A Simulator for QoS Service Choreography support

Alfonso Phocco Diaz, Daniel Batista, Dejan Milojicic

University of São Paulo, Brazil HP Labs Palo Alto, USA

{alfonso7, batista}@ime.usp.br; dejan.milojicic@hp.com

Abstract

Service choreography allows the composition of services in a collaborative way, because of global description and decentralized coordination using interactions P2P among participants. However, since infraestructures and implementations aren't mature enough to enact choreographies, it is difficult to evaluate and analyse the fulfillment of QoS requirements when using choreographies. In this paper we propose a simulator of choreographies in order to evaluate parameters of QoS. We also propose a QoS model.

Problem statement

Due to the increasing number of devices joining the Internet, a centralized approach like an orchestration may not be sufficiently scalable in terms of network bandwidth to deal with the ever escalating number of devices and services that may be available. Within such a scenario, a decentralized approach, like choreographies, may turn out to be more capable of dealing with such high complexity [3].

Web services choreography is an efficient way to implement inter-organizational business processes, as the participants' business interactions are mutually independent (autonomous and heterogeneous). A choreography is a description of peer to peer interactions among existing services, i.e., in this model there isn't the role of a central controller, unlike of orchestration and current ESB[1] technologies. The various services communicate with each other directly [1].

During the enactment of services choreographies, the state of network elements (devices and links) plays a fundamental role. There must be guarantees of Quality of Service - QoS so that there are advantages of using a decentralized business model. A common method to define guarantees between a service provider and a client (which may also be a service) is by means of a Service Level Agreement - SLA.

Currently, to implement and enact a real service choreography is still difficult due immature technology support, especially by lack of choreography aware engine execution [6]. Thus, mechanisms to define QoS measurements, QoS requirements establishment, monitoring, and so on, aren't well developed to choreographies.

Our solution

The objective of our work is to develop a simulator in order to enact services choreographies and enable QoS support taking into account the infrastructure and environment aspects. To achieve it, we use a QoS model over service, message and communication attributes involved in a service choreography.

Based on [8] and [7] works, we propose a model QoS in three aspects: (a) service , (b) message and (c) communication. The table 1 shows the QoS model composed by QoS attributes, metrics and respective failure types, that our simulator should support.

To implement an entire simulator is a very complex task, due that, we decided to use an existing simulador as the base of our simulator.[EH] Could be better to explicit show which one has been chosen in this paragraph.

The SimGrid framework [4] is a simulation-based framework for evaluating cluster, grid and P2P mechanisms. SimGrid uses tasks to perform the simulation. Such tasks have an intrinsic cost to be transmitted over the network and an execution cost. Resources are described through an XML file in which it is possible to list available resources and their characteristics such as computing power, as well as available links and routes to other resources. In another XML file the deployment of the simulated entities is described, i.e, where each simulated entity, also referred to as a Process, is deployed.

[1] ESB: Enterprise Service Bus

Table 1: QoS model

QoS aspect	QoS attribute	Metric	Failure type
Service	Execution time	ms	timeout
Service	Throughput	#requests/s	service not available
Message	Message format	-	failure probability
Communication	Latency	ms	communication error
Communication	bandwidth	Mb/s	communication error

Since the SimGrid allows the simulation of distributed environments, we used it as the base to implement our simulator.

Execution environment

To attest the efficacy of our simulator we adopt a choreography scenario about Content Delivery Network (CDN) providing streaming multimedia objects [2]. The Figure 1 shows the reference scenario [2] : a user requires, and eventually receives, a complex service managed through a choreography of different Web-Services, one of which (WS_3 in the figure) controls the provisioning of streaming content. WS_1 and WS_2 show that several Web-Services are internally orchestrated. Such a scenario is used for assessment of choreography simulator.

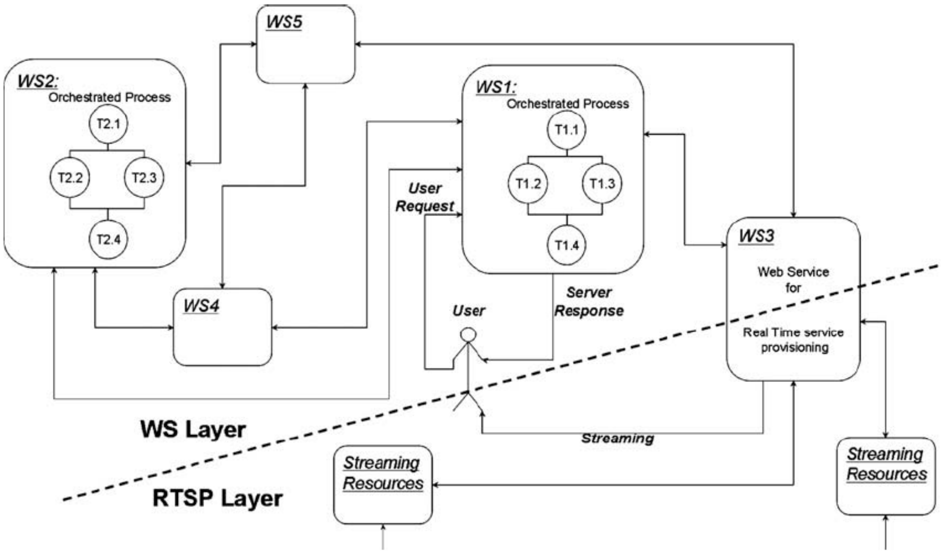

Figure 1: Choreography scenario about a CDN application[2]

