Aleksi Pekkala

Migrating a web application to serverless architecture

Master's Thesis in Information Technology

February 1, 2018

University of Jyväskylä

Department of Mathematical Information Technology

Author: Aleksi Pekkala

Contact information: alvianpe@student.jyu.fi

Supervisor: Oleksiy Khriyenko

Title: Migrating a web application to serverless architecture

Työn nimi: Web-sovelluksen siirtäminen serverless-arkkitehtuuriin

Project: Master's Thesis

Study line: Master's Thesis in Information Technology

Page count: 13+0

Abstract: This document is a sample gradu3 thesis document class document. It also functions as a user manual and supplies guidelines for structuring a thesis document.

The abstact is typically short and discusses the background, the aims, the research methods, the obtained results, the interpretation of the results and the conculsions of the thesis. It should be so short that it, the Finnish translation, and all other meta information fit on the same page.

The Finnish tiivistelmä of a thesis should usually say exactly the same things as the abstract.

Keywords: serverless, FaaS, architecture, cloud computing, web applications

Suomenkielinen tiivistelmä: Tämä kirjoitelma on esimerkki siitä, kuinka gradu3-tutkielmapohjaa käytetään. Se sisältää myös käyttöohjeet ja tutkielman rakennetta koskevia ohjeita.

Tutkielman tiivistelmä on tyypillisesti lyhyt esitys, jossa kerrotaan tutkielman taustoista, tavoitteesta, tutkimusmenetelmistä, saavutetuista tuloksista, tulosten tulkinnasta ja johtopäätöksistä. Tiivistelmän tulee olla niin lyhyt, että se, englanninkielinen abstrakti ja muut metatiedot mahtuvat kaikki samalle sivulle.

Sen tulee kertoa täsmälleen samat asiat kuin englannikielinen abstrakti.

Avainsanat: serverless, FaaS, arkkitehtuuri, pilvilaskenta, web-sovellukset

Contents

1 Introduction

> It is a good idea to start the Introduction with the main thesis statement or research question of the thesis. After that, it is a good idea to clarify things by defining any necessary terms. Definitions after the thesis statement! Also, don't babble in the introduction. The introduction is also a good place to discuss why your thesis statement is scientifically or practically relevant and interesting. Ideally, it would be relevant and interesting from both the scientific and the practical point of view. It would also be excellent if you explained, in the introduction, what your contribution is; that is, what such knowledge your thesis contains that you have investigated personally instead of reading it from somewhere. The contribution could well be, that you have personally checked the truth of a claim you found in a book or article. At the end of the Introduction, it is customary to briefly explain the structure of the thesis – what each chapter is about.

In a nutshell: serverless computing is a relatively new form of cloud computing that enables efficient resource utilization and elasticity, thus bringing about potential savings in hosting costs and increases in developer productivity (Roberts 2006). As a rather new development there's a lack of research in serverless computing. Multiple authors, e.g. Baldini, Castro, et al. (2017), Fox et al. (2017) and Eyk et al. (2017) note the identification of FaaS-specific patterns (composing functions into larger systems) as an important research topic. The object of this thesis is to fill that gap by a) surveying existing patterns in literature and b) evaluate, extend and propose new patterns by migrating parts of a web application into serverless architecture.

1.1 Research questions

Introduce the following research questions:

1. Why should a web app be migrated to FaaS?

2. What kind of patterns are there for building serverless web application backends?

3. Do the existing patterns have gaps or missing parts, and if so, can we come up with improvements or alternative solutions?

4. How does migrating a web app to FaaS effect its quality?

1.2 Outline

The thesis is structured as follows: the second chapter serves as an introduction to the concept of serverless computing. The chapter describes the main benefits and drawbacks of the platform, as well as touching upon its internal mechanisms and briefly comparing the main service providers. Extra emphasis is placed on how the platform's limitations should be taken into account when designing web application backends.

The third chapter consists of a survey into existing serverless design patterns. The applicability of other cloud computing, distributed computing and enterprise integration patterns is also evaluated in the chapter.

The fourth chapter describes the process of migrating an existing web application to serverless architecture. The patterns discovered in the previous chapter are utilized to implemented various typical web application features on a serverless platform. In cases where existing patterns prove insufficient or unsuitable as per the target application's characteristics, modifications or new patterns are proposed. This chapter forms the exploratory or constructive part of the thesis.

The outcome of the migration process is evaluated in the fifth chapter. The potential benefits and drawbacks of the serverless platform outlined in chapter 2 are used to reflect on the final artifact. The chapter includes approximations on measurable attributes such as hosting costs and performance as well as discussion on the more subjective attributes like maintainability and developer experience.

The final chapter of the thesis aims to draw conclusions on the migration process and the resulting artifacts. The chapter contains a summary of the research outcomes and ends with recommendations for further research topics.

2 Serverless computing

> The goal of the theoretical part of a thesis is to develop the theoretical background required in the thesis. The idea is that a reader of the thesis should, based on just the thesis itself, be able to understand all the special concepts and methods used in the thesis. A good thesis also gives well-argued reasons for why exactly these concepts and methods are in use in the thesis (with the main alternatives given in the literature mentioned).

2.1 What is serverless?

Definition and implications of serverless computing. Main features or tenets. Benefits, use cases, notable users? Roberts (2006) has a thorough description of the platform.

Two historical/evolutionary views: on-premises infra -> grid -> IaaS -> PaaS -> serverless and on-prem -> VMs -> containers -> serverless.

