Distributed and Scalable QoS Optimization for Dynamic Web Service Composition

Mohammad Alrifai

L3S Research Center
Leibniz University of Hannover, Germany
alrifai@L3S.de

Supervised by: Prof. Dr. tech. Wolfgang Nejdl
L3S Research Center
Leibniz University of Hannover, Germany
nejdl@L3S.de

Abstract. Web service composition requests are usually combined with end-to-end QoS requirements, which are specified in terms of non-functional properties (e.g. response time, throughput and price). The goal of QoS-aware service composition is to select the best combination of services that meet these end-to-end requirements, while maximizing the value of a pre-defined utility function. This problem can be modeled as a multi-dimension multi-choice 0-1 knapsack problem, which is known as NP-hard in the strong sense. Existing solutions that rely on general purpose solvers suffer from poor performance, which render them inappropriate for applications with dynamic and real-time requirements. Moreover, global optimization techniques assume a centralized system model, which contradicts with the distributed and loosely-coupled environment of web services. The aim of this thesis is to develop scalable QoS optimization solutions that fit better to the distributed environment of web services. The idea is to decompose global constraints into local constraints that have to be fulfilled by a set of distributed service brokers. A solution that combines global optimization and local selection techniques is proposed.

1 Introduction

The service-oriented computing paradigm and its realization through standardized Web service technologies provide a promising solution for the seamless integration of business applications to create new value-added services. Industrial practice witnesses a growing interest in this ad-hoc service composition. With the growing number of alternative web services that provide the same functionality but differ in quality parameters, the composition problem becomes a decision problem on the selection of component services with regards to functional and non-functional requirements. In this work, we look at the non-functional requirements, namely quality of service parameters in composing web services.

1.1 Motivating Scenario

Consider for example the personalized multimedia delivery scenario in Figure 1. A PDA user requests the latest news from a service provider. Available multimedia content includes a news ticker and topical videos in MPEG 2 only. The following services are are required to serve the user's request: a transcoding service for the multimedia content to fit the target format, a compression service to adapt the content to the wireless link, a text translation service for the ticker, and also a merging service to integrate the ticker with the video stream. The user request can be associated with some end-to-end QoS requirements (like bandwidth, latency and price). The service composer has to ensure that the aggregated QoS values of the selected services match the user requirements. Dynamic changes due to changes in the QoS requirements (e.g. the user switched to a network with lower bandwidth) or failure of some services (e.g. some of the selected services become unavailable) can occur at run-time. Therefore, a quick response to adaptation requests is important in such applications.

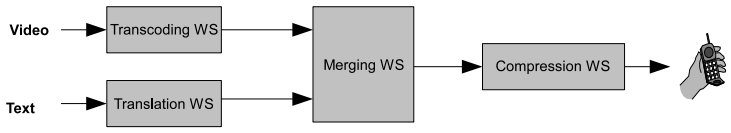

Fig. 1. Composition of Multimedia Web Services

1.2 Local vs. Global QoS Optimization

Two general approaches exists for the QoS-driven service composition: *local* optimization and *global* optimization. In the local optimization approach, one service is selected from each service class independently based on its local utility value. This approach is very efficient as the time complexity of the local optimization approach is linear with respect to the number of service candidates. However, local optimization cannot satisfy end-to-end QoS requirements (like maximum total response time). On the other hand, the global optimization approach aims at solving the problem on the composite service level. This approach seeks the service composition, which maximizes the overall utility value, while guaranteeing global constraints. The global optimization problem can be modeled as *Multi-Choice Multidimensional Knapsack problem* (MMKP), which is known to be NP-hard in the strong sense [1]. Therefore it can be expected that any exact algorithm that solves MMKP has an exponential effort [2], which is not suitable for real time applications. Moreover, global optimization approaches rely on centralized computation, which is not feasible for the distributed and dynamic environment of web services.

1.3 Expected Contribution

The aim of this PhD thesis is to address the performance and scalability issues of QoS aware service selection by applying a scalable distributed heuristic that

combines global and local optimization techniques. The approach contributes the following results to the state of the art:

1. Decomposition of global QoS constraints into local constraints is modeled as a mixed integer program. The size of the resulting program is independent on the number of service candidates and hence can be solved more efficiently than existing MIP-based solutions.
2. Selection of component services is solved using guided local optimization. The local optimization is performed for each service class independently and in parallel to further improve the performance.
3. Extensive evaluation of the approach by comparing its performance and quality with existing exact and heuristic solutions. The evaluation will be done by simulation as well as in a large real world environment, e.g. PlanetLab.

