The Third International Conference on Internet and Web Applications and Services

Worst-Case Workflow Performance Optimization

Julian Eckert, Stefan Schulte, Michael Niemann, Nicolas Repp and Ralf Steinmetz
Multimedia Communications Lab (KOM)
Department of Electrical Engineering and Information Technology,
Technische Universität Darmstadt, Germany
e-mail: julian.eckert@kom.tu-darmstadt.de

Abstract— **Performance evaluation and execution management of service-oriented workflows became quite important in order to avoid performance degradation. Performance measurement is crucial to ensure that workflow execution remains feasible and that SLA violations due to overload are avoided. Network calculus as a well-known system theory for deterministic queuing systems can be used to describe the worst-case performance behavior of a workflow in order to plan workflow control in advance.**

Concerning business processes with high repetition rates the workflow controller has to be able to serve all incoming requests with an optimal composition of Web services. Thus, this paper presents a formal worst-case calculation model using the concepts of network calculus. Furthermore, optimization problems based on the worst-case scenario are introduced in order to minimize the worst-case delay and to maximize the throughput of the Web services invoked with minimal costs.

Index Terms—**Performance Optimization, Quality of Service, Service-oriented Architecture, Network Calculus, Web service Workflow.**

I. Introduction

Competitive markets and the deregulation of markets forced enterprises to build cost-efficient business processes which must be state of the art from both a technical as well as from a business perspective. Hence, the performance of a business process at runtime has become a major issue regarding its competitiveness. Thus, enterprises require continuous business process management that supports business process intelligence, which facilitates the flexible composition of a business process consisting of several services.

Nowadays, enterprises are faced with an IT architecture consisting of large amounts of heterogeneous legacy systems, middleware platforms, programming languages, operating systems, and communication channels, which are barely manageable [18]. Quality of Service (QoS) and costs have become a major issue concerning an effective business process management which also meets customer expectations [2]. Thus, enterprises have to plan their business processes in advance to be able to adapt them to changing business needs.

Concerning cross-organizational workflows in which multiple parties and even external partners are involved, flexible business processes can be achieved by integrating internal legacy systems, as well as by coupling external business partners. The on-demand integration of multiple loosely coupled services, as well as the integration of internal legacy systems is provided by a Service-oriented Architecture (SOA) [19]. Each step of the business process can be realized by a specific service. Workflows may use Web services as an open standard to realize business processes [3]. In order to avoid the risk of poor workflow performance, business process intelligence, capacity planning of workflows, business process automation, and performance analysis are equally important.

Performance analysis of a Web service workflow becomes important in order to be able to plan the workflow execution in advance. Beside the QoS-aware composition of one workflow execution, performance analysis concerning the worst-case performance behavior assuming a high amount of incoming workflow execution requests becomes similarly important. This behavior can be modeled by applying network calculus to the concept of Web service workflows.

This paper focuses on optimization problems for the worst-case performance of service-based workflows in order to minimize the delay and to maximize the throughput by using network calculus. Network calculus is a system theory for deterministic queuing systems. It was developed in the 1990s and is widely used in the context of deterministic QoS in packet switched networks.

The remainder of this paper is structured as follows. In section II, related work is introduced followed by a detailed description of the system model including the concept of arrival and service curves as well as the applied cost model in section III. The application of these concepts to service-based workflows is depicted in section IV. Optimization problems for throughput maximization and delay minimization etc. are described in section V and recommendations are given in section VI. The paper closes with a conclusion and an outlook on future work.

II. Related Work

Modeling workflows has been widely studied, e.g., in [1]. Statistical models are used very often, although they are not always appropriate, as in critical business processes (e. g. loan handling, claims handling) deterministic QoS becomes more and more important in order to fulfill the customer's needs. By using only statistical models the achievement of performance goals cannot be guaranteed.

