

Probabilistic versus possibilistic risk assessment models for optimal service level agreements in grid computing

Christer Carlsson · Robert Fullér

Received: 2 October 2010 / Revised: 24 April 2011 / Accepted: 19 December 2011 /
Published online: 11 January 2012
© Springer-Verlag 2012

Abstract We present a probabilistic and a possibilistic model for assessing the risk of a service level agreement for a computing task in a cluster/grid environment. These models can also be applied to cloud computing. Using the predictive probabilistic approach we develop a framework for resource management in grid computing, and by introducing an upper limit for the number of failures we approximate the probability that a particular computing task is successful. In the predictive possibility model we estimate the possibility distribution of the future number of node failures by a fuzzy nonparametric regression technique. Then the resource provider can use the probabilistic or the possibilistic model to get alternative risk assessments.

Keywords Grid and cloud computing · Service level agreement (SLA) · Predictive probabilities · Predictive possibilities

1 Introduction

There is an increasing demand for computing power in scientific and engineering applications which has motivated the deployment of high performance computing (HPC) systems that deliver tera-scale performance. Current and future HPC systems

C. Carlsson (✉) · R. Fullér
IAMSR, Åbo Akademi University, Joukahainengatan 3-5A, Åbo, 20520 Turku, Finland
e-mail: christer.carlsson@abo.fi

R. Fullér
e-mail: robert.fuller@abo.fi

R. Fullér
Institute of Intelligent Engineering Systems, John von Neumann
Faculty of Informatics Óbuda University, Bécsi út 96/b, 1034 Budapest, Hungary
e-mail: fuller.robert@nik.uni-obuda.hu

that are capable of running large-scale parallel applications may span millions of nodes. For parallel programs, the failure probability of nodes and computing tasks assigned to the nodes has been shown to increase significantly with the increase in number of nodes. Large-scale computing environments, such as the grids CERN LCG, NorduGrid, TeraGrid and Grid'5000 gather (tens of) thousands of resources for the use of an ever-growing scientific community. Many of these Grids have the problems of any large-scale computing environment, which contributes to making Grids relatively unreliable computing platforms. Iosup et al. (2007) collected and present material from Grid'5000 which illustrates this contention. In a thorough study Schroeder and Gibson (2006) analyse failure data collected over 9 years at Los Alamos National Laboratory (LANL) which includes 23,000 failures recorded on more than 20 different systems, mostly large clusters of SMP and NUMA nodes. Their study includes root cause of failures, the mean time between failures, and the mean time to repair. Iosup et al. (2007) fit statistical distributions to the Grid'5000 data using maximum likelihood estimation (MLE) to find a best fit for each of the model parameters. They found that the best fits for the inter-arrival time between failures, the duration of a failure, and the number of nodes affected by a failure, are the Weibull, Log-Normal, and Gamma distributions, respectively. The results for inter-arrival time between consecutive failures indicate an increasing hazard rate function, i.e. the longer a computing node stays in the system, the higher the probability for the node to fail. For the LANL dataset Schroeder and Gibson (2006) studied the sequence of failure events and the time between failures as stochastic processes. This includes two different views of the failure process: (1) the view as seen by an individual node; (2) the view as seen by the whole system. They found that the distribution between failures for individual nodes is well modeled by a Weibull or a Gamma distribution; both distributions create an equally good visual fit and the same negative log-likelihood. For the system wide view of the failures the basic trend is similar to the per node view during the same time. The Weibull and Gamma distributions provide the best fit, while the lognormal and exponential fits are significantly worse. A significant amount of the literature on grid computing addresses the problem of resource allocation (Brandt et al. 2005; Czajkowski et al. 2005; Magana and Serrat 2007). The presence of disparate resources that are required to work in concert within a grid computing framework increases the complexity of the resource allocation problem. Jobs are assigned either through scavenging, where idle machines are identified and put to work, or through reservation in which jobs are matched and pre-assigned with the most appropriate systems in the grid for efficient workflow.

Cloud computing could be seen as a disruptive technology in relation to grid computing when it was introduced to the business world in 2008. There are several formal definitions for cloud computing but the one given by Marston et al. (2011) captures the essence:

It is an information technology service model where computing services (both hardware and software) are delivered on-demand to customers over a network in a self-service fashion independent of device and location. The resources required to provide the requisite quality-of-service levels are shared,

dynamically scalable, rapidly provisioned, virtualized and released with minimal service provider interaction. Users pay for the service as an operating expense without incurring any significant initial capital expenditure, with the cloud services employing a metering system that divides the computing resource in appropriate blocks.

Cloud computing uses the “self-service model” but as shown by Rimal et al. (2011) cloud computing also builds on mutual contracts, as the Service level agreements (SLAs) in grid computing. Not surprisingly, early experience with cloud computing has brought out the problems with *outages* that make cloud reliance questionable. Rimal et al. (2011) have collected some statistics showing that Microsoft Azure was out of operation for 22 h on one occasion; Google AppEngine for about 5 h on several occasions; FlexiScale was out for 18 h due to core network failure on one occasion, etc. Our risk assessment models for SLAs can be applied to cloud computing as well as to grid computing; in the following we will use grid/cloud computing unless a particular solution applies to grid computing only.

