QoS-based Decision Services in Grids

Dimosthenis Kyriazis[1], Peer Hasselmeyer[2], Konstantinos Tserpes[1], Andreas Menychtas[1] and Theodora Varvarigou[1]

[1]Dept .of Electrical and Computer Engineering, National Technical University of Athens, 9 Heroon Polytechniou Street, 15773 Athens, Greece

E-mail: {dkyr, tserpes, a_menychtas, dora}@telecom.ntua.gr

[2]IT Research Division, NEC Laboratories Europe, NEC Europe Ltd, Sankt Augustin, Germany

E-mail: hasselmeyer@ccrl-nece.de

Abstract

Complex and mission-critical applications require the use of distributed environments, which will allow their execution in a reliable and cost-efficient way. Grid environments made this case feasible by providing heterogeneous resources. Nevertheless, Quality of Service (QoS) aspects are regarded as fundamental for enabling the execution of business applications in Grids. In that frame, we introduce decision services that establish QoS guarantees and fulfill the applications' requirement on becoming QoS-aware. A workflow mapping component is presented here that allows the mapping of workflow processes to Grid provided services, assuring at the same time end-to-end provision of QoS based on user-defined parameters and preferences; and a resource brokering service that reconciles the interests of providers and consumers based on Service Level Agreements (SLAs), which are negotiated between service providers and the resource broker. We also present a comparison of these decision services.

1. Introduction

Grid computing is increasingly considered as a next-generation infrastructure able to provide distributed and heterogeneous resources in order to deliver computational power to resource demanding applications in a transparent way [1]. However, the advent of Grids and their current maturity level raised issues regarding the QoS provision with the use of decision services. In that frame, we introduce two mechanisms that enable the service selection by exploiting QoS information at the same time. In the following paragraphs we start by introducing the terminology used within the first mechanism, named Workflow Mapping and we proceed further on for the second one, named Resource Broker.

Following the above, managing the workflow operations within Grid environments requires the orchestration of distributed resources [2], which can be achieved with application composition via workflows. A Workflow Model / Specification is used to define a workflow both at task and structure level as also presented in Figure 1.

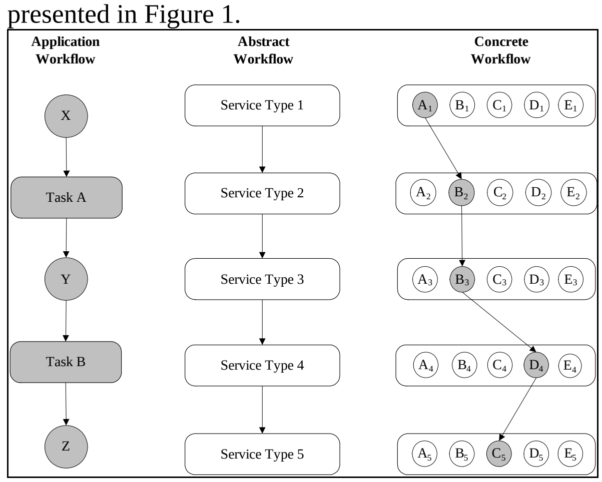

Figure 1. Workflow Definitions

Besides the application workflow, there are two types of workflows, namely *Abstract* and *Concrete* [3], [4]. In an abstract model, the tasks are described in an abstract form without referring to specific Grid resources for task execution. In the concrete model, the tasks of the workflow bind to specific resources and therefore this model provides service semantic and execution information on how the workflow has been composed both for the service instances and for the overall composition (e.g. dataflow bindings).

Following the workflow models' definitions, tasks in an abstract model are portable and can be mapped onto any suitable Grid services by using appropriate discovery and mapping mechanisms. To this end, we present two complimentary decision services which can create concrete workflows from abstract ones. The first service, called the "*Workflow Mapping Component*" selects providers based on QoS optimization considerations. The second decision service, called the "*Resource Broker*", selects providers based on guaranteed QoS levels using SLAs. The presented services receive as input parameters the ones published via the SLAs and therefore the complexity of the tasks doesn't affect the selection process. Potential change of the QoS parameters at runtime is not taking into account by the presented components since they run at the selection phase and not at the execution one.

The remainder of the paper is structured as follows. Section 2 presents related work in the fields of selection processes and decision services. Section 3 introduces the workflow mapping component while section 4 presents the resource broker. A comparison of the aforementioned components is included in section 5. Finally, section 6 concludes with a discussion on future research and potentials for the current study.

2. Related Work

There are various approaches for decision services that handle QoS information in order to arrive at a selection of service providers. Generally, the way QoS is perceived to work in current Grid architectures is as part of the SLA negotiation process [5]. To this direction, the Globus Architecture for Reservation and Allocation (GARA) [6] addresses QoS at the level of facilitating and providing basic mechanisms for QoS support, such as resource configuration, selection, etc.