A semantic approach for providing a QoS-aware composition of Web services using ontologies and artificial intel-

978-0-7695-3163-2/08 $25.00 © 2008 IEEE
DOI 10.1109/ICIW.2008.118

ligence planners can be found in [16]. The QoS-aware composition of services to a composite service or a workflow is described in [8] with a genetic algorithm-based heuristic approach. The work of Berbner et al. [4], [5], [6] focuses on the execution and the optimization of one workflow and describes a heuristic-based detailed workflow execution approach based on Web services which enables the selection of specific Web services at runtime as well as replanning mechanisms. A method that uses a predictive QoS model to compute the QoS for workflows in terms of performance, cost and reliability is shown in [9]. The question arises what happens if multiple workflow execution requests arrive at the workflow controller. Is the workflow controller able to handle all incoming requests? The analysis of the average performance behavior of Web service workflows in a scenario with multiple execution requests in a specific time period by using the concept of queuing theory is shown in [12]. This approach facilitates the optimization of the utilization of the invoked Web services in a workflow considering a volume rate cost model. Beside the average execution behavior of QoS-aware Web service workflows, a worst-case consideration by adapting results from the network calculus to Web service workflows is shown in [11].

III. System Model

The considered scenario consists of many workflow requestors that want to execute a specific workflow in a specific time period. The intermediary, who acts as a workflow controller has to combine several Web services to form a workflow. This composed workflow has to be able to serve all incoming requests within a specific time period at good performance properties. The workflow consists of m different tasks which have to be executed sequentially. For each task i, the workflow controller has to choose a Web service which fulfills the required functionality out of $j=1,\ldots,n$ available Web services per task i. All of these Web services are able to fulfill the required functionality of task i. The workflow controller has to create an execution plan for the optimal utilization of Web services in order to, e. g., increase the throughput and to minimize the costs for the Web service usage. It has to select the most efficient Web services for the execution of the workflow requests in order to achieve a flexible and QoS-aware workflow composition. Furthermore the Web services are invoked sequentially.

Hereby, the selection process of Web services of the workflow controller consists of two steps. On the one hand the workflow controller has to select the services which fulfill the required functionality of the specific task of the process. On the other hand non-functional properties such as QoS or costs have to be considered as well in order to optimize the overall performance properties of a workflow.

Usually non-functional properties are described in the Service Level Agreements (SLAs), defined in RFC 3198 [20], which represent bilateral contracts between a service provider and a service consumer. In the SLAs service properties, such as availability, performance, or pricing information and other contract information, are included. Assuming that a variety of services fulfill the required functionality for the execution of one task, the non-functional properties play an important role in their differentiation.

In critical business processes the strict maintenance of non-functional requirements such as, e. g, throughput and cost becomes more and more important. A further performance analysis requires a description of the performance of each service, a description of the behavior of the arrivals of the workflow execution requests as well as an adequate cost model for the service usage.

A. Arrival curves

Arrival curves, which constrain the arrival process, can be used to describe the behavior of the incoming requests for a workflow execution.

Definition 1 [Arrival curve]

Given a wide-sense increasing function α defined for $t \geq 0$, we say that a flow R is constrained by α if and only if for all $s \leq t$:

$$R(t) - R(s) \leq \alpha(t-s) \tag{1}$$

We define that R has α as an arrival curve, or also that R is α-smooth. In this context several arrival curves can be assumed. The Token Bucket arrival curve is appropriate as it captures the incoming request arrivals as shown by equation (2). The maximum size of bulk arrivals can be modeled as well as the sustained service rate. At the beginning of a planning period several requests already exist (bulk) and the subsequent execution requests arrive with a constant rate r_a.

$$a(t) = b + r_a t \quad for\ t > 0 \tag{2}$$

B. Service Curves

The relevant parameters used to describe a Web service include the response time and the rate at which requests can be served. Therefore, the latency-rate (LR) service curve [7], [17] is well-suited to describe the performance of a Web service. A service curve is defined as follows.

Definition 2 [Service curve].

Consider a system S and a flow through S with input function R and output function R^0. We say S offers a *service curve* β to the flow, if and only if β is wide-sense increasing and $R^0 \geq R \otimes \beta$ for all $t \geq 0$.

The operator $\otimes$ represents the convolution which is specified in more detail in [7]. In general, a service curve β as shown in Fig. 1 has a specific arrival rate r and may have a latency l.

