



What are the benefits of Microservices and how do they compete with other serverless architectures?

Name: Daniel Rhys Newman

Supervisor: Michael Dacey

student number: 1702330

email: danielrhysnewman@gmail.com

1. Introduction

The purpose of this project is to analyse the uses and benefits of microservices. Many programmers argue that microservices are becoming outdated compared to functions as a service systems, and so this project exists to analyse whether or not microservices are still useful to developers. To do this the author has looked at existing literature and created their own microservice system to test the legitimacy of these claims.

4. Methodology

To come to conclusions about the benefits of microservices and how these benefits effect the development process for microservice systems, the following hypotheses were made about microservices.

- Microservices run quickly and require little processing time
- Microservices charge a developer only for the services they use
- Microservices allow developers to cut up work load into smaller chunks
- Microservices are scalable to large website traffic
- Microservices allow developers to focus on main code as server logic is pre-built
- If an error occurred to one service, the other services can still operate
- The more functions that are added, the harder the program is to understand
- higher chance of failure in microservices compared to regular websites
- network issues could lead to complete website outage.

These hypotheses are what will be tested to create the results.

6. Testing

Testing the systems turned out successful, both GET and POST request for each teacher worked as intended. Testing the processing time of each service averages a processing time to 52 milliseconds per process, which is very quick considering these services need to communicate with a server. The longest amount of time required to process a request in the testing was 73 milliseconds and the shortest amount of time required to run a process was 31 milliseconds. The processing time fluctuates between tests but still remains relatively short, showing that microservices are still fast regardless of server communication.

2. Aims and Objectives

Aim:

To investigate the benefits and drawbacks of microservices and see how they effect the design process of a project.

Objectives:

- To conduct a review of literature
- To analyse methods of how to create results
- To use AWS to create microservices
- To discuss findings from the implementation

5. Implementation

In order to implement a microservice system, using AWS Lambda the author created a school system that would hold teacher information and mark them as present when the teacher submitted a request.

To do this a function was created per teacher that would respond to a GET request and a POST request. If a GET request was requested for a teacher it would get that teacher's information (Their name and classroom number). If a POST request was made by a teacher it would post to the server that the teacher was present.

These functions exist on a Amazon Web Service server and undergo a server communication process whenever these functions are requested. The author then placed these functions into a bigger function which makes up the whole microservice.

So this way if a user clicked on a button to get a certain teachers information the server would look through the available functions to choose the most appropriate one.

3. Research Overview

Serverless computing is a popular method of creating functions, used by businesses such as Google, Netflix and Amazon. [1] In these systems the developer pays only for what services run from the server rather than traditionally paying for whole servers entirely.

Microservices are a type of serverless architecture. Some programmers believe it is outdated compared to functions as a service systems. Microservices are smaller chunks of services bundled together to be called when needed from a server. So if a user requests to select an option on a website the microservice containing that function will run.

Functions as a service systems then are smaller microservices, so instead of being larger collections of functions like microservices they instead are just one function per server request. This can save some money for businesses as these services will require less processing time and less processing power needed to run each service.

7. Results

Based on the research conducted about microservices and the statistics gathered from the implementation of the microservice systems the author has gathered a number of results showing the potential and use for microservices as well as some other things to consider.

Microservices do indeed process quickly as shown in the testing section where these services average 52 milliseconds per process.

Based on research and official Amazon Web Services statistics a service that requires 100 milliseconds of processing time running 500MB of memory would need to be processed 1,200,041 times to be the equivalent of \$1USD. In testing as these processes took 52 milliseconds on average and so this figure could be doubled. [2]

In implementation many services were created in separate chunks and then placed together, this proves that microservices allow programmers to split up their workload to make development easier.

Microservices are scalable in terms of gathering more website traffic. Server providers like Amazon Web Services allow more server attention to be spent on services that are receiving more traffic at any given time. As well as this developers can focus their attention away from server side logic as the server providers themselves can issue new updates to be done automatically.

Microservices do not technically become harder to understand the more functions are made for them, in the implementation suitable naming conventions for the functions were chosen which made searching for functions much easier. To add to this, microservices work best the smaller they are.

Though it is true that more errors are likely to occur in a microservice system these will not prevent the whole system from becoming unusable as these functions will be self contained, meanwhile if the same were to happen to a normally written website it would likely all crash at once. This shows one of microservices greatest strengths, fault tolerance.



[1] C. Richardson, 'Who is using microservices', 2020. [Online]. Available: <https://microservices.io/articles/whoisusingmicroservices.html>. [Accessed: 03-May- 2020].

[2] Amazon, 'AWS Pricing', AWS Lambda, 2020. [Online]. Available: <https://aws.amazon.com/lambda/pricing/>. [Accessed: 03-May- 2020].