Furthermore, there are traditional selection and scheduling methods, as in [7] but they only consider system performance, but they have neglected the user's grade of service demand. In [8] and [9] three algorithms are presented that use two parameters - cost and deadline time - in order to express quality of service dimensions. These parameters are used to implement a selection scheme, which refers to applications that consist of parametric processes that are independent of each other regarding their execution sequence. Literature [10] proposes some optimization scheduling algorithms under the limitation of time and cost in the Nimrod-G model, which supports user-defined deadline and budget constraints. DAGMan [11] extends the Condor Job Scheduler to handle "inter-job" dependencies and allows the user to schedule programs based on dependencies.

The difference between the systems describe above and the workflow mapping mechanism lies in the fact that the ones presented here address the case of selecting services and nodes based on QoS parameters by dealing only with the specific cases of minimizing one of the parameters whilst furthermore all the parameters have the same weight attribute. In our study, the QoS parameters are dealt with in a combined way as well, while we also introduce the case in which the user sets preferences and therefore one of the parameters may play a higher role in the selection process (a weight attribute is attached to it).

On the other hand, traditional Grid resource brokers work on a best-effort basis and do not usually offer the possibility of reserving resources in advance. The workload management system of the European DataGrid is an example of a broker [14], which bases scheduling decisions on different parameters such as queue length. The system described in [16] selects the "best" resource based on execution time history. The examples given in the WS-Agreement specification [15] already hint at the use of SLAs as a means of advance reservation (also in [12]). Nevertheless, SLAs have not been widely used for that purpose. An exception is the system described in [13] which uses WS-Agreement as resource reservation protocol.

The presented resource broker differs from the ones presented in this section in the following ways: reservations do not only cover job submission, information on resource state is not required, and prediction of execution time is not needed. The broker described in section 5 addresses the objectives of service providers and consumers by taking into account their constraints during the selection and SLA negotiation process.

3. Workflow Mapping Component

3.1. Use Cases based on User Preferences

The QoS parameters are taken into account for the definition of the concrete workflow and are prerequisites for the achievement of end-to-end QoS provisioning. In addition to hard limits on QoS properties, service consumers usually have a preference for optimizing a particular parameter. If an important deadline needs to be met, the consumer's preference will be on a shorter execution time than on a lower price. This preference might be opposite when operating under a tight budget. In general, such preferences can include multiple QoS parameters. Their relative importance can be expressed as weights on the parameters. Based on that the following use cases are identified: *Thresholds* to the values of the QoS parameters, *Optimization of a Specific Parameter* - by considering as major factor a specific QoS parameter (e.g. availability), and *Optimum Solution* - by setting equal importance of the parameters for the selection process.

3.2. Mechanism Overview

The objective of the presented mechanism is to identify and describe the process that needs to be implemented in order to define the concrete workflow given an application workflow and the essential QoS parameters. A workflow mapping mechanism is an integral part of the QoS provisioning, since this is the only way to estimate, calculate and conclude the mapping of workflows and the selection of the available service types and instances in order to deliver an overall quality of service across a federation of providers.

Given that each workflow contains service types that can be executed by a set of service instances (candidates), which are annotated with QoS information, the workflow mapping mechanism allows the selection of the service instances for each service type based on the application workflow, the user constraints and preferences and the QoS parameters for each service instance. The above information serves as input to the workflow mapping component that implements an algorithm presented in the following paragraph.

3.3. Algorithm description

The algorithm that implements the QoS-based workflow mapping starts by selecting the cheapest service instance per service type in order to check if it is feasible to conclude to a concrete workflow that meets the user's cost-constraint (if any). In sequel, the service instances are sorted with regard to their QoS parameters. This results to a sorted list with the "best offers" of service instances per service type. Based on this list replacements to the initial workflow may occur in order to achieve the optimum mapping. Afterwards, a scheduling scheme is applied to meet the user's time constraint or if a time optimization preference was stated. Following, we describe in brief the algorithm steps (a detailed description can be found in [17]):

-*Step 1:* Calculation of a value for each service instance that characterizes its QoS level based on the published QoS parameters in the SLAs.

-*Step 2:* Initial workflow mapping with regard to the user defined constraints / parameters and preferences.

-*Step 3:* Adoption of a service instance per service type and calculation of additional cost in order to increase the overall QoS level.

-*Step 4:* Creation of a list with the "best offers" for the workflow in order to optimize the initial selection.

-*Step 5:* Application of a scheduling scheme that allows more than one instances to be selected per service type so as to meet the user's time constraints.

4. Resource Brokering Service

4.1. Service Level Agreement Negotiation

In general, SLAs are used for codifying service properties in the form of terms. Terms typically describe the provided service and the QoS levels that have been agreed on. In this context, service consumers often need certain minimum QoS levels on service access and therefore they determine a set of non-functional QoS properties that are important to them and put them into SLAs established with their service providers. Since SLAs express the intent of future service access and can therefore be used to predict future resource utilization, service providers can use this data to plan ahead and dedicate the right amount of resources to the services used and calculate prices for the services offered.

