A new Workflow Mapping Mechanism for Grids

Dimosthenis Kyriazis, Konstantinos Dolkas, Andreas Menychtas and Theodora Varvarigou

Dept .of Electrical and Computer Engineering, National Technical University of Athens,

9, Heroon Polytechniou Str, 15773 Athens, Greece

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

***Abstract*— The beneficial perspective of heterogeneous and distributed environments such as Grid environments lies on solving computational intensive problems in a reliable and cost-efficient way. In that frame, and considering the business needs for Quality of Service we present a novel Workflow Mapping Mechanism for Grids. This mechanism is able to assure end- to-end provision of QoS over Grid infrastructures by providing utilization and management of Grid services. The simulation process showed promising results and in combination with the end-to-end QoS aspect it helps towards making the adoption of different business models feasible on Grids. This approach is part of the ongoing research effort in the framework of NextGRID IST Project.**

***Index Terms*— Business Grids, Quality of Service, Workflow Management**

I. Introduction

GRID computing is increasingly considered as the next phase of distributed computing. Built on pervasive Internet standards, grid computing enables organizations to share computing and information resources across department and organizational boundaries in a secure and highly efficient manner. Grids support the sharing, interconnection and use of diverse resources, integrated in the framework of a dynamic computing system capable to deliver computational power to resource demanding applications in a transparent way [1], [2].

In that frame, workflow is an architecturally important factor for dynamic interoperability and adaptation to different business models and deployment contexts. A Workflow Mapping Mechanism is an integral part of the provision of Quality of Service (QoS), and especially the end-to-end Quality of Service, since this is the only way to estimate, calculate and conclude to 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. A mechanism that has been studied and evaluated through simulations is described in the following paragraphs.

Supporting end-to-end Quality of Service with workflow mapping mechanisms requires the pursuit of a number of QoS aspects that need to be addressed. In order to tackle each one of them in this paper, we will start from the examination of the requirements and parameters that need to be fulfilled in order to enable the dynamic mapping of workflows for different business processes. Furthermore, this set of parameters are categorized based on their source (for example there can be application parameters or user-defined parameters) and consist as input for the workflow mapping mechanism. For each service instance, the aforementioned QoS parameters are taken into account within the mapping mechanisms. The major objective of the work presented in our paper is to demonstrate how the provision of end-to-end QoS can be implemented and to indicate which workflow mapping mechanism is the one that should be employed in order to achieve the latter.

The general concept of workflow and the naming that will be used further on in this paper for the workflow schemas is addressed in §II, while in §III the workflow mapping mechanism is introduced as a module containing the appropriate algorithm that is needed in order to embed QoS information into workflows. In §IV the parameters that will be used - and are considered as initial QoS parameters - are presented. In the current approach Availability and Cost are taken into account as sample parameters, however our approach can be generalized in order to include more parameters. Use Cases analyzed based on user parameters / preferences are presented thereafter (§V). The research topic and the focus of our work is included in §VI where the algorithm used for the novel workflow mapping is described. Finally §VII briefly presents our conclusions and summarizes the future work of this study.

II. Workflow Definitions

Since workflow is a wide concept and technology, in the following paragraphs workflow definitions are stated both in general and in the terminology that is being used afterwards in this document. Regarding the general definition, Workflow Management Coalition provides the following definition [3]:

Workflow is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.

A workflow can be defined as the orchestration of a set of activities to accomplish a complicated goal, while in Grid environments; workflow includes application processes, business processes, and infrastructure processes.

Workflow is an architecturally important factor for dynamic interoperability and adaptation to different business models, which can be addressed as workflow policies, and deployment contexts. The final outcome of the workflow mapping mechanism will be the Concrete Workflow, which is a composition of services providing 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, control flow structures). The concrete workflow can be seen as a "path", a selection of service instances from service types on specific order so as to achieve the goal of implementing an application workflow.
Mainly, the aim of our research is to identify and describe the process that needs to be implemented in order to define the concrete workflow given a typical application workflow and the essential QoS parameters. As mentioned above, the concrete workflow is "constructed" from an application workflow. An intermediate step is the "construction" of an Abstract Workflow, which is a composition of services providing only service semantic information on how the workflow has been composed. On the following figure (Fig. 1) the aforementioned workflow definitions are presented:

