Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 88 (2015) 36575-36579

ISSN: 2229-712X

Optimizing the QoS evaluation in service based systems

T.Priyaradhikadevi[1] and S. Parvathi[2]

[1]Department of Computer Science and Engineering Mailam Engineering College Mailam – 604304, Tamil Nadu, India.

[2]Department of Computer Science and Engineering, Sengunthar College of Engineering, Tiruchengode – 637 205, Tamilnadu, India.

ARTICLE INFO

Article history:

Received: 29 October 2012;

Received in revised form:

18 November 2015;

Accepted: 23 November 2015;

Keywords

Service based systems,

QoS evaluation,

High-quality workflow,

Service-based system.

ABSTRACT

In service based systems, QoS are the basic construct that aims to facilitate the quality and usability of service based system. Service based systems takes a very long time to get the result under workload, of quality evaluation when dealing with a workflow with several thousand services and multiple parameters. To address this need, we introduce a tool-supported framework for the development and to improve the evaluation performance of QoS by optimizing the QoS requirements. OQoSE approach develops adaptable service based systems which increases the evaluation result of QoS efficiently. OQoSE method which used to review every service and parameter time then it is used to discover the probabilities that are adaptable under various services with multiple parameters.

Introduction

The new paradigm of service development and deployment imposes great challenges to the quality of the services. Since an application system usually involves services from different organizations, to guarantee that these services can collaborate correctly and seamlessly, the quality of services is extremely important. External factors like resource sharing and service reuse also affect the quality of services. These specific requirements distinguish service development from traditional software development. Service based systems are playing a important role in application domains where this applications undergo a continual changes to requirements environment. Networks must provide secure, predictable, measurable, and sometimes guaranteed services. Achieving the required Quality of Service (QoS) by managing the delay, delay variation, bandwidth, and packet loss parameters on a network becomes challenging.

OQosE approach used to increase the efficiency, performance in the result of quality evaluation in a efficient manner when dealing with various services and multiple parameters. OQOSE for service based systems is designed by redefining the technique of QoSMOS adapating to various services. Adaptable service based system designed based on various services and multiple parameters by improving the Markov Decision processes. Our approach defines the language for optimizing the multi model consistency. QoSMOS approach applies to service based systems probabilistically quantifiable and externally observable QoS properties, such as reliability, availability and performance. OQoSE approach designed to overcome the probabilistic result of QoS properties by analyzing the probabilities and prioritizing which are implemented in service based systems.

To improve the performance and effectiveness of QoS Management in service based systems, we have defined the Reliable and effective architecture for service based systems called OQoSE(Optimizing the Quality of Service Evaluation)

(a) QoS Evaluation is improved with viewing techniques to discover the requirement evaluation and the QE specification system.

(b) Similar QoS with Analyzing and Consulting techniques provided by the STAT equity checker.

(c) Requirement Filtering is done in order to improve the quality and usability of service based systems by the Requirement Filterer.

(d) Service based system yields an better performance with various number of services and parameters provided by the Quality Evaluator.

(e) Result of service based system based on GPAC.

The OQoSE framework which deals with QoS requirement for improving the Quality Evaluation by means of two complementary mechanisms. The first mechanism conists of identifying the proper service which yields the best results with number of services and parameters. The second mechanism controlled by OQoSE consists of configuring the administrator system internally by allocating and optimizing the resources based on the requirements if the requirements of administrator which causes high workload in service system then Service based system configures the administrator system by optimizing the resources. A Set of online traces collection mechanisms and discuss their performances in terms of required CPU/RAM resources and introduced network overhead. Its main benefit of service based system to allocate the resources based on actual workload of the system.