A latency-rate service curve of a Web service can be described analytically by equation (3). The rate at which requests are served is denoted by r and the time needed for initialization is defined by l.

$$\beta(t) = r(t-l) \quad ; \beta(t) = 0 \quad t < l \tag{3}$$

It can be stated that the invocation of Web services usually implies a dedicated latency before requests can be executed. The latency performance of SOAP implementations is depicted in more detail in [10].

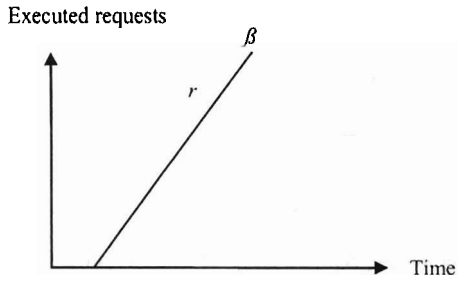

Fig. 1. Service curve

C. Cost model for Web services

The pricing and accounting information are usually described in the SLAs of the services. The pay-per-use and the flat-rate model are the most discussed pricing models for Web services in the literature, e. g. [13], [14], [15], [21]. Our further analysis is based on the pay-per-use pricing model approach with the assumption that an increasing execution rate leads to higher costs. The indices i and j represent the considered task i and the category j of the workflow. This implies that the smaller the response times the higher are the caused costs.

A service provider offers a Web service $WS_{i,j}$ with a specific functionality and a specific response time $t_{resp,i,j}$ to the service consumer. The execution rate $r_{i,j}$ at which the requests can be served is the reciprocal of $t_{resp,i,j}$. The Web service provider charges the service consumer a specific amount as fixed costs for the usage of the Web service and a variable portion for each Web service invocation. The broken down costs for each service invocation are denoted by $c_{i,j}$. The assumed cost model of this analysis is a common cost model and implies that it is the interest of the service consumer to use the most cost-efficient services in order to realize the execution of the workflow.

IV. Application to Service-Based Workflows

In the considered system, incoming workflow execution requests may arrive in a bulk at a specific time or can arrive constantly. Our analysis assumes that at the beginning ($t=0$) there is a bulk of arrivals and then requests arrive with a constant rate r.

Concerning a business process consisting of m different tasks, the workflow controller has to search for m different Web services which fulfill the required functionality of each task i. Assuming that for each task i multiple Web services $WS_{i,j}$ exist which fulfill the required functionality, the differences of these services are the latency l and the rate r. Equation (4) depicts the analytical description of service curves for different Web services with different QoS properties, whereas the indices i and j denote the Web services of process task i and category j.

$$\beta_{i,j}(t) = r_{i,j}(t - l_{i,j}) \quad ; t \geq l_{i,j} \tag{4}$$

The aggregated service curve $\beta_{w,k}(t)$ which is the convolution of the service curves of the services used in the business process can analytically also be described as shown in equation (5). The parameter w depicts the specific workflow and k the executable service composition (with $k=1,\ldots,o$).

Theoretically, for a workflow consisting of m different basic activities and n different available Web services per activity, $o=n^m$ different workflow compositions may be possible.

$$\beta_{w,k}(t) = r_{w,k}(t - l_{w,k}) \quad ; t \geq l_{w,k} \tag{5}$$

with

$$r_{w,k} = min\{r_{1,j}, r_{2,j}, \ldots, r_{m,j}\} \tag{6}$$

$$l_{w,k} = \sum_{i=1}^{m} l_{i,j} \tag{7}$$

The execution rate of the aggregated service curve $r_{w,k}$ is the minimum of all execution rates $r_{i,j}$ (with $i=1,\ldots,n$) of the service curves of the chosen Web services (equation (6)). The latency of the aggregated service curve is the summation of all latencies $l_{i,j}$ (with $i=1,\ldots,n$) of the chosen Web services (equation (7)) considering sequential service invocations as assumed in this paper. These results occur due to the findings of the application of network calculus to Web services. By using this analytical description of service curves and for the convolution of service curves the worst-case performance properties can be computed as shown in the next section.