In grid/cloud computing a resource provider [RP] offers resources and services to other users based on agreed SLAs. The research problem we have addressed is formulated as follows:

- the RP is running a risk being in violation of his SLA if one or more of the resources [nodes in a cluster, grid or cloud] he is offering to prospective customers will fail when carrying out the tasks
- the RP needs to work out methods for a systematic risk assessment [RA] in order to judge if he should offer the SLA or not if he wants to work with some acceptable risk profile

In the context we are going to consider (a generic grid/cloud computing environment) resource providers are of various types which mean that the resources they manage and the risks they have to deal with are also different; we have dealt with the following RP scenarios (but we will report only on extracts due to space):

- RP1 manages a cluster of n_1 nodes (where $n_1 < 10$) and handles a few (<5) computing tasks;
- RP2 manages a cluster of n_2 nodes (where $n_2 < 150$) and handles numerous (≈ 100) computing tasks; RP2 typically uses risk models building on stochastic processes (Poisson–Gamma) and Bayes modeling to be able to assess the risks involved in offering SLAs;
- RP3 manages a cluster of n_3 nodes (where $n_3 < 10$) and handles numerous (≈ 100) computing tasks; if the computing tasks are of short duration and/or the requests are handled online then RP3 could use possibilistic models that will offer robust approximations for the risk assessments;
- RP4 manages a cluster of n_4 nodes (where $n_4 < 150$) and handles numerous (≈ 100) computing tasks; if the computing tasks are of short duration and/or the requests are handled online, then *hybrid models* which combine stochastic processes and Bayesian modeling with possibility models could be used for handling this type of cases.

During the execution of a computing task the fulfilment of the SLA has the highest priority, which is why an RP often is using resource allocation models to safeguard against expected node failures. When spare resources at the RP's own site are not available outsourcing will be an adequate solution for avoiding SLA violations.

The risk assessment modeling for an SLA violation builds on the development of predictive probabilities and possibilities for possible node failures in combination with the availability of spare resources. The rest of the paper will be structured as follows: in Sect. 2 we will work out the basic conceptual framework for risk assessment, in Sects. 3, 4 we will introduce the Bayesian predictive probabilities as they apply to the SLAs for RPs in grid/cloud computing, in Sects. 5, 6 we will work out the corresponding predictive possibilities, in Sect. 7 we show a lower limit for the probability of success of computing tasks in a grid/cloud and, finally, in Sect. 8 there is a summary and conclusions of this study.

2 Risk assessment

There is no universally accepted definition of business risk but in the RP context we will understand risk to be a potential problem which can be avoided or mitigated (Carlsson and Weissman 2009). The potential problem for an RP is that he has accepted an SLA and may not be able to deliver the necessary computing resources in order to fulfil a computing task within an accepted time frame T . *Risk assessment* is the process through which a resource provider tries to estimate the probability for the problem to occur within T and *risk management* the process through which a resource provider tries to avoid or mitigate the problem. In classical decision theory risk is connected with the probability of an undesired event; usually the probability of the event and its expected harm is outlined with a scenario which covers the set of risk, regret and reward probabilities in an expected value for the outcome. The typical statistical model has the following structure,

$$R(\theta, \delta(x)) = \int L(\theta, \delta(x))f(x|\theta)dx \quad (1)$$

where L is a loss function of some kind, x is an observable event (which may not have been observed) and $\delta(x)$ is an estimator for a parameter θ which has some influence on the occurrence of x . The risk is the expected value of the loss function. The statistical models are used frequently because of the very useful tools that have been developed to work with large datasets.

3 Predictive probabilities

In the following we will use node failures in a cluster of a grid/cloud as the focus, i.e. we will work out a model to predict the probabilities that n nodes will fail in a period covered by an SLA ($n = 0, 1, 2, 3, \dots$). In the interest of space we have to

do this by sketches as we deal with standard Bayesian theory and modeling (Carlsson and Weissman 2009).

The *first step* is to determine a probability distribution for the number of node failures for a time interval $(t_1, t_2]$ by starting from some basic property of the process we need to describe. Typically we assume that the node failures represent a *Poisson process* which is non-homogenous in time and has a rate function $\lambda(t)$, $t > 0$.

The *second step* is to determine a distribution for $\lambda(t)$ given a number of observations on node failures from r comparable segments in the interval $(t_1, t_2]$. This count normally follows a $\text{Gamma}(\alpha, \beta)$ distribution and the posterior distribution $p(\lambda_{t_1,t_2})$, given the count of node failures, is also a Gamma distribution according to the Bayes' theory. Then, as we have been able to determine λ_{t_1,t_2} we can calculate the predictive distribution for the number of node failures in the next time segment; Bayes' theory shows that this will be a Poisson–Gamma distribution.

The *third step* is to realize that a computing task can be carried out successfully on a cluster (or a grid) if all the needed nodes are available for the scheduled duration of the task. This has three components: (1) a predictive distribution on the number of nodes needed for a computing task covered by an SLA; (2) a distribution showing the number of nodes available when an assigned set of nodes is reduced by the predicted number of node failures and an available number of reserve nodes is added (the number of reserve nodes is determined by the resource allocation policy of the RP); (3) a probability distribution for the duration of the task.

The *fourth step* is to determine the probability of an SLA failure. Consider a grid/cloud of k clusters, each of which contains n_c nodes, leading to the total number of nodes $n = \sum n_c$, where $c = 1, \dots, k$. Let $\lambda(t)$, $t > 0$, denote generally a time non-homogeneous rate function for a Poisson process $N(t)$. We will assume that we have the RP4 scenario as our context, i.e. we will have to deal with hundreds of nodes and hundreds of computing tasks with widely varying computational requirements over the planning period.

To illustrate the predictive nature of Bayesian characterizations of uncertainty, we use a simple example model and contrast the strategy with the maximum likelihood method. Consider a data set comprising information from six exchangeable sampling units, each of which can raise a number X_i of events, such as errors in coding, software, transmission etc, for $i = 1, \dots, 6$. When $X_i = 0$, it is concluded that no errors occurred within the i th sampling unit. If it is anticipated that the number of realized errors is small in relation to the number of potentially possible errors, a Poisson distribution is one common characterization of the uncertainty related to the intensity at which the errors occur among comparable sampling units. Using this distribution as the likelihood component for each of the observations, we obtain the joint conditional distribution of the data as,

$$p(x_1, \dots, x_6 | \lambda) \propto \exp(-6\lambda) \lambda^{\sum_{i=1}^6 x_i}, \tag{2}$$