Specifically, we summarize the approaches according to:

i) Viewing the QoS Specification (QoS Requirement identification);

ii) Evaluating the QoS Requirements through models and algorithms;

iii) Redefining the requirements which yield the better performance for workflows of various numbers of services and parameters;

iv) Implementing the optimizatation method to be applied for SBS;

Tele:

E-mail addresses: priyaradhikadevi@gmail.com

Verifying and executing the Approaches

QoS Requirement Identification

Formal and precise specification of Quality of Service constraints is the basis for monitoring and metering QoS metrics. It prescribes which QoS metrics to monitor, where and how to do this monitoring, how to eventually calculate aggregate QoS metrics, what the expected values of QoS metrics are, and eventually what to do if they are not met. Consequently, formal and precise specification of QoS metrics is particularly useful in performance management. QoS requirement is done through the various languages used to formal specification for classes of services for web service. Requirement identification play key role to evaluate the performance of service based system. Identification of requirement then evaluating finally executing the efficient result.

QoS Evaluation Models and Algorithm

Trust-mechanism-based service selection models, we propose a novel ANN-based (Artificial Neural Network) service selection model called the ANNSS model for QoS evaluation. An ANN-based evaluation standard for the service quality of service provider is given out so that user can acquire an effective guidance and choose the most appropriate service. QoS Evaluation model

QoS Quality Evaluation

Producing the Better Perfomance with various number of services and parameters through Taverna. A Taverna a tool for building and running workflows of services. Taverna is an application by integerating the web services. It allows to construct workflows which analyses the various number of services and parameters.

QoS Optimiztation and Implementation

Specifying QoS properties on services and selecting services for their composition in a way that maximizes the QoS values. We apply the integration of concerns paradigm to allow combined specification of QoS and functional properties by using Quantitative Constraint Automata, and to integrate QoS aspects into service-oriented applications development processes for service selection.

Organization: The rest of the paper is organized as follows. In Section 2 we shortly describe the main formalisms used throughout the paper, namely requirement specification and violation and Markov Models. Section 3 describes the OQoSE architecture, OQoSE generic architecture consists of viewing phase, analysis phase, consulting phase and result phase and final text includes conclusion.

Preliminaries

Formal definition of Qos Specification

Qos Evaluation Solution

Trust-mechanism-based service selection models, we propose a novel ANN-based (Artificial Neural Network) service selection model called the ANNSS model for QoS evaluation. An ANN-based evaluation standard for the service quality of service provider is given out so that user can acquire an effective guidance and choose the most appropriate service. QoS Evaluation model reducing the Better Perfomance with various number of services and parameters through Taverna. A Taverna a tool for building and running workflows of services. Taverna is an application by integerating the web services. It allows to construct workflows which analyses the various number of services and parameters.

QoS Quality Evaluation

Formal and precise specification of Quality of Service constraints is the basis for monitoring and metering QoS metrics. It prescribes which QoS metrics to monitor, where and how to do this monitoring, how to eventually calculate aggregate QoS metrics, what the expected values of QoS metrics are, and eventually what to do if they are not met. Consequently, formal and precise specification of QoS metrics is particularly useful in performance management. QoS requirement is done through the various languages used to formal specification for classes of services for web service. Requirement identification play key role to evaluate the performance of service based system. Identification of requirement then evaluating finally executing the efficient result.

The most used Markov models include:

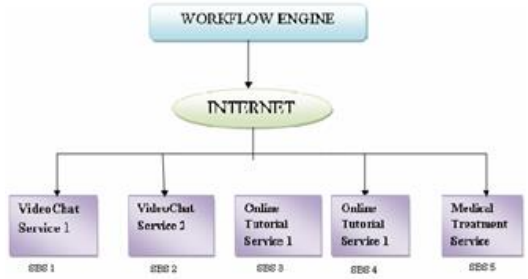
• Discrete Time Markov Chains (DTMC), which are the simplest Markovian model where transitions between states happen at discrete time steps;

• Continuous Time Markov Chains (CTMC) where the value associated with each outgoing transition from a state is intended not as a probability but as a parameter of an exponential probability distribution (transition rate);

Markov Decision Processes (MDP) that are an extension of DTMCs allowing multiple probabilistic behaviours to be specified as output of a state. These behaviours are selected non-deterministically.

