Collaboration and the Knowledge Economy: Issues, Applications, Case Studies
Paul Cunningham and Miriam Cunningham (Eds)
IOS Press, 2008 Amsterdam
ISBN 978–1–58603–924-0

Economically Enhanced Risk-aware Grid SLA Management

Karim DJEMAME[1], James PADGETT[1], Iain GOURLAY[1], Kerstin VOSS[2], Dominic BATTRE[3], Odej KAO[3]
[1]*School of Computing, University of Leeds, Leeds LS2 9JT, UK*
Tel: +44 113 3436590, Fax: + 44 113 3435468,
Email: karim@comp.leeds.ac.uk, jamesp@comp.leeds.ac.uk, iain@comp.leeds.ac.uk
[2]*Paderborn Center for Parallel Computing, Fürstenallee 11, 33102 Paderborn, Germany*
Tel: +49 5251 60-6321, Fax: +49 5251 60-6297, Email: kerstinv@upb.de
[3]*Technische Universität Berlin, Department of Telecommunication Systems, Einsteinufer 17, 10587 Berlin, Germany, Tel: +49 30 31424230/21988, Fax: +49 30 31421060*
Email: Dominic.Battre@tu-berlin.de, Odej.Kao@tu-berlin.de

Abstract: Grid computing has emerged as a global infrastructure for the next generation of e-Science and commercial applications. This paper reports on recent results on risk-awareness integration in the EU funded project AssessGrid (Advanced Risk Assessment and Management for Trustable Grids) architecture as well as the potential economic issues underlying risk-aware SLA management. Specifically, it will focus on the resource provider, broker and end-user.

1. Introduction

Advances in Grid computing research have in recent years resulted in considerable commercial interest in utilising Grid infrastructures to support commercial applications and services. 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 infrastructures, in order to move beyond the best-effort approach to service provision that current Grid infrastructures follow.

AssessGrid (Advanced Risk Assessment and Management for Trustable Grids) addresses the key issue of risk by developing a framework to support risk assessment and management for all three Grid actors (end-user, broker, and resource provider) [1,2]. To integrate risk awareness and support risk management in all Grid layers, new components are introduced: the provider benefits from access to a consultant service that provides statistical information to support both risk assessment and the identification of infrastructure bottlenecks. The broker makes use of a confidence service that provides a reliability measure of a resource provider's risk assessment, based on historical data. In addition, a workflow assessor supports the broker by providing risk assessments for entire workflows. The end-user interface is realised as a Grid portal that provides the functionality required to support the end-user in using the Grid. The end-user must be able to select whether to negotiate with a provider or with a broker.

On the other hand, AssessGrid is contributing to the establishment of commercial adoption of Grid technology by providing a framework where an end-user can specify a Service Level Agreement (SLA) and choose a provider based on quality and price given, therefore creating a market place and putting the providers in competition with each other.

This paper reports on recent results on risk-awareness integration in the AssessGrid architecture as well as the potential economic issues underlying risk-aware SLA management. Specifically, it will focus on the resource provider, broker and end-user.

2. AssessGrid

AssessGrid focuses on the integration of risk management functionality within Grid middleware. It does this by addressing the concerns of end-users and providers through encouraging greater commercial interest in Grid usage through incorporation of risk assessment mechanisms into Grid infrastructures as well as automated SLA agreement processes utilizing risk information. Incorporation of risk-aware components within the SLA negotiation process as an additional decision support parameter for the end-user is of primary importance. Risk is an ideal decision support parameter within the AssessGrid scenario since it combines both the quantifiable probability of SLA failure with the non-deterministic expected loss, a parameter known only to the beneficiary of the services stated in the SLA. The usage scenarios addressed by the AssessGrid architecture consider 3 principle actors: end-user, broker and provider.

2.1 Resource Provider

The AssessGrid architecture integrates risk-awareness into a Resource Management System (RMS), OpenCCS RMS [3] by extending existing modules (negotiation manager and scheduler) to handle information about Probabilities of SLA Failures (PoFs) and incorporating new components: a risk assessor and a consultant service with associated database. The consultant service uses monitoring information from the historical database in order to generate statistical data required by the risk assessor as input. The risk assessor is responsible for estimating the probability of an SLA failure, for each SLA, in order to determine whether an SLA should be agreed. The probability of an SLA violation is influenced by the availability of spare resources, which can be used in the case of a resource outage, as well as the provider's fault-tolerance capabilities.