where λ is the unknown Poisson intensity parameter. In our example we have a single unknown quantity, the Poisson parameter λ , for which the uncertainty is quantified prior to the arrival of the observations x_1, \dots, x_6 . The standard prior used

for this purpose is the Gamma(α , β) family of distributions, which has the following density representation,

$$p(\lambda|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda).$$

The *posterior* distribution of the intensity parameter λ , i.e. the conditional distribution of λ given the data $\mathbf{x} = (x_1, \dots, x_6)$, is then also a Gamma distribution and its density function equals,

$$p(\lambda|\mathbf{x}) = \frac{(\beta + 6)^{\alpha+z}}{\Gamma(\alpha + z)} \lambda^{\alpha+z-1} \exp(-(\beta + 6)\lambda),$$

where $z = \sum_{i=1}^6 x_i$ is the value of the sufficient statistic for the data under the Poisson model. Under a squared error loss, the Bayesian estimate of λ equals the posterior mean, which can be written as $(\alpha + z)/(\beta + 6)$.

In many technological applications it is not of primary interest to estimate model parameters, but rather use a statistical model for handling uncertainty in a given situation. For instance, in our example it might be of interest to provide a probability statement regarding whether a future number of events X_{n+1} would exceed a certain threshold, deemed crucial for the acceptability of the error rate in the system. The Bayesian answer to this question is the predictive distribution of the anticipated future observations, given the knowledge accumulated so far. Formally, this distribution is defined as,

$$p(x_7|x_1, \dots, x_6) = \int_0^\infty p(x_7|\lambda)p(\lambda|x_1, \dots, x_6)d\lambda, \quad (3)$$

where $p(\lambda|x_1, \dots, x_6)$ is the posterior distribution,

$$\begin{aligned} p(\lambda|x_1, \dots, x_6) &\propto \frac{\exp(-6\lambda)\lambda^y \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda)}{\int_0^\infty \exp(-6\lambda)\lambda^y \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda)d\lambda} \\ &\propto \frac{\exp(-6\lambda)\lambda^y \lambda^{\alpha-1} \exp(-\beta\lambda)}{\int_0^\infty \exp(-6\lambda)\lambda^y \lambda^{\alpha-1} \exp(-\beta\lambda)d\lambda}, \end{aligned}$$

which is recognized as the Gamma($\alpha + z$, $\beta + 6$) distribution. By evaluating the integral (3) with respect to the above Gamma distribution, we obtain a special case of a Poisson–Gamma distribution, which is a mixture of Poisson distributions.

4 Predictive probabilities in grid/cloud computing management

Using the predictive probabilistic approach, our aim here is to develop a framework for resource management in grid/cloud computing. The works cited earlier illustrate the substantial level of variation that exists over different grids/clouds with respect to the degree of exploitation, maintenance policies and system reliability. Several

factors are of importance when the uncertainty about the expected number of resources available at a given time point is considered. In the current work we consider explicitly the effects of computing task characteristics in terms of execution time and the number of resources required, as well as failure rate and maintenance for individual system components. A basic predictive model with a modular structure is derived, such that several ramifications with respect to the distributional assumptions can be made when deemed necessary for any particular grid environment. Consider a grid/cloud of k clusters, each of which contains n_c operative units termed nodes, with the total number of nodes in the grid/cloud denoted by, $n = \sum_{c=1}^k n_c$. For simplicity of notation, we will below treat the clusters as exchangeable units concerning predictive statistical learning.

To introduce a stochastic model for tendencies of node failures, let $\lambda(t) > 0, t \geq 0$, denote generally a time-nonhomogeneous rate function for a Poisson process $N(t)$. The rate function specifies the expected number of failure events in any given time interval $(t_1, t_2]$ according to,

$$\lambda_{t_1, t_2} = \int_{t_1}^{t_2} \lambda(t) dt.$$

Under many circumstances it is feasible to simplify the Poisson model by assuming that the rate function is constant over time, or over certain time segments. In the Poisson process it is assumed that the waiting times between successive events are exponentially distributed with the rate parameter λ governing the expected waiting times. Consider a time interval $(t_1, t_2]$ with the corresponding rate parameter λ_{t_1, t_2} , such that x events have been observed during $(t_1, t_2]$. Then, the likelihood function for the rate parameter is given by,

$$p(x | \lambda_{t_1, t_2}) = \frac{e^{-\lambda_{t_1, t_2}} (\lambda_{t_1, t_2})^x}{x!}.$$

By collecting data (counts of events) x_1, \dots, x_r from r comparable time segments, we obtain the joint likelihood function,

$$p(x_1, \dots, x_r | \lambda_{t_1, t_2}) \propto e^{-r\lambda_{t_1, t_2}} (\lambda_{t_1, t_2})^{\sum_{i=1}^r x_i}.$$

In accordance with the predictive example considered in the previous section, using the Gamma prior distribution for λ_{t_1, t_2} we then obtain the predictive Poisson–Gamma probability of having y events in the future on a comparable time interval, which equals,

$$\frac{(\beta + r)^{\alpha+z} \Gamma(\alpha + z + y)}{\Gamma(\alpha + z) y! (\beta + r + 1)^{\alpha+z+y}},$$

where $z = \sum_{i=1}^r x_i$.

When a computing task is slated for execution in a grid/cloud environment, its successful completion will require a certain number of nodes to be available over a given period of time. To assess the uncertainty about the resource availability, we

need to model both the distribution of the number of nodes and the length of the task required by the tasks. The most adaptable choice of a distribution for the number of nodes required, say M , is the multinomial distribution, $p(M = m) = p_m$, $m = 1, \dots, u$, where u is an *a priori* specified upper limit for the number of nodes. Such a vector of probabilities will be denoted by \mathbf{p} in the sequel. Here u could, e.g. equal the total number of nodes in the system, however, such a choice would lead to an inefficient estimation of the probabilities, and therefore, the upper bound value should be carefully assessed using empirical evidence.