Evidence the solution works

The services were modeled as a set of working threads that receive a task sent over the network, execute it and then send another task over the network to act as a web service response. The available methods and computational effort needed to execute them, the amount of worker threads, medium size of the given responses, and service name are configurable through a deployment XML file. The choreography topology (host, communication channels and links) is configured through a platform XML file. Above this infrastructure, a monitor is developed. This monitor is responsible for measuring the QoS attributes specified in table 1 of individual services as well as to aggregate them in order to calculate global or composed QoS attributes as total time response.

Competitive approaches

Many simulators for distributed environments were proposed, i.e., GridSim framework [3], Pi4SOA [10], and SimGrid framework [4]. The GridSim framework [3] is a distributed environment simulation engine based upon on events. It implements entities to emulate users. Users' requests are scheduled through a broker that allocates them into the simulated resources. Pi4SOA [10] presents a policy-based infrastructure to dynamically verify and control the collaboration process in SOA(Service Oriented Architecture). It is also used as a starting point to develop an event driven policy enforcement in [9].

As shown, there is no simulations solutions for supporting service choreography enactment with QoS support.

Current status

Since there was no simulator systems for service choreographies, we have developed a initial simulator to compare the performance of an orchestrations and a choreography [5]. There, choreographies revealed to be a better choice with regard to performance and to perform the enactment of choreographies. Currently, we are evaluating the performance based on simulations, particularly interested in studying the effects of QoS composition on throughput of services and networks aspects such as latency and bandwidth. Furthermore, we are developing the simulator with focus in choreography enactment and QoS support.

Next steps

As a next step we will compare the results obtained in our simulator with measurements of a real choreography. Also, we will develop a monitoring module in order to perform QoS measurements, QoS aggregation, and so on. In this way, we will have the infrastructure for achieving SLA establishment, QoS violation detection, and other QoS issues, on service choreographies.

References

[1] A. Barker, C. D. Walton, and D. Robertson. Choreographing Web Services. *IEEE Transactions on Services Computing*, 2(2):152–166, 2009.

[2] F. Buccafurri, P. Demeo, M. Fugini, R. Furnari, a. Goy, G. Lax, P. Lops, S. Modafferi, B. Pernici, and D. Redavid. Analysis of QoS in cooperative services for real time applications. *Data & Knowledge Engineering*, 67(3):463–484, Dec. 2008.

[3] R. Buyya, M. Murshed, C. Campus, and G. Campus. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. *CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE (CCPE)*, 14:1175–1220, 2002.

[4] H. Casanova, A. Legrand, and M. Quinson. SimGrid: a Generic Framework for Large-Scale Distributed Experiments. In *Proceedings of the Tenth International Conference on Computer Modeling and Simulation*, UKSIM '08, pages 126–131. IEEE Computer Society, 2008.

[5] F. P. Guimaraes, E. H. Kuroda, and D. M. Batista. Performance Evaluation of Choreographies and Orchestrations with a New Simulator for Service Compositions. In *Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD)*, pages 140–144, 2012.

[6] O. Kopp, L. Engler, T. V. Lessen, and F. Leymann. Interaction Choreography Models in BPEL: Choreographies on the Enterprise Service Bus. In *SBPM ONE 2010 the Subjectoriented BPM Conference (2010)*, 2010.

[7] N. Looker, M. Munro, and J. Xu. Simulating errors in web services. *Network*.

[8] R. S. Pandey. A Meta-Model Based Proposal for QOS of WSCDL Choreography. In *Proceeedings of the International MultiConference of Engineering and Computer Scientists IMECS 2010*, volume I, pages 2–8, Hong Kong, 2010. Newswood Limited.

[9] W. T. Tsai, X. Zhou, and Y. Chen. SOA Simulation and Verification by Event-Driven Policy Enforcement. *41st Annual Simulation Symposium (anss-41 2008)*, pages 165–172, Apr. 2008.

[10] X. Zhou, W. Tsai, X. Wei, Y. Chen, and B. Xiao. Pi4SOA: A Policy Infrastructure for Verification and Control of Service Collaboration. *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'06)*, pages 307–314, 2006.