Composite service optimization through decentralized coordination

Mario Henrique Cruz Torres, Tom Holvoet
{MarioHenrique.CruzTorres, Tom.Holvoet}@cs.kuleuven.be
Department of Computer Science
Katholieke Universiteit Leuven
3001 Heverlee, Belgium

ABSTRACT
Service Oriented Computing (SOC) is a computing paradigm that enables the creation of loosely-coupled applications that expose functionality through the offering of services. Composite services are made by the composition of different services and need to maintain certain levels of quality of service (QoS). One type of solution to maintain the required QoS in a composite service is to effectively select the best component services that will participate in the composition. We show a middleware solution that is able to maintain a required QoS, using a decentralized coordination mechanism. This coordination mechanism autonomically adapts in order to improve the overall efficiency of the services being composed.

Categories and Subject Descriptors: C.2.4 [Computer Systems Organization]: Distributed Systems
General Terms: Algorithms
Keywords: ACO, SOC, bio-inspired, coordination mechanisms, multi-agent system, service allocation, choreography, orchestration

1. INTRODUCTION
Service Oriented Computing is a paradigm that enables the creation of loosely coupled applications offered as services, which are invoked through the network. It is possible to assemble a set of different component services in order to create a service composition. The reuse obtained by service composition minimizes the software production cycle, allowing organizations to focus on their core businesses. Composite services are designed in terms of service interfaces and bound to specific providers at run-time, achieving an even higher level of flexibility and dynamism [2].

However, the extra flexibility given by runtime binding to service providers brings a new sort of problems, such as uncertain QoS. The QoS uncertainty occurs because each service provider has its own characteristics of performance, connectivity, pricing model, etc. In order to create performant composite services it is paramount to be able to maintain the required QoS, what can be done through proper component service selection.

In this paper we focus on the problem of adapting service compositions to maintain a required QoS. Due to the characteristics of this problem we propose a decentralized and autonomous coordination mechanism to solve it.

Several works deal with the problem of component service selection to adapt composite services, for instance [5], [1], [4]. [5] proposes a middleware, called AgFlow, which offers two approaches for component service selection, a local and a global one. Both approaches provided by AgFlow use integer linear programming to implement the selection algorithms, what is best suited for small-scale problems as the computational costs increase exponentially with the problem size. Another issue with AgFlow is that it is not suited for very dynamic environments, since it needs to execute the optimization algorithms every time some change occurs. Our approach deals with dynamic changes in the environment by the normal execution of the algorithms, since the solution is built iteratively. A very interesting approach is proposed by [1], which uses the notion of a skyline query to select services for the composition. The skyline query selects only those services that are not dominated by any other service, regarding their QoS, thus reducing the number of candidate services to be considered in the composition. An issue with this approach is that it relies on a centralized solution and can flood the network if many composite services use the same technique. Our approach works in a decentralized way, allowing multiple parties benefit from the information stored in the overlay network.

Another work is presented in [4]. Different techniques for component service selection are discussed and evaluated. The simulation results show that the most efficient approach is the proxy-based, followed by the collaborative approach. In the proxy-based approach, all the service invocations go through a proxy that can then, load balance and select the best available services. In the collaborative approach different composite services collaborate, sharing QoS information about component services, to allow a better component service selection. Our solution shares characteristics with both the proxy-based and collaborative approaches, what we show in next section.

2. DECENTRALIZED COORDINATION
The biggest benefit of the proxy-based approach discussed in [4] is that is has access about all the service invocations happening between a number of services at a given time. In our solution there is no single entity capable of having the knowledge about all the service invocations, however our solution provides an overlay network that is used by
the services to indirectly give information about successful service invocations, failures, etc.

The motivation for our solution comes from the fact that Ant Colony Optimization (ACO) algorithms are very efficient for graph traversal problems [3]. In our solution, component and composite services participate in a service overlay network. There are two participation modes: (i) service provider, which are the component services, (ii) service requester, which are the composite services. A service provider is represented by a ResourceAgent. ResourceAgents are also responsible for storing the QoS information of the services they represent. Besides representing service providers, the ResourceAgents maintain the pheromone level information. Pheromones indicate the quality of a certain path in the overlay network. ResourceAgents have a mechanism to eliminate stale pheromone information. This mechanism is called pheromone evaporation and eliminates stale pheromone information at a specific rate, called pheromone evaporation rate. ResourceAgents can autonomously adapt their own pheromone evaporation rates, to indicate that their services are being over requested, allowing that way a load balancing of their services usage.

OrgAgents represent the service requestors. OrgAgents are responsible for selecting the services that will participate in a composition and are also responsible for maintaining the QoS of the composite service they represent. OrgAgents send a number of ExplorationAnts to search for services in the environment. These ants crawl the overlay network, looking for services needed by theOrgAgent. The ExplorationAnts check the component services QoS, asking this information to the ResourceAgents, and evaluate the quality of the component service using an heuristic. The heuristic takes the pheromone levels, response time, execution time, and service price of the component service into account.

Each ExplorationAnt has a survival time, stopping its execution after it. The ExplorationAnts return the gathered information, in the form of a service path, to theOrgAgent, which, in turn, evaluates these information selecting the best path. Once the OrgAgent decides to engage in a particular service path, it sends out IntentionAnts to walk along this path. IntentionAnts drop pheromones along the composition path, indicating the OrgAgent’s intention to use the services along that particular path, reinforcing then the quality of the services along the path.

3. MIDDLEWARE PROTOTYPE

We implemented our decentralized coordination mechanism, described in the previous section, in a middleware prototype. The middleware provides tools to abstract the coordination mechanisms, located in the middle layer, away from the messages inspection and service binding, done at the bottom layer. The middleware prototype was implemented as a service engine in the Apache ServiceMix 1 Enterprise Service Bus (ESB). It can be seen as the middle layer located between the composite service layer and the web-services middleware layer, as depicted in Figure 1.

In the current prototype the service overlay network is created based on configuration parameters given to the middleware at setup time.

---

1http://servicemix.apache.org

4. CONCLUSIONS

In this paper we show our approach for selecting component services to adapt service compositions. In our approach virtual ants can explore different service paths and drop information, called pheromones, about the quality of these paths. This pheromone information is then used by other ants to perform their own searches for the best component services to participate in their compositions. That way, all the entities can make use of the aggregated information available in the overlay network.

The main contribution of this work is to show how to combine two different service selection strategies, more specifically a proxy-based and a collaborative-based approaches, into a decentralized coordination mechanism, that takes advantages of both approaches.

As future work we will investigate alternatives to the creation of the service overlay network. We will also investigate the active networking paradigm, which is also suited to work in dynamic environments.

5. ACKNOWLEDGMENTS

This research is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund K.U.Leuven.

6. REFERENCES


