A Survey on Workflow Middleware

Emad Heydari Beni

University of Antwerp
[firstname]@heydari.be

Abstract

A workflow consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information[8]. The Workflow paradigm has been adopted in a wide range of software systems in order to make business logic more flexible, manageable and agile towards initial design and future changes. With the advent of Cloud Computing, enterprises have been making efforts for migration or deployment of their workflows and their middleware in the Cloud. On the way to achieve this goal, some challenges should be considered and planned, such as multi-tenancy, scalability, load balancing, hybrid Clouds, failure management, security and some other aspects. In this paper we present several leading workflow middleware with short overview of their features and we elaborate on those, which have support for deployment in a Cloud environment. Furthermore, we discuss some of the aforementioned challenges in more details.

I. Introduction

Workflows are used to declare some activities in order to federate disparate systems and services. Over the last decades, lots of commercial and scientific workflow execution and management engines have been developed and shipped to the production. Workflows are defined using process execution languages such as *BPEL* and are usually combined with business ruling systems to detect business situations for taking actions to the raised events.

With evolution of various computing paradigms from local machines to Grids and Clouds, the tendency of middleware migrations to the larger scale set-ups is an undeniable trend. In this paper, we survey the current scientific and industrial leading state-of-the-art workflow middleware, and briefly introduce them mentioning those with Cloud deployment support. Afterwards, we present the necessary challenges of workflow cloud migration that need to get addressed such as multi-tenancy, scalability, failure management, and hybrid workflows.

[1]Business Process Execution Language.
[2]Business Rule Management System

II. Workflow Middleware

A workflows is declared by a process execution language such as BPEL[1]. BPEL is specifically used for specifying actions within business processes with web services. Another example of a process definition language can be *XML Process Definition Language* (XPDL). In contrast to execution perspective of BPEL, XPDL contains information about the graphical description of the processes model, which is mainly designed for exchanging process definitions in-between different workflow vendors.

Workflows are usually combined with business ruling systems. In a nutshell, business rules detect business situations as events, and workflows take actions to the raised events. There are basically two types of rule engines: inference rule engines, and reactive rule engines. The former is classical conditional ruling system and the latter is based on event-driven architecture. There are various standards for interaction with rule engines. One of the famous, industry de facto, specifications is *JSR-94*, which is implemented by jBoss Drools, Ora-

cle Business Rules, Fico Blaze Advisor BRMS[2], and many other tools.

I. Leading Technologies of workflow middleware

Workflows are instantiated and executed within the workflow management systems (WfMS). WfMS's are for running, orchestrating and monitoring of business workflows using aforementioned definition languages. There are large number of workflow middleware either open-source or commercial ready to use with different features. In this section we introduce some of them.

JBPM, part of the jBoss family, is an open-source workflow engine with BPMN[3] and jPDL[4] as its process definition languages. It enables users to create, deploy, execute and manage business processes along with Graphical representation of their models. Another example is *Activiti BPM Platform* which is developed by the key developers of jBPM with more solid design. There are smooth supports for Spring framework, JTA, and Cloud scalability as well as asynchronous continuations for this promising workflow engine.

The Stardust Process Engine is a relatively new workflow system based on WfMC reference model. It is fully implemented in Java and J2EE with pluggable security layer meaning that it is integrable into arbitrary existing security infrastructures. It supports concurrent execution of activities as well as transactions with two-phase commit.

Microsoft Workflow Foundation (WF) is another workflow system with an in-process engine within .NET Framework. Workflows can be defined with XAML[5] or one of the .NET languages. In addition to common features among workflow engines such as scheduling and activities' execution, it supports workflow persistence meaning that it is capable of persisting idle workflows' data in a datastore and reload them in given time which increases the scalability.

There are other vendors with proven workflow management systems such as Oracle BPM of the Fusion middleware, Apache Orchestration Director Engine (ODE), Bonita BPM and many other solutions.

