CONTROL-EF

Htoo Lwin (120832)
Fathima Shafana (121985)

Background

- The recent COVID-19 pandemic has forced many educational institutions at all levels to conduct lectures online in lieu of physical classes.
- Most such online lectures are recorded by the institution for student review or administrative purposes and are hence are usually opened to student access.
- Students can use these videos to either review course material or learn about new concepts from courses not directly related to their curriculum.

Problem Statement

- Navigating through the lecture content can prove to be time consuming and difficult without any prior processing of the videos.
- The main difficulties are **twofold**:
  1. Students can take some time manually going through video lists to find the lectures of topics they would like to learn (if the videos are publicly accessible in the first place).
  2. Students can also have a hard time searching through a particular video for a time when the lecturer is talking about the topic they want to learn about, especially if that student is reviewing the lecture.

Proposed Solution

- An application that helps users to get the exact timestamp of the keyword searched for.
- An architecture that is easily scalable and has **good read performance.**

Aims & Objectives

1. To provide a platform for students to easily access video lectures in an educational institution.
2. To provide students with the ability to quickly search for videos or timestamps using different facets such as tags, keywords.

Quality Attribute Analysis

Availability	**H**	The platform's main purpose is to provide aid in the learning process, so it should be available as much as possible. It should not fail the student when he needs it.
Performance	**H**	When a student is learning, it is important not to lose focus. So, the platform's search function should work as quickly as possible.
Portability	**L**	As the majority of quality studying is done in front of a desk, the software architecture should be designed for desktop computers (Windows PC, Mac OS).
Security	**M**	The main function - searching - only retrieves data, which is not dangerous. But uploading and transcript editing should be done only by authorized personnel. The software should provide that.
Scalability	**H**	As this platform is basically a database, any kind of expansion should be seamless and unnoticed by the user..
Testability	**M**	The functionality is quite simple and input methods are limited. The upload formats (video and audio) are also limited. The only functionality that should be tested thoroughly is the transcript generation.

Architecture Design and Implementation

Architectural Challenges and Other Discussion

- Database Performance
    - **Transcript search performance should be consistent regardless of data volume**.
        - Has to search every row of transcript table
    - Our servers won’t handle video/transcript processing so lightweight! Only handles **Transcript search**.

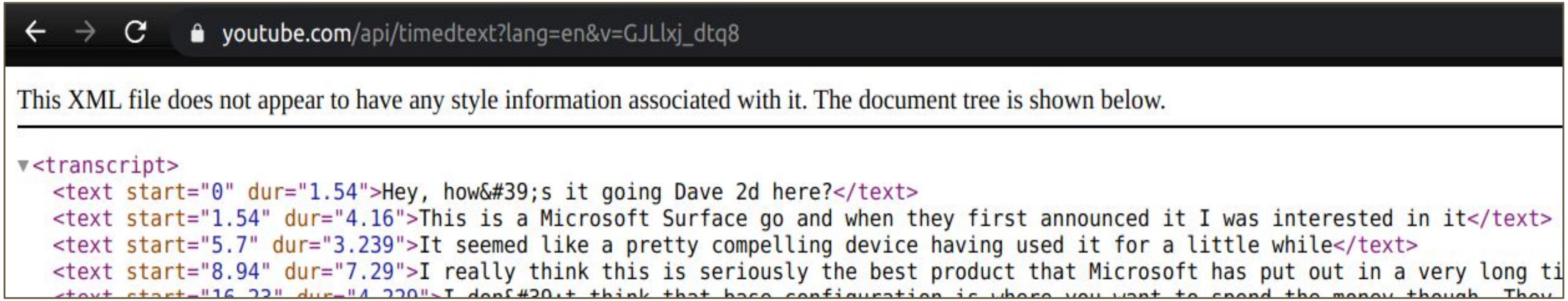
Architectural Challenges and Other Discussion

- Transcript-Timestamp Mapping in Database
    - Inspired by YouTube’s caption format
- Integration
    - YouTube API authentication and integration
        - API keys/OAuth clients (register our app as client?)
    - Upload and get transcripts

Integration Challenges

- Transcript retrieval
  - No clear documentation on public APIs
    - Different public APIs and different parameters
    - Unreliable
  - Secure APIs need OAuth 2.0 clients for authentication
  - Official documentation only supports secure APIs.
- Upload Video
  - Same issue with authentication

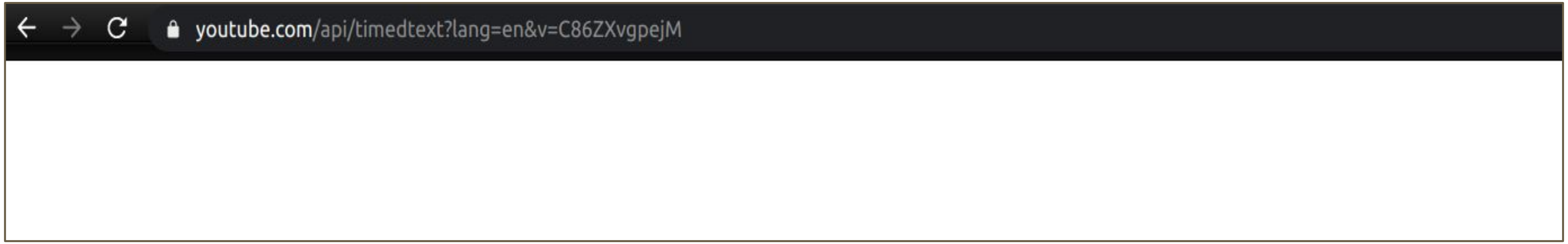
Integration Challenges - Public API troubles



