

ENHANCED VIDEO STREAMING ARCHITECTURE USING CLOUDLETS

Kande Arachchilage Janitha Praneeth Bandara

(219318P)

Thesis/Dissertation submitted in partial fulfilment of the requirements for the degree
Master of Science in Computer Science

Department of Computer Science & Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

June 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).

Signature:

Date: 26/06/2023

The above candidate has carried out research for the PhD/MPhil/Masters thesis/dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the supervisor: Eng. Prof. Indika Perera

Signature of the supervisor:

Date: 26/06/2023

ACKNOWLEDGEMENT

This research has been made with the intention of deriving a solution in the video streaming world which is improving day by day with the efforts of different fields of expertise. For the enhancement of this research, there were a number of people who spend their time and helped with their capabilities. I would like to offer my heartfelt gratitude to all of them including my supervisor Eng. Prof. Indika Perera, Department of Computer Science & Engineering, University of Moratuwa. And special thanks to my parents and my colleagues who encouraged me all the time.

ABSTRACT

As mobile computing and cloud computing work together to provide new technologies and capabilities to end users, clouds are overloading with data-intensive and compute-intensive processes. Video streaming is such an application trending in the technology world. Video content has been a significant form of communication as well as media for entertainment. With the cloud's potential, editing, and processing video streams can be easily done. Even so, while watching a real-time video stream, a single buffering situation can be distracting to most users. Real-time stream processing is a computationally and memory-intensive operation that is also latency sensitive. Standalone clouds cannot handle such latency.

Cloudlets are important and new architectural components in mobile and cloud architecture, representing the intermediate stage of the Cloudlet-mobile device hierarchy. This thesis investigates to resolve live video stream processing which is computer & memory intensive and can be done smoothly using Cloudlets. Cloudlets can provide a platform for developing and deploying new video editing algorithms and applications, which can be executed on the same video stream to produce different outputs. This approach can provide users with a new level of flexibility and control over their video content, enabling them to create customized streams that match their preferences. The prototype of the system was developed using agile development methods, Java and Python programming languages, and the Google Cloud platform.

The software system is evaluated based on maintainability, usability, and sustainability. The system's architecture has the potential to open up new business models and gain market share in the future. The system can be further improved by making it more automatic and internally running.

The system has the potential to improve the field of video streaming architecture by improving the user experience, and the ability to use resources more efficiently. Its plug-and-play capability will make it easier for users to access and use the platform, which will increase adoption rates and user satisfaction. The system is planned to have certain security elements that will assist in the protection of users' personal information and the prevention of unauthorised access to their accounts. This can help to build trust with users and reduce the risk of data breaches.

Keywords: Cloud Computing, Cloudlets, Video Streaming

TABLE OF CONTENT

Declaration	i
Acknowledgement.....	ii
Abstract	iii
Table of content.....	iv
List of Figures	vii
List of Tables.....	ix
Chapter 1. Introduction	1
1.1. Context of Study.....	1
1.2. Statement of the Problem	1
1.3. Aims	2
1.4. Objectives	2
1.5. The Methodology Chosen	2
1.6. Summary of Chapters	3
Chapter 2. Literature review	4
2.1. Cloud Computing	4
2.1.1. Cloud Architectures (LaaS, PaaS, SaaS).....	4
2.1.2. Cloud Computing Deployment Models	5
2.1.3. Cloud Security.....	5
2.1.4. Cloud Performance.....	6
2.1.5. Cloud Standards	7
2.1.6. Cloud Service Providers.....	7
2.2. Edge Computing and Cloudlets.....	8
2.2.1. Edge Computing and Cloudlets:	8
2.2.2. Existing Cloudlets	9
2.2.3. Cloudlet vs Cloud.....	10
2.2.4. Transient Cloudlet Customization	10
2.2.5. Cloudlet Architectures	10
2.2.6. Cloudlet Resource Management	11
2.2.7. Performance Evaluations Technique.....	12
2.2.8. Cloudlet Deployment and Management Considering Video Streaming	12
2.2.9. Programming Languages, Framework, and Libraries Review.....	13
2.2.10. Cloudlet Usage on Video Streaming Enhancement	13
2.3. Real-Time Streaming	13
2.3.1. Video Streaming.....	14
2.3.2. Limitations in Video Streaming with Clouds.....	15
2.3.3. Security Issues with Cloud-Based Video Streaming	15
2.3.4. Performance Metrics on Evaluating Video Streaming.....	16
2.3.5. Quality of Service (QoS).....	17
2.3.6. User Behaviour and Preferences	17
2.3.7. Optimize Technique for Video Streaming	18

