



Revista Ingenierías Universidad de Medellín

ISSN: 1692-3324

Universidad de Medellín

Ruiz-Paz, Sergio Fabián; Santaolaya-Salgado, René; Fragoso-Díaz, Olivia G.

Model of dynamic orchestration for SaaS*

Revista Ingenierías Universidad de Medellín, vol. 16, no. 31, 2017, July-December, pp. 143-153

Universidad de Medellín

DOI: <https://doi.org/10.22395/rium.v16n31a7>

Available in: <https://www.redalyc.org/articulo.oa?id=75055115008>

- How to cite
- Complete issue
- More information about this article
- Journal's webpage in redalyc.org

UNEM
redalyc.org

Scientific Information System Redalyc

Network of Scientific Journals from Latin America and the Caribbean, Spain and Portugal

Project academic non-profit, developed under the open access initiative

Model of dynamic orchestration for SaaS*

Sergio Fabián Ruiz Paz**
René Santaolaya Salgado***
Olivia G. Fragoso Díaz****

Received: 14/12/2016 • Accepted: 12/05/2017
DOI: 10.22395/rium.v16n31a7

Abstract

Cloud Computing is an ubiquitous model that enables clients to access different services in a fast and easy manner. In this context, one of the most used models is *Software as a Service* (SaaS), which means that software is deployed and provisioned to the customer via internet through a web browser on a pay per use mode. However, given its complexity and characteristics, such as reusability, scalability, flexibility and customization, SaaS may be defined by workflows, which consist of atomic services, or micro-services hosted geographically in different places. SaaS execution under this type of composition may lead to abnormal behavior or failures in the end user applications at runtime. This paper presents a new model of dynamic orchestration for SaaS, which aims to reduce failures or abnormal behavior of the services involved in the execution process of business application.

Keywords: atomic services; cloud computing; dynamic orchestration; microservices; software as a service; workflow.

* Research article in progress. Origin: "Dynamic Orchestration Model of Business Processes in the Cloud Computing". Date of realization: August 2014-July 2018. Financing entities: Conacyt, TecNM/Cenidet in Mexico, under the project: "SOA Service Oriented Architectures Applied for Improving E-Learning Technology" 5517.15.16-P-P.

** M. Sc. Department of Computer Science, National Center for Research and Technological Development, Cuernavaca, México. fruiz@cenidet.edu.mx

*** Ph. D. Department of Computer Science, National Center for Research and Technological Development, Cuernavaca, México. rene@cenidet.edu.mx

**** Ph. D. Department of Computer Science, National Center for Research and Technological Development, Cuernavaca, México. ofragoso@cenidet.edu.mx

Modelo de orquestación dinámica para SaaS

Resumen

La nube computacional como modelo ubicuo, permite el suministro de servicios a clientes que acceden a ella de forma fácil y rápida. El Software como Servicio (SaaS) es uno de los modelos de mayor uso, mediante el cual las aplicaciones se despliegan y aprovisionan por los clientes vía internet mediante un navegador web en un modelo de pago por uso. Sin embargo, por su complejidad y características —reusabilidad, escalabilidad, elasticidad y personalización— el SaaS es definido por flujos de trabajo compuestos de microservicios o servicios atómicos alojados geográficamente en diferentes lugares. El SaaS en este contexto puede presentar comportamientos anormales en los resultados o fallos en la aplicación final del usuario en tiempo de ejecución. En este artículo se presenta un modelo de orquestación dinámica cuyo objetivo es disminuir las fallas o comportamientos anormales de los servicios que participan en el proceso de ejecución de las aplicaciones de negocios.

Palabras clave: servicios atómicos; nube computacional; orquestación dinámica; micro-servicios; software como servicio; flujos de trabajo.

Modelo de orquestração dinâmica para SaaS

Resumo

A computação em nuvem é um modelo onipresente que permite o fornecimento de serviços a clientes que têm acesso a ela de forma fácil e rápida. O software como serviço (SaaS) é um dos modelos de maior uso, por meio do qual os aplicativos se estendem e armazenam pelos clientes via internet, com um navegador web pago por uso. Contudo, por sua complexidade e características — reuso, escalabilidade, elasticidade e personalização —, o SaaS é definido por fluxos de trabalho compostos de microserviços ou serviços atômicos alojados geograficamente em diferentes lugares. Nesse conteto, o SaaS pode apresentar comportamentos anormais nos resultados ou falhas na aplicação final do usuário em tempo de execução. Neste artigo, apresenta-se um modelo de orquestração dinâmica, cujo objetivo é diminuir as falhas ou os comportamentos anormais dos serviços que participam do processo de execução dos aplicativos de negócios.