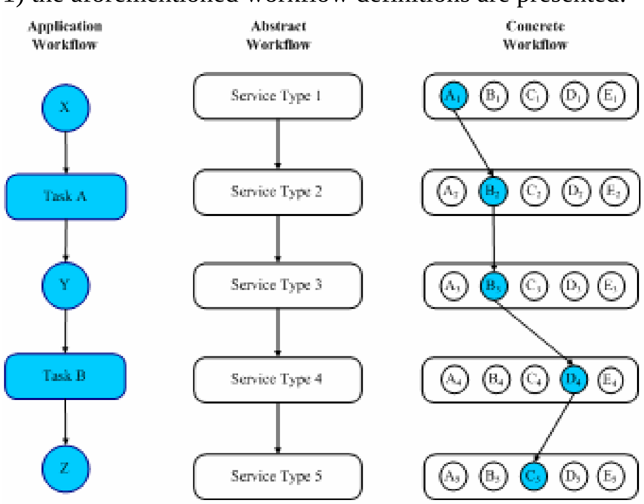

Fig. 1: Workflow Definitions

III. Workflow Mapping Mechanism Overview

Aiming on assuring end-to-end provision of QoS with the use of workflow mapping mechanisms, there are various approaches on how to handle the QoS information. The Globus Architecture for Reservation and Allocation (GARA) [4] addresses QoS at the level of facilitating and providing basic mechanisms for QoS support, namely resource configuration, discovery, selection, and allocation.

The main goal of the presented workflow implementation is to allow application developers to enhance business processes into workflow as procedures and in combination with QoS parameters to enable dynamic interoperability and adoption of different business models within Grid infrastructures. A business process example can be the selection service, which should be integrated into a workflow to allow its execution at specific service instances.

The workflow mapping mechanism that is introduced and described in this paper consists of two (2) major steps, which are also represented in Fig. 2:

1. Importance Sorting of the services which "produce" the Abstract Workflow. It is important to know which of the services play a higher role and this factor should be encompassed into the application hints provided by the application developers (e.g. which service is of higher importance: movement of large amount of data or execution time). The final result of this step is the construction of the Abstract Workflow, with the list of available service instances per service type. This list is a subgroup of all available services based on the pre-completed negotiation process and the SLAs.

2. Execution of the algorithm / mechanism (described in §VI of this paper) with use of the QoS parameters and the abstract workflow outcome of the previous step and as a result definition of the Concrete Workflow.

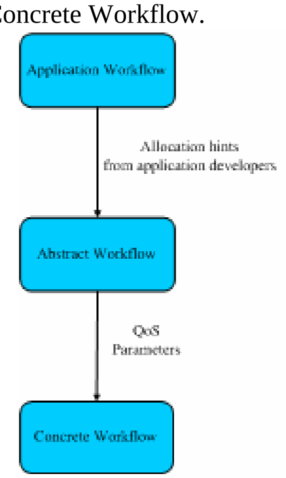

Fig. 2: Workflow Mapping Mechanism Overview

IV. Initial Quality of Service Parameters

As stated in the workflow mapping mechanism overview (§III), QoS parameters are taken into account for the definition of the Concrete Workflow and are prerequisites for the achievement of end-to-end QoS Workflow. In this section of our paper, the parameters that have been used in our study are presented.

There are numerous parameters that are of particular interest in grid architectures and the work that should be performed on defining the QoS parameters and their role in the workflow mapping mechanism. Our work has been focused initially on Availability, expressed as numerical percentage for each service instance and Cost, expressed as numerical account units for each service instance and these parameters are presented as input on the algorithm in §VI. Nevertheless, these parameters are selected as representative indicative parameters for the overall workflow process and the set of the parameters can be extended in order to include more than two (2).

In the following paragraphs, a classification of the QoS parameters is stated as a direct consequence of the logical categorization of the parameters, which means that parameters under the same category are sharing common properties. Currently, this work has concluded to the classification of QoS parameters in three (3) major categories:

- User-defined Parameters, which relate to requirements / constraints that the user who initiates the workflow would like to pose, such as cost restrictions (e.g. maximum overall cost).
- Application Parameters, which relate to the offered QoS from the application perspective. For example, the application configuration could play a significant role to the availability of the task to be executed.
- Resource Parameters, which relate to all types of resources, including computational, storage and network resources. For example, from that perspective the network infrastructure can be regarded as a set of interacting resources that are offering a specific QoS level.