Establishing an SLA involves a negotiation phase in which provider and consumer try to establish SLA terms that both can agree on. Both parties have their own objectives and priorities regarding SLA terms. Consumers pose specific QoS requirements (e.g. deadline) while service providers want to maximize revenues. To this direction, the negotiation phase is used to reconcile both parties' requirements and restrictions. The outcome of this phase is either an agreement (the SLA) or the termination of the negotiation. In the latter case, the consumer needs to find another provider that can fulfill its requirements.

4.2. Resource Brokering

The resource broker is designed to select a particular service instance from a set of candidates, just as the

QoS workflow mapping component does. In comparison to the QoS component it performs selection in a different way, though. The idea behind the broker was to develop a novel resource brokering and scheduling model that is based on SLA negotiation. The aim of the model is to schedule resource usage in a way that fulfils requirements and constraints of both service providers and consumers. The SLA guarantees the service consumer access to a service with certain quality of service while at the same time it helps service providers manage service provisioning and resource workloads. The SLAs considered here cover only short-term resource access and can therefore be considered advance resource reservations.

During the brokering process, the broker obtains offers from candidate service providers and finally selects the one with the best offer. The metric for “best" depends on the client of the broker and could be price, time, availability, etc. As the terms within SLA offers are the deciding factor for doing business with a particular provider, the broker’s scheduling decisions can be seen as being based on those SLAs.

The resource broker communicates with service providers via the SLA negotiation interface. As shown in Figure 2 that interface is provided by an agreement service running at the service provider’s site. A provider’s agreement service hides the details of the providers’ resources and their management. To make reasonable decisions it is of major importance to obtain information regarding the resources for which it is establishing SLAs for as well as their current state. Different providers will follow different business processes for resource pricing and decisions on accepting and rejecting new SLAs. The agreement service is therefore not only resource-specific; it is also provider-specific. Furthermore, our model replaces the necessity of publishing resource state information (contrary to existing resource brokers) by guarantee terms in SLAs [18].

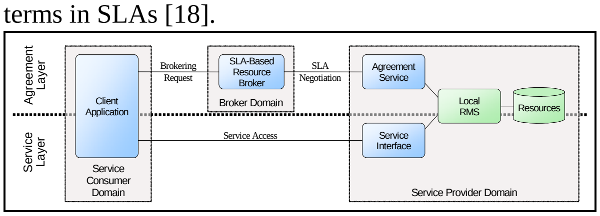

Figure 2. SLA-Based Resource Brokering

The broker’s SLA negotiation is triggered by the service consumer. As described for the workflow mapping component, consumers have certain needs, which are posed in terms of QoS parameters. These needs can be expressed in the form of requirements on SLA terms. The resource broker obtains these requirements and decides whether or to what degree a given SLA template fulfils them.

As shown in Figure 3, the consumer supplies a list of candidate service providers as well as a set of requirements on SLA terms to the resource broker. Each candidate service provider is denoted by an endpoint reference to the agreement service presenting the SLA negotiation interface to the broker. The requirements on SLA terms are used for selecting and prioritizing SLA templates.

For mapping an abstract workflow to a concrete one, the broker is used repeatedly, once for each abstract workflow step. The broker successively replaces abstract service types with concrete service instances until the whole workflow has been made concrete. In the process, all service instances are guaranteed by SLAs which in turn establish certain guarantees for the overall workflow.

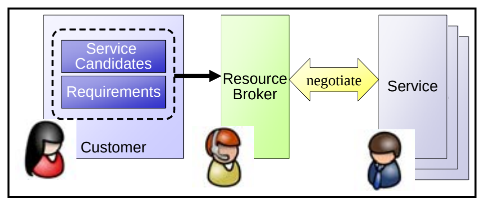

Figure 3. Customer/broker data exchange

5. Comparison

As decision services, both the workflow mapping component as well as the resource broker perform the same function: select a particular service from a set of candidates. The first difference between the two components is that the mapping component initially selects a set of candidates and afterwards performs various optimizations on this selection in order to fulfill the user’s QoS expectations. The broker, on the other hand, bases selection decisions on a service’s ability to present an acceptable SLA.

When looking at the input and output data of the two components their similarity can be seen directly. Both need a set of service candidates and the service consumer’s QoS requirements. The output is then the chosen service instance, selected based on whatever metric the respective component uses. Since the components use different selection criteria, different sets of information are needed for performing the selection: the mapping component needs QoS information on service candidates while the broker needs SLA templates.