Palavras-chave: computação em nuvem; fluxos de trabalho; microserviços; orquestração dinâmica; serviços atômicos; software como serviço.

INTRODUCTION

Cloud Computing has become a new technological paradigm that is being widely used by the information technology and telecommunications industry because of the easy manner in which it provides services on demand, elasticity, flexibility and computing resources. Cloud Computing has a wide range of technological solutions, supported by three deployment models: *Software as a Service* (SaaS), *Platform as a Service* (PaaS) and *Infrastructure as a Service* (IaaS), for the industry, institutions, organizations, small, medium and large scale business; as well as for hundreds of thousands of individual users. In order to meet the users' objectives, improving productivity, reducing costs and improving its services among other benefits in a model "pay per use", these models facilitate access on demand in a public or private, free or paid mode. All these resources are provided without any service installed in the user's local environment, eliminating the cost and effort for installing, maintaining and updating services.

SaaS, as reported in [1-3], is the most productive service of Cloud Computing, it is commonly used today by consumers and given its impact on the economy and on solving business problems, it is predicted to have a considerable growth in the upcoming years. Nevertheless, the current complexity of SaaS produced by the need to satisfy a growing and dynamic social demand of processes in economy, technology and culture of Cloud users, has brought significant challenges to ensure that applications deployed in the SaaS model meet satisfactorily users' needs.

As mentioned in [4], services orchestration "is one of the main elements to exploit the potential of Cloud Computing". Specifically, it is a primordial activity in the management of services, it should be present in the stages of deployment, provisioning and implementation of SaaS in such a dynamic environment like the one presented in the cloud, in the delivering of services. However, the overhead for running the orchestration, as well as scalability of heterogeneous resources and other constraints related to the environment –such as coordination, allocation and coordination of resources at services– remains a problem to be addressed.

Composition and behavior of SaaS

SaaS was previously thought as a generalist software by most providers, with the intention to respond to a great number of users with related requirements on any domain context. However, the growing demand for custom software for specific domains requirements, such as economics or cultural environments, lead to more complex software development.

To provide solutions to their users, SaaS deploys services of small, medium and large scale. As mentioned in [5], small-scale SaaS services also called atomic services

or micro-services, have the ability to provide the features of reusability, scalability, flexibility and personalization in a workflow schema, which matches a user business process. In this manner, it makes available the advantage of using micro-services in manual or automatic composition, based on alternative workflows that form part of an application of a business process to services providers. Thereby, reducing costs for maintenance, development effort and solutions deployment time. On the client's side the advantages are: short time service delivery, cost reduction due to pay per use mode, and short terms in investment return.

Based on the above description, SaaS includes three concepts that are worth describing. The first one is *Services Composition*: the composition of a service from multiple services or software packages expressed in certain logic, according to a workflow [6], this logic follows a business process that meets the requirements of a certain user. The second one is *Workflows*, which are sets of computational steps that need to be run by third-party packages, and that can be represented as software components or functions [6]. These packages are commonly known as atomic services or micro-services, as mentioned in [5]. Finally the third one is *Automatization of Services*, which refers to one of the complex tasks in services' management, and which supposes the capability that services may have to adapt to Cloud Computing (elasticity, extensibility), automatically and effectively, to the current needs of users, as mentioned in [7]. Those three concepts together give rise to the need for establishing atomic services or micro-services with the degree of self-sufficiency necessary to meet a unique responsibility, to be lightly coupled, isolated, highly available, scalable, reusable, reactive, and with a high throughput.

In order to meet the user demands, the SaaS model initiates when a client executes an application (service) in an Internet environment using a web browser. Given the characteristics and architecture that running services have, normally the services trigger a set of calls to other services which have been previously defined by the workflow, either from the same supplier or other service providers.

Also as mentioned in [8], each software service can act as a service provider or a service requester. SaaS provider exposes the functionality of its services to other applications via public brokers. All these processes are done in order that a service can incorporate both data and functionality from other services. Both user and service providers can benefit from the economy of scalability in deployment and management of SaaS services. This gives rise to a sequence of requests from SaaS services and resources to achieve effective execution of an application, which is seen as a user service request for a specific task.