The abovementioned parameters and their classification are shown in Fig. 3:

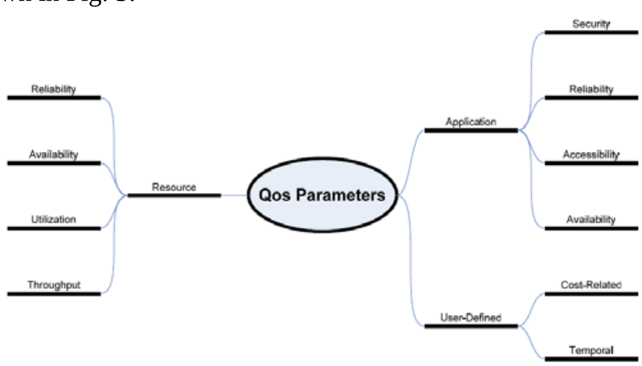

Fig. 3: QoS Parameters Categorization

There has to be mentioned that the initiation of a workflow may involve the invocation of a service that in turn will trigger the employment of a resource, and in this case the produced SLA may include QoS parameters information related to the resource such as the memory status. These parameters should also be taken into account from the workflow mapping mechanisms.

V. USE CASES BASED ON USER PREFERENCES

Besides the parameters and their classification, the users who initiate the workflow specify their preferences / constraints which also play a significant role for the definition of the concrete workflow. These constraints imply the user preference that can be expressed or seen as a "weight" factor on the QoS parameters (the latter is named as slope in the following paragraphs of this paper). As mentioned above, for the purposes of this study Availability and Cost are initial indicative parameters and based on that the following use cases are presented:

- "Cheapest". In this case the service instance per service type with the lower Cost Value is selected.
- "Most" Available. In this case the service instance per service type with the higher Availability Value is selected.
- Constraint / Threshold to Cost. In this case the service instances with "optimum" availability related to the cost but without exceeding the overall constraint / threshold of the cost are selected. Major factor is the cost which is represented with a higher "weight" (expressed as Cost Slope in the selection algorithm description – §VI).
- At least % of Availability. In this case the service instances with at least a value of availability related to the cost but without selecting service instances that do not match the availability constraint are selected. Major factor is the availability which is represented with a higher "weight" (expressed as Availability Slope in the selection algorithm description – §VI).
- Constraint / Threshold to Cost and at least % of Availability. In this case the user preferences include constraints both for the Cost and for the Availability. The weights for the two (2) parameters are equal and the algorithm execution results to a concrete workflow where the threshold to the cost will most probably be reached in order to achieve the highest end-to-end availability, which will of course match at least the availability constraint.
- Optimum Solution. In this case the weights for the two (2) parameters are equal and the algorithm execution results to a concrete workflow where the service instances selected offer the optimum value of availability for the corresponding cost.

VI. ALGORITHM DESCRIPTION

In the following paragraphs of the specific section the algorithm that is used for the workflow mapping and the selection of service instances is being described in detail. Even that the problem of service instances selection from one or more available service types is a problem which does not have a unique solution, the optimum selection is attempted to be achieved with the use of the following algorithm. This algorithm has been produced after various experiments and should not be regarded as a unique solution to the problem.

The basic steps of the algorithm are named and described firstly while a detailed explanation for each step follows:

- Step 1: Calculation of auxiliary values that are needed for the algorithm completion (for example the value alternant Cost / Availability).
- Step 2: Initial workflow mapping with integration and use of the user defined constraints / parameters (as described in the Use Cases of §V) for the service types.
- Step 3: Discovery for each service type the most "valuable" choice of service instance and calculation of

additional Cost in order to increase Availability. The latter can be seen as an optimization step in order to achieve better levels of availability while the user constraints and parameters are not violated.

- Step 4: Discovery of the optimum replacement(s) for the final workflow mapping with identification of the most "valuable" replacement of service instances (if any) and recalculation of the overall Cost and Availability.

A. Step 1: Calculation of auxiliary values

1. Based on the user preferences / requirements the slope of the conversion functions for Cost and Availability is defined.

2. The aforementioned is looped for each service type individually.

3. Calculation of the minimum and maximum Cost and Availability values for the specific service type among the service instances of it.