V. Optimization Approach

This section presents an optimization approach in order to minimize the worst-case delay of the invoked Web services of the workflow and to maximize the throughput of the workflow. Constraints as costs and other side conditions are introduced as well.

Differences of the invoked Web services occur concerning the service rate $r_{i,j}$, the latency $l_{i,j}$, and the costs $c_{i,j}$. The higher the service rate $r_{i,j}$, the higher are the costs $c_{i,j}$. For all n Web services ($j=1,\ldots,n$) of task i a binary variable $x_{i,j}$ is introduced to model whether a Web service $WS_{i,j}$ is used for the workflow execution or not. In order to avoid that more than one Web service of one task is used at the same time, for the binary variable should hold:

$$\sum_{j=1}^{n} x_{i,j} = 1 \quad \forall i = 1,\ldots,m \tag{8}$$

The challenge is which Web services have to be selected at which process step in order to reduce the delay and to

maximize the throughput.

A. Throughput maximization

The characteristic of the convolution of several service curves $\beta_{i,j}(t)$ is that the resulting service curve $\beta_{w,k}(t)$ has the slope of the service curve with the smallest slope as shown in equation (6). This implies that the maximum achievable execution rate $r_{w,max}$ of the workflow is the minimum of all maximum execution rates $r_{i,j}$ of the available Web services per process step. An upper bound for the maximum achievable throughput in the considered workflow is described by equation (9).

$$r_{w,max} = \min_{i}\{\max_{j}\{r_{i,j}\}\} \tag{9}$$

It is not necessary to invoke the Web service with the highest execution rate $r_{i,j}$ per process step in order to achieve a maximum throughput of the entire workflow, because the overall throughput is bounded by the process step with the weakest throughput. Thus, the workflow controller is able to reduce his costs by rejecting the invocation of Web services with execution rates $r_{i,j}$ higher than the highest achievable execution rate $r_{w,max}$.

Assuming the workflow controller has a fixed budget, denoted by C_{max}, a service composition with a maximum throughput can be computed by solving the optimization problem with the objective function (10) and the constraint (11).

$$Max\ F(\vec{x}) = \underset{i=1}{\overset{m}{MIN}}\left(\sum_{j=1}^{n} r_{i,j}\, x_{i,j}\right) \tag{10}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}\, x_{i,j} \le C_{max} \tag{11}$$

The aim of the workflow controller is to maximize the throughput of the entire workflow constrained by a fix budget C_{max}.

B. Delay analysis

The delay of an execution request specifies the elapsed time between the arrival of an execution requests and the point of time at which the request is processed. The worst-case delay for a given workflow and as specific arrival behavior as shown in Fig. 2 can be easily computed by equation (12) and (13) which is derived from equation (5). In equation (12) and (13), b determines the number of execution requests at $t=0$ of the corresponding arrival curve.

$$d_{w,k} = \inf\{\tau : \beta(\tau) \ge b\} \tag{12}$$

$$d_{w,k} = l_{w,k} + \frac{b}{r_{w,k}} \tag{13}$$

with $r_w \ge r_a$

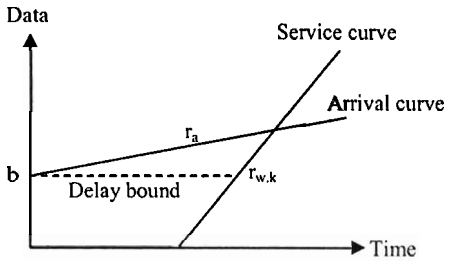

Fig. 2. Delay bound

In order to compute the worst-case delay $d_{w,k}$ of the considered workflow it has to be ensured that the workflow execution rate $r_{w,k}$ is higher than the rate at which the requests arrive r_a. If this requirement is not met, the workflow controller will not be able to serve all incoming execution requests, because more execution requests arrive than the composed workflow is able to serve.

With this computation it is possible to compute the worst-case delay for each possible composition of the services to a workflow.