Participant services can be of different providers and may have different structures (e.g. different formats for export or import data, API's industry standard provider or API's private from another vendor) that define a heterogeneous service scheme and can promote abnormal behavior in the performance of a service.

Abnormal behaviors of SaaS services are the main challenges for services and research providers, given that these behaviors impact on the degree of service availability which is a main feature of the SaaS model and one of the main objectives to be achieved in Services Level Agreements (SLA).

The availability of SaaS is directly related to the degree to which a service can be used [9]. The elements related to the degree in which a service can be used are resources used by the atomic services or microservices as part of a workflow application business process and the interoperability of services and capacity that they have, as well as the capacity that they have of performing the function for which they were created.

Therefore, searching, selecting, recommending and coordinating of virtualized and non-virtualized services as well as coordinating and allocating of necessary resources for the proper performance of services in a dynamic way are some of the challenges to be addressed by researchers in this field.

Orchestration of services and its challenges

The orchestration of services is one of the main elements to exploit the potential of Cloud Computing since it can involve different types of services (software, computer, storage and network), and can be performed in different layers of the model of Cloud Computing depending of the domain context of interest. Thereby enabling the effective preparation and adaptation of the services used by the user. The way services are considered and coordinated by the orchestration depends on the place where they are required.

The term *service orchestration* is closely related to the concept of *service choreography* due to the close relationship that both keep in services composition, due to the complex collaboration among services. This relationship has been presented from the Service Oriented Architecture (SOA) up to SaaS in Cloud Computing. However, there is a strong difference among them. Service choreography as mentioned in [10], is a global description of the participating services, which is defined by the exchange of messages, rules of interaction and agreements between two or more endpoints. On the other hand, the concept of service orchestration, described in detail in the following section and according to [10], represents a single centralized executable process. The orchestrator (executable process) is responsible for invoking and combining the

services. Generally, service orchestration can be set on to levels as indicated in [4]: Orchestration of Software Services (SaaS) and Orchestration of Infrastructure Services (IaaS).

The SaaS consists of establishing the business logic and tasks execution order, and can cover the applications and organizations to define a transactional model of multi-step long-term process. Since services are associated with physical resources where they are executed, orchestrating SaaS includes organization, coordination and management of deployment, and automated configuration of one or more necessary and interrelated components for the provision of services in a given time. For that reason it is important to analyze a series of technical services management and failure recovery aspects.

The IaaS involves the creation and manipulation of resources of Cloud Computing (e.g. computation, storage and network) and consists mainly of effectively coordinating the deployment of a set of services in a virtualized infrastructure, in order to meet operational and quality objectives for the end users and Cloud Computing providers.

One of the current challenges in SaaS according to [4], is the dynamic allocation of resources that ensures Services Level Agreement (SLA) in the Cloud resources to make profitable decisions when placing the workload and dynamic coordination of resources, so as to optimize the use of resources and services while reducing overall response time of each service and resource.

In the context of Cloud Computing where our proposed solution is located, service orchestration involves the choreography of services. That is, it is required in the preparation and adaptation of the services required in a workflow that establishes the rules of interaction and agreements for effective communication between business process services.

This paper presents a model of dynamic orchestration for services, which adheres to the dynamic cloud computing environments. This model ensures the correct execution of services that are part of the workflow corresponding to a business process in the SaaS model.

The rest of this paper is organized as follows: Section 1 describes the proposed model of dynamic orchestration for services, as well as the features that help to reduce failures in business process applications deployed in the SaaS model. Section 2 describes the analysis and discussion of results of the dynamic orchestration model based on the SaaS delivered solutions. Finally, section III presents the findings of the proposed model as well as some remaining challenges to be addressed.

1. MODEL OF DYNAMIC ORCHESTRATION

In the previous section we presented some challenges for SaaS when services do not comply in a timely manner or do not produce the results expected by the user, this may happen due to the lack of service availability in the presence of different factors affecting performance.

This section presents a dynamic orchestration model of services that form workflow sequences of business applications. The goal of this model is to reduce failures of services involved in the workflow of business process. This model of dynamic orchestration of services is sensitive to behavioral factors from the domain applications.

The intention is to apply the model to monitor the system forces presented in the microservices that participate in workflows and collect information associated with the business processes of the users at runtime, such as:

- Factors Related to Services: Cost, processing time, responsiveness for service calls, reputation.
- Service Independent Factors: Bandwidth, latency servers, throughput, service availability, number of reentrant service requests.
- Service Performance Factors: In the service selection this concept refers to the assessment of certain quality measures of services from the applications domain, such as: service reliability, service response time, frequency of use, frequency of failures, cost of service.