4. Calculation of the conversion values for Cost and Availability based on the minimum Cost and Availability values that have been calculated above with the use of the following functions:

$$F_{Cost}(x) = 1 + \left(\frac{x - MinimumCostValue}{MaximumCostValue} \right)^2 * CostSlope$$

$$F_{Availability}(x) = 1 + \left(\frac{x - MinimumAvailabilityValue}{MaximumAvailabilityValue} \right)^2 * AvailabilitySlope$$

These conversion functions are non-linear with positive slope. They enact the influence of the change in the Cost and Availability values to the user defined requirements / parameters. The above functions came as a result of our study of various experiments and simulations.

5. Calculation of converted Cost and Availability Values for each service instance as following:

$$NewCostValue = InitialCostValue * F_{Cost}(InitialCostValue)$$

$$NewAvailabilityValue = InitialAvailabilityValue * F_{Availability}(InitialAvailabilityValue)$$

6. Calculation of the following Converted Index that will be used in sequel:

$$ConvertedIndex = \frac{NewCostValue}{NewAvailbilityValue}$$

The slopes factors in the aforementioned equations enact how important Cost and Availability are considered by the user and may get values $0 < slope < 1$ for both Cost and Availability. The higher the slope is, the more important the value is for the user.

Furthermore:

$$AvailabilitySlope = 1 - CostSlope$$

Finally, there has to be mentioned that the service instances that do not match the user defined requirements are not taken into account in the described algorithm.

B. Step 2: Initial workflow mapping

1. The initial workflow mapping is done with regard to the user defined parameters / requirements for both Cost and Availability

2. For each service type, the service instance that matches the minimum required Availability with the minimum Cost is selected

3. If the overall Cost Value calculated for all service instances selected (from all service types) exceeds the end-to-end user defined Cost parameter, the workflow mapping ends. Otherwise it continues with the following step.

C. Step 3: Discovery for each service type the most "valuable" choice of service instance

1. For each service type, the service instances that are more "valuable" (they are considered more "valuable" based on the converted index Cost / Availability) than the one selected are aggregated.

2. If for a service type there is not any service instance that can be considered as more "valuable" from the one selected, the specific service instance is excluded from the algorithm execution.

3. If for all service types there is not any service instance that can be considered as more "valuable" from the one selected, the algorithm ends and the initial workflow mapping is considered the final one.

4. The algorithm continues with the selection of the service types for each service instance that have the lower value of the converted index Cost / Availability.

5. For each new selection, the Cost and Availability values difference between the initial service instance selection and the new one is calculated.

6. These differences are aggregated in order to conclude to the service instances that can be considered as the most "valuable" based on the user defined requirements / parameters.

D. Step 4: Discovery of the optimum replacement(s) for the final workflow mapping

1. For each difference that has been calculated above, the converted index Cost / Availability is calculated. Basically, Step 1 of the algorithm is re-executed considering as values for the service instances the aforementioned differences.

2. The service instances with the lower converted index are selected.

3. Based on the new selection of service instances, the overall Cost for all service types is calculated.

4. If the Cost user defined constrained / parameter is matched, the replacement of the new selection takes place; otherwise the new selection is excluded from the available

service instances.

5. The algorithm is looped and continues from Step 3 for all service types of the Abstract Workflow.

VII. Conclusion

In this paper we presented a mechanism for workflow mapping related to the QoS information which would significantly increase the effort to provide the Grid environment with a dynamic QoS capability. The aim is to define a concrete workflow based on various parameters and to achieve the optimum selection of service instances that will allow end-to-end provision of Quality of Service.

The results of the experiments performed were encouraging and besides the identification of a possible selection service, the required set of parameters (application, resource, user-defined) and their integration into the mechanism was revealed. Of course, the simulation will be continued in order to obtain further knowledge and to improve the algorithm. However, we now have a good basis to start building on dynamic QoS service selection issues.

Future work in this task will include the attempt to comprise a set of parameters in the selection process, while other pertinent issues such as the integration of the QoS information to dynamic SLA negotiations will be examined. These steps are regarded as significantly important for the dynamic end-to-end provision of QoS.

Furthermore, the overall mechanism can be improved by including the dependencies between service instances (for example level on parameter values – cost). We expect soon to be in the state to explain how this is working in real-life, business environments since this work is performed within the frame of the NextGRID IST Project.

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] W. Leinberger, V. Kumar, “Information Power Grid: The new frontier in parallel computing?” IEEE Concur., Vol. 7, No. 4, pp. 75-84, October-December 1999.

[3] Workflow Management Coalition, Terminology & Glossary, Document Number WFMC-TC-1011, Issues 3.0, February 1999.

[4] 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, pp.27-36, 1999.