2.3.8.	Content delivery networks (CDNs).....	18
2.3.9.	Video compression algo (H.264, VP9...)	18
2.3.10.	Video transcoding	19
2.3.11.	Live video editing, Image processing	19
2.3.12.	Technologies on video streaming with Cloudlets	20
2.3.13.	Distributed Video Processing.....	20
2.4.	Augmenting Cognition with Cloudlets.....	21
	Kimberley prototype.....	22
	Drawbacks of VM-based Cloudlets.....	22
2.5.	Software development methodology	23
2.6.	Chosen Methodology	24
2.7.	Research Value on Video Streaming with Cloudlets	24
2.8.	Summary	24
Chapter 3.	Methodology	25
3.1.	Introduction	25
3.2.	Requirement Specification	25
	3.2.1. Functional Requirements	25
	3.2.2. Non-Functional Requirements	26
3.3.	System Design	26
	3.3.1. System Architecture	27
	3.3.2. Entity Relationship Diagram.....	32
	3.3.3. Use case diagram.....	33
	3.3.4. Activity diagram.....	33
	3.3.5. Class Diagrams	35
	3.3.6. Wireframe design	36
	3.3.7. Summary	38
Chapter 4.	Development and Implementation	39
4.1.	Introduction	39
4.2.	System implementation	39
	4.2.1. Development Environment	39
	4.2.2. Development Tools and Frameworks	40
	4.2.3. Implementation details	42
	4.2.4. Hardware implementation of the Cloudlet.....	45
	4.2.5. Cloud Services usage	46
	4.2.6. Web Hosting	50
	4.2.7. User Interface	52
	4.2.8. User Loggins implementation	54
4.3.	System Implementation Challenges	56
	4.3.1. Main challenges faced.....	56
	4.3.2. Solution Adopted	57
4.4.	GitHub repositories	57
	4.4.1. Cloudlet.....	57
	4.4.2. Microservice.....	57
4.5.	Summary	58

Chapter 5. Testing and Results	59
5.1. Introduction	59
5.2. Test planning	59
5.2.1. Unit testing	59
5.2.2. Integration testing	60
5.2.3. Rest API Testing	60
5.3. Test procedure	60
5.3.1. Test suite	61
5.3.2. Test Report	62
5.3.3. Error reports	64
5.4. How the prototype works	65
5.4.1. Microservice on the Google Compute Engine	65
5.4.2. Cloudlet application running on the Cloudlet prototype.....	66
5.5. Results	67
5.5.1. Improved User Experience.....	67
5.5.2. Workload Distribution	69
5.5.3. Plug and Play Capability	69
5.5.4. Open for Extension	69
5.5.5. Improved Security	70
Chapter 6. Critical Evaluation	71
6.1. Introduction	71
6.2. Software Evaluation Methodology (Criteria-based)	71
6.3. Achievements	73
6.4. Product Evaluation	73
6.5. User Experience Evaluation	74
6.5.1. Objective of the survey	75
6.5.2. Target Audience	75
6.5.3. Survey Design & Format	75
6.5.4. Results	76
6.6. Quality and Performance Evaluation.....	79
6.6.1. Resolution	79
6.6.2. Frame Rate	80
6.6.3. Start-up Time	82
6.6.4. PNSR.....	82
6.6.5. Conclusion	83
6.7. Future Enhancement.....	84
6.8. Lesson learned.....	84
6.8.1. Development Framework.....	84
6.8.2. Chosen methodology.....	84
Chapter 7. Conclusions	85
References	87
Appendix	91
Appendix 1: Gantt Chart	91

