Control-Ef: A Transcript Search System based on Spring MVC Architecture

Htoo Lwin
Computer Science
Asian Institute of Technology
st120832@ait.asia

Abdul Raheem Fathima Shafana
Computer Science
Asian Institute of Technology
st121985@ait.asia

***Abstract*—The recent COVID-19 pandemic has forced many educational institutions at all levels to conduct lectures online and most of these lectures are recorded for student review and/or administrative purposes. However, navigating through such lectures can be time consuming and hence, this study proposes an online transcript search system that is based on the Spring Boot MVC framework. This study also proposes a database architecture based on PostgreSQL and Apache Cassandra designed to provide consistent high read performance and scalability as the application grows. The performance of this system in searching transcripts is then stress tested rigorously against a "vanilla" architecture containing a simple PostgreSQL database. The results show that for the maximum amount of transcript data (500,000 records) that we tested with, PostgreSQL offers superior read performance to Apache Cassandra. It is believed that this trend will remain until the number of data records reach upwards of a million. This study also put forward that further testing would detect the bottleneck and may be able to find a potentially more robust architecture.**

***Index Terms*—Control-Ef, Video transcripts, Database Performance, Spring Boot, PostgreSQL, Apache Cassandra, Stress Testing**

I. Introduction

Many countries around the world have migrated from traditional face-to-face education to online education with the outbreak of COVID19 pandemic [1]. Recorded videos of lectures are considered as an important educational strategy in such contexts that could potentially improve the breadth and depth of students' learning since it allows students to recall the educational content [2]. However, the overload of information is also witnessed in return that daunt the students to find the appropriate content [2] and the specific content of their interest.

When there is only a limited time for preparing for an examination, watching the entire video of a recorded lecture is quite critical and time-consuming. In that context, the Search Engine Optimization, the search, and navigation to the specific content within the video would assist the students to learn the specific information within a short period of time. Many case studies have also proved that the Search Engine Optimization leverage the students positively towards learning [3]. The availability of transcripts of video lectures helps both hearing impaired and non-native speakers of English in addition to providing an efficient search. [3].

This paper introduces a transcript search system named Control-Ef based on Spring MVC Architecture that assist the students in learning. The objectives of this system are twofold: to help students search and obtain their desired video and to assist in navigating to the exact timestamp with the desired content within the video. Furthermore, the main architectural challenge that this study tries to solve is the performance of the system as it scales up. The actual problems of the students in learning through recorded video is analyzed and designed based on the Software Design paradigm. The developed system is intended assists the students in multitude of ways that helps them comprehend the content of the video, improves the accessibility and makes the learning more engaging and more specific.

The basic workflow of the system is designed using three-tier structure of Spring MVC architecture. The system is set up using Spring Boot and implemented using Java and the database platform is PostgreSQL. Video playing and transcript support is carried out via the YouTube API. Only video metadata and the transcripts themselves will be recorded in the system's database. This means that the focus of our system performance lies in the querying of transcript data and not in the accessing or processing of videos. This is one major criteria that this study attempts to test. Hence, this paper proposes two alternative database architectures that are rigorously tested and compared against one another.

II. Related Works

A. Online Learning and Video Transcripts

A video system that stores and manages the video resources has been developed by [4] which is based on MVC framework. This system also supports authority management and security. The author proposes that MVC framework is the appropriate architectural pattern in developing a system that enables the users to upload, view and download video contents. Some of the popular Open Course Ware (OCW) resources such as [5] and [6] incorporated several innovative digital affordances to improve learning. Availability of video transcripts is a core feature in both of these platforms. A survey on these two OCW platforms [3] proved that the video transcripts provide integrated learning experience and helps to translate between languages.

In another study [7], the visual transcripts have been used to improve the quality of learning through video lectures. This system provides the visual transcript and text in a linear layout to enable students to browse or search through the videos. However, this work is quite limited to only blackboard style teaching. Zoom is another video-conferencing platform that provides automatic audio transcription for the cloud recordings of its meeting [8]. The transcript is also divided into sections with timestamps that lets the users to navigate to the searched keyword. However, this is application specific and available only to Zoom meetings.

B. Software Architecture Patterns

The major focus of this study is the performance of transcript searches. Hence, a short review of big data architectures has been conducted to choose appropriate architecture for the system that could handle speedy searches across a bulk of transcripts. [9] utilized Apache HBase as a distributed datastore for millions of clinical data records from IoT devices as shown in Figure 1. Hbase was chosen due to its ability to scale horizontally in a distributed manner.