2 Related Work

The QoS-based web service selection and composition in service-oriented applications has recently gained the attention of many researchers [3–6]. In [4] the authors propose an extensible QoS computation model that supports open and fair management of QoS data. The work of Zeng at al. [3] focuses on dynamic and quality-driven selection of services. The authors use global planning to find the best service components for the composition. They use linear programming techniques to find the optimal selection of component services. Similar to this approach Ardagna et al. [5] extend the linear programming model to include local constraints. Linear programming methods are very effective when the size of the problem is small. However, these methods suffer from poor scalability due to the exponential time complexity of the applied search algorithms [7, 2]. In [6] the authors propose heuristic algorithms to find a near-to-optimal solution more efficiently than exact solutions. The time complexity of the heuristic algorithm for the combinatorial model is polynomial, and exponential for the graph model. Despite the significant improvement of these algorithms compared to exact solutions, both algorithms do not scale with respect to the number of web services and remain out of the real-time requirements.

3 A Distributed Approach for Web Service QoS Optimization

We divide the QoS-aware service composition problem into two sub-problems that can be solved more efficiently in two subsequent phases. In the first phase, we use global optimization techniques to find the best decomposition of global QoS constraints into local constraints on the component service level. In the second phase, we use local selection to find the best component services that satisfy the local constraints from the first phase.

We assume an architecture consisting of a *service composer* and a number of *service brokers*. The service composer instantiates a composite service in collaboration with the service brokers. Each service broker is responsible for managing

QoS information of a set of web service classes. A list of available web services is maintained by the service broker along with registered measurements of their non-functional properties, i.e. QoS attributes, like response time, throughput, price etc. For the sake of simplicity we assume in this paper that each service class is maintained by one service broker. The two phases of our approach are described in the next subsections in more details.

3.1 Decomposition of Global QoS Constraints

In order to avoid discarding any service candidates that might be part of a feasible composition, the decomposition algorithm needs to ensure that the local constraints are relaxed as much as possible while meeting global constraints. To solve this problem, we divide the quality range of each QoS attribute into a set of discrete quality values, which we call "*quality levels*". We then use mixed integer programming (MIP) to find the best combination of these quality levels for using them as local constraints. The size of our MIP model is much smaller than the size of the MIP model in [3, 5] as the number of decision variables in our case is much smaller than the number of variables in their model. Therefore, our MIP model can be solved much faster.

Quality Levels: In this paper, we use a simple method for constructing the quality levels. For each service class S_i, we divide the quality range of each of the m QoS attributes into d quality levels: $q_{ik}^1, ..., q_{ik}^d$, $1 \leq k \leq m$ as follows:

$$q_{ik}^z = \begin{cases} Qmin(i,k) & \text{if } z = 1 \\ q_{ik}^{z-1} + \frac{Qmax(i,k) - Qmin(i,k)}{d} & \text{if } 1 < z < d \\ Qmax(i,k) & \text{if } z = d \end{cases} \quad (1)$$

where $Qmin(i,k)$ and $Qmax(i,k)$ are the local minimum and maximum values, respectively, for the kth attribute of the service class S_j. We then assign each quality level q_{ik}^z a value between 0 and 1, which indicates the probability p_{ik}^z that using this quality level as a local constraint would lead to finding a solution. The probability p_{ik}^z for the zth level of q_k at S_i is computed as follows:

$$p_{ik}^z = h/l \quad (2)$$

where h is the number of service candidates satisfying q_{ik}^z and l is the total number of service candidates at S_i.

MIP Formulation: The goal of our MIP model is to find the best decomposition of QoS constraints into local constraints. Therefore, we use a binary decision variable x_{ik}^z for each local quality level q_{ik}^z such that $x_{ik}^z = 1$ if q_{ik}^z is selected as a local constraint for the QoS attribute q_k at the service class S_i, and $x_{ik}^z = 0$ otherwise. To ensure that only one quality level is selected from the set

of d levels of the QoS attribute q_k at the service class S_i, we add the following set of constraints to the model:

$$\sum_{z=1}^{d} x_{ik}^{z} = 1, \forall i, \forall k, 1 \leq i \leq n, 1 \leq k \leq m$$

where n is the number of service classes and m is the number of QoS constraints. Note that the total number of variables in the model equals to $n * m * d$, i.e. independent on the number of service candidates per class l. By ensuring that the number of quality levels d is small enough such that $m * d \leq l$ we can ensure that the size of our MIP model is smaller than the size of the model used in [3, 5]. The selection of the local constraints must ensure that global constraints are still satisfied. Therefore, we add the following set of constraints to the model:

$$\sum_{i=1}^{n} \sum_{z=1}^{d} q_{ik}^{z} * x_{ik}^{z} \leq c'_k, \forall k, 1 \leq k \leq m$$