OpenCCS makes advance reservations during the SLA negotiation. Resource reservation is used in order to determine the probability of failure of meeting the SLA by considering the resource stability. The RMS functionalities to reduce risk and fulfil the SLA include: 1) checkpointing and migration; 2) dedicated spare resources; 3) pool of spare resources; 4) profit considering scheduling after resource failures, and 5) job outsourcing to another provider.

A provider offers access to resources and services through formal SLA offers specifying the requirements as well as PoF, price, and penalty. Providers need well-balanced infrastructures, so that they can maximise the offerable 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. We report on a number of economic issues that have been identified which affect the provider. These issues can be categorized as belonging to SLA negotiation and post-negotiation phases.

2.2 Broker

After the SLA negotiation has returned an SLA offer, the broker is responsible for performing reliability checks on the PoFs contained in the SLA offers. Without this check, the end-user has no independent view on the provider's assessment, which cannot be assumed to be impartial. SLA offers that are deemed to be unreliable are subjected to an additional risk assessment by the broker using historical data related to the provider making the offer. Where multiple SLA offers are returned by the SLA negotiation process, the broker can rank these according to a price, penalty, PoF matrix depending on the priorities of the end-user.

2.3 End-User

An end-user is a participant from a broad public approaching the Grid in order to perform a specific task that comprises of one or more services. In order to make a request for such services, the end-user must indicate the task and associated requirements formally within an SLA template. The information contained within the template is used to negotiate access for the end-user with providers offering these services, such that the task may be completed. The inclusion of risk information within the SLA negotiation process allows the end-user to make informed, risk-aware decisions on the SLA offers received so that any decision is acceptable and balances cost, time and risk.

The end-user is provided with a number of abstract applications that make use of Grid services deployed within the Grid fabric layer. SLA requests and offers are exchanged between end-user and broker or provider, in order to agree an SLA that grants permission to invoke a Grid service in the fabric layer. Within each layer, the organisation performing the role of each actor must define a policy statement governing the acceptable bounds of negotiation. This restricts end-users and contractors to request or offer SLAs that fall outside of the organization's acceptable limits. For example, in addition to specifying budget constraints, there may be a restriction on a provider's penalty conditions to limit the financial loss incurred because of an SLA violation. Taking these policy limits into consideration, an end-user can negotiate an SLA to run a Grid service by defining requirements as well as the requested QoS in an SLA request. During the definition process the end-user evaluates the importance of the application in terms of its urgency and the consequences of delayed results or failure. A further validation of the policy limits must be made against the SLA offers received from the broker or providers.

Where several SLA offers have been negotiated on behalf of the end-user, the broker can return a ranked list - according to price, penalty, and PoF. The challenge for the end-user is to find an SLA offer which offers the best service in terms of price, penalty, and PoF. We report on a mathematical model to help the end-user make the best offer selection based on quality criteria. The end-user defines a ranking of the quality criteria (e.g. PoF is more important than price) in order to measure each of the offers according to its closeness to the criteria.

3. WS-Agreement Extension

The SLA negotiation is based on the WS-Agreement protocol [4], which has been extended in AssessGrid project to allow flexible SLA negotiation schemes between contractors and service providers. Modifications consist in the addition of two operations: *commit()* and *createAgreements()*, with a significant change in the semantics of *createAgreement()* operation. The signature of this operation remains intact. However, the changes in its semantics are important enough to consider that AssessGrid version of WS-Agreement protocol is actually a modification to the WS-Agreement protocol rather than an extension, as the “single round” acceptance model in the original WS-Agreement specification posed an unavoidable limitation to AssessGrid requirements. For more details see [5]. The actual AssessGrid version of the SLA template is based on that defined in the original WS-Agreement specification [4].

