SLA-based routing for EAI middleware

A step towards self-optimizing BPM

Yves Caseau

Bouygues Telecom, F-Boulogne-Billancourt
ycaseau@bouygues telecom.fr

ABSTRACT. This topic of this paper is the management and optimization of business processes within an integration middleware. These issues arise when IT is organized and measured according to the key business processes, using components that participate to the processes through their connection to a global infrastructure (EAI) and the orchestration of a process flow engine (BPM). We study the impact of routing and control strategies and show that it is possible to demonstrate a self-adaptive and self-healing behavior. Defining the QoS (Quality of Service) of a process is a matter of throughput, latency and availability. Throughput measures the number of individual processes that can be performed in a fixed amount of time like a day. Latency is the time taken for the end-to-end execution of a single process, which is most often what the customer sees and reflects the quality of service that is perceived. Availability is the percentage of the time when the system is able to start a new process. Meeting such a SLA (Service Level Agreement), and even monitoring this SLA, is a hard task because the engineering (sizing and capacity analysis) is done on the individual systems and the infrastructure, whereas the QoS is measured on chains of connected systems. However, the reverse-engineering from the SLA at the process level is even harder. We have coined a new name, OAI, Optimization of Application Integration, for this challenge. The first step of our research project was to study the impact of the message passing strategy. We used a finite-event simulation with multiple scenarios that reproduce burst (congestions) and system failures to evaluate the robustness of different approaches. The first lesson that one may draw from these experiments is that priority routing works. The four algorithms that use process priority as part of the sorting strategy are able to maintain the SLA of high priority processes much better than the first four algorithms. The second lesson is that First-Come-First-Served is not a good default algorithm. On the other hand, sorting messages according to expected completion times (from the SLA) is the best overall approach when the event flow is doable by the EAI system, while the combination of priority and SLA sorting is the best approach. The other approach to self-adaptive middleware is to control the message flow. We found that control rules are interesting, but they are secondary to the routing policy: it is more efficient to deal with congestion problems with a distributed routing strategy rather than with a global rule schema. Rules should complement the routing scheme and since there is no value in doing the same work twice, the priority handling is best done at the component level.

KEYWORDS: (web) service level agreement, self-optimizing, autonomic, middleware, message passing, EAI, business processes, BPM.