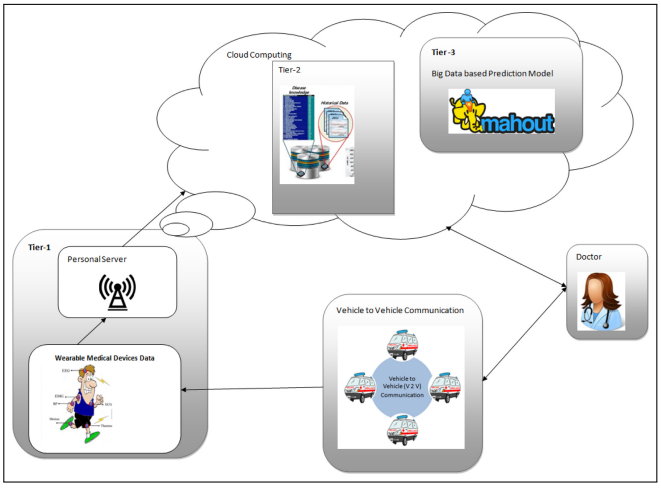


Fig. 1: Architecture as proposed by Kumar et. al. [9]. The relevant HBase component is in Tier 2.

Another study by [10] also used Apache Hbase as a large-scale datastore, in this case storing large volumes of video metadata. In addition to metadata, they also stored the actual raw video data on an HDFS Hadoop system. In contrast to this system, meta data of Control-Ef is stored in a relational PostgreSQL database and YouTube is used to store the videos.

In another study [11], the authors proposed a system once again for IoT data acquisition in which Apache Cassandra, a non-relational column datastore, was used to store large amounts of sensor data with a focus on high availability. A multi-node Cassandra cluster was setup for their study. Apache Cassandra was chosen because it offers high availability, scalability and fault tolerance. Their architecture is shown in Figure .

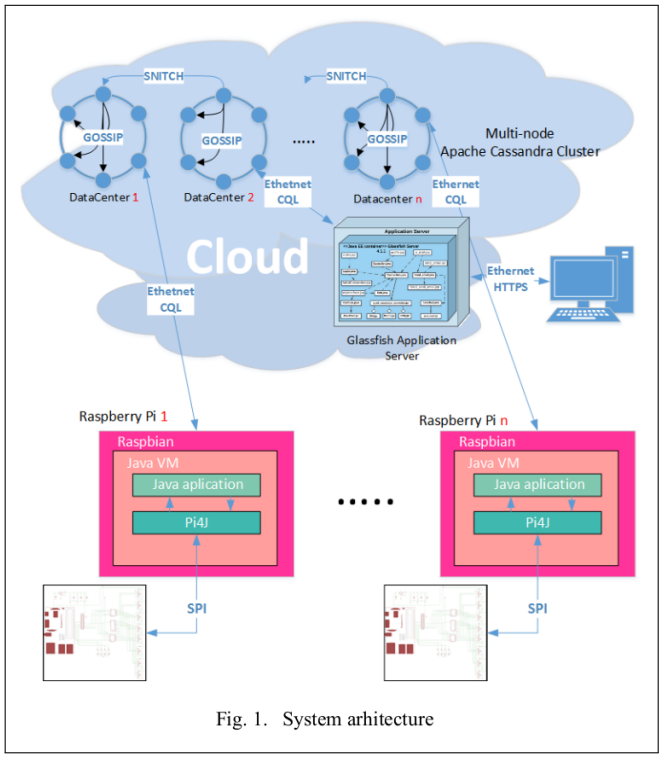


Fig. 1. System arhitecture

Fig. 2: System architecture of Ferencz et. al. [11].

III. Design and Implementation

The Control-Ef system has been implemented to overcome the challenges encountered in existing transcript search systems. This system is lightweight in nature and has improved performance over its predecessors. Since the YouTube API is used for both the video player and transcript generator, the system is faster as much of the processing has been outsourced to a separate service, i.e. YouTube in this case. As this system does not make an API call to a separate transcript generation, the communication and administrative overhead such as extra API calls, API key management, etc are reduced. However, one major disadvantage lies in the fact that this system is highly reliant on YouTube and suffers from the risk of Single Point of Failure.

A. Architecture

Since the transcript generated is of high volume and highly unstructured, one of the main goal of the system was to choose an architecture that can provide a speedy response to the query made by the user, in this context is the provision of exact timestamp of the keyword searched when multitude of users are logged in and many videos with similar keywords exist in the system. Thus, choosing a distributed, scalable management system to support big data generated was crucial. From extensive research on literature, multi-node Apache Cassandra cluster and the classical PostgreSQL were chosen as the data storage component to store the highly unstructured transcript data. The choice of Cassandra over other No-SQL database is that all the nodes in the cluster is able to perform

all read-write operations [12]. Thus, the potential architecture of Control-Ef is either woven around Apache Cassandra or PostgreSQL to provide improved performance and resiliency to the system as shown in Figure 3 or **??**.