OQoSE Architecture

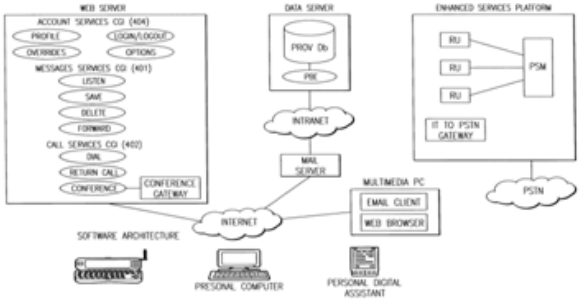
Physical Design of an individual service that encompasses all the resources used by a service. OQoSE service architecture serves as a point of reference for evolving the service or gauging the impact of any change in the service and improving the systems adaptability to various services. OQoSE architecture design paradigm is composition-centric so it can potentially address QoS requirements by recomposing the service in different configurations



Service Based System Architecture

Generic OQoSE Architecture

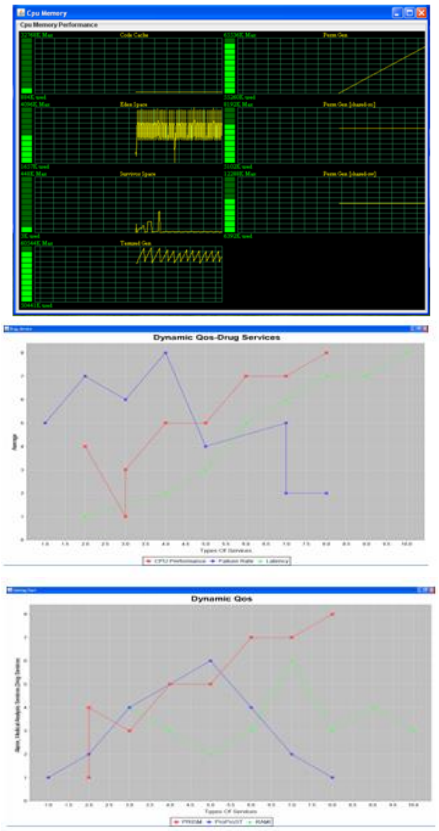
As illustrated in figure 3, OQoSE augments the standard SBS architecture with a component termed an Involuntary manager. This component employs the adaptive computing viewing-analysis-consulting-Result (VACR) loop to ensure that the SBS adapts continually in order to achieve a set of high-level, multi-objective QoS requirements specified by its administrator.



Videoconferencing Workflow

OQoSE approach which yields effective result in quality evaluation of SBS when dealing with various number of services and multiple parameters.

Performance Graph



OQoSE VACR Loop for the Video Conferencing SBS

QoS management has been extensively studied in network-based multimedia applications as well as web-based applications

Viewing Stage

The first stage of the loop involves viewing either or both of:

1)The Result and performance of the service based systems. The response time and failure rates parameters is viewed by administrator and user.

2)The heavy load of individual service systems (e.g., Educational Tutorial service and video Chat service)resources allocated to these services (e.g., CPU, memory and bandwidth) these services is limited only to administrator; these characteristics cannot be viewed by user and developer.

The developer of service based system provide an useful operational model for service based system to build or update is an initial version.

The Viewing phase of OQoSE and the continuous changes to requirements by the administrators used to discovering the requirements which affect the QoS attributes and improving the performance of Service based systems by optimizing the requirements.

Analysis Stage

Qos requirements are processed in the analysis stage Qos requirements are important for the effectiveness of quality evaluation. Requirements should be identified and then optimized by different phases. QoS requirements are operated by the operational model of the viewing phase. Analysis step includes some intial processing steps in these instructions QoS requirements which are specified by the administrator basically requirements specification is in high level language are converted involuntarily into predefined QoS requirements.

Example:

The Qos Requirements for the videoconferencing service based system from our simulation includes performance and reliability related requirements. For the relatiability related requirements consists of average number of videochat request associated with a particular person who is chatting with the requester of the video service based system (i.e., the lifetime of the system) is ten; these high-level requirements are detailed below:

R0 The probability P0 that a video chat dial request failure ever occurs during the lifetime of the system is less thatn P0=0.12.

R1 The probability P1 that a video chat service failure ever occurs during the lifetime of the system is less than P1=0.13.