youtube.com/api/timedtext?lang=en&v=GJLlxj_dtq8

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<transcript>
   <text start="0" dur="1.54">Hey, how&#39;s it going Dave 2d here?</text>
   <text start="1.54" dur="4.16">This is a Microsoft Surface go and when they first announced it I was interested in it</text>
   <text start="5.7" dur="3.239">It seemed like a pretty compelling device having used it for a little while</text>
   <text start="8.94" dur="7.29">I really think this is seriously the best product that Microsoft has put out in a very long ti
   <text start="16.23" dur="4.229">I don&#39;t think that base configuration is where you want to spend the money though. They
```

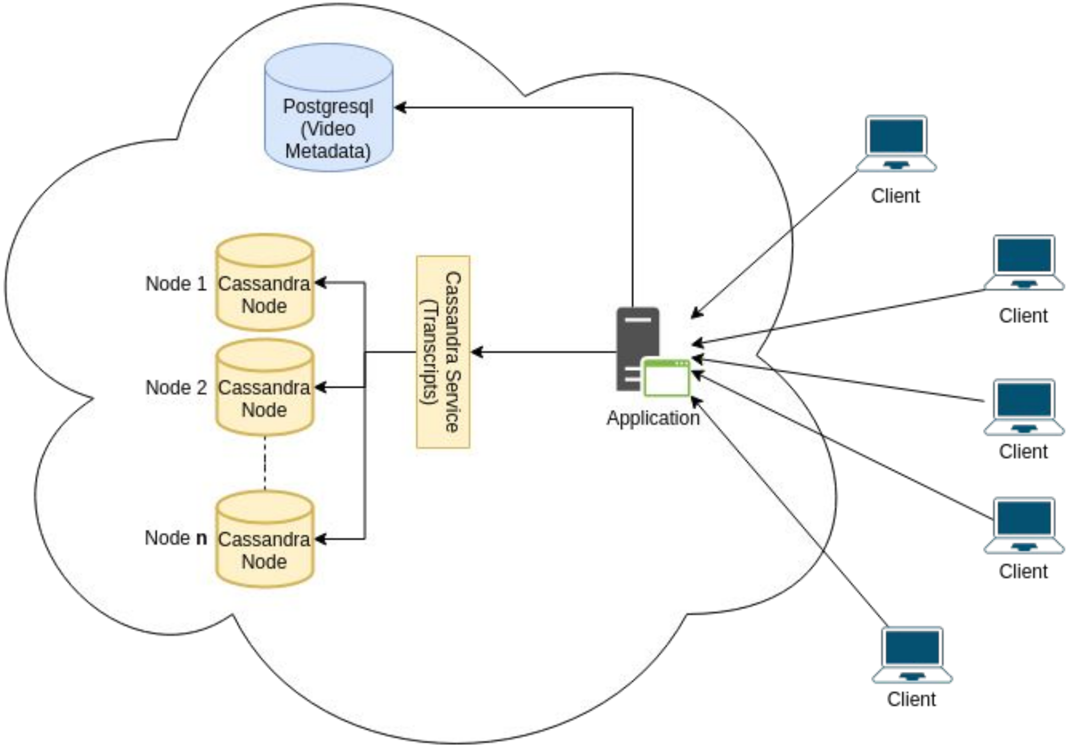


youtube.com/api/timedtext?lang=en&v=C86ZXvgpejM

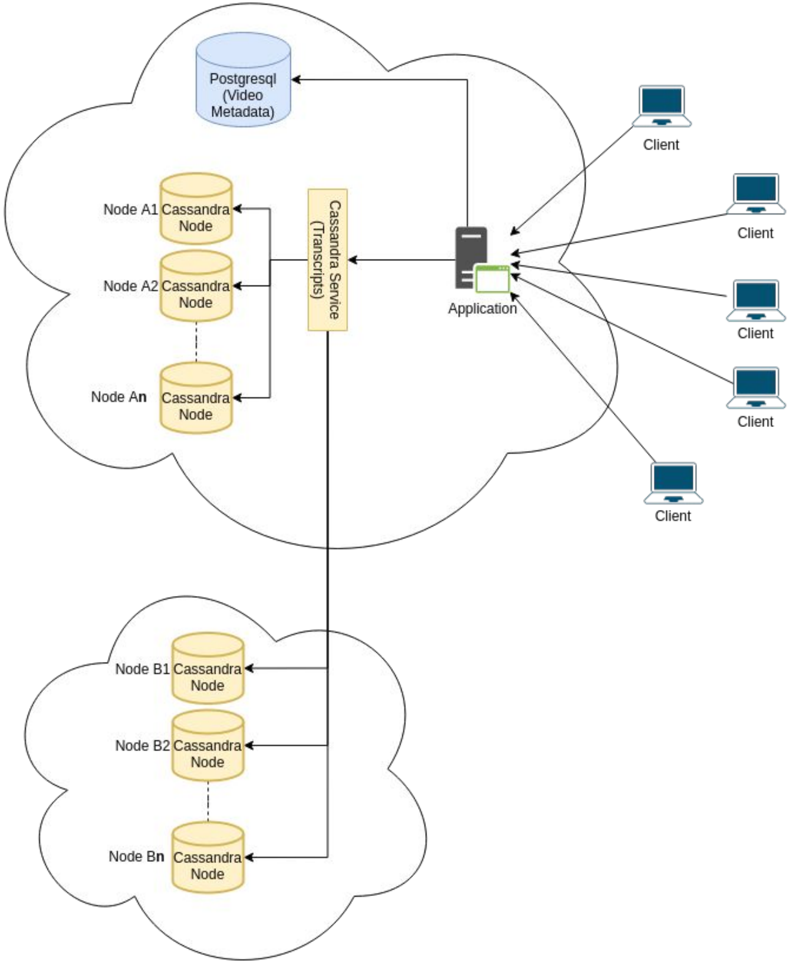
Architectural Challenges and Other Discussion

- Scalability (to support more users)
    - Our architecture should be able to easily scale and store more transcripts and videos
    - Our servers won’t handle video/transcript processing so lightweight! Only handles Transcript search.

Proposed Architectures - Single Server



Proposed Architectures - Multi Server

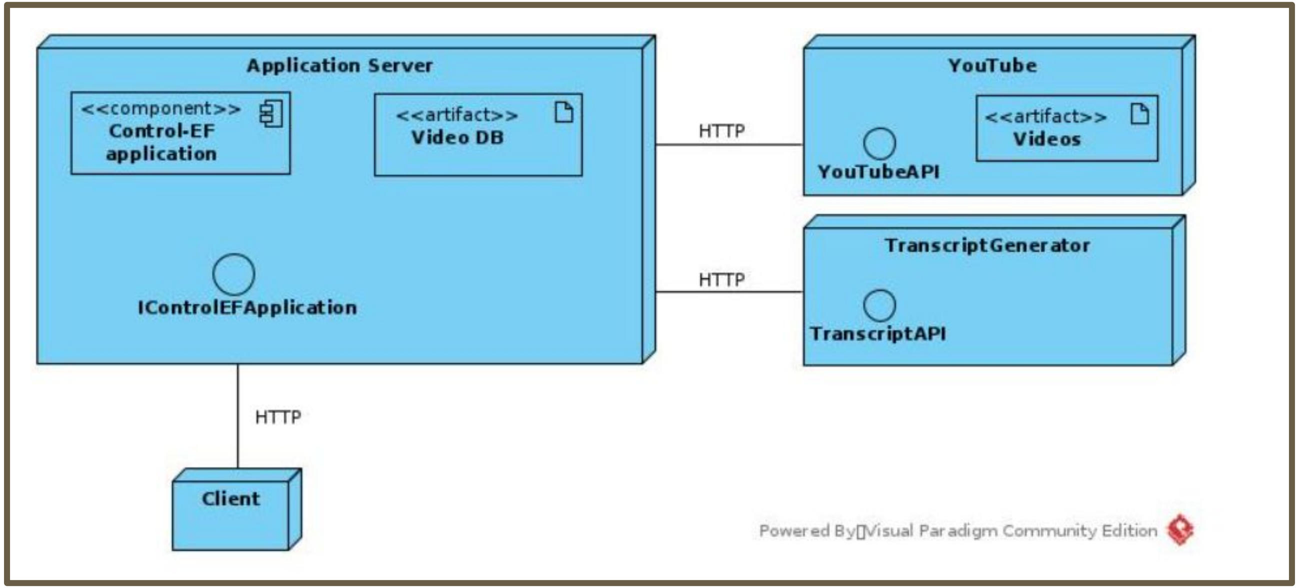


Architecture Patterns

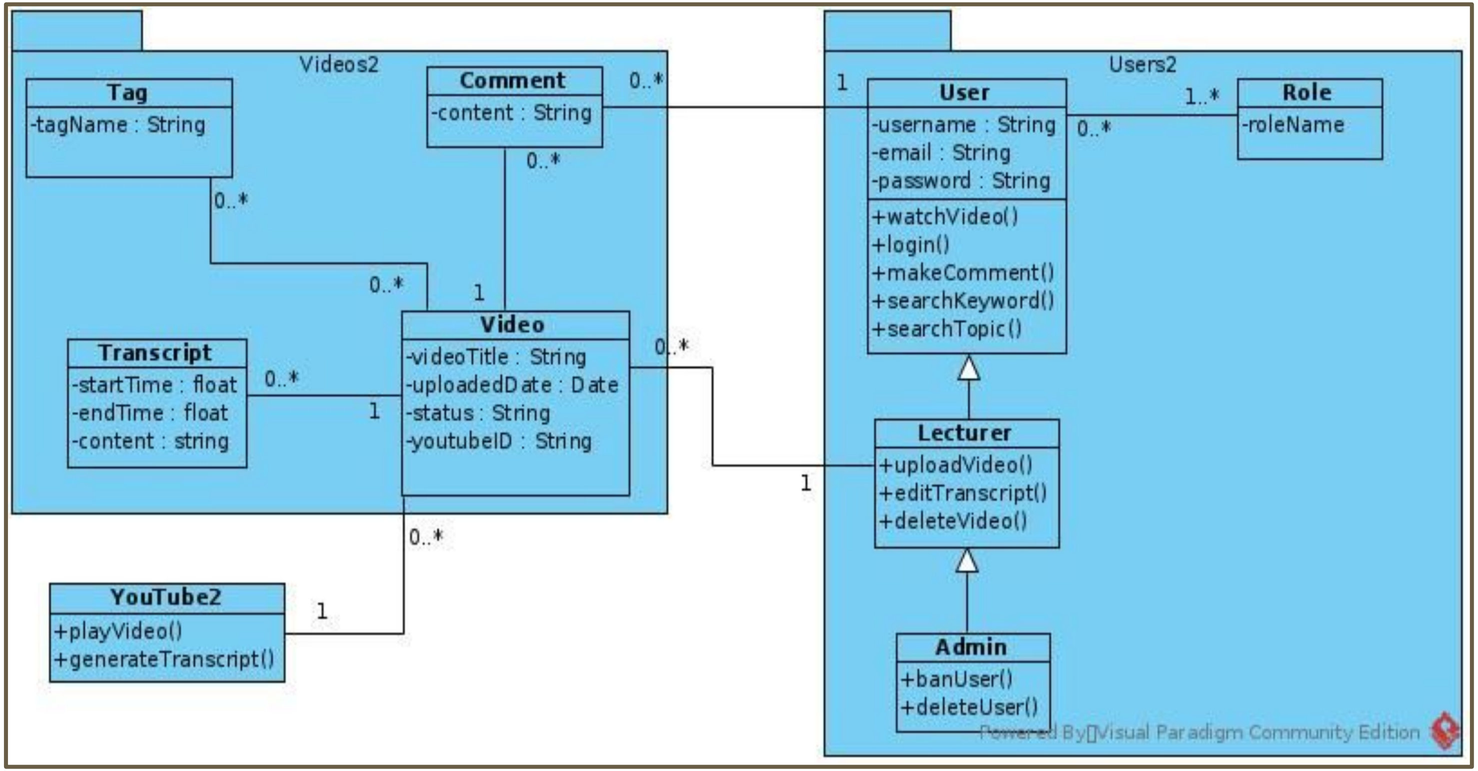
1. Model-view-controller (MVC)
2. Service-oriented Architecture
3. Peer-to-Peer(?)

- The Control-EF application will utilize the **model-view-controller** (MVC) architecture pattern in main using the MVC framework Spring Boot.
- It will also incorporate some best practices from **service-oriented architecture** (SOA). Students will consume the videos from YouTube and transcripts from our database.

Architectural Design - Deployment Diagram



Class Diagram



Tools and Technologies

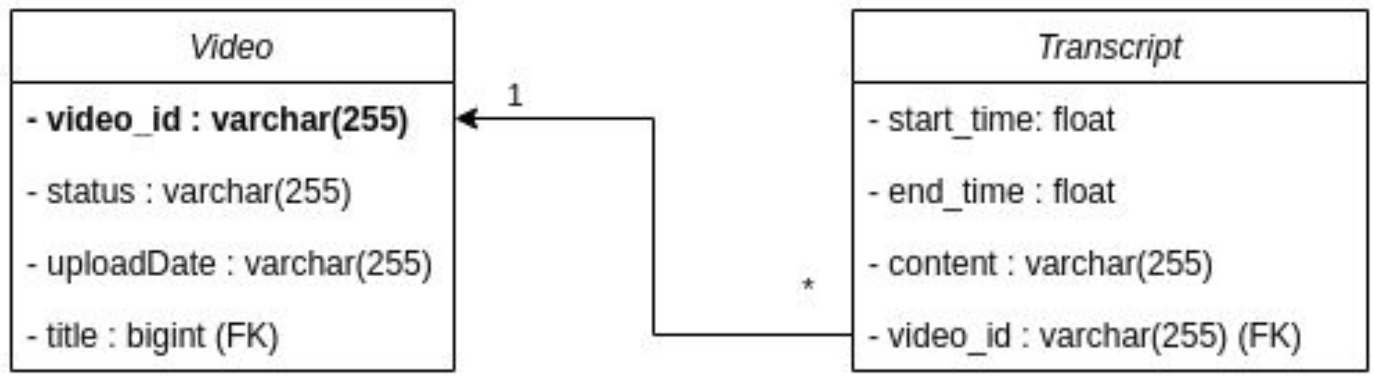
Database: Postgres (use JDBC Postgres Driver)

Language: Java 8

Application Framework: Spring Boot

Testing Tools: Apache JMeter & Cassandra-Stress

Video-Transcript Mapping



Video

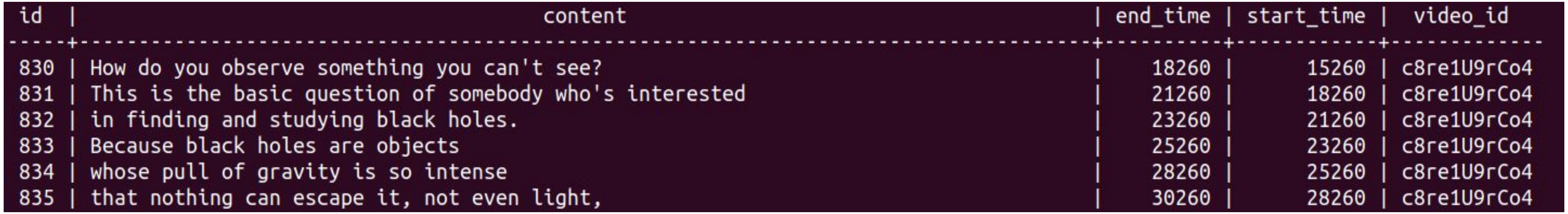
- video_id : varchar(255)

- status : varchar(255)
- uploadDate : varchar(255)
- title : bigint (FK)

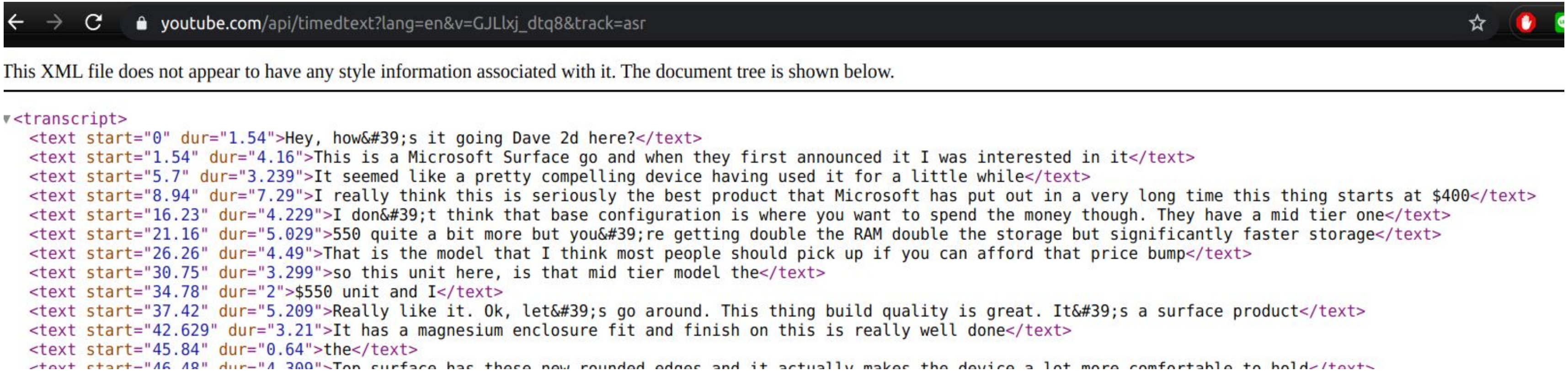
Transcript

- start_time: float
- end_time : float
- content : varchar(255)
- video_id : varchar(255) (FK)

Transcripts (Ours vs. Youtube)