II. The Cloud-based workflow middleware

There are several enterprises with Cloud-based workflow management system solutions federating disparate (Cloud) services in order to enable their users for various types of workflows. In this section we briefly present a few of them.

Amazon Simple Workflow is the cloud-based workflow management solution of the Amazon. Beyond the ordinary features of the workflows management systems, you can benefit from a fault-tolerant service with durable workflow state persistence capability. Besides, workers and deciders (i.e. ruling system) are stateless which makes them scalable. Workers should be polled and checked by Workflows and the result should be sent back to the Amazon. Then, in this way you can leverage your workflow to your in-house systems. *Data Pipeline* is the other workflow solution of the Amazon for specific purpose. This workflow service is a data-oriented engine designed for data movement in-between Amazon compute and storage services or on-premise datastores at specified points. Using its Cloud-native services along with its fault-tolerant, repeatable workflow architecture it enables data scientists to develop data-intensive projects.

Microsoft Workflow Manager is another example of a cloud-based workflow management system with elastic scalability. The programming model, run-time and activity methodologies are based on Windows Workflow Foundation(WF). It comes with multi-tenant environment with ability to organize and manage tenants of the services. It has the capability

[3]Business Process Model and Notation
[4]Native process definition language of jBPM
[5]Extensible Application Markup Language

of updating the process definitions, ensuring the integrity of the running instances. Furthermore, It supports (out/in)bound messaging capabilities with guarantee of safe delivery done by eventually consistency model relying on Service Bus. Inbound messages are being sent repeatedly as long as there is no message from Notification endpoint. Afterwards, the messages are saved in Service Bus to get delivered to the other workflows. When a workflow receives a message, the execution of its process can be continued. Reliable messaging can also be delegated to BizTalk server which is better for integration and long-running transactions.

There are other vendors trying to provide Cloud-based solutions. Salesforce Workflow and Approvals, Effektif, and KiSSFLOW are the cases of these efforts. Amongst open-source, scientific and domain-independent workflow management systems, *Taverna* is an example of a system which benefits from Amazon computing nodes and Grid Computing.

III. Workflows in the Cloud

In this section we briefly discuss some of the probable challenges one may face while deploying workflows in a Cloud environment.

I. Challenges of Cloud deployment

Scalability and Load Balancing. One of the prominent features of the Cloud-based services is scalability. As a matter of fact that Cloud promises unlimited resources including storage and processing power, there should be the possibility of leveraging the infrastructures. And, this is achievable by having horizontally scalable application-level components alongside the lower levels. Design decision regarding database, messaging medium, and the workflow engine should be as decentralized as possible While developing a workflow for a cloud environment. The smaller pieces of workflows such as workers, services and ruling engines should be stateless to be capable of working in parallel. When scalability is in place, Load Balancing is also considerably important.

Multi-tenancy. This is basically about serving multiple tenants (i.e. users) using a single instance of a software. This principle brings several benefits to providers. It cuts the upfront and operational costs by consuming underutilized hardware resources along with the simplification of maintenance. Multi-tenancy can be either done in infrastructure-level or application-level. The latter should be considered for the deployment of the workflows in the Cloud. Every tenant has some requirements and in order to meet those needs, a high degree of per-tenant customizability is considerably important. Since numerous tenants share the same instance of the application or perhaps data store, performance isolation (QoS) and security should be also taken into consideration.

Failure Management. Since the Cloud-based solutions aims to provide services with minimum human involvement, autonomous workflows in terms of Self-diagnosis, self-healing, self-configuration using intelligent algorithms and Control Systems concept (i.e. autonomic control loops) should be developed to address mentioned challenges. Engines should be resilient to failure and able to restart activities if necessary using QoS-aware approaches. In order to be capable of resuming a workflow after a failure or even for a disaster recovery, *Event Sourcing pattern* should be utilized, meaning that a workflow engine can durably store and keep track of workflow states, and in case of failure it can resume the processes or transactions using the existing event logs.