C. Worst-case delay minimization

When determining the execution plan, one objective of the workflow controller should be to minimize the worst-case delay of the entire workflow. The workflow controller has to select a feasible composition of services to a workflow that minimizes the overall worst-case delay. Thus, equation (14) describes the objective function of this optimization problem.

$$Min\ F(k) = l_{w,k} + \frac{b}{r_{w,k}} \tag{14}$$

This objective function is not enough to describe this problem, because without a constraint the workflow controller would always choose the services with the highest execution rates $r_{i,j}$. This implies that he would choose the most expensive Web services and is faced with the highest costs. Thus, an important constraint in this optimization model is that the overall costs of the workflow execution are constrained by a certain boundary C_{max} as shown in equation (11), in order to realize a cost-efficient Web service composition.

This optimization approach facilitates that the workflow controller is able to minimize his worst-case delay under the assumption of a fixed budget.

D. Cost minimization

Another optimization problem occurs if the workflow controller has to execute a specific amount of workflow executions e until a fixed deadline t_e as depicted in Fig. 3. The selection of Web services has to be done in order to execute all requests at minimal costs.

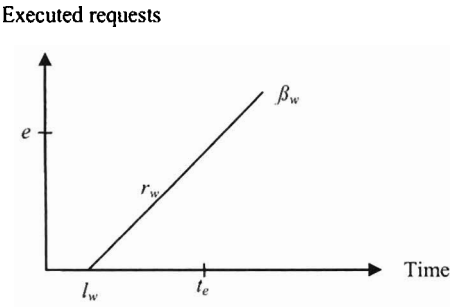

Fig. 3. Fixed amount of workflow executions

The optimization problem can be formulated by the main objective function shown in equation (15) and the constraints depicted in equation (16). The main objective has to be to reduce the costs as much as possible with the constraint that the composed workflow is able to serve all incoming requests e until the time t_e.

$$Min\ F(\vec{x}) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j}\, x_{i,j} \tag{15}$$

The constraint in this case is as follows:

$$\beta_{w,k}(t_e) \geq e \tag{16}$$

In the case that there exists more than one possible solution for this optimization problem the workflow controller may choose this composition which minimizes the overall worst-case delay and maximizes the overall throughput.

With the optimization approaches mentioned in this paper, the workflow controller is able to optimize the worst-case behavior of his composed workflow in advance and is able to avoid performance degradation before they occur.

VI. Recommendations

In the previous section optimization approaches concerning delay, throughput and cost are described. These optimization approaches facilitates that the orchestrator is able to optimize the worst-case behavior of the workflow execution before the workflow execution starts, i. e., it has to invoke those Web services that meet the requirements of the optimization model.

Concerning throughput maximization, the throughput of the composed workflow that can be achieved is bounded by the lowest execution rate of the invoked services in the entire workflow. A purely throughput optimization approach would be to determine the maximum achievable execution rate and to invoke only those Web services for the other tasks that have a higher or the same execution rate and that minimize the overall costs.

A reasonable procedure concerning delay minimization would be to estimate a maximum acceptable delay for the service consumer and to minimize the costs by invoking those Web services that guarantee this worst-case behavior.

Concerning business processes, the described optimization approaches fit to business processes with a sequential execution of tasks. Workflows with a high repetition rate and a high business value, e.g. claims handling, loan handling, and accounting, require a continuous workflow control. The proposed optimization approaches support the workflow orchestrator to optimize the worst-case behavior of business processes in order to meet customer requirements.

VII. Conclusion

In this paper we propose several optimization approaches for Web service workflows in order to maximize the throughput and to minimize the delay in a worst-case consideration. This work applies results from network calculus to Web service workflows and extents previous results with the analytical description of the performance behavior of Web service workflows. The optimization approaches show how the delay and the throughput of a workflow execution can be optimized. The workflow controller is able to optimize his execution plan in advance and to invoke only those Web services which fulfill the worst-case requirements.

Our further research aims at extending our approach with heuristics and enhancements of the proposed optimization model.

VIII. Acknowledgments

This work is supported in part by the E-Finance Lab e.V., Frankfurt am Main, Germany (http://www.efinancelab.com).

