. Cooke, *Probabilistic risk analysis*. 2001, Cambridge: Cambridge University Press.
24. Abernethy, R.B., *The new Weibull handbook: reliability & statistical analysis for predicting life, safety, risk, support costs, failures, and forecasting warranty claims, substantiation and accelerated testing, using Weibull, Log normal, Crow-AMSAA, Probit, and Kaplan-Meier models*. 5th ed. 2006, North Palm Beach, Fla.: R.B. Abernethy.
25. Murthy, D.N.P., M. Xie, and R. Jiang, *Weibull models*. Wiley series in probability and statistics. 2004, Hoboken, N.J. [Great Britain]: Wiley Interscience.
26. Lawless, J.F., *Statistical models and methods for lifetime data*. 2nd ed. Wiley series in probability and statistics. 2003, Hoboken, N.J. [Great Britain]: Wiley-Interscience.
27. *Access to Knowledge through the Grid in a Mobile World (Akogrimo).* Available from: http://www.mobilegrids.org/.
28. *Grid Resources for Industrial Applications (GRIA).* Available from: http://www.gria.org/.
29. *TrustCoM Project*. Available from: http://www.eu-trustcom.com/.

30. *GridTrust*. Available from: http://www.gridtrust.eu.
31. *NextGRID: Architecture for Next Generation Grids*. Available from: http://www.nextgrid.org.
32. *Grid Economics and Business Models (GridEcon)*. Available from: http://www.gridecon.eu.
33. *The AssessGrid Project*. Available from: http://www.assessgrid.org/.
34. Karim, D., et al., *Introducing Risk Management into the Grid*, in *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*. 2006, IEEE Computer Society.
35. Lin, T.T.Y. and D.P. Siewiorek, *Error log analysis: statistical modeling and heuristic trend analysis.* Transactions on Reliability, IEEE, 1990. **39**(4): p. 419.
36. Jun, X., Z. Kalbarczyk, and R.K. Iyer. *Networked Windows NT system field failure data analysis*. in *Proceedings of the 1999 Pacific Rim International Symposium on Dependable Computing.* 1999.
37. Iosup, A., et al. *On the dynamic resource availability in grids*. in *Proceedings of the 8th IEEE/ACM International Conference on Grid Computing (GRID 2007),* 2007