International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Survey on Service Management Mechanisms in Service Oriented Computing

Anish Bhagwat

[1]Pune Institute of Computer Technology, Pune, Maharashtra, India

Abstract: ***Software applications have evolved over time and the Service based applications are the latest trend in IT industry today. Such applications are characterized by combination of loosely coupled services which work together to provide the required functionalities. These applications need to absorb and adapt to the changing requirements at runtime to ensure business continuity. In order to do so, the key job is to monitor the services, detection faults and apply possible solutions to resolve the fault. In this paper, we discuss various approaches that are in place to tackle such kind of situations.***

Keywords: service-oriented computing, service-oriented architecture, monitoring, cloud computing

1. Introduction

One of the most emerging paradigm for distributed computing is the Service Oriented Computing (SOC). It is responsible for bringing about a major change in how the software applications are designed, constructed and delivered [1]. It basically combines the set of approaches that represent computing in a Service Oriented Architecture (SOA) which is an architectural pattern that delivers utility of an application as a service via some underlying protocol. In such a computing scenario, services are the key independent, loosely-coupled elements that can be combined together to build collaborative applications in a distributed environment. This independent nature of services creates a room for supporting changes in the service configurations during run-time. The dynamic reconfiguration and provisioning of services allows for creating adaptable and evolvable systems.

The main examples of such kind of service oriented computing in practice today include cloud services, web services, grid services, etc. Cloud services are the most recent among these in which applications are dynamically provided as a service, rather than as a product, to the end users over the network based on the user demands. Although these services are separated by their varied applicability, the one thing they share in common is their dynamic nature which gives rise to an important challenge of managing these services effectively during the run time so that they can adapt to the changes in business environments and efficiently address the new user demands. The traditional approaches of quality assurance during the development phase need to be replaced by new ones which can ensure operation time quality of service.

Such a need to ensure better quality standards at runtime gave rise to a new practice called the IT Service Management (ITSM) which aims at providing services and reliable support allowing smooth functioning of the organization's business processes. The major intention behind this approach is to change the perspective of looking at IT infrastructure as a group of services on the whole rather than individual software / hardware components. ITSM aims at managing the delivery of services with respect to the changing needs of organization.

For proper service management of IT services, it is essential for the service to be flexible enough to adapt to the changes in organization's operating environment which give rise to new requirements. For adaption to changes, the key element is monitoring of the existing services. A monitoring system, as its name suggests, observes the behavior of a service to ensure its conformance with the specifications provided. Today, any organization that rely on IT services for its operation do have a monitoring system in place which continuously keeps on tracking the services to check if its performance matches with the requirements specified ensuring better service quality levels. It essentially is used to track faults in service, provide run-time organization and gather data that are important from the application's evolution point of view. The focus of this paper is on monitoring part of the IT service management (ITSM) practice.

Most of the monitoring systems used today are capable of observing the service and alerting if any kind of fault / warning is detected in the service being monitored. Also, it updates back when the service is back to normal state of operation. However, between this period of time it is required for a person to diagnose the provisioned service, find the root cause of the fault detected, decide on the corrective action that can be taken against the fault and then eventually make the changes to fix the fault & bring the service back to normal state. The human intervention required to perform this service diagnosis consumes time and hence may affect the ongoing business operations that use this specific IT service.

In this paper, we discuss about a variety of work done towards designing systems that provide better service management through different approaches.

2. Literature Survey

Over the last decade, a lot of work has been done on designing and developing systems that can provide effective

service management for fault detection and monitoring. Most of this work focuses on detection of faults in a service.

Web services provide means of web communications between applications for providing greater availability. In such a context, where multiple services depend on each other to deliver a single utility to the end user, any event of failure in a specific service or even performance degradation could greatly affect the user's experience. [2] discusses the issue of service failure detection and replacement focusing on the time of replacement and the frequency of failure.

With the advancement in technology, virtualization came into existence gradually. The work done in [3] focuses on building a service oriented monitoring framework using REST that monitors and reports faults in physical and virtual infrastructure.