IX. References

[1] W. M. P. v. d. Aalst, K. M. v. Hee, *Workflow Management: Models, Methods, and Systems*, MIT press, Cambridge MA: 2002.

[2] V. A. F. Almeida and D. A. Menascé, "Capacity Planning: An Essential Tool for Managing Web Services," *IT Professional*, vol. 4, no.4, 2002, pp. 33-38.

[3] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services. Concepts, Architectures and Applications*, Springer, Berlin: 2004.

[4] R. Berbner, T. Grollius, N. Repp, O. Heckmann, E. Ortner, R. Steinmetz, "An approach for the Management of Service-oriented Architecture (SoA) based Application Systems," in *Enterprise Modelling and Information Systems Architectures (EMISA 2005)*, 2005.

[5] R. Berbner, O. Heckmann, and R. Steinmetz, "An Architecture for a QoS driven composition of Web Service based Workflows," in *Proceedings of the Networking and Electronic Commerce Research Conference*, 2005.

[6] R. Berbner, M. Spahn, N. Repp, O. Heckmann, R. Steinmetz., „Dynamic Replanning of Web Service Workflows," in *IEEE International Conference on Digital Ecosystems and Technologies*, 2007.

[7] J.-Y. Le Boudec and P. Thiran: Network Calculus. Number 2050 in Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2001).

[8] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms," in *Proceedings of Genetic and Evolutionary Computation Conference*, 2005, pp. 1069-1075.

[9] J. Cardoso, A. Sheth, J. Miller, J. Arnold, K. Kochut, "Modeling Quality of Service for workflows and web service processes," in *Web Semantics: Science, Services and Agents on the World Wide Web Journal*, 1 (2004), pp.281-308.

[10] D. Davis and M. Parashar, "Latency Performance of {SOAP} Implementations," in *Proceedings of the 2 IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2002, pp. 407-412.

[11] J. Eckert, K. Pandit, N. Repp, R. Berbner, and R. Steinmetz, "Worst-Case Performance Analysis of Web Service Workflows," in *Proceedings of the 9th International Conference on Information Integration and Web-based Application & Services*, 2007.

[12] J. Eckert, S. Schulte, N. Repp, R. Berbner, R.. Steinmetz "Queuing-based Capacity Planning Approach for Web Service Workflows Using Optimization Algorithms," in *IEEE International Conference on Digital Ecosystems and Technologies*, 2008.

[13] D. Gouscos, M. Kalikakis, and P. Georgiadis, "An Approach to Modeling Web Service QoS and Provision Price," in *Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops*, 2003, pp. 121-130.

[14] O. Günther, G. Tamm, and F. Leymann, "Pricing web services," *Int J. Business Process Integration and Management*, vol. 2, no. 2 2007, pp. 132-140.

[15] G. E. Mathew, J. Shields, and V. Verma, "QoS Based Pricing for Web Services," in *Proceedings of the 5th International Conference on Web Information Systems Engineering Workshops*, 2004, pp. 264–275.

[16] M. Naseri, A. Towhidi, "QoS-Aware Automatic Composition of Web Services using AI Planners," in *Proceedings Second International Conference on Internet and Web Applications and Services*, 2007.

[17] K. Pandit, "Quality of Service Performance Analysis based on Network Calculus," PhD Thesis, Dept. of Electrical Engineering, Technische Universität Darmstadt, 2006.

[18] M. P. Papazoglou and W. J. van den Heuvel, "Leveraging legacy assets," M. P. Papazoglou, S. Spaccapietra, and Z. Tari, *Advances in Object-Oriented Modeling*, MIT Press, Cambridge MA: 2000, pp. 131-160.

[19] M. P. Papazoglou, "Service-Oriented Computing: Concepts, Characteristics and Directions," in *Proceedings of the 4th International Conference on Web Information Systems Engineering*, 2003, pp. 3-12.

[20] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, "Terminology for Policy-Based Management," RFC 3198, 2001.

[21] I. Yahav; A. Gal; and N. Larson, "Bid-Based Approach for Pricing Web Service," in *Proceedings of Cooperative Information Systems*, 2006, pp. 360-376.