WEBASSEMBLY AND V8 ISOLATE-BASED APPROACH FOR SERVERLESS COMPUTING ARCHITECTURE

Asanka Lakmal Basnayake

209314P

Master of Science in Computer Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

July 2023

WEBASSEMBLY AND V8 ISOLATE-BASED APPROACH FOR SERVERLESS COMPUTING ARCHITECTURE

Asanka Lakmal Basnayake

209314P

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science in Computer Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

July 2023

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books)

Name: B.M.A.L.Basnayake

Signature: Date: 2023/07/27

The supervisor/s should certify the thesis/dissertation with the following declaration. The above candidate has carried out research for the Masters thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the Supervisor: Prof. Indika Perera

Signature of the supervisor: Date: 27/07/2023

ACKNOWLEDGEMENT

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Prof. Indika Perera, for providing abundant guidance, support, and encouragement throughout this
research. Also, I would like to thank my colleagues for sharing knowledge, providing support, and providing constant encouragement. Also, I would like to thank tech communities and tech articles for sharing knowledge, providing support, and providing constant encouragement.

ABSTRACT

User have the flexibility to execute short-lived applications with the help of serverless computing frameworks without concern about load balancing by scaling zero to large-scale traffic and further without considering configuring platform and runtime software, etc. Stateless containers are used in existing serverless platforms to isolate functions by preventing memory sharing directly with other serverless functions. Duplicating and serialising data repeatedly per new user request hit on the server by invoking VMs and containers leads to high performance and resource usage. The necessity of a light-weighted isolation approach is required to overcome these issues. We introduce WebAssembly and Chrome V8 isolate based, a new isolation mechanism for serverless computing. This isolates the memory of each function using linear memory with software fault isolation (SFI), which is a portable low-level bytecode called WebAssembly (which major browser vendors have collaboratively designed). The proposed runtime for serverless functions uses Chrome V8 Isolate-based isolation and virtualizes file systems with a similar approach to POSIX host interfaces using WebAssembly System Interface (WASI). WebAssembly (Wasm) and Chrome V8-based software fault isolation have been proposed in the study as an alternative method for conventional container-based serverless approaches, with near-native speed, a small memory footprint compared to containers, and optimised invocation time. WebAssembly-based isolation is compared against the container-based approach to validate its applicability within the study.

Keywords: WebAssembly, Serverless, Containers, V8

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
WASM	WebAssembly
API	Application Programming Interface
WASI	Webassembly System Interface
JS	JavaScript
OS	Operating System
POSIX	Portable Operating System Interface
IPC	Inter-Process Communication
BSD	Berkeley Software Distribution
NPM	Node Module Package
VM	Virtual Machine
SFI	Software Fault Isolation
MB	Megabyte
KB	Kilobyte
I/O	Input/Output

LIST OF APPENDICES