Chapter 13

Control Engineering for Scaling Service Oriented Architectures

Yixin Diao  
IBM, USA

Joseph L. Hellerstein  
Google, USA

Sujay Parekh  
IBM, USA

ABSTRACT

Scaling Service Oriented Architectures (SOAs) requires a systematic approach to resource management to achieve service level objectives (SLOs). Recently, there has been increasing use of control engineering techniques to design scalable resource management solutions that achieve SLOs. This chapter proposes a methodology for scaling SOAs based on control engineering. The methodology used here extends approaches used in scaling software products at IBM and Microsoft.

INTRODUCTION

Service Oriented Architectures (SOAs) provide a way to build applications that facilitate re-use and sharing. This is accomplished by using web services (or other similar techniques) to construct applications from services deployed in a computing infrastructure so that the applications are themselves services that can be used by other applications. For example, an email application could be structured to use services for message presentation, database look-up, and full text search. Further, if the email application itself exposes a web services interface, this application might in turn be used by a workflow system to route work requests.

While appealing, SOAs present significant challenges, especially in scaling. A central concern is the extent to which an SOA application is assured that the services it depends on will deliver the performance required for the application to meet its service level objectives (SLOs). For example, an email server built on an SOA architecture may have SLOs for client access to their mailboxes, but achieving these SLOs requires that the database service provide suitably short
response times for database queries. Although there are always concerns about the performance of embedded components in complex applications, SOAs present new challenges because the underlying services may experience rapid changes in resource demands due to concurrent use by multiple applications.

This chapter proposes an approach to engineering scalable resource management solutions for SOA environments. The proposed approach is forward thinking in that it is an extrapolation of our experience with building resource management solutions for software products, especially IBM’s DB2 Universal Database Server and Microsoft’s .NET. Our definition of scalability is quite broad. It encompasses the traditional perspective of solutions that scale to higher request rates. But it also includes scaling in terms of the diversity of requests and operating environments, such as the challenges faced by the .NET Thread Pool that is deployed on almost all of the one billion computers running the Windows Operating System. Because it is difficult to obtain accurate models of real world systems, our approach to scaling resource management does not depend on having detailed system models. Rather, we rely on simple performance models and employ feedback control to sense and correct for errors in modeling and control as well as changes in workloads and resources. Thus, our approach relies heavily on control engineering to design effective closed loop systems such as is done in mechanical, electrical, and aerospace engineering.

To illustrate the challenge of resource management in SOAs, we begin with an example of SOA scaling as reported in James (2008). The service described in this article automatically constructs professional video from still photographs and sound tracks. The service is deployed on a multi-layer SOA consisting of an Application Layer, a Management Layer, and a Cloud Layer. The Application Layer was developed by Animoto, a company founded in 2006. The animation application consists of services providing video analysis, music analysis, customization, and rendering. The Management Layer, which is provided by RightScale, maps the application to the underlying computing and storage infrastructure. This requires considerations for application provisioning, load balancing, scaling the number of servers, and application deployment. The Cloud Layer, which uses Amazon.com EC2 and S3 services, provides the compute and storage infrastructure.

Each layer has its own set of resource management objectives, according to its purpose. At the Cloud Layer, the objectives are to maximize the utilizations of computing and storage resources. The objectives of the Management Layer are to minimize the cost of the deployment by minimizing the number of servers allocated and minimize application response times. The latter can be achieved through load balancing, but this objective is sometimes in conflict with the objective of minimizing costs. At the Application Layer, the objectives are to manage the trade-offs between resource demands (compute cycles and storage) and service quality. In this SOA, quality metrics include response time to produce the video and the quality of the rendering.

The foregoing illustrates the complex interactions that take place in multi-layer SOAs to achieve SLOs. Dealing with these interactions in an ad hoc way often results in scaling issues that degrade service quality and demand costly human interventions by service providers. Control engineering provides a systematic approach to dealing with these challenges. In particular, control engineering is a way to analyze complex dynamic systems and design robust feedback-based solutions to achieve scalable resource management in a dynamic and diverse environment like SOA (Astrom, 2006).

The approach we propose is forward thinking in that it is based on our experience with engineering resource management solutions of the component technologies of SOAs (e.g., databases and operating system concurrency), but not SOAs per se. However, our belief is that the same fundamentals apply to SOAs as well. Also, having scalable
Related Content

A Framework on Data Mining on Uncertain Data with Related Research Issues in Service Industry
Edward Hung (2011). *Service Intelligence and Service Science: Evolutionary Technologies and Challenges* (pp. 114-128).
[www.igi-global.com/chapter/framework-data-mining-uncertain-data/47359?camid=4v1a](www.igi-global.com/chapter/framework-data-mining-uncertain-data/47359?camid=4v1a)

Adoption of a Comprehensive Web-Based Wealth Management Service
[www.igi-global.com/article/adoption-comprehensive-web-based-wealth/46069?camid=4v1a](www.igi-global.com/article/adoption-comprehensive-web-based-wealth/46069?camid=4v1a)

Computational Intelligence for Information Technology Project Management
Robert J. Hammell, Julie Hoksbergen, James Wood and Mark Christensen (2010). *Intelligent Systems in Operations: Methods, Models and Applications in the Supply Chain* (pp. 80-104).
[www.igi-global.com/chapter/computational-intelligence-information-technology-project/42656?camid=4v1a](www.igi-global.com/chapter/computational-intelligence-information-technology-project/42656?camid=4v1a)

Toward a Service (Eco)Systems Perspective on Value Creation
[www.igi-global.com/article/toward-service-ecosystems-perspective-value/71942?camid=4v1a](www.igi-global.com/article/toward-service-ecosystems-perspective-value/71942?camid=4v1a)