```
 id  |                          content                          | end_time | start_time |  video_id
-----+-----------------------------------------------------------+----------+------------+------------
 830 | How do you observe something you can't see?               |    18260 |      15260 | c8re1U9rCo4
 831 | This is the basic question of somebody who's interested   |    21260 |      18260 | c8re1U9rCo4
 832 | in finding and studying black holes.                      |    23260 |      21260 | c8re1U9rCo4
 833 | Because black holes are objects                           |    25260 |      23260 | c8re1U9rCo4
 834 | whose pull of gravity is so intense                       |    28260 |      25260 | c8re1U9rCo4
 835 | that nothing can escape it, not even light,               |    30260 |      28260 | c8re1U9rCo4
```



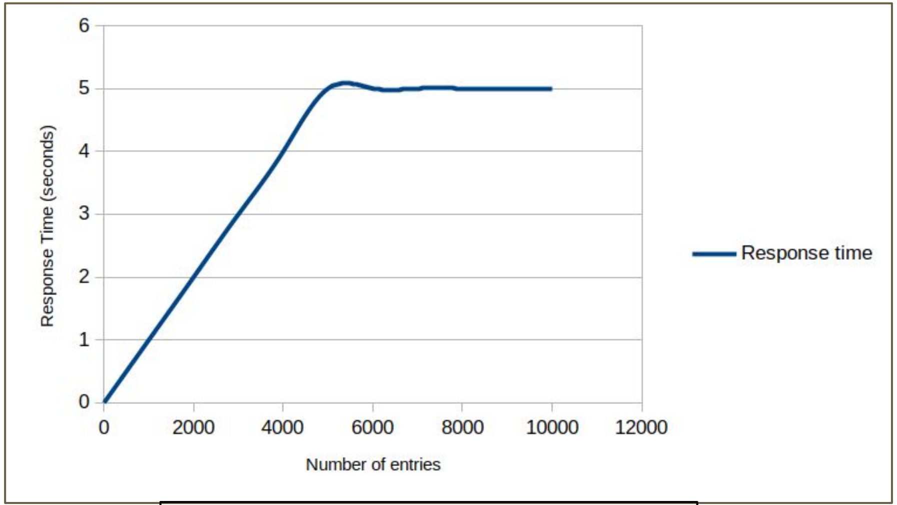
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▾<transcript>
   <text start="0" dur="1.54">Hey, how&#39;s it going Dave 2d here?</text>
   <text start="1.54" dur="4.16">This is a Microsoft Surface go and when they first announced it I was interested in it</text>
   <text start="5.7" dur="3.239">It seemed like a pretty compelling device having used it for a little while</text>
   <text start="8.94" dur="7.29">I really think this is seriously the best product that Microsoft has put out in a very long time this thing starts at $400</text>
   <text start="16.23" dur="4.229">I don&#39;t think that base configuration is where you want to spend the money though. They have a mid tier one</text>
   <text start="21.16" dur="5.029">550 quite a bit more but you&#39;re getting double the RAM double the storage but significantly faster storage</text>
   <text start="26.26" dur="4.49">That is the model that I think most people should pick up if you can afford that price bump</text>
   <text start="30.75" dur="3.299">so this unit here, is that mid tier model the</text>
   <text start="34.78" dur="2">$550 unit and I</text>
   <text start="37.42" dur="5.209">Really like it. Ok, let&#39;s go around. This thing build quality is great. It&#39;s a surface product</text>
   <text start="42.629" dur="3.21">It has a magnesium enclosure fit and finish on this is really well done</text>
   <text start="45.84" dur="0.64">the</text>
   <text start="46.48" dur="4.309">Top surface has these new rounded edges and it actually makes the device a lot more comfortable to hold</text>
```