4. Economic Issues in Risk-aware SLA Management

The integration of risk-awareness into the Grid provides a number of benefits within an economy framework but also gives rise to numerous research problems. In the following we present an overview of the AssessGrid developments and their impact for continuing research in economic issues.

4.1 Resource Provider

A number of economic issues have been identified which affect the provider. These issues can be categorised as belonging to the pre-runtime (i.e. during SLA negotiation), run-time and post-runtime phases.

In the pre-runtime phase a risk aware negotiation requires that a provider place an advance reservation for the SLA and calculates the PoF. Based on this, a provider determines the price and penalty fee that will be offered to an end-user. In order to increase the chances that a potential end-user accepts an offer, a provider might offer a service with better conditions, i.e. lower price, lower PoF, or higher penalty. A provider's decision whether to agree or reject an SLA depends on the fees and the requested PoF in comparison with the current status of its infrastructure.

For contractors (end-users or brokers), an important provider selection criterion is the price. The SLA template contains pricing information for actions such as data transfer, CPU usage, and storage. Within the AssessGrid model these prices are variable since the price depends on the SLAs PoF value.

The market mechanism will influence the pricing since each provider has only a limited resource set with variable utilisation. Consequently, prices for resource usage will not be fixed but will depend on the economics of supply and demand. Reservations which are well in advance will usually result in a reduced price since there will be access to a greater number of free reservation slots. Equally, immediate resource usage may also result in reduced prices, as providers try to increase their utilisation if demand is low. However, end-users risk resources unavailability if they wait too long before reserving resources.

After an SLA has been agreed by the provider and the end-user, the provider has to ensure during runtime that the SLA will not be violated. Accordingly, the provider's risk management activities are controlled by estimating the penalty payments in the case of an SLA violation. Providers using the AssessGrid developments will be able to initiate precautionary fault tolerance mechanisms in order to prevent SLA violations. The penalty fees, in addition to the PoF (i.e. risk) are the decisive factors in determining which fault tolerance mechanisms are initiated.

In the post-runtime phase the provider has to evaluate the final SLA status to determine whether a penalty fee has to be paid. Even in the case the SLA had been fulfilled the costs for the fulfilment have to be checked since the initiation of a fault tolerance mechanism also consumes resources and therewith results in additional costs. The results of the evaluation process will point out on the one hand whether adjustments in the offer making policies are necessary in order to increase the provider's profit. On the other hand statistics can be generated which show whether initiated fault tolerance mechanisms had been able to prevent an SLA violation.

4.2 Broker

In the following section, we report on 1) how risk assessment is implemented at the broker layer [6], and 2) the workflow scheduling algorithm supported by the broker. Workflow scheduling is one of the key issues in the management of workflow execution and refers to the process of mapping and managing execution of inter-dependent tasks on distributed resources.

4.2.1 Risk Assessment

The Risk Assessor component provides the logic used by the broker's Confidence Service to compute a reliability measure and if necessary a risk assessment. It does this using past SLA data from a historical database. The Risk Assessor uses this data to determine the reliability of the offered PoF value from the provider's SLA. It returns a reliability object which contains the providers name, the number of SLAs on which the reliability is based, a reliability measure and an adjusted PoF. The algorithm for computing the reliability measure is based on a comparison of the total number of observed SLA failures in the historical database with the expected number of failures assuming the provider's PoFs are accurate. This is measured in terms of the number of standard deviations and compared with the expected behaviour of providers that are reliable - in the sense that any systematic errors in their risk assessments are within pre-specified bounds. A key feature of this algorithm is to account for the fact that old SLA data may be less relevant than newer SLA data. A provider's behaviour with regard to risk assessments could change as a consequence of a variety of factors, e.g.

- A provider's infrastructure is updated - this may have an effect on the reliability of subsequent risk assessments.
- A provider's risk assessment methodology or model parameterisation may change.
- A provider's policy may change, for example due to economic considerations. For example, they may decide that they can make more profit if they start to give overoptimistic estimates to end-users/brokers, in order to persuade them to agree offers.