How does serverless relate to concepts such as FaaS, SOA, cloud-native, microservices, FaaS, containers, ...?

2.2 Service providers

Lynn et al. (2017) provide an overview and multi-level feature analysis of seven enterprise serverless computing platforms.

2.3 Execution models and triggers

Describe the two supported execution models, synchronous and asynchronous, and how they relate to application design. The former is used to build a typical request-response flow, e.g. a REST API endpoint, whereas the latter relates to pub-sub and other event-driven flows. Give examples on the kind of triggers supported by serverless platforms (HTTP calls, messaging, database events, ...).

2.4 Drawbacks and limitations

What to take into consideration when migrating to serverless?

- cold start (Lloyd et al. 2018)

- the need for circuit breakers (risk of DDoSing yourself) when interacting with non-serverless components like a database. Mention novel serverless database services like Google's Cloud Spanner and AWS Aurora.

- SLAs

- vendor lock-in

Baldini, Cheng, et al. (2017) identify three competing constraints: functions should be considered as black boxes; function composition should obey a substitution principle with respect to synchronous invocation; and invocations should not be double-billed.

Roberts (2006) and Adzic and Chatley (2017) respectively list a number of limitations.

3 Serverless design patterns

Survey of serverless design patterns. How to compose individual functions into larger systems?

Sbarski and Kroonenburg (2017) introduce the following five patterns: Command, Messaging Priority queue, Fan-out and Pipes and filters.

Enterprise integration patterns by Hohpe and Woolf (2004).

Serverless programming model embraces modularization and isolation.

Adzic and Chatley (2017) suggest embracing the platform by 3 typical aspects: use distributed authorization, let clients orchestrate workflows and allow clients to directly connect to AWS resources.

4 Migration process

Decide on parts to migrate.

A simple REST API to showcase API Gateway and synchronous invokation, shouldn't require any big changes to application code.

Interacting with an external SaaS service like Twilio, Auth0. Demonstrate event-driven invocation.

Something to do with timed events, cron.

An analytics pipeline.

Transactions, eventual consistency. Demonstrate double-billing.

Possibly split into another evaluation chapter? Evaluating costs, performance, etc.

5 Evaluation

Evaluation the outcome of the migration process. Performance, costs, ...?

6 Conclusion

The last chapter of a thesis is the Conclusion (some authors use Conculsions, instead). Keep it short, and discuss what one can conclude about the thesis statement or research question given in the Introduction, in light of all that has been written in the thesis. The Conclusion is also the place to discuss any limitations and weaknesses of the thesis (especially those that cast doubt on the reliability of the results given in the thesis), if they have not been already discussed, for example in a Discussion chapter. It is also customary to state, what further research might be beneficial in light of this thesis.

If the Conclusion threatens to become too long, it is a good idea to split the interpretation of the results into its own chapter, often called Discussion, making Conclusion short and sweet.

Bibliography

Adzic, Gojko, and Robert Chatley. 2017. “Serverless computing: economic and architectural impact”. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering,* 884–889. ACM.

Baldini, Ioana, Paul C. Castro, Kerry Shih-Ping Chang, Perry Cheng, Stephen J. Fink, Vatche Ishakian, Nick Mitchell, et al. 2017. “Serverless Computing: Current Trends and Open Problems”. *CoRR* abs/1706.03178. arXiv: 1706.03178. http://arxiv.org/abs/1706.03178.

Baldini, Ioana, Perry Cheng, Stephen J. Fink, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Philippe Suter, and Olivier Tardieu. 2017. “The Serverless Trilemma: Function Composition for Serverless Computing”. In *Proceedings of the 2017 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software,* 89–103. Onward! 2017. Vancouver, BC, Canada: ACM. ISBN: 978-1-4503-5530-8. doi:10.1145/3133850.3133855. http://doi.acm.org/10.1145/3133850.3133855.

Eyk, Erwin van, Alexandru Iosup, Simon Seif, and Markus Thömmes. 2017. “The SPEC cloud group’s research vision on FaaS and serverless architectures”. In *Proceedings of the 2nd International Workshop on Serverless Computing,* 1–4. ACM.

Fox, Geoffrey C., Vatche Ishakian, Vinod Muthusamy, and Aleksander Slominski. 2017. “Status of Serverless Computing and Function-as-a-Service (FaaS) in Industry and Research”. *CoRR* abs/1708.08028. arXiv: 1708.08028. http://arxiv.org/abs/1708.08028.

Hohpe, Gregor, and Bobby Woolf. 2004. *Enterprise integration patterns: Designing, building, and deploying messaging solutions.* Addison-Wesley Professional.

Lloyd, Wes, Shruti Ramesh, Swetha Chinthalapati, Lan Ly, and Shrideep Pallickara. 2018. “Serverless Computing: An Investigation of Factors Influencing Microservice Performance”. *The IEEE International Conference on Cloud Engineering (IC2E).* Forthcoming.

Lynn, Theo, Pierangelo Rosati, Arnaud Lejeune, and Vincent Emeakaroha. 2017. "A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms". In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom),* 162–169. IEEE.

Roberts, Mike. 2006. "Serverless Architectures". Visited on February 1, 2017. `https://martinfowler.com/articles/serverless.html`.

Sbarski, Peter, and S Kroonenburg. 2017. *Serverless Architectures on AWS: With examples using AWS Lambda.* Manning Publications, Shelter Island.