From the pairing of this information, the necessary adjustment to the workflow previously defined is generated, ensuring the proper functioning of services and meeting users objectives.

Figure 1 describes the process of dynamic orchestration of services that participate in the proposed model.

The model of dynamic orchestration of services consists of several steps and coordinates services to ensure the proper functioning of SaaS. The steps may be defined as follows:

- The dynamic orchestration model takes as input the workflows of business process (SaaS) from an application.
- Each atomic service or microservice participating in a workflow is analyzed to determine its behavior.

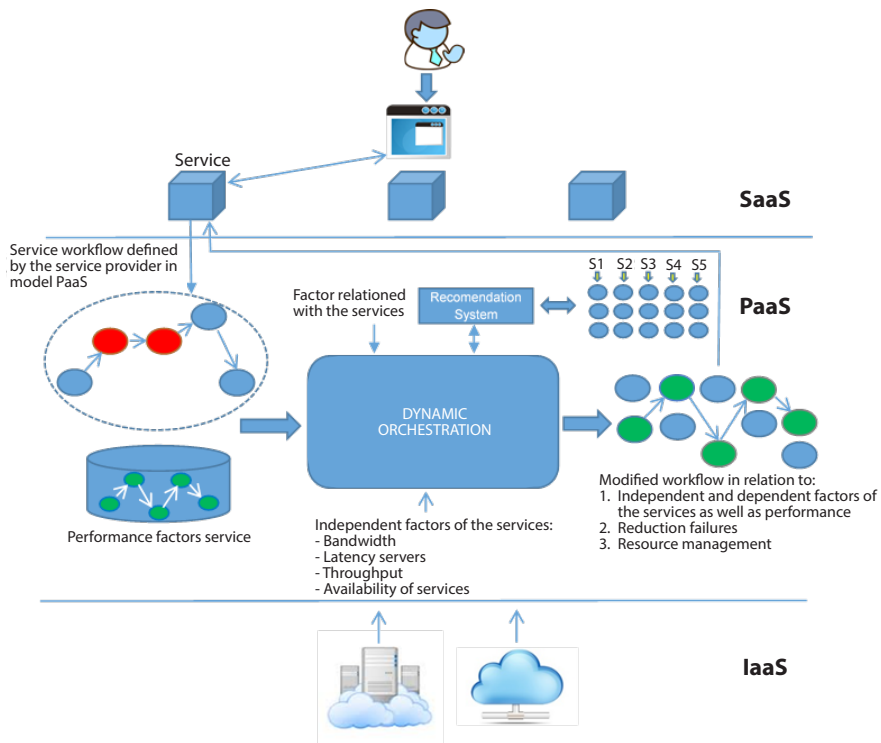


Figure 1. Model of Dynamic Orchestration

Source: authors

- The analysis results of each atomic service determine the search and selection of candidate atomic services that may participate in the workflow.
- Once the selection of candidate atomic services is established, immediately a recommendation module analyzes the information on their static, dynamic and behavioral factors to define their behavior in the workflow.
- Services with the best features are selected and recommended for the dynamic orchestration.
- From the candidate atomic services, the orchestration model determines the workload of available resources, the resources allocated to each service, as well as the performance of the service to determine the probability of failure of each of them, facilitating the selection of alternative candidate services for the workflow.
- Once the integration of selected atomic services is established, the orchestration model performs the control flow and generates the internal reconstruction of the document of services composition to establish a new version of the process

workflow of business application. This articulated process is integrated with the microservices that at the moment, have the best condition.

2. ANALYSIS AND DISCUSSION OF RESULTS

The proposed dynamic orchestration model in this paper is addressed specifically to support the successful execution of services involved in the execution process of business applications, with the aim to reduce failures presented in SaaS runtime when the user is interacting with the services. To achieve this aim, the model takes into account dependent and independent factors of the services as well as certain important measures of service performance factors, as described in detail in section 1. To control these factors may have a significant impact in the quality attribute known as Degree of Utilization which is a part of the services feature named Availability, which all services should have as mentioned in [9], in the Service Performance section. All this in order for the dynamic orchestration model to control the execution of services by identifying and orchestrating only those services that have a significant degree of utilization, assuring with it that the user faces true availability of services.