If a provider's behaviour changes (such that a reliable provider becomes unreliable or vice-versa) then a reliability measure that accounts equally for all SLA data could be misleading. Similarly, when a provider is evaluated as unreliable and the broker makes its own PoF estimate, this is unlikely to be accurate when a provider's behaviour has recently changed. In order to address this problem, the reliability measure takes the form of a weighted average. SLAs are split into equally sized categories, according to how recently they were executed. The reliability measure is computed over each category and then weighted, where the category weightings increase linearly, moving forward in time. Hence the most recent category has the largest weight. Simulation results indicate that this approach is superior to both the basic measure (without weights) and a moving average with fixed window size, for providers that change behaviour. If the reliability measure is less than some pre-specified threshold value (chosen according to the level of confidence required that a provider is unreliable) then the broker assumes that the provider's assessment is accurate and its PoF estimate is therefore equal to the provider's. Otherwise, the broker performs its own estimate under the assumption that a single parameter can be used to address systematic errors of the form, $P_{fail} \approx P_{offerred}(1+\delta)$ where δ may be positive or negative. Similarly to the reliability measure, the SLAs are split into categories and only SLAs with an offered PoF within *x*% of the value in the current SLA offer are considered. If the total number of SLAs considered is less than some threshold value then *x* is incremented and the process repeated until a sufficiently large sample is found.
The value of δ is then computed as a weighted average across the categories and used to estimate the PoF for the SLA offer under consideration.

4.2.2 Workflow Scheduling

The broker can schedule workflows given a correctly formatted SLA request. The workflow scheduling algorithm currently implemented is detailed in the following:

1. Generate a DAG from the end-user's workflow request. Each vertex corresponds to a task in its pending state and edges correspond to dependencies between tasks. The distance of an edge from vertex *i* to vertex *j* is the estimated time taken to transit between those states, i.e. the estimated execution time for task *i* to execute. An end vertex is created and an edge from each exit task to the end vertex, with length(s) corresponding to the execution time of the exit task(s). There is a limiting assumption that the workflows considered will have only one entry and exit task. Scenarios involving multiple entry and exit tasks are the subject of future consideration.

2. Estimate the required workflow execution time, based on the longest path from the entry task to the end vertex. Based on this, the total slack time available is estimated and a slack time ratio is computed to determine the proportion of slack time to be allocated to each task. The amount of slack time allocated to a task is linearly proportional to the task's expected execution time.
3. Compute the maximum acceptable PoF for each task P_{task}, using the approach outlined below and compute the earliest start and latest finish time for each task.
4. Filter providers and send requests for quotes for each task.
5. Once responses are received, compute the reliability and broker's PoF estimate for each quote. Those where the broker's PoF estimate exceeds the task threshold will be rejected.
6. Check that there is at least one quote for every task. If this is not the case then no workflow mapping can be found. Otherwise,
7. For each task, choose the best quote according to the cost function. If it is no longer available, move to the next best quote and so on. If no agreements can be secured for a task, roll back and cancel all previously made agreements, i.e., no workflow mapping can be found.

In the simplest algorithm, the PoF threshold is the same for each task and is computed as follows: the maximum acceptable PoF for the workflow is specified by the end-user as P_{max}, or, equivalently, the minimum acceptable probability of success $P_{succ} = 1 - P_{max}$. If all tasks are treated as equivalent in terms of acceptable PoF, then note that if the probability of success for each of the n tasks is $P_{task}(success)$ then the probability of success for the entire workflow is,

$$P_{wf}(success) = (P_{task}(success))^n \quad (1$$

The minimum probability of success for each task is then set to $\sqrt[n]{P_{succ}}$, i.e. the PoF for each task must satisfy,

$$P^{i}_{fail} < (1 - \sqrt[n]{P_{succ}}),\ i = 1 \ldots n$$

An unexplored problem is the economic issue underlying the handling of workflows for the broker. Since the broker is responsible in this case for the SLA fulfilment, it has to react on failures (negotiate with providers for a repeated job execution) in order to prevent paying penalties. Other essential economic issues are the pricing mechanisms for brokers and providers which must take account of the probability of failure in a risk-aware Grid environment.

