

A Scalable Multi-Tenant Architecture for Business Process Executions

Milinda Pathirage, WSO2 Inc., Sri Lanka

Srinath Perera, WSO2 Inc., Sri Lanka

Indika Kumara, WSO2 Inc., Sri Lanka

Denis Weerasiri, WSO2 Inc., Sri Lanka

Sanjiva Weerawarana, WSO2 Inc., Sri Lanka

ABSTRACT

Cloud computing, as a concept, promises cost savings to end-users by letting them outsource their non-critical business functions to a third-party in pay-as-you-go style. However, to enable economic pay-as-you-go services, the end-users need Cloud middleware that maximizes sharing and support near-zero cost for unused applications. Multi-tenancy, which let multiple tenants to share a single application instance securely, is a key enabler for building such a middleware. On the other hand, Business processes capture Business logic of organizations in an abstract and reusable manner, and hence play a key role in most organizations. This paper presents the design and architecture of a scalable Multi-tenant Workflow engine while discussing in detail the potential use cases of such architecture. Primary contributions of this paper are motivating workflow multi-tenancy, and the design and implementation of a scalable multi-tenant workflow engine that enables multiple tenants to run their workflows securely within the same workflow engine instance without modifications to the workflows. Furthermore, the workflow engine supports process sharing and process variability across the tenants and discusses its ramifications.

Keywords: Business Processes, Cloud, Multi-Tenancy, Service Oriented Architecture (SOA), Workflows

INTRODUCTION

By enabling computing power and storage as a utility that is consumed on demand, Cloud Computing has made a significant impression. Many believe that it can bring major savings

following the footsteps of electricity, which is one of the major success stories of Utilities (Carr, 2005). These savings can come in many forms like outsourcing, on-demand resource provision, economics of scale, pay-as-you-go, etc. Let us look at a few such benefits.

For most organizations, IT infrastructure incurs significant costs while supporting non-

DOI: 10.4018/jwsr.2012040102

key functions of those organizations. However, outsourcing non-key functions and focusing on competitive advantages are considered the guiding principal for organizations. Therefore, most organizations want to outsource their IT functions as much as possible. Enabling outsourcing by moving IT functionality to the Cloud is one of the key promises of the Cloud.

Cloud comes in three flavors: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and Business Process as a Service (BPaaS). Since SaaS comes as customizable turnkey solutions, it is the most promising approach for outsourcing among the three. Examples of such SaaS applications are GMail, Salesforce etc., where end users can use them with almost zero operational effort. However, this model does not work for all applications. For specific applications like Customer Relationship Management (CRM) or Email, where behavior is well-defined and universally accepted, SaaS applications have greater chances for matching the exact scenarios of the users. However, most IT infrastructures have custom behaviors that do not fit with off-the-shelf SaaS applications. To handle such cases, user can go to IaaS. However, IaaS can only takeoff Hardware maintenance cost, which is anyway a small fraction. Therefore, IaaS users have to build, deploy, and run the applications themselves. PaaS provides a middle ground by enabling users to run custom applications in a third-party hosting environments.

Example of such applications are a Web Application, J2EE application, Web Service, Workflows, or a combination that runs in a third-party hosting environment. There are many kinds of PaaS services available ranging from databases like database.com to messaging services like Amazon Simple Queuing Service (SQS) to Windows Azure and Google AppEngine.

Due to factors like millions of online users, mobile devices, Facebook and viral marketing, the Slashdot effect--that is a large spike of requests for an application in a small period of time--has become common. Conventional wisdom is to allocate resources for systems

for maximum load, but with steep load spikes, such allocations are unrealistic and wasteful. On these settings, Cloud provides an alternative solution by letting the users allocate resources on demand.

However, SaaS, PaaS and above cases have pay-as-you-go model as a key assumption. That is a user can ask for resources, use them only when he needs them, and release resources when he does not. If this assumption holds, unused applications should cost the user almost nothing. For this model to sustain, such unused applications should also cost almost nothing for the application provider as well.

We argue that to support the pay-as-you-go model, both SaaS and PaaS middleware should be able to support applications owned by many users (we call them tenants) within the same server while allocating resources on demand. It is possible to do this through IaaS where one can run a VM per each user. However, many of the applications and users are not active (in use) often. If a hosting provider has 10,000 tenants and if only few hundred are in use at a given time, then always running a VM for each is a waste. Since booting-up a VM often takes time and does not complete fast enough to serve the first request, keeping VMs in disk and booting VMs on demand is often impractical.

As a solution to this problem, multi-tenancy has been proposed and is considered a key enabling technology for the Cloud. The idea is that the same server instance can support multiple tenants, and that enables hosting organizations to mix and match heavily-used and lightly-used tenants together. This enables them to run the overall infrastructure with much less resources. In summary, with multi-tenancy, the unit of sharing is smaller than a virtual machine, enabling increased sharing of resources.

SOA plays a major role in Enterprise IT infrastructures. For example, according to 2009 ZAP survey (Saat & Discher, 2008), about 50% of organizations had already used SOA, while more than 25% more planned to use it within the next year. Therefore, supporting SOA platform as a service that enables users to outsource their existing SOA applications through a PaaS could

19 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

www.igi-global.com/article/scalable-multi-tenant-architecture-business/70388?camid=4v1

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Computer Science, Security, and Information Technology. Recommend this product to your librarian:

www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

Test Pair Selection for Test Case Prioritization in Regression Testing for WS-BPEL Programs

Lijun Mei, Yan Cai, Changjiang Jia, Bo Jiang and W.K. Chan (2013). *International Journal of Web Services Research* (pp. 73-102).

www.igi-global.com/article/test-pair-selection-for-test-case-prioritization-in-regression-testing-for-ws-bpel-programs/86263?camid=4v1a

Enhancing the Testability of Web Services

Daniel Brenner, Barbara Paech, Matthias Merdes and Rainer Malaka (2009). *Managing Web Service Quality: Measuring Outcomes and Effectiveness* (pp. 223-244).

www.igi-global.com/chapter/enhancing-testability-web-services/26081?camid=4v1a

WSMoD: A Methodology for Qos-Based Web Services Design

M. Comerio, F. De Paoli, S. Grega, A. Maurino and Carlo Batini (2010). *Web Services Research for Emerging Applications: Discoveries and Trends* (pp. 16-44).

www.igi-global.com/chapter/wsmoD-methodology-qos-based-web/41516?camid=4v1a

XML Data Binding for C++ Using Metadata

Szabolcs Payrits, Péter Dornbach and István Zólyomi (2012). *Innovations, Standards and Practices of Web Services: Emerging Research Topics* (pp. 232-249).

www.igi-global.com/chapter/xml-data-binding-using-metadata/59926?camid=4v1a