Moreover, the proposed model performs a dynamic orchestration which guarantees the availability of services in a workflow at runtime, therefore supporting the execution of service during user interaction. Likewise, an alternative execution plan is also provided by the model, further ensuring the successful execution of the service as discussed in the needs to address the orchestration of services, mentioned by [4] and explained in section 2.

3. CONCLUSION

Due to the dynamic environment that Cloud Computing presents, it is difficult to accurately predict changes in the environment that may affect SaaS status and that may compromise their performance.

Abnormalities and errors in the performance of services must be controlled, and techniques and strategies to control them should be defined accordingly.

The orchestration models also depend heavily on what point of the orchestration (IaaS, PaaS or SaaS) of services requires maintaining control and at what stage (development, deployment, or execution) of the service and the Cloud Computing model it becomes necessary to perform the orchestration.

In SaaS the challenge that must be addressed is the availability of services, so as to ensure the operation at runtime of the applications of the service user.

There is a need for ensuring that atomic services or microservices of workflows business processes are actually available at runtime, and that they meet certain performance criteria, according to the available infrastructure and the features offered by them, as well as specific user needs.

Thus, the proposed model of dynamic orchestration of services becomes an important factor in SaaS performance monitoring. On the one hand, it allows to use a proactive technique that anticipates failure, choosing those microservices that must be replaced when failures present, and those that will be replaced by analyzing the attributes that are presented in services at runtime. On the other hand, it also allows service management regarding the allocation of resources by employing techniques of workloads and workflows management to ensure proper execution.

The model of dynamic orchestration of services is sensitive to static, dynamic and behavioral factors, for proper and effective coordination of atomic services or microservices workflow business processes.

In this model of dynamic orchestration of services, the techniques used for planning, recovery, monitoring and recommendations aim at reducing overload in the model execution. However, evaluating the effectiveness of the model is not an objective in this research.

An important part of this research is the automatization of the model, which is obtained by developing monitoring modules and recommendation of sensible services to static and dynamic factors.

Some of the challenges that remain on the orchestration of services in a dynamic environment are:

- Dynamic orchestration of services with version control of atomic services or microservices facing the presence of abnormalities at runtime.
- An effective dynamic light orchestration of services to make proper use of the resources at runtime.
- A dynamic orchestration of services that is automatically activated during deployment, provisioning and implementation of atomic services or microservices.
- Dynamic orchestration of services must consider all the conditions established between the users and services provider in the SLA.
- Dynamic orchestration of services that consider the migration of virtual instances, virtual machines and physical machines that host microservices must be coordinated during their deployment, provisioning and execution.

REFERENCES

- [1] IBM Center, "Champions of Software as a Service: How SaaS is fueling powerful competitive advantage," Armonk, NY, pp. 1-11, 2014.
- [2] M. Mistretta, "El estado actual del *cloud computing* en México," *InformationWeek*, septiembre, n.º 230. pp. 16-20, 2013.
- [3] Cisco, "Cisco Global Cloud Index : Forecast and Methodology, 2014–2019," White Pap., pp. 1–41, 2014
- [4] K. Bousselmi, Z. Brahmi, and M. M. Gammoudi, "Cloud Services Orchestration: A Comparative Study of Existing Approaches," in 2014 28th International Conference on Advanced Information Networking and Applications Workshops, pp. 410–416, 2014.
- [5] J. Lewis and M. Fowler, "Microservices," *martinfowler.com*, 2014. Available: <http://martinfowler.com/articles/microservices.html>, March 25, 2014.
- [6] J. Wang, P. Korambath, I. Altintas, J. Davis, and D. Crawl, "Workflow as a Service in the Cloud: Architecture and Scheduling Algorithms," *Procedia Computer Science*, vol. 29, pp. 546–556, 2014.
- [7] K. Klai and H. Ochi, "Model Checking of Composite Cloud Services," in 2016 IEEE International Conference on Web Services (ICWS), pp. 356–363, 2016.
- [8] J. M. Myerson, "The role of Software as a Service in cloud computing SaaS maturity," no. April, pp. 1–8, 2009.
- [9] European Commission, "Cloud Service Level Agreement Standardization Guidelines," *Digital Agenda for Europe*. [Online]. Available: ec.europa.eu/digital-agenda/en/news/cloud-service-level-agreement-standardisation-guidelines, June 2014.
- [10] J. Terpak, P. Horovcak, and M. Lukac, "Mathematical models creation using orchestration and choreography of web services," 2016 17th Int. Carpathian Control Conf., pp. 739–742, 2016.