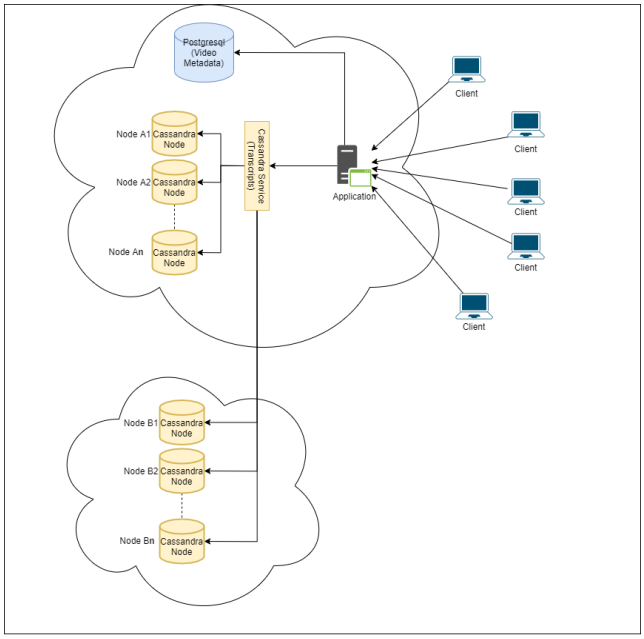


Fig. 3: Proposed System Architecture with Cassandra.

The video metadata and user data will still be stored in the PostgreSQL database. The two key challenges in implementing the system with Apache Cassandra are the possibility of automatic scaling across multiple nodes and the supportability of Cassandra to the structure of timestamp.

IV. Experimental Design

The goal of this system is to provide a fast search **response time** for students to study from recorded videos. **Response time** refers to the time taken for a user to receive a response from the system on the user interface. Therefore, performance testing, in specific, stress testing, has been designed to study on the response times of the system for both of the proposed architectures. The test plan is to simulate an increase in the number of transcript data entries in the Cassandra cluster and the PostgresQL Database until they reach 500, 000 entries. This can be accommodated with a far fewer number of videos since a video can have multiple entries for the timestamps. The rate of change in the response time of the system's transcript search can then be estimated as the number of entries as the system increases. The goal of the test is to compare the performance of both databases for retrieving the timestamp(s) of the specific keyword upon search.

A. Test Plan

The test plan is to simulate an increase in the number of transcript data entries starting from 2000 records in both databases until they reach 500,000 entries for a single user. The test is carried out at multiple iterations for a read query and the average latency is considered as the time taken for the keyword search across the database. Through this stress testing, the system would be able to choose an architecture that withstands for the increasing number of transcript entries.

B. Hypothesis and Testing Tools

The study hypothesize that Cassandra would outperform the PostgreSQL database because of its ability to scale up efficiently for increasing number of transcripts. On this ground, stress testing was carried out with two of the popular automated testing tools to validate our hypothesis. The testing for PostgreSQL database was done by Apache JMeter 5.4.1 tool. Since, Cassandra is not supported with native JMeter, Cassandra-Stress tool was used to test the performance of Cassandra cluster.

C. Test Environment and Procedure

Both tests run on Guppy Server available at CSIM of Asian Institute of Technology rather than local machines. This was done to standardize the testing environment for both instances. Both databases lie as docker containers on Guppy. PostgreSQL uses one container while Cassandra uses two containers as there are two nodes in the test Cassandra Cluster.The connection to containers were established through SSH tunnel.

The test environment was prepared accordingly and the stress testing was run for a range of transcripts namely **2,000; 5,000; 10,000; 25,000; 50,000; 100,000; 300,000 and 500,000** separately. The test on databases were conducted in 200 iterations separately and the mean latency was accounted for database performance for a specific number of records. The test architectures for both PostgreSQL and Apache Cassandra are shown in Figure 4b and Figure 4a respectively.

The test was performed with the objective to obtain the outcome as shown in Figure 5.