Database Stress Testing

Testing Objectives

1. To test our proposed database architecture for consistent database performance for searching transcripts.



Our expected (and desired) result.

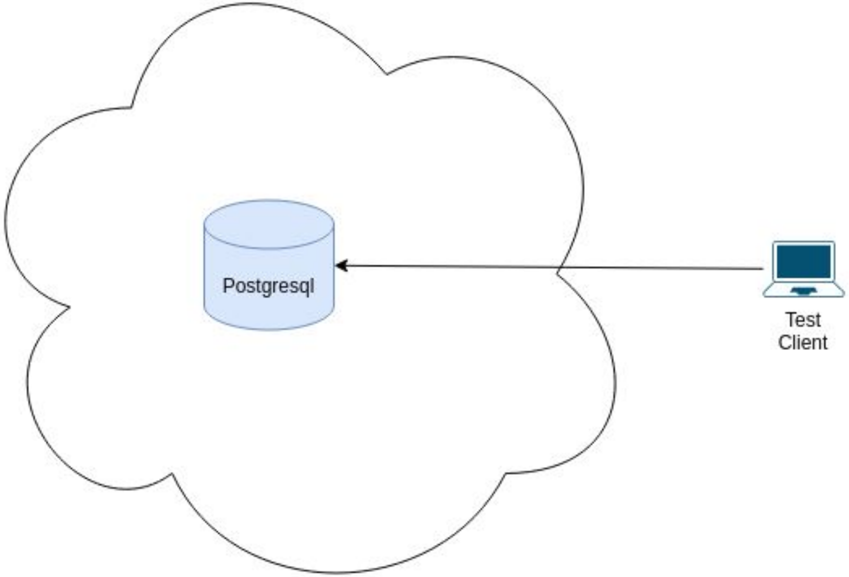
Test Methodology

- Compare Postgres search performance vs. Cassandra search performance
- Simulate keyword searching in real scenario with this query:
  - `select * from transcripts where content like '%a%'`
- Test at different amount of data records
  - 2k, 5k, 10k, 25k, 50k, 100k, 300k and 500k records
- **Find the mean latency in the response time of 200 read queries in milliseconds and compare**
  - **Evaluation metric - mean response time**
- Run tests on both Postgres and Cassandra