An advantage of the use of the multinomial distribution in this context is its ability to represent any types of multimodal distributions for M , in contrast to the standard parametric families, such as the Geometric, Negative Binomial and Poisson distributions. For instance, if there are two major classes of computing tasks or maintenance events, such that one class is associated with relatively small numbers of required nodes, and the other with relatively large numbers, the system behavior in this respect is well representable by a multinomial distribution. On the other hand, standard parametric families of distributions would not enable an appropriate representation, unless some form of a mixture distribution were utilized. Such a choice would complicate the inference about the underlying parameters due to the fact that the number of mixture components would be unknown *a priori*. In general, estimation of parameters and calculation of predictive probabilities under a mixture model requires the use of a Monte Carlo simulation technique, e.g. the EM- or Gibbs sampler algorithm (Robert and Casella 2005).

A disadvantage of the multinomial distribution is that it contains a large number of parameters when u is large. However, this difficulty is less severe when the Bayesian approach to the parameter estimation is adopted. Given observed data on the number of nodes required by computing tasks, the posterior distribution of the probabilities \mathbf{p} is available in an analytical form under a Dirichlet prior, and its density function can be written as,

$$p(\mathbf{p}|\mathbf{w}) = \frac{\Gamma(\sum_{m=1}^u \alpha_m + w_m)}{\prod_{m=1}^u \Gamma(\alpha_m + w_m)} \prod_{m=1}^u p_m^{z_m + w_m - 1},$$

where w_m corresponds to the number of observed tasks utilizing m nodes, α_m is the *a priori* relative weight of the m th component in the vector \mathbf{p} , and \mathbf{w} is the vector $(w_m)_{m=1}^u$. The corresponding predictive distribution of the number of nodes required by a generic computing task in the future equals the Multinomial–Dirichlet distribution, which is obtained by integrating out the uncertainty about the multinomial parameters with respect to the posterior distribution. The Multinomial–Dirichlet distribution is in our notation defined as,

$$p(M = m^* | \mathbf{w}) = \frac{\Gamma(\sum_{m=1}^u \alpha_m + w_m)}{\prod_{m=1}^u \Gamma(\alpha_m + w_m)} \cdot \frac{\prod_{m=1}^u \Gamma(\alpha_m + w_m + I(m = m^*))}{\Gamma(1 + \sum_{m=1}^u \alpha_m + w_m)}.$$

To simplify the inference about the length of a task affecting a number of nodes we assume that the length follows a Gaussian distribution with expected value μ and variance σ^2 . Let the data t_1, \dots, t_b represent the lengths (say, in minutes) of b tasks.

This leads to the sample mean $\bar{t} = \sum_{i=1}^b t_i$ and variance $s^2 = b^{-1} \sum_{i=1}^b (t_i - \bar{t})^2$. Assuming the standard reference prior (see Bernardo and Smith 1994) for the parameters, we obtain the predictive distribution for the length of a future task, say T , which has the t -distribution with parameters \bar{t} , $((b - 1)/(b + 1))s^2$, $b - 1$, i.e. the probability density of the distribution equals,

$$p(t|\bar{t}, ((b - 1)/(b + 1))s^2, b - 1) = \frac{\Gamma(b/2)}{\Gamma(\frac{b-1}{2})\Gamma(1/2)} \left(\frac{1}{(b+1)s^2}\right)^{\frac{1}{2}} \times \left[1 + \frac{1}{(b+1)s^2}(t - \bar{t})^2\right]^{-\frac{b}{2}}.$$

The probability that a task lasts longer than any given time t equals $P(T > t) = 1 - P(T \leq t)$, where $P(T \leq t)$ is the cumulative distribution function (CDF) of the t -distribution. The value of the CDF can be calculated numerically using functions existing in most computing environments. However, it should also be noted that for a moderate to large b , the predictive distribution is well approximated by the Gaussian distribution with the mean \bar{t} and the variance $s^2 \frac{(b+1)}{(b-3)}$. Consequently, if the Gaussian approximation is used, the probability $P(T \leq t)$ can be calculated using the CDF of the Gaussian distribution.

We now consider an approximation to the probability that a particular computing task is successful. This happens if there will always be at least a single idle node available in the system in the case of a node failure. Let $S = 1$ denote the event that the task is successful, and $S = 0$ the opposite event. We formulate the probability of the success as the sum of the probabilities $P(\text{“none of the nodes allocated to the task fail”}) + \sum_{m=1}^{m_{\max}} P(\text{“}m\text{ of the nodes allocated to the task fail \& at least }m\text{ idle nodes are available as reserves”})$. Here m_{\max} is an upper limit for the number of failures considered. The value can be chosen by judging the size of the contribution of each event, determined by the corresponding probability. Thus, the sum can be simplified by considering only those events that do not have vanishingly small probabilities. A conservative bound for the success probability can be derived by assuming that the m failures take place simultaneously, which leads to,

$$\begin{aligned} P(S = 1) &= 1 - P(S = 0) \\ &= 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur \& less than } m \text{ free nodes available}) \\ &= 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur})P(\text{less than } m \text{ free nodes available}) \\ &\geq 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur})P(\text{less than } m \text{ free nodes at any time point}) \end{aligned} \tag{4}$$

The probability $P(m \text{ failures occur})$ is directly determined by the failure rate model discussed above. The other term, the probability $P(\text{less than } m \text{ free nodes at any time point})$, on the other hand, is dependent both on the level of workload in the system and the eventual need of reserve nodes by the other tasks running simultaneously. Thus, the failure rate model can be used to calculate the probability

distribution of the number of reserve nodes that will be jointly needed by the other tasks (that are using a certain total number of nodes) during the computation time that has the distribution specified above for a single node. This result shows how to decide the number of nodes to keep in reserve in order to be able to complete a computing task successfully when the RP is willing to accept different levels of risk to fail an SLA. The predictive probabilities model has been extensively tested and verified with data from the LANL cluster (Schroeder and Gibson 2006). In the *AssessGrid* project we collected results for the RP1 scenario where the RP is using a cluster with only a few nodes; the test runs have been carried out also for the scenarios RP2–RP4 (Carlsson and Weissman 2009).