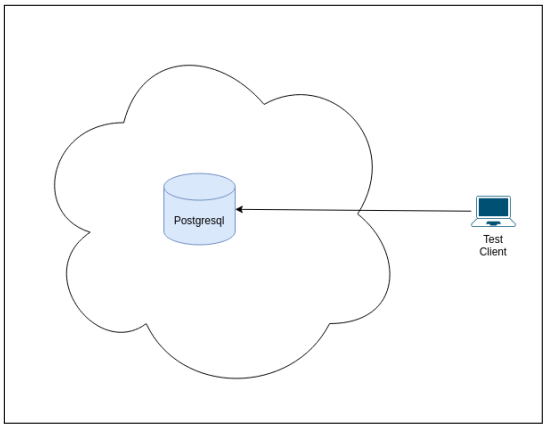
V. Results

The tests were conducted according to the specified test plan and the results as shown in Table I were obtained.

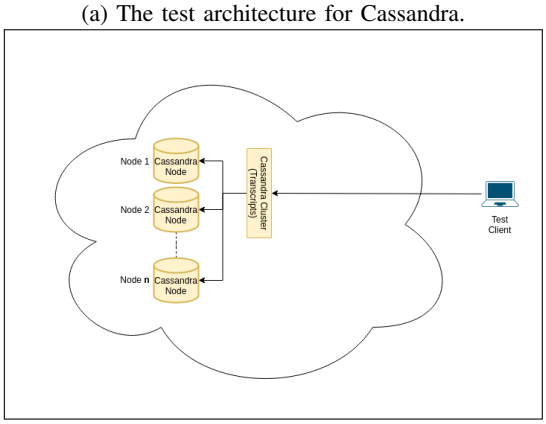
Data Records	Mean Latency (ms)	
	Cassandra	PostgresQL
2000	80.2	5
5000	75	9
10000	191	17
25000	595.3	77
50000	1210.5	238
100000	3114	362
300000	8677.6	1965
500000	19795.3	2871

TABLE I: Results of the stress test.

The graphical illustrations of the results are presented below for easy comparison. Figure 6 shows the comparison at discrete data intervals while Figure 7 shows the overall trend as the number of records increases. As can be seen in both graphs, the discrepancy between PostgreSQL and Cassandra response times grows as the data volume grows. In Figure 6



(a) The test architecture for Cassandra.



(b) The test architecture for PostgreSQL.

Fig. 4: The test architecture.

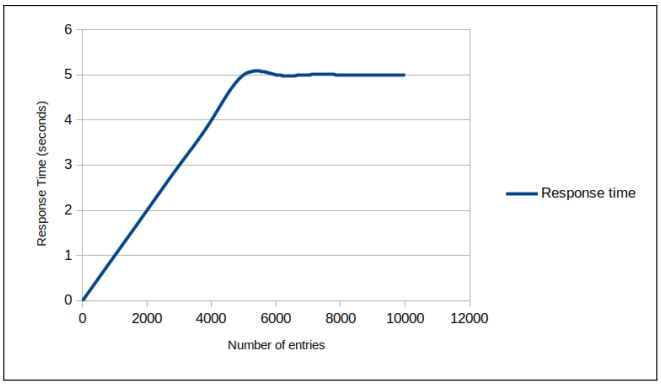


Fig. 5: Expected Outcome of the test.

at the 50,000 data mark, the response time of Cassandra is nearly six times than that of PostgreSQL.

VI. Discussion

The test results failed to prove the hypothesis since PostgreSQL outperformed the Cassandra for increasing data records. However, with the available resources, the test has been conducted for a maximum of 500,000 records whereas Cassandra has proven benefits in contexts where there are data of high dimension, heavy write workloads and complex real-

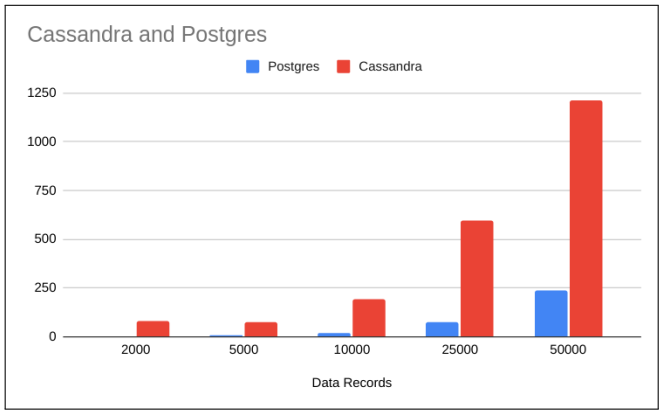


Fig. 6: Comparison of latency between Cassandra and Postgres.