Service oriented architecture is an architectural pattern in which loosely coupled services work together to provide some functionality. These architectures support runtime evolution of software through reconfiguration of misbehaving services. A spectrum based fault localization technique is normally used for automated service diagnosis. In [4] a novel method was presented to improve service diagnosis by monitoring not only the services but also the invocation links between these services. This work shows to what extent can the service diagnosis be improved by inclusion of invocation links in monitoring.

Approach discussed in [5] addresses the problem of monitoring converged services. Converged services can be thought of as an integration of traditional telecommunication services and web services. This paper focuses on the new challenges such converged services face from their management perspective. A runtime monitoring scheme is proposed which first uses code instrumentation techniques to understand the structure of the converged services and then accordingly apply fault detection.

All the above work has made an important contribution towards providing a better service management by improved monitoring and fault detection. However, each of this approach still requires human intervention at some point of time for service management.

3. Summary and Conclusion

The recent advancements in technology have resulted in a rapid shift towards a service oriented computing. In such a scenario, providing efficient service management has arose as an important need in the cloud computing environments.

The earlier work done as discussed above focuses mainly on monitoring strategies and fault detection. However, with current state of art, there is a need for a self-healing type of an automated system which would prove efficient for service management on the large scale cloud environments.

References

[1] Saralaya, S. and D'Souza R., "A Review of Monitoring Techniques for Service Based Applications", in 2nd International Conference on Advanced Computing, Networking and Security (ADCONS), IEEE 2013.
[2] Pereira E. and Pereira R., "Fault monitoring and detection of distributed services over local and wide area networks", in 12th International Conference on Parallel and Distributed Systems, Vol. 2, IEEE 2006.
[3] Katsaros G., Kubert R. and Gallizo G., "Building a Service-Oriented Monitoring Framework with REST and Nagios", in International Conference on Service Computing (SCC), IEEE 2011.
[4] Cuiting Chen, Gross H.G. and Zaidman A., "Improving Service Diagnosis through Invocation Monitoring", in 13th International Conference on Quality Software (QSIC), IEEE 2013.
[5] Adrada D., Salazar E., Rojas J. and Corrales J.C., "Automatic Code Instrumentation for Converged Service Monitoring & Fault Detection", in 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE 2014.
[6] P. Falcarin and L. Walter, "An Aspect-Oriented Approach for Dynamic Monitoring of a Service Logic Excecution Environment," IEC Annu. Rev. Commun., vol. 59, pp. 237–242, 2006.
[7] N. Goel and R. K. Shyamasundar, "Automatic Monitoring of SLAs of Web Services," in Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific, 2010, pp. 99–106.
[8] C. Chen, H.-G. Gross, and A. Zaidman, "Spectrum-based fault diagnosis for service-oriented software systems," in Proc. of the Int'l Conf. on Service-Oriented Computing and Applications (SOCA). IEEE, 2012.
[9] C. Chen, A. Zaidman, and H.-G. Gross, "A framework-based runtime monitoring approach for service-oriented software systems," in Int'l Workshop on Quality Assurance for Service Based Applications (QASBA). ACM, 2011, pp. 17–20.
[10] L. Ardissono, R. Furnari, A. Goy, G. Petrone and M. Segnan, "Fault tolerant web service orchestration by means of diagnosis," Proc. 3rd European Conf. Software Architecture (EWSA 06), Springer, 2006, pp. 2-16, doi:10.1007/11966104_2.
[11] N. Delgado, A. Q. Gates, and S. Roach, "A taxonomy and catalog of runtime software-fault monitoring tools," IEEE Trans. Softw. Eng., vol. 30, Dec. 2004, pp. 859–872, doi: 10.1109/TSE.2004.91.

Author Profile

Anish Bhagwat received the B.E. in Information Technology, in 2012 from PIIT, New Panvel affiliated to University of Mumbai. Perusing M.E. degree in Computer Engineering from Pune Institute of Computer Technology, Pune affiliated to Savitribai Phule Pune University.