Test Environment

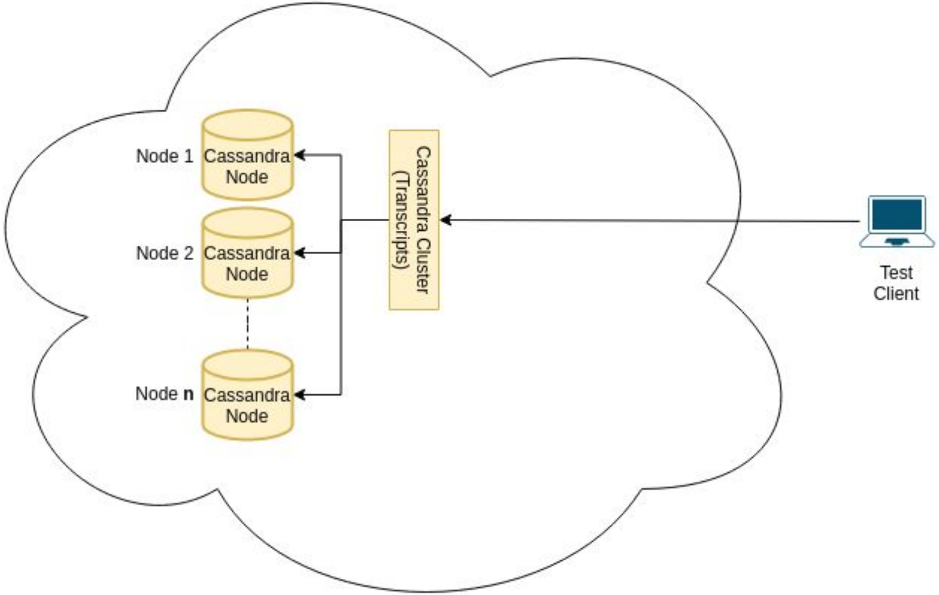
- The tests are run on the Guppy server available at CSIM here at AIT.
- Why Guppy?
  - To maintain the same and standard test environment.
  - Local Machines are of different specifications.
  - Control other variable except the Database
  - Databases themselves are located in Docker containers on Guppy
- For cassandra, two nodes were used = two containers

Test Architecture - Postgres

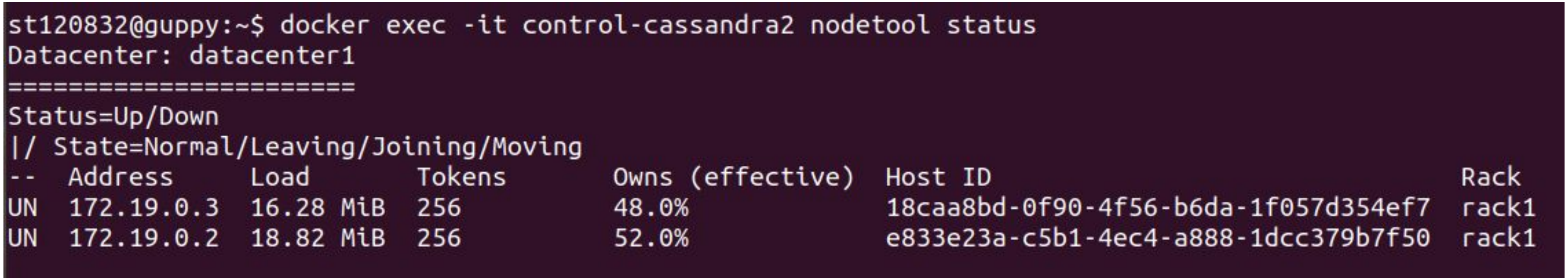


Test Architecture - Cassandra

Two nodes



Test Architecture - Cassandra contd.



```
st120832@guppy:~$ docker exec -it control-cassandra2 nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load       Tokens       Owns (effective)  Host ID                               Rack
UN  172.19.0.3  16.28 MiB  256          48.0%             18caa8bd-0f90-4f56-b6da-1f057d354ef7  rack1
UN  172.19.0.2  18.82 MiB  256          52.0%             e833e23a-c5b1-4ec4-a888-1dcc379b7f50  rack1
```

Two nodes with 500k records in
the same datacenter/cluster

Test Architecture - Cassandra contd.

See this link in the Github Repository for instructions on how to set up this cluster

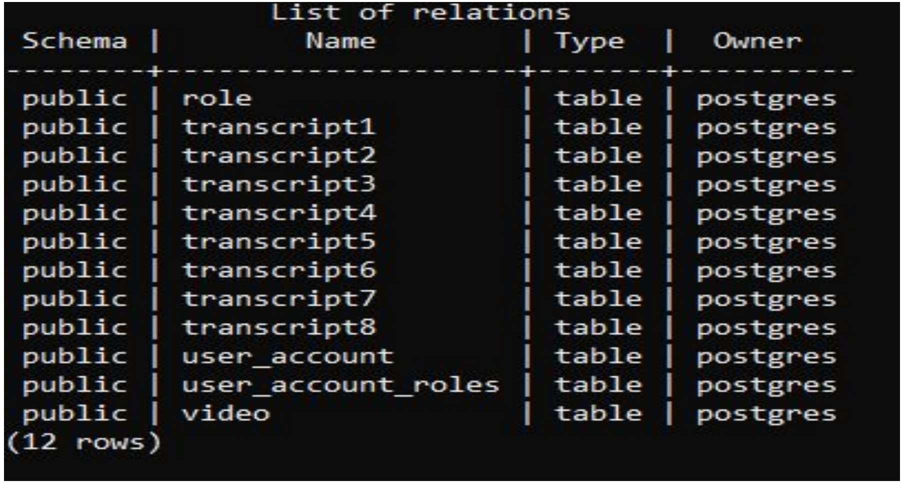
Test Tools

1. **Apache JMeter 5.4.1**
   - JMeter is a widely used tool utilized for analyzing and measuring the performance of a variety of services.
   - In the case of this study, it is used for the particular case of database stress testing.
   - JMeter natively supports most relational databases through JDBC driver plugins and this is the case for PostgreSQL as well.
2. **Cassandra-Stress Tool**
   - Cassandra is not supported natively by Apache JMeter
   - Plugins such as this are sorely out of date
   - Cassandra Stress tool was found to be more suitable for our testing purposes and is officially supported by the Apache foundation.

Test Scenario 1: PostgreSQL

Phase 1: Prepare test data sets for Stress Testing

- According to the Test Plan, a series of tables with records ranging from 2000, 5000, 10000, 25000, 50000, 100000, 300000 and 500000 were built in PostreSQL database as transcript1, transcript2, transcript3, transcript4, transcript5, transcript6, transcript7 and transcript8 respectively.