R2 The probability P2 that a return call or conference request generates an request which fails is less than P2=0.006.

R3 Assuming that request generated by message service have low priority while request generated by call service have high priority, it is notified that the probability P3 that a high priority request fails is less than P3 = 0.003.

In addition to reliability, we considered one of the following QoS Quality Requirements is performance requirements specified by administrator of the video assistance system:

R4: The probability P4 that the number of message service request exceeds 75% of the request queue capacity for the video chat service.

R5: The probability P5 of a message service request being dropped due to the request queue being fall during a day of operation is less than 0.05.

QoS requirements require information of the reliability and perfomance metrics specified by service based system administrators. SBS controllers need to have an information of the service-level agreements that the SBS users expect from the system are not fulfilled.

Consulting Stage

The Consulting stage of the OQoSE uses the results of the analysis stage to construct a organize for automatically configure the SBS. The different types of self adaptation are possible by the OQoSE approach and implemented in the result stage.

Service based system change the workflow by adaption approach. OqoSE framework which provides self adaption technique for all service based system including those that employ developer and administrator. Abstract overflow is implemented in SBS functionality consists of abstract services developed. Consulting stage map the abstract services within this original workflow to concrete services by the analysis results process that takes during the consulting stage.

Self adaptation is achieved in the SBS by configuring the resource allocation to individual services. SBS implement the internally administered services possibly adapt the resources allocation to these services by varying the workloads and in the QoS requirements for the system. Self adaptation is achieved by performance-related QoS requirements with minimal cost and through redundancy purposes achieving dependence QoS requirements by running services across a variable number of servers.

OQoSE architecture map the abstract to concrete services using one of the mapping patterns detailed below:

Abstract service uses a concrete service with suitable perfomance, reliability and cost characterisitcs by single mapping (SGL).

In sequential of concrete service is mapped to an abstract service by sequential one-to-many mapping (SEQ).

Review of Existing and Proposed Approaches

Authors & Citation	Problem Domain	QoS Requirements	QoS Evaluation	QoS Optimization or Adaption Method	Validation
Marzolla at al. 2007 [79]	QoS-driven workflow management	Response Time	Analytic solving of BCMP queueing systems	Exact methods and bound analysis	Experiments based on gen. examples
Sato at al. 2007 [93]	QoS-driven workflow management	Reliability	Analytic solving of Markov Models	Exact solution based on CTMC analysis	Experiments based on gen. examples
Guo at al. 2007 [52]	QoS-driven workflow management	Avaliability	Simple Aggregation Functions sequential, choice, parallel and iterative web service composition	Hill Climbing based selection redundancy mechanisms	Experiments based on gen. examples
QoSMOS	Service Selection/ Resource allocation and Service Parametrisation	Perfomance and Reliability	Analytic solving of Markov Models (DTMC and MDP)	Exact solution based on probabilistic model checking with PRISM experiments	Experiments based on gen. examples and the TeleAssistance case study
OQoE for SBS	Efficiency in Multimodel Consistency, Improving the result of multiple services and various parameters.	Efficiency,Relaibaility, and performance	Analytic solving of Markov Models and BCMP queueing systems.	Efficient Solution based on reviewing the services to improve the Quality evaluation.	Experiments based on gen. examples and the videoconferencing case results.

A set of concrete service is mapped to an abstract service by an parallel one-to-many mapping (PAR), which are called during the execution of the workflow.

OQoSE approach supports the various mapping pattern for each abstract service. When the workflow is executed, these services of the service based system are used one at a time starting with initial service in the sequence and carrying on the sequence until the error response is obtained. Reliability-related QoS of an service based system improved used by an abstract services.

Result Stage

Increasing number of workflows engines supports dynamic workflow changes supporting this functionality in OqoSE involves exploiting the advantages of such existing engines. The Consulting stage is allocated a resources to concrete services in the VACR loop. Dedicated resource management approach which affects the SBS by modifying the parameters of an application server; starting, stopping changing virtual machine. Various number of servers running the services are modified at runtime by the service invocations across all these servers which are alloated to individual services.

