Towards High Maturity in SaaS Applications Based on Virtualization: Methods and Case Study

Shijun Liu, Shandong University, China
Yong Zhang, Shandong University, China
Xiangxu Meng, Shandong University, China

ABSTRACT

This paper proposes a high level maturity in SaaS applications based on virtualization technologies, in which the hardware virtualization focuses on providing the flexible on-demand hosting environment, the application virtualization used to meet the scalable application deployment requirements and data virtualization solves data integration and unified data access problems to ensure multi-tenant efficiency. In order to realize the configurability, a method based on metadata and JMX is used to manage the metadata so that changed metadata can be hot deployed immediately during runtime. A case study of a SaaS application for supply business in automobile industry is introduced with the discussion of approaches to higher level maturity model.

Keywords: Maturity Model, Metadata, Service Computing, Software as a Service, Virtualization

INTRODUCTION

In modern manufacturing field, such as automobile industry, an entire manufacturing process often cooperated by assembly factory and part suppliers. Agile Supply Chain Management increasingly becomes an effective and important measure to enhance competitive advantage of enterprises that needs the support of agile information system to integrate their supply chain more effectively and quickly (Tarokh et al., 2007).

With the rapid development of software and network technology, SaaS (Software as a Service) (Dubey & Wagle, 2007) has been widely accepted as a popular way to carry out the software service delivery. SaaS applications have been adopted by more and more enterprises. Despite the slumping worldwide economy, the adoption of SaaS is growing and evolving in the enterprise application software markets, including customer relationship management (CRM), enterprise resource planning...
(ERP) and supply chain management (SCM) (Guseva, 2008).

SaaS model offers many advantages for software customers. Instead of software licenses, maintenance and operational costs that occur in the traditional on-premise model, companies consume IT-services in SaaS model can use the software on demand, just like they would use any other utility such as electricity or water. SaaS user pays only for the usage of the software for example in a pay-per-usage model.

One central advantage of the SaaS model for hosting providers is that they can exploit economies of scale by offering the same software as a service to different customers in the same computing environment and thus increase the utilization of the corresponding IT infrastructure. In order to be consumed in a one-to-many way, the software must be built following the model of a multi-tenant architecture. Multi-tenancy means that the software behaves for every tenant as if each customer was running a separate instance of the software. However, since each customer is different, the application must be customizable to meet the needs of each particular customer. In general, applications can be scaled up by moving the application to a larger, more powerful server and scaled out by running the application on more servers. SaaS applications have to serve a very large number of concurrent users, so scaling out is almost always the best way to add capacity. The ability to rapidly scale a SaaS application can often mean the difference between success and failure for a SaaS provider.

A well-designed SaaS application is generally distinguished by mainly three qualities: scalability, configurability, and multi-tenant efficiency (Chong & Carraro, 2006). But to reach the high level of maturity several problems must be solved. Firstly, to scale a SaaS application to an arbitrarily large number of customers, the number of servers, storages and instances on the back end must could be increased, decreased or migrated as necessary to match demand, without requiring additional re-architecting of the application. Secondly, to provide high configurable to a SaaS application means not only the individual customized interface but also the functions and data structures. Moreover, how to custom is different if the instance is described from different scratch lines. Multi-tenant-efficient means changes or fixes to a SaaS application can be rolled out to thousands of tenants as easily as a single tenant, which depends on an apparent, distributed and liable hosting environment. The solution of these problems requires a new framework and technology, where virtualization technology could be used.

In this paper, we provide an architecture to ensure high maturity of SaaS applications, in which metadata and JMX (Java Management Extensions) (Beng & Sullins, 2003) are used to realize the configurability, scalability is discussed both in application layer and data layer, and data virtualization is the applied to maintain the accuracy and consistence in transparent data storage.

RELATED WORK

Chong and Carraro (2006) presents four maturity levels for SaaS applications. In the first level, each customer has his own customized version of the hosted application, and runs his own instance of the application. At the second level of maturity, an instance of the application is deployed for every tenant. This means that for every tenant the same code is deployed separately, but can be configured on a per-tenant basis through configuration files. In level 3 the code is only deployed once for all tenants and additional metadata files are deployed for each tenant that configures the code to behave in a tenant-specific manner. At the fourth and final level of maturity, the vendor hosts multiple customers on a load-balanced farm of identical instances, with each customer’s data kept separate, and with configurable metadata providing a unique user experience for each customer.

According to Chong’s definition, the key attributes of SaaS architectures are configurability, multi-tenant efficiency, and scalability. Each
Related Content

The Use of Mobile Applications in Shopping: A Focus on Customer Experience
www.igi-global.com/article/the-use-of-mobile-applications-in-shopping/178492?camid=4v1a

The Legal Implications of Cloud Computing
www.igi-global.com/chapter/legal-implications-cloud-computing/70145?camid=4v1a
Hybridization of SOM and PSO for Detecting Fraud in Credit Card
[www.igi-global.com/article/hybridization-of-som-and-pso-for-detecting-fraud-in-credit-card/182657?camid=4v1a](www.igi-global.com/article/hybridization-of-som-and-pso-for-detecting-fraud-in-credit-card/182657?camid=4v1a)

[www.igi-global.com/article/developing-and-validating-measurement-instrument-for-various-aspects-of-digital-economy/217439?camid=4v1a](www.igi-global.com/article/developing-and-validating-measurement-instrument-for-various-aspects-of-digital-economy/217439?camid=4v1a)