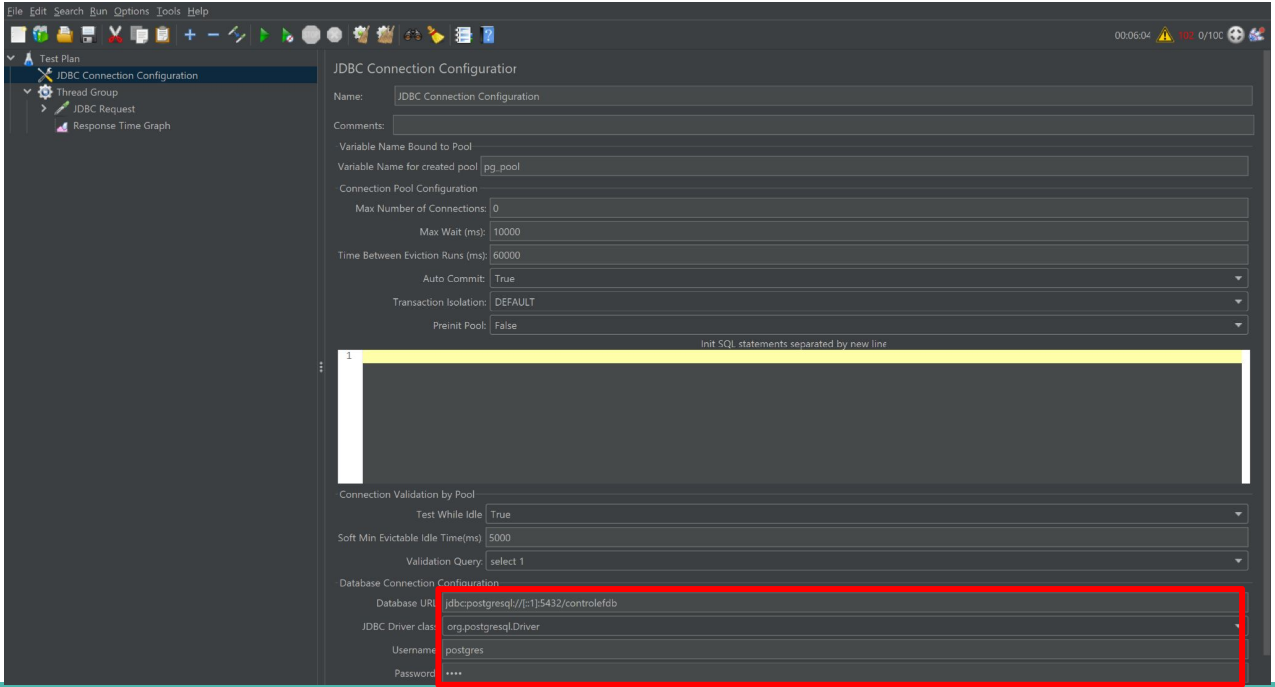


```
                  List of relations
 Schema |          Name          | Type  |  Owner
--------+------------------------+-------+----------
 public | role                   | table | postgres
 public | transcript1            | table | postgres
 public | transcript2            | table | postgres
 public | transcript3            | table | postgres
 public | transcript4            | table | postgres
 public | transcript5            | table | postgres
 public | transcript6            | table | postgres
 public | transcript7            | table | postgres
 public | transcript8            | table | postgres
 public | user_account           | table | postgres
 public | user_account_roles     | table | postgres
 public | video                  | table | postgres
(12 rows)
```

Test Scenario 1: PostgreSQL contd.

Phase 2: Stress Testing with Apache JMeter 5.4.1

- Configure JDBC connection for controlefdb


File Edit Search Run Options Tools Help
00:06:04
102
0/100
Test Plan
JDBC Connection Configuration
Thread Group
JDBC Request
Response Time Graph

JDBC Connection Configuratior

Name: JDBC Connection Configuration

Comments:

Variable Name Bound to Pool

Variable Name for created pool pg_pool

Connection Pool Configuration

Max Number of Connections: 0

Max Wait (ms): 10000

Time Between Eviction Runs (ms): 60000

Auto Commit: True

Transaction Isolation: DEFAULT

Preinit Pool: False

Init SQL statements separated by new line

Connection Validation by Pool

Test While Idle True

Soft Min Evictable Idle Time(ms): 5000

Validation Query: select 1

Database Connection Configuration

Database URL jdbc:postgresql://[::1]:5432/controlefdb

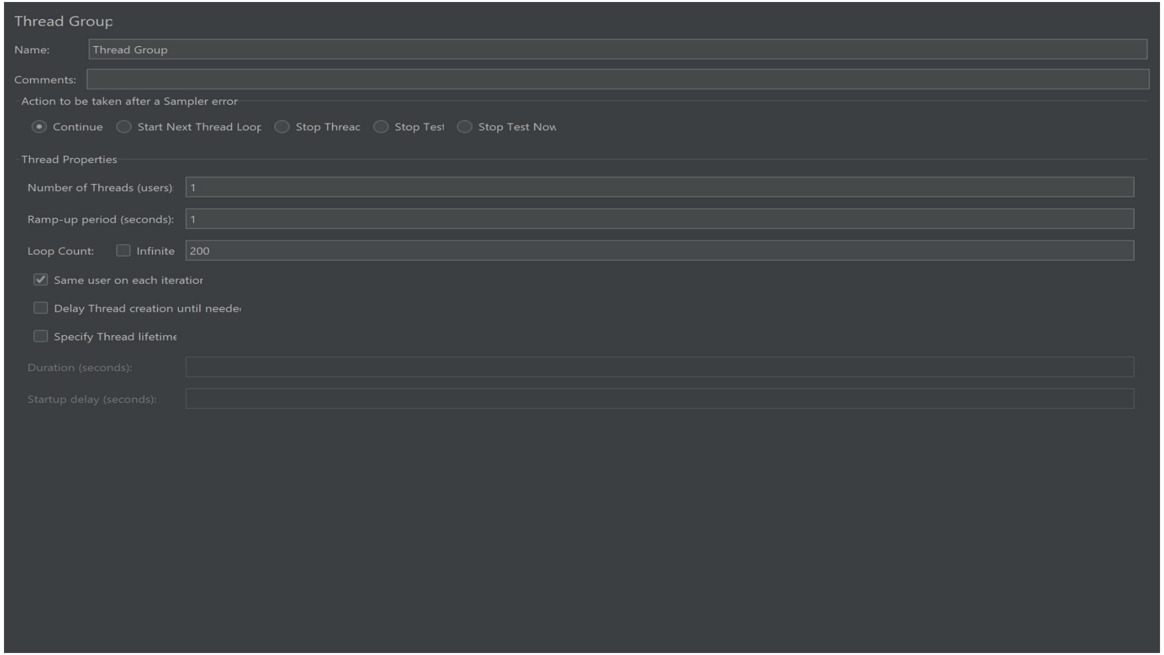
JDBC Driver class org.postgresql.Driver

Username postgres

Password ••••

Test Scenario 1: PostgreSQL contd.

- Create Thread Group



Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

Number of Threads (users): 1

Ramp-up period (seconds): 1

Loop Count: □ Infinite 200

☑ Same user on each iteration

□ Delay Thread creation until needed

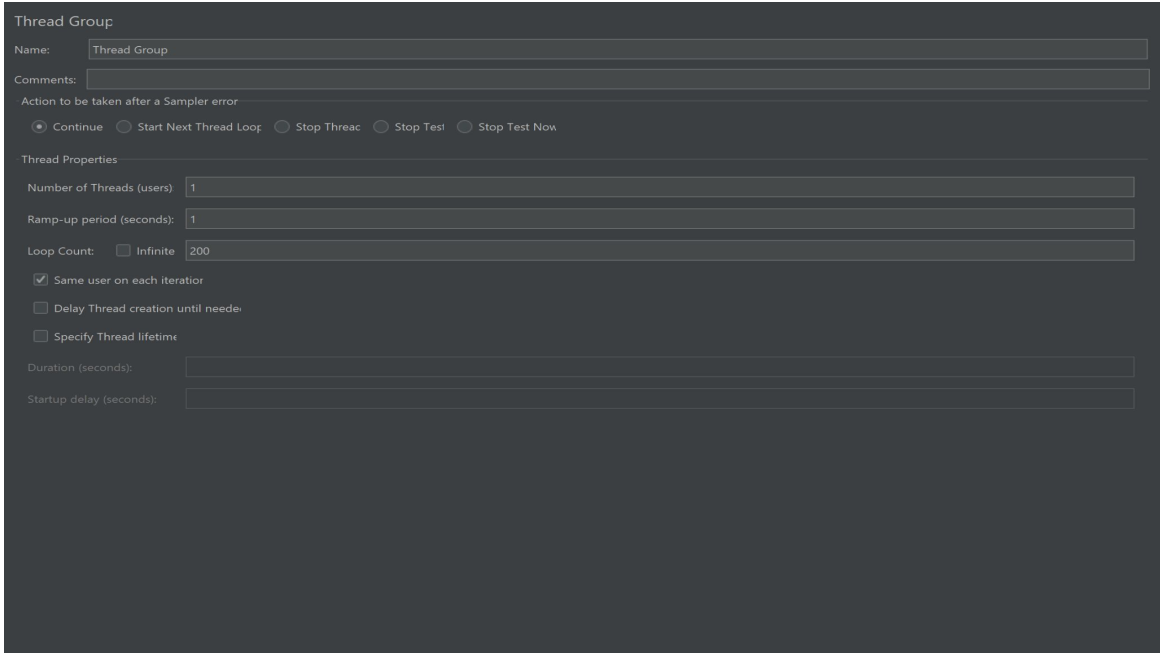
□ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Test Scenario 1: PostgreSQL contd.