5 Predictive possibilities

In this Section we will introduce a possibilistic method for simple predictive estimates of node failures in the next planning period. We will interpret the new model as an alternative for predicting the number of node failures. In this way we will have a possibilistic estimates of the expected number of node failures. An RP may use either one estimate for his risk assessment or use a combination of both. In the *AssessGrid* project (Carlsson and Weissman 2009) we have implemented both models as one software solution to give the user two alternative routes. We will now sketch the possibilistic regression model which is a simplified version of the fuzzy nonparametric regression model with crisp input and fuzzy number output introduced by Wang et al. (2007). This is essentially a standard regression model with parameters represented by triangular fuzzy numbers—typically this means that the parameters are intervals and represent the fact that the information we have is imprecise and/or uncertain. Fuzzy sets were introduced by Zadeh (1965) to represent/manipulate data and information possessing non-statistical uncertainties. Let X be a nonempty set. A fuzzy set A in X is characterized by its membership function $\mu_A: X \rightarrow [0, 1]$, where $\mu_A(x)$ is interpreted as the degree of membership of element x in fuzzy set A for each $x \in X$. It should be noted that the terms membership function and fuzzy subset get used interchangeably and we will write simply $A(x)$ instead of $\mu_A(x)$.

A fuzzy number A is a fuzzy set of the real line \mathbb{R} with a normal, fuzzy convex and continuous membership function of bounded support (Carlsson and Fullér 2002). Fuzzy numbers can be considered as possibility distributions. If A is a fuzzy number and $x \in \mathbb{R}$ then $A(x)$ can be interpreted as the degree of possibility of the statement “ x is A ”. A fuzzy number A is said to be a triangular fuzzy number, denoted $A = (a, \alpha, \beta)$, with center a , left width $a - \alpha > 0$ and right width $\beta - a > 0$ if its membership function is defined by $A(t) = (t - \alpha)/(a - \alpha)$ if $\alpha \leq t \leq a$, $A(t) = (t - \beta)/(a - \beta)$ if $a < t \leq \beta$, and $A(t) = 0$ otherwise. If A is symmetrical, $\alpha = \beta$, then we use the notation $A = (a, \alpha)$. A triangular fuzzy number with center a may be seen as a fuzzy quantity “ x is approximately equal to a ”.

Let \mathbb{F} denote the set of all triangular fuzzy numbers. We consider the following univariate fuzzy nonparametric regression model,

$$Y = F(x) + \varepsilon = (a(x), \alpha(x), \beta(x)) + \varepsilon. \tag{5}$$

where x is a crisp independent variable (input) with domain \mathbb{D} , where $\mathbb{D} \subset \mathbb{R}$. The output $Y \in \mathbb{F}$ is a triangular fuzzy variable, $F: \mathbb{D} \rightarrow \mathbb{F}$ is an unknown fuzzy regression function with its center, lower and upper limit $a(x)$, $\alpha(x)$, $\beta(x)$, and ε may also be considered a fuzzy error or a hybrid error containing both fuzzy and random components.

We will take a sample of a dataset (in our case, a sample from the LANL dataset) which covers inputs and fuzzy outputs according to the regression model. Let x_i, Y_i be a sample of the observed crisp inputs and fuzzy outputs of model (1). The main goal of fuzzy nonparametric regression is to estimate $F(x)$ at any $x \in \mathbb{D}$ from $(x_i, Y_i), i = 1, 2, \dots, n$. The membership function of an estimated fuzzy output should be as close as possible to that of the corresponding observed fuzzy number (Wang et al. 2007). From this point of view, we shall estimate $a(x), \alpha(x), \beta(x)$ for each $x \in \mathbb{D}$ in the sense of best fit with respect to some distance that can measure the closeness between the membership functions of the estimated fuzzy output and the corresponding observed one. There are actually a few such distances available. We use the distance proposed by Diamond (1988) as a measure of the fit because it is simple to use and, more importantly, can result in an explicit expression for the estimated fuzzy regression function when the local linear smoothing technique is used. Let $A = (a, \alpha_1, \beta_1), B = (b, \alpha_2, \beta_2)$ be two triangular fuzzy numbers. Then the squared distance between A and B is defined by Diamond (1988), $d^2(A, B) = (\alpha_1 - \alpha_2)^2 + (a - b)^2 + (\beta_1 - \beta_2)^2$. Using this distance we will extend the local linear smoothing technique in statistics to fit the fuzzy nonparametric model (5). Let us now assume that the observed (fuzzy) output is $Y_i = (a, \alpha_i, \beta_i)$, then with the Diamond distance measure and a local linear smoothing technique we need to solve a locally weighted least-squares problem in order to estimate $F(x_0)$, for a given kernel K and a smoothing parameter h , where

$$K_h(|x_i - x_0|) = K\left(\frac{|x_i - x_0|}{h}\right), \tag{6}$$

The kernel is a sequence of weights at x_0 to make sure that data that is close to x_0 will contribute more when we estimate the parameters at x_0 than those that are farther away, i.e. are relatively more distant in terms of the parameter h . Let

$$\hat{F}_{(i)}(x_i, h) = (\hat{a}_{(i)}(x_0, h), \hat{\alpha}_{(i)}(x_0, h), \hat{\beta}_{(i)}(x_0, h)). \tag{7}$$

be the predicted fuzzy regression function at input x_i . Compute $\hat{F}_{(i)}(x_i, h)$ for each x_i and let

$$CV(h) = \frac{1}{l} \sum_{i=1}^l d^2(Y_i, \hat{F}_{(i)}(x_i, h)). \tag{8}$$

We should select h_0 so that it is optimal in the following expression

$$CV(h_0) = \min_{h > 0} CV(h). \tag{9}$$

By solving this minimalization problem, we get the following estimate of $F(x)$ at x_0 (Wang et al. 2007),