5. Current Implementation

AssessGrid current software prototype integrates risk-awareness in all three layers of the Grid (end-user, broker and resource provider). The prototype supports direct negotiation between an end-user and a provider, using the broker layer to provide a reliability measure of the provider's risk assessment. In addition, negotiation between an end-user and a provider, using a broker as a mediator is also supported. In this scenario, SLA negotiation between the end-user and multiple providers is performed using the broker Service. This improves the scope of the end-user SLA request and also provides a list of SLA offers ranked by PoF. Also, the broker's functionality is enhanced by allowing it to function as a high-level provider. The broker can now offer its own SLAs combining SLAs from providers into a single SLA, which is useful for workflow mapping.

In this scenario end-users agree SLAs with the broker directly, who in turn agrees SLAs with all providers involved in executing the end-user's tasks or workflow. In the case of single tasks, these may need to be executed redundantly - to achieve a very low PoF (e.g. in cases where a single provider is unable to offer one low enough). For workflow orchestrations the broker is used to map entire workflows to individual SLAs agreed directly with resource providers.

6. Business Benefits

The economic benefit of using a broker within the SLA negotiation process affects all three Grid actors and provides the opportunity for an economy model where SLAs for software services are bought and sold based on differentiated classes of service. The broker's role also creates a competitive market place. In AssessGrid, three business models are identified:

1. *The trusted consulting party model*: a contract is defined through an SLA with the provider and end-user through direct negotiation and the end-user can query the broker's Confidence Service to obtain a reliability measure for an SLA offer.
2. *The intermediate party model*: the end-user submits an SLA request to the broker, which then forwards the request to suitable providers. The broker returns SLA offers to the end-user, ranked by price, penalty, and PoF. The end-user is then free to select and commit to an SLA offer by interacting directly with the corresponding provider.
3. *The virtual provider model*: the end-user agrees an SLA with the broker, which in turn agrees SLAs with all providers involved in executing the end-user's application. The broker can be used to map entire workflows to resources.

7. Conclusion

This paper has reported on recent results on risk-awareness integration in the AssessGrid architecture as well as the potential economic issues underlying risk-aware SLA management.

Resource providers are able to assess an SLA's Probability of Failure (PoF) before committing to it, through risk estimation models. This can be used to identify bottlenecks in their own infrastructure. The ability to assess the risk associated with an SLA request before it commits, enables a resource provider to build a planning based RMS schedule using the computed PoF values.

The problem of evaluating the reliability of the PoF estimates for SLA offers received from providers is also key. A broker that acts on behalf of end-users to find and negotiate for suitable resources benefits from risk management mechanisms.

Further work will focus on further enhancing the software prototype functionality. For example the method used to rank quotes within the broker layer will be improved by considering three criteria, price penalty and PoF, rather than just PoF. Additional effort is needed to address workflow fault tolerance and pricing strategies. Although the current prototype can react to failures in task SLAs belonging to a workflow, the broker needs to determine whether re-submission of the task (in order to ensure completion of the workflow) is appropriate financially. In this case it needs to evaluate whether it is more profitable to pay a penalty fee to the end-user and accept that the SLA has failed. In the latter case it will try to resubmit and retain the rewards set out in the SLA.

References

[1] AssessGrid - Advanced Risk Assessment and Management for Trustable Grids, 2008. http://www.assessgrid.eu

[2] Introducing Risk Management into the Grid. K. Djemame, I. Gourlay, J. Padgett, G. Birkenheuer, M. Hovestadt, K. Voss and O. Kao. In Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam, The Netherlands, December 2006
[3] OpenCCS: Computing Center Software, 2008. http://www.openccs.eu
[4] Web Services Agreement Specification (WS-Agreement). A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Kakata, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu. OGF Document Series, 2007
[5] AssessGrid Second Software Prototype, Deliverable D3.2 - Broker Scenario – K. Voss, D. Battre, J. Padgett, I. Gourlay, I. Rosenberg, July 2008
[6] I. Gourlay, K. Djemame and J. Padgett. Reliability and Risk in Grid Resource Brokering. In Proceedings of the 2008 IEEE International Conference on Digital Ecosystems and Technologies, Phitsanulok, Thailand, February 2008