- Create Thread Group



Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

Number of Threads (users): 1

Ramp-up period (seconds): 1

Loop Count: □ Infinite 200

☑ Same user on each iteration

□ Delay Thread creation until needed

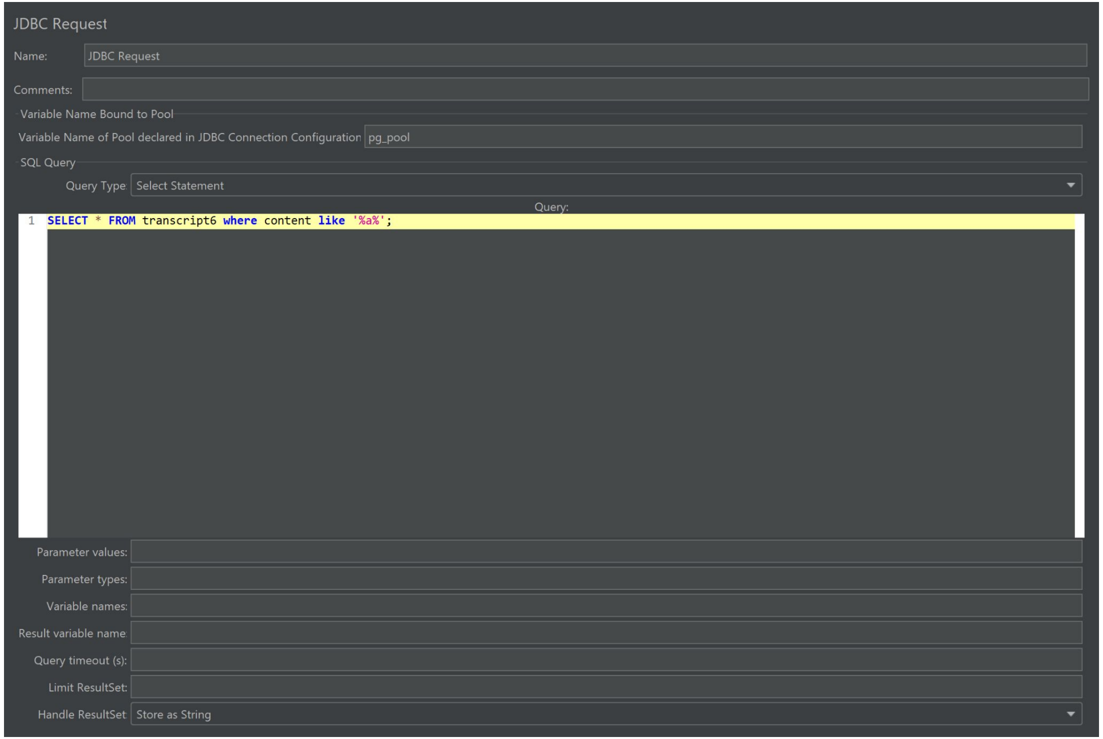
□ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Test Scenario 1: PostgreSQL contd.

- Add JDBC Request as Sampler



JDBC Request

Name: JDBC Request

Comments:

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: pg_pool

SQL Query

Query Type: Select Statement

Query:

```
SELECT * FROM transcript6 where content like '%a%';
```

Parameter values:

Parameter types:

Variable names:

Result variable name:

Query timeout (s):

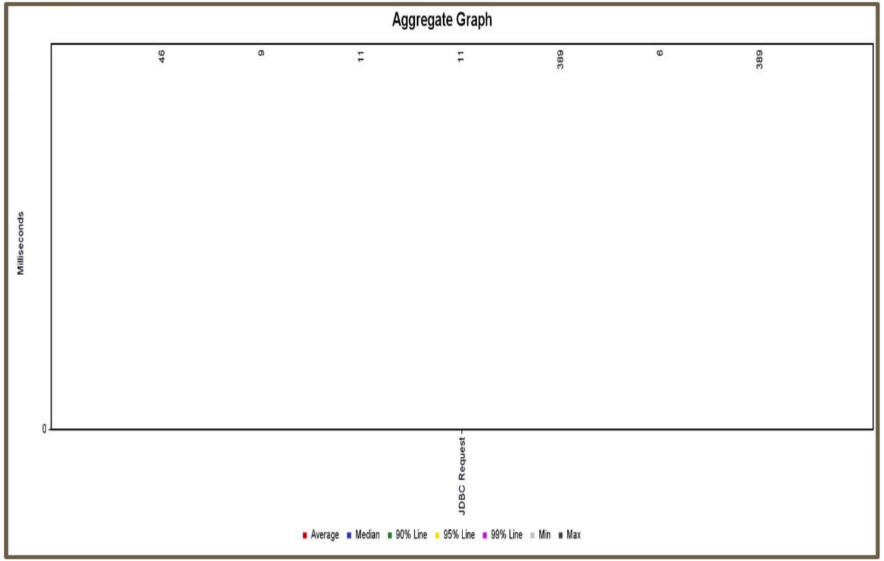
Limit ResultSet:

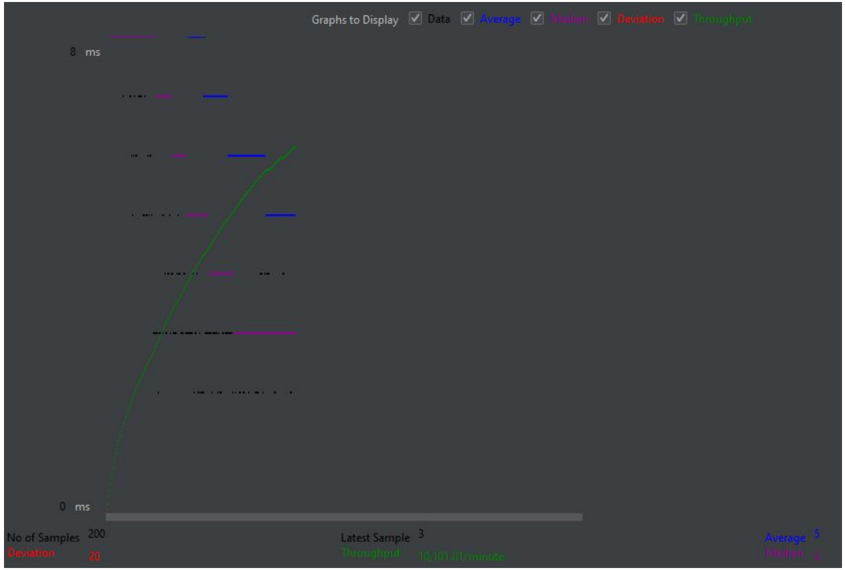
Handle ResultSet: Store as String

Test Scenario 1: PostgreSQL contd.