$$\hat{F}(x_0) = (\hat{\alpha}(x_0), \hat{\alpha}(x_0), \hat{\beta}(x_0)) = (\mathbf{e}_1^T \mathbf{H}(x_0, h) \mathbf{a}_Y, \mathbf{e}_1^T \mathbf{H}(x_0, h) \alpha_Y, \mathbf{e}_1^T \mathbf{H}(x_0, h) \beta_Y), \tag{10}$$

where $\mathbf{H}(x_0, h) = (\mathbf{X}^T(x_0) \mathbf{W})^T(x_0, h) \mathbf{X}(x_0)^{-1} \mathbf{X}^T(x_0) \mathbf{W}(x_0, h)$,

$$\mathbf{X}(x_0) = \begin{pmatrix} 1 & x_1 - x_0 \\ 1 & x_2 - x_0 \\ \vdots & \vdots \\ 1 & x_n - x_0 \end{pmatrix} \mathbf{a}_Y = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \alpha_Y = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \beta_Y = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}$$

and $\mathbf{W}(x_0, h) = \text{Diag}(K_h(|x_1 - x_0|), \dots, K_h(|x_n - x_0|))$ and $\mathbf{e}_1 = (1, 0)^T$.

Note 1: If the observed inputs are symmetric, in a way that, $a_i - \alpha_i = \beta_i - a_i$, then $\hat{F}(x_0)$ is also a symmetric triangular fuzzy number. In fact, if the i th symmetric fuzzy output is denoted by $Y_i = (a_i, c_i)$, where $c_i = a_i - \alpha_i = \beta_i - a_i$ denotes the spread of Y_i , then the spread vector of the n observed fuzzy outputs can be expressed as $\mathbf{c}_Y = \mathbf{a}_Y - \alpha_Y = \beta_Y - \mathbf{a}_Y$. According to (10) we get, $\hat{\alpha}(x_0) - \hat{\alpha}(x_0) = \hat{\beta}(x_0) - \hat{\alpha}(x_0) = \mathbf{e}_1^T \mathbf{H}(x_0, h) \mathbf{c}_Y$ and, therefore, $\hat{F}(x_0) = (\mathbf{e}_1^T \mathbf{H}(x_0, h) \mathbf{a}_Y, \mathbf{e}_1^T \mathbf{H}(x_0, h) \mathbf{c}_Y)$ is a symmetric triangular fuzzy number.

6 Predictive possibilities in grid computing management

Suppose there are n observations, $\{x_1, \dots, x_n\}$, where x_i denotes the number of node failures in the i th planning period, and l of them are different. Without loss of generality we will use the notation $\{x_1, \dots, x_l\}$ for the l different observations. Let X_{n+1} denote the (unknown) possibility distribution of the number of node failures in the next planning period. Then X_{n+1} is defined over the set of non-negative integers $\{0, 1, 2, 3, \dots\}$. It is clear that from the first n observations we can not predict the exact value of the $(n + 1)$ th observation, but using a fuzzy nonparametric regression technique we can estimate the possibility of the statement “the number of node failures in the $(n + 1)$ th planning period will be x_0 ” where x_0 is a non-negative integer. Let us introduce a notation $v_i = |\{j : \text{the } j\text{th observation is equal to } x_i\}|$. In our case the center of the symmetric triangular fuzzy number will be $a_i = x_i$, and its lower and upper limits are,

$$\alpha_i = x_i - \frac{v_i}{n}, \quad \beta_i = x_i + \frac{v_i}{n}.$$

Since we work with the symmetrical case, we shall use the notation $c_i = v_i/n$. We should choose a kernel function and h . Let $h = 1/n$ and let

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right),$$

the Gaussian kernel. Let $k_i = K_h(|x_n - x_0|)$. We calculate now the matrix $\mathbf{e}_1^T \mathbf{H}(x_0, h)$ by,

$$\begin{aligned}
 (\mathbf{X}^T(x_0)\mathbf{W})^T(x_0, h)\mathbf{X}(x_0) &= \begin{pmatrix} \sum_{i=1}^l k_i & \sum_{i=1}^l k_i(x_i - x_0) \\ \sum_{i=1}^l k_i(x_i - x_0) & \sum_{i=1}^l k_i(x_i - x_0)^2 \end{pmatrix} \implies \\
 (\mathbf{X}^T(x_0)\mathbf{W})^T(x_0, h)\mathbf{X}(x_0)^{-1} &= \frac{1}{(\sum_{i=1}^l k_i)(\sum_{i=1}^l k_i(x_i - x_0)^2) - (\sum_{i=1}^l k_i(x_i - x_0))^2} \\
 &\quad \times \begin{pmatrix} \sum_{i=1}^l k_i(x_i - x_0)^2 & -\sum_{i=1}^l k_i(x_i - x_0) \\ -\sum_{i=1}^l k_i(x_i - x_0) & \sum_{i=1}^l k_i \end{pmatrix}
 \end{aligned}$$

Let $s = \sum_{i=1}^l k_i$, $t = \sum_{i=1}^l k_i(x_i - x_0)$ and $v = \sum_{i=1}^l k_i(x_i - x_0)^2$. Then,

$$(\mathbf{X}^T(x_0)\mathbf{W})^T(x_0, h)\mathbf{X}(x_0)^{-1}\mathbf{X}^T(x_0) = \begin{pmatrix} v - t(x_1 - x_0) \dots v - t(x_l - x_0) \\ s(x_1 - x_0) - t \dots s(x_l - x_0) - t \end{pmatrix}$$

and

$$\begin{aligned}
 \mathbf{H}(x_0, h) &= (\mathbf{X}^T(x_0)\mathbf{W})^T(x_0, h)\mathbf{X}(x_0)^{-1}\mathbf{X}^T(x_0)\mathbf{W}(x_0, h) \\
 &= \begin{pmatrix} k_1(v - t(x_1 - x_0)) \dots k_l(v - t(x_l - x_0)) \\ k_1(s(x_1 - x_0) - t) \dots k_l(s(x_l - x_0) - t) \end{pmatrix},
 \end{aligned}$$

and $\mathbf{e}_1^T\mathbf{H}(x_0, h) = (k_1(v - t(x_1 - x_0)) \dots k_l(v - t(x_l - x_0)))$, and finally,

$$\mathbf{e}_1^T\mathbf{H}(x_0, h)\mathbf{a}_Y = \sum_{i=1}^l x_i k_i(v - t(x_i - x_0)), \quad \mathbf{e}_1^T\mathbf{H}(x_0, h)\mathbf{c}_Y = \sum_{i=1}^l c_i k_i(v - t(x_i - x_0)).$$