Hybrid Clouds. Some workflows within enterprises due to different types of reasons such as security or data localization with respect to regulatory compliances should not be entirely outsourced to the public Clouds. Sometimes there are steps in workflows that are in-house hardware-dependent. The workflows and their middleware should support service variability, and service selection based on run-time performance KPI's[6] or fluctuating costs of on-

[6]Key Performance Indicator

demand services. These cluster of features cannot be done without strong interoperability strategies using hybrid Clouds.

There are some other challenges such as enhancement of existing Workflow engines in terms of security, multi-tenancy and process definition changeability.

IV. Related works

There are large number of related works targeting the Workflows or similar contexts to address mentioned challenges in this paper.

[1] proposes a reference service framework for integrating scientific workflow management systems into various Cloud platforms consisting of Cloud resource management, resource scheduling, Swift scientific workflow management system integration. [2] presents the design and architecture of a multi-tenant workflow engine ensuring the secure execution of workflows within the same engine. [3] proposes a decentralized BPEL workflow engine using continuation-passing style. [4] developed a middelware called AdaptUbiFlow for service selection and adaptation in workflows.[5] presents a scheduling algorithm for Cloud-based workflow applications with tasks which are geographically distributed. They claims that data transmission is the bottleneck which can be addressed by variability of workflow data dependencies with respect to the cost and execution time. They used Active BPEL engine and Amazon EC2 for their experiments. [6] discusses strategies based on multi-commodity flow problems for handling quality of service and performance isolation of workflows in hybrid Cloud. [7] developed a multi-tenant SaaS application using jBoss workflow technologies (i.e. jBPM) focusing on customizability of tenant-specific requirements. They consider various challenges caused by jBPM shortcomings or design flaws such as centralized workflow engine, lack of concurrent executions of paths, and some other challenges.

V. Conclusion

In this paper, we have reviewed the current leading workflow middleware technologies, and those with support of Cloud-based deployment. Moreover, we discussed the challenges and the concepts that should be taken into account for workflow deployment in a cloud environment such as scalability, load balancing, multi-tenancy, failure management, and hybrid Clouds. To conclude, a set of related works in the same context is presented.

References

[1] Yong Zhao, Youfu Li, Ioan Raicu, Shiyong Lu, Cui Lin, Yanzhe Zhang, Wenhong Tian and Ruini Xue. *A Service Framework for Scientific Workflow Management in the Cloud*. Services Computing, IEEE Transactions, 2014.

[2] Milinda Pathirage, Srinath Perera, Indika Kumara, Sanjiva Weerawarana. *A Multi-tenant Architecture for Business Process Executions*. Web Services (ICWS), IEEE International Conference, 2011.

[3] Weihai Yu. *Running BPEL Processes without Central Engines*. Tapir Akademisk Forlag, 2007.

[4] Frederico Lopes, Thiago Pereira, Everton Cavalcante,Thais Batista, FlÃĆÂt'via C. Delicato, Paulo F. Pires, Paulo Ferreira. *Selection and Adaptation in Workflows for Ubiquitous Computing*. Embedded and Ubiquitous Computing (EUC), 2011.

[5] Ernst Juhnke, Tim DÃűrnemann, David BÃűck, Bernd Freisleben. *Multi-objective Scheduling of BPEL Workflows in Geographically Distributed Clouds*. IEEE CLOUD, page 412-419. IEEE, 2011.

[6] Hendrik Moens, Eddy Truyen,Stefan Walraven,Wouter Joosen, Bart Dhoedt, Filip De Turck. *Network-aware impact determination algorithms for service workflow deployment in hybrid clouds*. International Con-

ference on Network and Service Management, USA 2012.

[7] Fatih Gey, Stefan Walraven, Dimitri Van Landuyt, Wouter Joosen. *Building a Customizable Business-Process-as-a-Service Application with current State-of-Practice*. International 12th International Conference, Budapest 2013.

[8] Glossary. *Business Process Management (BPM)*. Center of Excellence (CoE) 2009.