LIST OF FIGURES

Figure 1-1: Stream offers a number of different streams in response to user requests.	1
Figure 2-1 Cloudlet Architecture	8
Figure 2-2 HTTP Live Streaming Architecture [20].....	15
Figure 2-3 Runtime Binding in Kimberley [35]	22
Figure 2-4 The Gabriel Architecture for Cognitive Assistance	23
Figure 3-1 New Architecture	27
Figure 3-2 Simba Cloud Application	27
Figure 3-3 Cloudlet Application	29
Figure 3-4 Frontend Application.....	30
Figure 3-5: Architecture	31
Figure 3-6: Entity Relationship Diagram (ERD) for basic login	32
Figure 3-7: Use Case Diagram (Casper)	33
Figure 3-8: Connect with nearby Cloudlet Activity Diagram.....	34
Figure 3-9: Invoke a plugin Activity Diagram.....	34
Figure 3-10: UML Diagram (Simba)	35
Figure 3-11: UML Diagram (Casper)	35
Figure 3-12: UML Diagram (Plugin System) - Python	36
Figure 3-13: Cloudlet application homepage	36
Figure 3-14: Microservice application homepage.....	37
Figure 3-15: Login	37
Figure 3-16: Register.....	37
Figure 3-17: Streaming page	37
Figure 4-1: Postman	41
Figure 4-2: Video enhancement on Cloudlet	42
Figure 4-3: BlackAndWhitePlugin.py [44].....	43
Figure 4-4: BlackAndWhitePlugin.info	43
Figure 4-5: REST request to involve a plugin.	45
Figure 4-6: Simba showing available Cloudlets.	45
Figure 4-7: Creating Google VM instance.....	47
Figure 4-8: VM Basic Information	47
Figure 4-9: VM Other Settings	47
Figure 4-10: SSH - VM Access	48
Figure 4-11: MariaDB - Database storing user login data.	48
Figure 4-12: GCP Bucket Creating.....	49
Figure 4-13: Files in the GCP bucket.....	49
Figure 4-14: GCP Pub/Sub Creation.....	50
Figure 4-15: Deployment of the Spring Boot application on GCP VM.	50
Figure 4-16: Configure Incoming HTTP Request Using the Firewall.....	51
Figure 4-17: External IP.....	51
Figure 4-18: Website accessible through External IP	52
Figure 4-19: Domain forwarding with GoDaddy.....	52
Figure 4-20: Cloudlet frontend application.....	53

Figure 4-21: Microservice frontend application	53
Figure 4-22: Streaming UI	54
Figure 4-23: Sign-up Page	54
Figure 4-24: Log-In Page	55
Figure 4-25: Invalid user	55
Figure 4-26: Invalid password	55
Figure 4-27: Account Lockout due to multiple tries with incorrect credentials	56
Figure 5-1: Unit Testing	60
Figure 5-2: User connecting with the microservice.	65
Figure 5-3: User interacting with the cloudlet.	66
Figure 5-4: Prototype Plugins	66
Figure 5-5: Video streaming through Cloudlet	66
Figure 5-6: YouTube video containing smoking scenes.....	67
Figure 5-7: Cloudlet filtering PG scenes.....	67
Figure 5-8: Black-And-White plugin	68
Figure 5-9: Ads Plugin	68
Figure 5-10: Face Detection Plugin	68
Figure 5-11: Memory Consumption.....	69
Figure 6-1: User feedback on User Interface	76
Figure 6-2: User feedback on Video Processing Plugins.....	76
Figure 6-3: User feedbacks on Video Plugin Performance.....	76
Figure 6-4: User feedback on Platform.....	77
Figure 6-5: User feedback on Areas that can be improved.....	77
Figure 6-6: User feedback on new plugins	78
Figure 6-7: User feedback on features like to see in the platform.	78
Figure 6-8: User feedback on Device Compatibility	78
Figure 6-9 User feedback on Overall Satisfaction	79
Figure 6-10: Black and White plugin frame rate	80
Figure 6-11: Parent guide frame rate	81
Figure 6-12: Ads plugin frame rate.....	81
Figure 6-13: Face detection plugin frame rate	81
Figure 6-14: Overall frame rate.....	82
Figure 6-15: Start-up time.....	82
Figure 0-1: Project Management (Casper).....	91
Figure 0-2: Project Management (Simba).....	91

LIST OF TABLES

Table 1: Hardware Requirements	39
Table 2: Software Requirements	39
Table 3: Test participant role	61
Table 4: Test suite	61
Table 5 Test case TS001:	62
Table 6: Test case TS002	62
Table 7: Test case TS003	63
Table 8: Test case TS004	63
Table 9: Test case TS005	64
Table 10: Criteria-Based Evaluation Results	73
Table 11: Resolution Test Result	79
Table 12: Frame rate Test Result	80