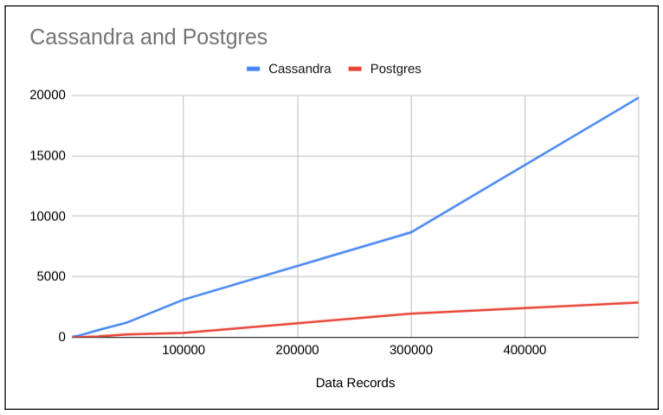


Fig. 7: Comparison of the latency trend between Cassandra and Postgres.

time data analytics. In the context of Control-Ef, the stress testing is performed for a read query since the main focus of the system is on transcript searches and the volume of data tested may be too little for Cassandra's big data benefits to be observed. Furthermore, the query is made only on a single table and the complexity in querying a join table is not done. These could be the potential reasons as why the performance of Cassandra is lower than that of PostgreSQL. However, further testing could detect bottleneck for relatively higher amount of data that could better identify a robust architecture that supports both scalability and performance.

VII. Conclusion and Future Works

In this study, an online transcript search system in the form of a web application is proposed based on the Spring MVC framework. Furthermore, two database architectures are proposed and compared with each other in terms of their read performance since the focus of the aforementioned app is on transcript searches. One architecture utilizes a "vanilla" relational database in the form of Postgres whereas the other uses Apache Cassandra, a columnar NoSQL database. A rigorous test experiment is conducted and the results show

that for the amount of data that we tested with, PostgreSQL proves to be the superior option in terms of read performance.

Future studies may perform bottleneck analysis on this test experiment to see if there are any factors (bottlenecks) contributing to the inferior performance of Cassandra. In addition, YouTube Data API would be integrated fully with the proposed Control-Ef system to further streamline the application as our future work.

ACKNOWLEDGEMENT

We would like to thank Dr. Chaklam Silpasuwanchai for his helpful advice and attention throughout the course of this study. Without his helpful input, we could not have succeeded.

REFERENCES

[1] K. Mukhtar, K. Javed, M. Arooj, and A. Sethi, "Advantages, limitations and recommendations for online learning during covid-19 pandemic era," *Pakistan journal of medical sciences*, vol. 36, no. COVID19-S4, p. S27, 2020.

[2] S. Kaup, R. Jain, S. Shivalli, S. Pandey, and S. Kaup, "Sustaining academics during covid-19 pandemic: the role of online teaching-learning," *Indian Journal of Ophthalmology*, vol. 68, no. 6, p. 1220, 2020.

[3] B. Crawford Camiciottoli, "The opencourseware lecture: A new twist on an old genre?" *Journal of English for Academic Purposes*, vol. 46, p. 100870, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1475158519306605

[4] F. Zhang, "Design and implementation of physical education video teaching system based on spring mvc architecture," in *Proceedings of the 2019 4th International Conference on Information and Education Innovations*, ser. ICIEI 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 116–119. [Online]. Available: https://doi.org/10.1145/3345094.3345113

[5] "Audio/video lectures." [Online]. Available: https://ocw.mit.edu/courses/audio-video-courses/

[6] "Open yale courses." [Online]. Available: https://oyc.yale.edu/

[7] H. V. Shin, F. Berthouzoz, W. Li, and F. Durand, "Visual transcripts: lecture notes from blackboard-style lecture videos," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1–10, 2015.

[8] "Using audio transcription for cloud recordings." [Online]. Available: https://support.zoom.us/hc/en-us/articles/115004794983-Using-audio-transcription-for-cloud-recordings-

[9] P. M. Kumar and U. D. Gandhi, "A novel three-tier internet of things architecture with machine learning algorithm for early detection of heart diseases," *Computers & Electrical Engineering*, vol. 65, pp. 222–235, 2018.

[10] M. N. Khan, A. Alam, and Y. Lee, "Falkon: Large-scale content-based video retrieval utilizing deep-features and distributed in-memory computing," in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2020, pp. 36–43.

[11] K. Ferencz and J. Domokos, "Iot sensor data acquisition and storage system using raspberry pi and apache cassandra," in *2018 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*. IEEE, 2018, pp. 000 143–000 146.

[12] A. Vaclavova and M. Kebisek, "Comparison of various nosql databases for unstructured industrial data," in *Proceedings of the Computational Methods in Systems and Software*. Springer, 2020, pp. 921–930.