The importance of OqoSE is to insure the compliance in high-level requirements of Service-based systems. Multiple workflows and dynamic workflow modification is emulated by the standard workflow engine of a OqoSE workflow. Dynamic resource allocation achieved by the SBS through load balancing the requests for a concrete service among a runtime chosen set of physical servers that run instances of the service.

Conclusion

In this paper we have presented OqoSE, a tool-supported framework for QoS quality evaluation optimizatation of self automatic service-based systems. OqoSE proposed an involuntary architecture that combines formal specification of QoS high-level requirements, QoS evaluation, viewing and parameter adaptation of the QoS approach and consulting and result of system adaptation.

QoS requirements uses the precise and formal specification of QoS high-level requirements with the markov decision process and the definition of a model based quality evaluation technique for QoS parameters. The viewing phase of OqoSE adaptation technique allows to update the quality models allows discovering requirements violations which enhance the QoS effectiveness by the SBS. The possible strategies are based on techniques for defined service selection, resource allocation. The quality models of OqoSE represent the overall system architecture, so it is possible to detect improper requirement generated by different causes and not only related to unexpected behaviors and also unexpected variations in the usage profile. The validation of the proposed framework has been performed through the application of OqoSE capabilities to a common case study of a service-based system for remote video assistance. The results obtained with a high number of numerical experiments and simulations proved the effectiveness of our approach.

References:

[1] J.-P. Katoen, M. Khattri, and I. S. Zapreev. A Markov reward model checker.In Proceedings of the Second International conference Quantitative Evaluation of Systems (QEST), pages 243–244. IEEE CS Press, 2005.

[2] Yau, S. S., Ye, N., Sarjoughian, H., Huang, D., Roontiva, A., Baydogan,M., Muqsith, M.: Towards Development of Adaptive Service-Based Software Systems. In: IEEE Trans. on Services Computing, vol. 99, pp. 247-260 (2009)

[3] Yau, S. S., Yin, Y., An, H. G.: An Adaptive Model for Tradeoff between Service Performance and Security in Service-Based Environments. In: Proc. of Int'l Conf. on Web Services, pp. 287-294 (2009)

[4] J. Hillston. A Compositional Approach to Performance Modelling. Cambridge University Press, 1996.

[5] J.T. Bradley and W.J. Knottenbelt. The ipc/HYDRA tool chain for the analysis of PEPA models. In Proc. 1st International Conference on the Quantitative Evaluation of Systems (QEST 2004), pages 334–335, Enscheda, Netherlands, September 2004.

[6] W J Knottenbelt, P G Harrison, M S Mestern, and P S Kritzinger. A probabilis-tic dynamic technique for the distributed generation of very large state spaces Performance Evaluation, 39(1–4):127–148, February 2000.

[7] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," Computer, vol. 40, pp. 38–45, 2007.

[8] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana: Web Services Description Language (WSDL) 1.1, W3C Note http://www.w3.org/TR/wsdl

[9] Huber, F. Schätz, B. Integrating Formal Description Techniques. In: FM'99 -- For- mal Methods, Volume II. Wing, J. et al. (eds.). Springer, 1999.

[10] Schätz, B. Towards Service-Based Systems Engineering: Formalizing and muChecking Service Specifications. Tech. Report TUMI-0602, TU München, 2002

[11] Antkiewicz , M. (2006a, September). Round-trip engineering of framework-based software using framework-specific modeling languages. In proceeding of 21st IEEE International Conference on Automated Software Engineering, 323 – 326, Washington, DC, USA.

[12] Hatcliff, J., Deng, W., Dwyer, M., Jung, G., & Prasad, V. (2003). Cadena: An integrated development, analysis, and verification environment for component- based systems. In Proceedings of the 25th International Conference on Software Engineering. Portland, OR.

[13] Lardieri, P., Balasubramanian, J., Schmidt, D., Thakar, G., Gokhale, A., & Damiano, T.(2007 to appear). A multi-layered resource management framework for dynamic resource management in enterprise DRE systems. In C. C Cavanaugh (Ed.) Journal of Systems and Software: special issue on Dynamic Resource Management in Distributed Real-time Systems.