

Microservices Data Mining for Analytics Feedback and Optimization

Kindson Munonye, Budapest University of Technology and Economics, Hungary

Péter Martinek, Budapest University of Technology and Economics, Hungary

ABSTRACT

When microservices-based architectures are adopted for an enterprise application, a basic requirement would be an evaluation of the performance with the objective of continuous monitoring and improved efficiency. This evaluation helps businesses obtain a quantitative measure of the benefits of a shift from monolith to microservices. Additionally, the metrics obtained could be used as a mechanism for continuous improvement of production application. This research proposes a model based on the principles of data mining called stream analytics feedback and optimization (SAFAO), which can be used to achieve a continuous optimization of microservices. Stream analytics is due to the fact that the analysis is performed on online application with continuously generated lived data. This approach has been tested in a simulated production environment based on Docker containers. The authors were able to establish empirical measures which were continuously extracted via a data mining methodology and then fed back into the running application through configuration management. The results show a continuous improvement in the performance of the microservices as indicated in the results presented in this research.

KEYWORDS

Analytics, Containers, Data Mining, Machine Learning, Microservices, Optimization

1. INTRODUCTION

A key aspect of enterprise applications is metrics collection which in turn is used to determine performance indicators that can be tuned and improved to provide higher software quality. For a monolith, these metrics could easily be collected via one entry point, for example a bootstrap *url* or application log files. In the case of microservices, however, there are now an increased number of interconnections and intercommunication between processes. A computation that once took place within a single process could now be fragmented into many micro-computations across different services, processes and even threads (Christudas, 2019).

In a monolith architecture, method invocations or interactions between processes are localized allowing for processes to make assumptions and optimizations for complete execution of its operations. However, for the same architecture in microservices, implementations and possible deployment is split into processes (Bakshi, 2017) and even locations. This increases the number metrics collection channels available for the application thereby requiring an enhanced approach to the metrics collection and feedback.

DOI: 10.4018/IJEIS.2021010102

Copyright © 2021, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

This is illustrated with the Hospital Information System (HIS) scenario in Figure 1 which shows a system of four components and a single entry point, the Admissions Service. Since this entry point interfaces with other services, it becomes a trivial task to gather system-wide metric from this service. This is because all needed data gathering and optimization could be achieved via a single interface. Data stream is sure to reach the remaining three services in a synchronous manner.

In the case of a microservice architecture, each component could be considered as a full-fledged independently deployable application and therefore can generate its own dataset. For example, in Figure 2, the metrics generated by the Patients services *http://path1:port(1)* need to be evaluated relative to the metrics generated by the Admissions service at *http://path3:port(3)* and so on. Then these data somehow needs to be fed back into the system for purpose of optimizing system performance.

Therefore, the objective of this research is to develop an approach that can be used to integrated data generation, collection, analysis, feedback and optimization in a microservices architecture. This model adopts a data mining approach for microservices monitoring, data collection, analysis and feedback for performance metrics tuning.

This research is arranged as follows: chapter 2 provides an overview of the theoretical and architectural framework of data mining in applications. Chapter 3 discusses the concept of metrics gathering and performance tuning for microservices. Some existing research are presented as well.

Figure 1. Single entry point for monoliths

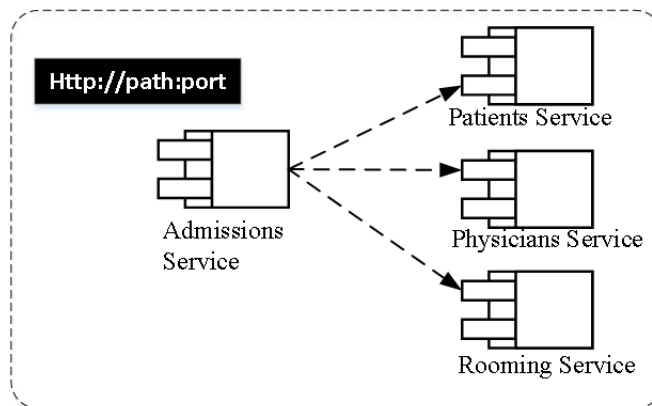
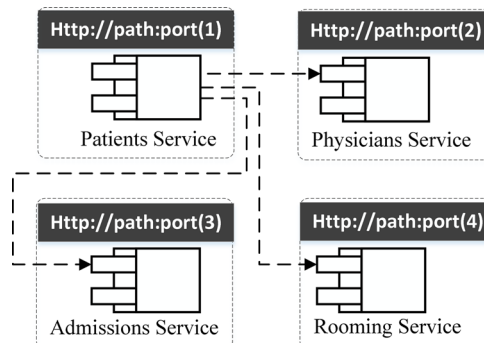


Figure 2. Multiple channels for –services architecture



20 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/microservices-data-mining-for-analytics-feedback-and-optimization/268361

Related Content

An ASP-Based Product Customization Service Systems for SMEs: A Case Study in Construction Machinery

Yan Su, Wenhe Liao, Yu Guo and Shiwen Gao (2008). *International Journal of Enterprise Information Systems* (pp. 1-17).

www.irma-international.org/article/asp-based-product-customization-service/2132

Managing Complexity and Institutionalization: Power and Politics in ERP Implementation in an Australian University

Brian Corbitt, Konrad Peszynski and Olaf Boon (2005). *Qualitative Case Studies on Implementation of Enterprise Wide Systems* (pp. 160-175).

www.irma-international.org/chapter/managing-complexity-institutionalization/28250

An Investigation of Enterprise Resource Planning Implementation in a Small Firm: A Study of Problems Encountered and Successes Achieved

Magdy Abdel-Kader and Thu Phuong Nguyen (2013). *Competition, Strategy, and Modern Enterprise Information Systems* (pp. 57-80).

www.irma-international.org/chapter/investigation-enterprise-resource-planning-implementation/70319

RFID Adoption: Issues and Challenges

E.W.T. Ngai and Angappa Gunasekaran (2009). *International Journal of Enterprise Information Systems* (pp. 1-8).

www.irma-international.org/article/rfid-adoption-issues-challenges/3947

IT Development

Andrew Targowski (2003). *Electronic Enterprise: Strategy and Architecture* (pp. 290-320).

www.irma-international.org/chapter/development/9668