Finally, the scope the components operate on is different. The mapping component works on whole workflows while the broker can only handle single workflow tasks. Although the broker can iterate over all tasks in a workflow and negotiate SLAs for all tasks, the result would still be different from what the mapping component does. The broker would still look at the task level only, while the mapping component performs optimization of the workflow both on the task level as well as on the overall workflow level.

6. Conclusions

In this paper we have introduced two QoS-based decision services, the workflow mapping component and the service-oriented Grid resource broker. Notwithstanding, it is within our future plans to attempt an orchestration to simultaneously use both presented components since this is expected to be of benefit. The differences of the two components highlight their complementary nature. They can be orchestrated by having the broker provide SLA-related information (e.g. SLA terms) to the mapping component in order to decide on how to map workflow processes to service instances taking into account this information as well.

Concluding, unlike other heterogeneous systems [34], Grids have not yet adopted effective schemes that will facilitate end-to-end QoS provisioning. In that rationale, we have presented two QoS-based decision services.

7. References

[1] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations” International Journal Supercomputer Applications, Vol. 15, No. 3, 2001.

[2] A. Mayer, S. McGough, N. Furmento, W. Lee, M. Gulamali, S. Newhouse, and J. Darlington, “Workflow Expression: Comparison of Spatial and Temporal Approaches”, Workflow in Grid Systems Workshop, GGF-10, Berlin, March 9, 2004.

[3] E. Deelman, J. Blythe, Y. Gil, and C. Kesselman, “Workflow Management in GriPhyN”, Kluwer, Netherlands, 2003.

[4] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M. H. Su, K. Vahi, M. Livny, “Pegasus: Mapping Scientific Workflow onto the Grid”, Across Grids Conference 2004, Nicosia, Cyprus, 2004.

[5] R. J. Al-Ali, K. Amin, G. von Laszewski, O. F. Rana, D. W. Walker, M. Hategan, N. J. Zaluzec: “Analysis and Provision of QoS for Distributed Grid Applications”, pp. 163-182, Journal of Grid Computing, 2004

[6] I. Foster, C. Kesselman, C. Lee, B Lindell, K. Nahrstedt, A. Roy, “A Distributed Resource Management Architecture that Supports Advance Reservation and Co-Allocation”, Proceedings of the International Workshop on QoS, 1999

[7] Shanshan Song, Yu-Kwong Kwok, Kai Hwang, “Security-Driven Heuristics and A Fast Genetic Algorithm for Trusted Grid Job Scheduling Parallel and Distributed Processing Symposium”, 19th IEEE International 04-08 April 2005

[8] R. Buyya, M. Murshed, D. Abramson, “A Deadline and Budget Constrained Cost-Time Optimization Algorithm for Scheduling Task Farming Applications on Global Grids”, Proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications(PDPTA702), 2002

[9] R. Buyya, D. Abramson, S. Venugopal, “The Grid Economy”, Proceedings of the IEEE Volume 93, Issue 3, pp. 698-714, Mar 2005

[10] D. Angulo, I. Foster, C. Liu, and L. Yang, "Design and Evaluation of a Resource Selection Framework for Grid Applications", Proceedings of IEEE International Symposium on High Performance Distributed Computing (HPDC-11), Edinburgh, Scotland, 2002

[11] DAGMan, http://www.cs.wisc.edu/condor/dagman/

[12] Karl Czajkowski, Ian Foster, Carl Kesselman, Volker Sander, and Steven Tuecke. “SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems”,Job Scheduling Strategies for Parallel Processing, Springer Verlag, 2002.

[13] Erik Elmroth and Johan Tordsson. “An Interoperable, Standards-based Grid Resource Broker and Job Submission Service”. In: Proceedings of the First International Conference on e-Science and Grid Computing (e-Science '05), Melbourne, Australia, December 2005.

[14] G. Avellino et al. “The EU DataGrid Workload Management System: towards the second major release”,.Proceedings of the International Conference on Computing in High Energy and Nuclear Physics, San Diego, CA, March 2003.

[15] Open Grid Forum. “Web Services Agreement Specification (WS-Agreement)”, March 2007 http://www.ogf.org/documents/GFD.107.pdf.

[16] Hui Li, David Groep, Je Templon, and Lex Wolters. “Predicting Job Start Times on Clusters”. In: Proceedings of the IEEE International Symposium on Cluster Computing and the Grid, Chicago, IL, 2004.

[17] Dimosthenis Kyriazis, Konstantinos Tserpes, Andreas Menychtas, Antonis Litke and Theodora Varvarigou, “An innovative workflow mapping mechanism for Grids in the frame of Quality of Service”, Future Generation Computer Systems, Elsevier, 2008

[18] Peer Hasselmeyer, “Removing the Need for State Dissemination in Grid Resource Brokering”. In: 5th International Workshop on Middleware for Grid Computing (MGC 2007), Newport Beach, CA, USA, November 2007.