The objective function of our MIP model is to maximize the probability that the selected local constraints will lead to finding a feasible composition. Therefore, using (2) the objective function can be expressed as follows:

$$\text{maximize} \prod_{i=1}^{n} \prod_{k=1}^{m} p_{ik}^{z} \ , 1 \leq z \leq d \tag{3}$$

We use the logarithmic function to linearize (3) in order to be able to use it in the MIP model:

$$\text{maximize} \sum_{i=1}^{n} \sum_{k=1}^{m} \sum_{z=1}^{d} ln(p_{ik}^{z}) * x_{ik}^{z} \tag{4}$$

By solving this model using any MIP solver methods, we get a set of local quality levels that we use in the second phase for guiding local search.

3.2 Local Search

We use the local constraints obtain from the first phase as upper bounds for the QoS values of component services. Web services that violate these local constraints are skipped from the search. We then sort the qualified services by their utility values and select the top service from each class. In order to evaluate the multi-dimensional quality of a given web service we use a Multiple Attribute Decision Making approach: i.e. the *Simple Additive Weighting (SAW)* technique [8] to compute the utility value of the service. The utility computation involves scaling the values of QoS attributes to allow a uniform measurement of the multi-dimensional service qualities independent of their units and ranges. We compute the distance between the quality value $q_k(s_{ji})$ of a given service candidate s_{ji} and the maximum value $Qmax(j, k)$ in its class S_j and compare

it with the distance between the maximum and minimum overall quality values that can be obtained by any composition: $Qmax'(k) = \sum_{j=1}^{n} Qmax(j,k)$, $Qmin'(k) = \sum_{j=1}^{n} Qmin(j,k)$. This scaling method ensures that the evaluation of service candidates is globally valid, which is important for guiding local search in order to avoid local optimums. The scaling process is then followed by a weighting process for representing user priorities and preferences.

We compute the utility $U(s_{ji})$ of the i-th service candidate in class S_j as:

$$U(s_{ji}) = \sum_{k=1}^{r} \frac{Qmax(j,k) - q_k(s_{ji})}{Qmax'(k) - Qmin'(k)} \cdot w_k \tag{5}$$

with $w_k \in \mathbb{R}_0^+$ and $\sum_{k=1}^{r} w_k = 1$ being the weight of q_k to represent user's priorities.

4 Conclusion and Future Work

This PhD thesis is aimed at the development of distributed and scalable solutions to the global QoS optimization problem for web service compositions. Unlike existing approaches that model the problem as a conventional combinatorial optimization problem, we model the problem as a distributed optimization problem by exploiting the special characteristics and structure of the web service environment. Current results of this work indicate a very promising improvement over existing solutions. The next steps of this work include extending the existing model to support different styles of web service compositions and QoS constraints. We also plan to evaluate the performance of our approach against existing exact and approximate solutions by extensive simulations as well as in a large real world environment, e.g. PlanetLab.

References

1. Pisinger, D.: Algorithms for Knapsack Problems. PhD thesis, University of Copenhagen, Dept. of Computer Science (1995)
2. Parra-Hernandez, R., Dimopoulos, N.J.: A new heuristic for solving the multichoice multidimensional knapsack problem. IEEE Transactions on Systems, Man, and Cybernetics, Part A **35**(5) (2005) 708–717
3. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: WWW. (2003) 411–421
4. Liu, Y., Ngu, A.H.H., Zeng, L.: Qos computation and policing in dynamic web service selection. In: WWW. (2004) 66–73
5. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. IEEE Trans. Software Eng. **33**(6) (2007) 369–384
6. Yu, T., Zhang, Y., Lin, K.J.: Efficient algorithms for web services selection with end-to-end qos constraints. TWEB **1**(1) (2007)
7. Maros, I.: Computational Techniques of the Simplex Method. Springer (2003)
8. Yoon, K..P., Hwang, C.L.: Multiple Attribute Decision Making: An Introduction (Quantitative Applications in the Social Sciences. Sage Publications (1995)