So we can write our estimation as,

$$\begin{aligned}
 \hat{F}(x_0) &= (\mathbf{e}_1^T\mathbf{H}(x_0, h)\mathbf{a}_Y, \mathbf{e}_1^T\mathbf{H}(x_0, h)\mathbf{c}_Y) \\
 &= \left(\sum_{i=1}^l x_i k_i(v - t(x_i - x_0)), \sum_{i=1}^l c_i k_i(v - t(x_i - x_0)) \right),
 \end{aligned}$$

which is the predicted possibility distribution of the symmetric triangular form of node failures. So, the estimate of the possibility that “the number of node failures in $(n + 1)$ th planning period will be x_0 ”, denoted by $\text{Pos}(X_{n+1} = x_0)$, is computed by,

$$\text{Pos}(X_{n+1} = x_0) = \begin{cases} 1 - \frac{|\sum_{i=1}^l x_i k_i(v - t(x_i - x_0)) - x_0|}{\sum_{i=1}^l c_i k_i(v - t(x_i - x_0))}, & \text{if } \frac{|\sum_{i=1}^l x_i k_i(v - t(x_i - x_0)) - x_0|}{\sum_{i=1}^l c_i k_i(v - t(x_i - x_0))} \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

The possibilistic model is a faster and more robust estimate (that is the possibility of a node failure always exceeds the probability of a node failure) and will therefore be useful for online and real-time risk assessments with relatively small samples of data.

7 A lower limit for the probability of success of computing tasks in a grid

In this section we will consider an approximation to the probability that a particular computing task in a grid is successful. This happens if there will always be at least a

single idle node available in the system in the case of a node failure. We will use the Fréchet–Hoeffding bounds for copulas to show a lower limit for the probability of success of a computing task in a cluster (or a grid/cloud). A two-dimensional copula is a function $C: [0, 1]^2 \rightarrow [0, 1]$ which satisfies $C(0, t) = C(t, 0) = 0$, $C(1, t) = C(t, 1) = t$ for all $t \in [0, 1]$, $C(u_2, v_2) - C(u_1, v_2) - C(u_2, v_1) + C(u_1, v_1) \geq 0$ for all $u_1, u_2, v_1, v_2 \in [0, 1]$ such that $u_1 \leq u_2$ and $v_1 \leq v_2$. Equivalently, a copula is the restriction to $[0, 1]^2$ of a bivariate distribution function whose margins are uniform on $[0, 1]$. The importance of copulas in statistics is described in the following theorem (Sklar 1959): Let X and Y be random variables with joint distribution function H and marginal distribution functions F and G , respectively. Then there exists a copula C such that $H(x, y) = C(F(x), G(y))$. Conversely, for any univariate distribution functions F and G and any copula C , the function H is a two-dimensional distribution function with marginals F and G . For any copula C we have $W(u, v) = \max \{0, u + v - 1\} \leq C(u, v) \leq M(u, v) = \min \{u, v\}$. In the statistical literature, the largest and smallest copulas, M and W are generally referred to as the Fréchet–Hoeffding bounds. Let us recall that in (4) we used the notations $P(\text{success})$ as $P(S = 1)$ and $P(\text{failure})$ as $P(S = 0)$; furthermore, let us use the abbreviation less-than- m -anytime for the event “less than m free nodes available at any point of time”. Then we can rewrite (4) as,

$$\begin{aligned}
 P(\text{success}) &= 1 - P(\text{failure}) = 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur \& less-than-}m\text{-anytime}) \\
 &= 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur})P(\text{less than } m \text{ free nodes available}) \\
 &\geq 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur})P(\text{less-than-}m\text{-anytime}) \\
 &\geq 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur}) \\
 &\quad \times \left\{ \sum_{j=1}^{t^*} P(j - 1 \leq \text{the length of a task} < j, \text{less-than-}m\text{-anytime}) \right. \\
 &\quad \left. + P(\text{the length of a task} \geq t^*, \text{less-than-}m\text{-anytime}) \right\} \\
 &\geq 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur}) \\
 &\quad \times \left\{ \sum_{j=1}^{t^*} P(\text{the length of a task} < j, \text{less-than-}m\text{-anytime}) \right. \\
 &\quad \left. + P(\text{the length of a task} \geq t^*, \text{less-than-}m\text{-anytime}) \right\}
 \end{aligned}$$

Let us introduce the following notations $G(m) = P(\text{less-than-}m\text{-anytime})$ and $F(t) = P(\text{the duration of a task is less than } t)$. Let t^* be chosen such that $1 - F(t^*) > 0.995$. Furthermore, denote the copula of F and G by H , where

$$H(t, m) = P(\text{the duration of a task is less than } t, \text{less-than-}m\text{-anytime}).$$

Then using the Fréchet–Hoeffding upper bound for copulas we find that,

$$\begin{aligned}
 P(\text{success}) &\geq 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur}) \\
 &\times \left\{ \sum_{j=1}^{t^*} \min \left\{ \frac{\Gamma(b/2)}{\Gamma(\frac{b-1}{2})\Gamma(1/2)} \left(\frac{1}{(b+1)s^2} \right)^{\frac{1}{2}} \left[1 + \frac{1}{(b+1)s^2} (j-\bar{j})^2 \right]^{-\frac{b}{2}}, \right. \right. \\
 &P(\text{less than } m \text{ free nodes at any time point}) \left. \left. \right\} \right. \\
 &\left. + \underbrace{P(\text{the length of a task is } \geq t^*, \text{less-than-}m\text{-anytime})}_{\leq 0.005} \right\}
 \end{aligned}$$

If we summarize these results we get (Carlsson et al. 2009),

$$\begin{aligned}
 P(\text{success}) &\geq 1 - \sum_{m=1}^{m_{\max}} P(m \text{ failures occur}) \\
 &\times \left\{ \sum_{j=1}^{t^*} \min \left\{ \frac{\Gamma(b/2)}{\Gamma(\frac{b-1}{2})\Gamma(1/2)} \left(\frac{1}{(b+1)s^2} \right)^{\frac{1}{2}} \left[1 + \frac{1}{(b+1)s^2} (j-\bar{j})^2 \right]^{-\frac{b}{2}}, \right. \right. \\
 &P(\text{less-than-}m\text{-anytime}) \left. \left. \right\}.
 \end{aligned}$$

8 Summary and conclusion

We developed a pure probabilistic and a possibilistic technique for assessing the risk of an SLA for a computing task in a grid/cloud environment. Using the predictive probabilistic approach we developed a framework for resource management in grid/cloud computing, and by introducing an upper limit for the number of possible failures, we approximated the probability that a particular computing task can be executed. We also showed a lower limit for the probability of success of a computing task in a grid. In the possibilistic model we estimated the possibility distribution defined over the set of node failures using a fuzzy nonparametric regression technique. The probabilistic models scale from 10 nodes to 100 nodes (and then on to any number of nodes); while the possibilistic models scale to 100 nodes. The RP can use both models to get two alternative risk assessments.

In the *AssessGrid* project we carried out a number of validation tests in order to find out (1) how well the predictive possibilistic models can be fitted to the LANL dataset, (2) what differences can be found between the probabilistic and possibilistic

predictions and (3) if these differences can be given reasonable explanations. In the testing we worked with short and long duration computing tasks scheduled on a varying number of nodes and the SLA probabilities of failure estimates remained reasonable throughout the testing. The smoothing parameter h for the possibilistic models should be determined for the actual cluster of nodes, and in such a way that we get a good fit between the probabilistic and the possibilistic models. The approach we used for the testing was to experiment with combinations of h and then to fit a distribution to the results; the distribution could then be used for interpolation.

Acknowledgments The work on the Bayes' models in Sects. 3, 4 has greatly benefited from a collaboration with Professor Jukka Corander, Department of Statistics, Åbo Akademi University, who is an eminent statistician.

References

- Bernardo JM, Smith AFM (1994) Bayesian theory. Wiley, Chichester
- Brandt JM, Gentile AC, Marzouk YM, Pébay PP (2005) Meaningful statistical analysis of large computational clusters. In: Proceedings of the 2005 IEEE international conference on cluster computing, Burlington, MA, pp 1–2. doi:[10.1109/CLUSTER.2005.347090](https://doi.org/10.1109/CLUSTER.2005.347090)
- Carlsson C, Fullér R (2002) Fuzzy reasoning in decision making and optimization. Springer, Berlin
- Carlsson C, Weissman O (2009) Advanced risk assessment, D4.1. The AssessGrid Project, IST-2005-031772, Berlin
- Carlsson C, Fullér R, Mezei J (2009) A lower limit for the probability of success of computing tasks in a grid. In: Proceedings of the tenth international symposium of Hungarian researchers on computational intelligence and informatics (CINTI 2009) Budapest, Hungary, pp 717–722
- Czajkowski K, Foster I, Kesselman C (2005) Agreement-based resource management. Proc IEEE 93(3):631–643. doi:[10.1109/JPROC.2004.842773](https://doi.org/10.1109/JPROC.2004.842773)
- Diamond P (1988) Fuzzy least squares. Inf Sci 46(3):141–157. doi:[10.1016/0020-0255\(88\)90047-3](https://doi.org/10.1016/0020-0255(88)90047-3)
- Iosup A, Jan M, Sonmez OO, Epema, DHJ (2007) On the dynamic resource availability in grids. In: Proceedings of the 8th IEEE/ACM international conference on grid computing (GRID 2007), pp 26–33. doi:[10.1109/GRID.2007.4354112](https://doi.org/10.1109/GRID.2007.4354112)
- Lintner J (1965) The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. Rev Econ Stat 47(1):13–37
- Magana E, Serrat J (2007) Distributed and heuristic policy-based resource management system for large-scale grids. Lect Notes Comput Sci 4543:184–187
- Marston S, Li Z, Bandyopadhyay S, Zhang J, Ghalsasi A (2011) Cloud computing—the business perspective. Decis Support Syst 51(1):176–189
- Rimal BH, Jukan A, Katsaros D, Goelven Y (2011) Architectural requirements for cloud computing systems: an enterprise cloud approach. J Grid Comput 9(1):3–26
- Robert CP, Casella G (2005) Monte Carlo statistical methods. Springer, New York
- Schroeder B, Gibson GA (2006) A large-scale study of failures in high-performance computing systems. In: DSN2006 conference proceedings, Philadelphia, pp 249–258. doi:[10.1109/DSN.2006.5](https://doi.org/10.1109/DSN.2006.5)
- Sklar A (1959) Fonctions de Répartition à Dimensions et Leurs Marges. Paris, France Publ. Inst. Statist. Univ. Paris 1959(8):229–231
- Wang N, Zhang W-X, Mei C-L (2007) Fuzzy nonparametric regression based on local linear smoothing technique. Inf Sci 177(18):3882–3900. doi:[10.1016/j.ins.2007.03.002](https://doi.org/10.1016/j.ins.2007.03.002)
- Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353. doi:[10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)