- Add listeners to view your results
  - View Results Tree
  - Graph Results
  - Aggregate Graph

Result: No. of Transcripts - 2 000





Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

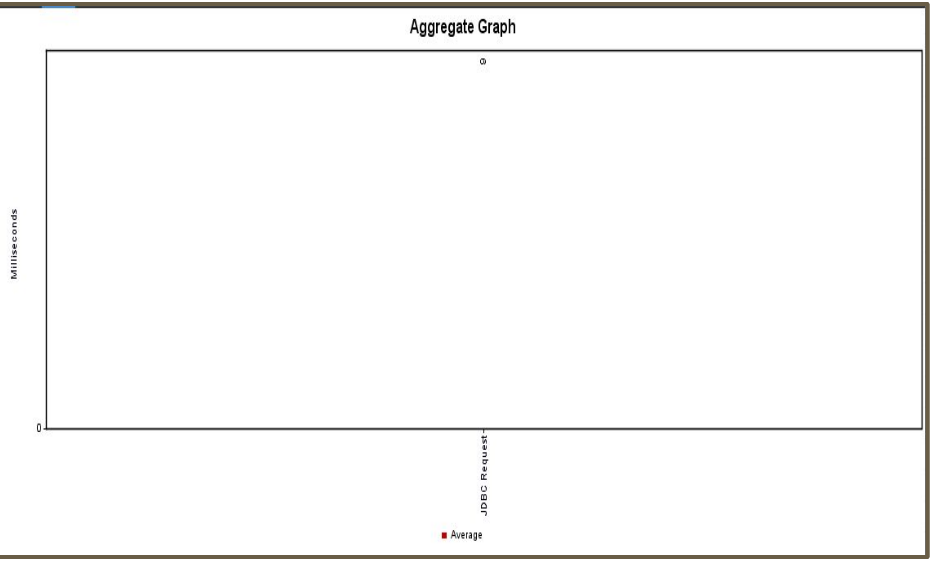
No of Samples 200
Deviation 20

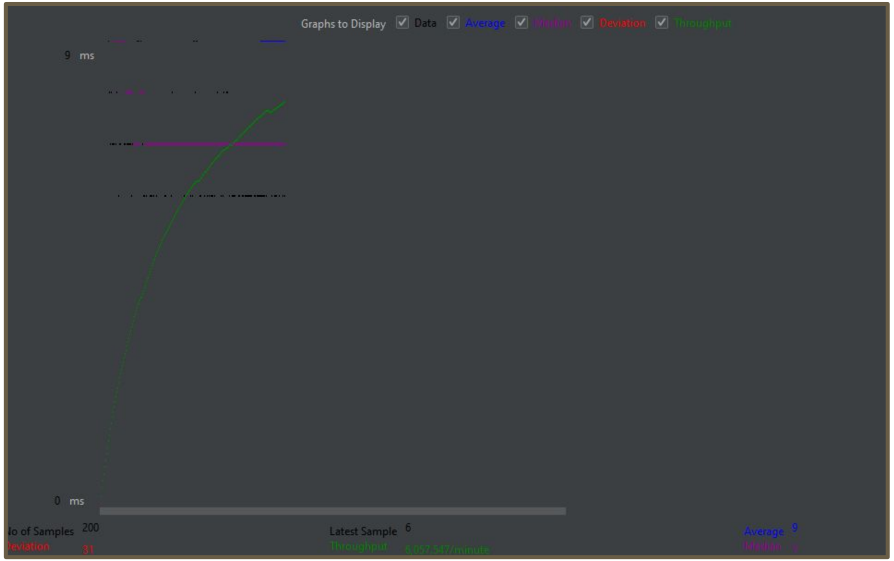
Latest Sample 3
Throughput 10,101.01/minute

Average 5
Median

Average Latency = 5 ms

Result: No. of Transcripts - 5 000

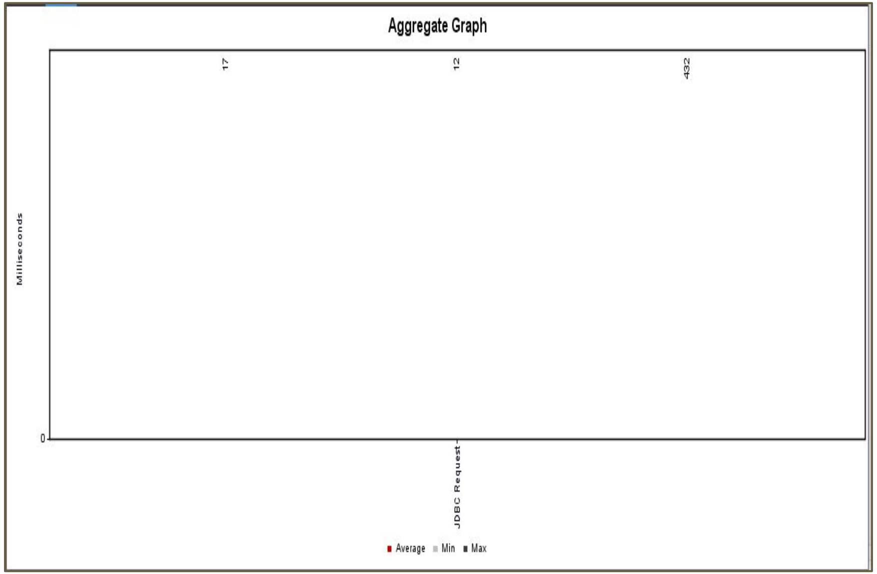


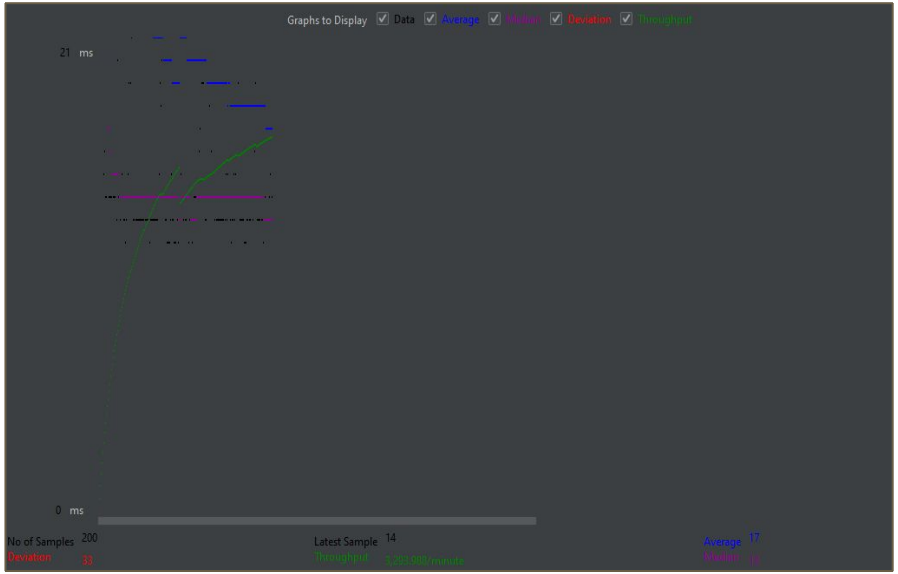

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

lo of Samples 200
Deviation 31
Latest Sample 6
Throughput 6,057.547/minute
Average 9
Median

Average Latency = 9 ms

Result: No. of Transcripts - 10 000




Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

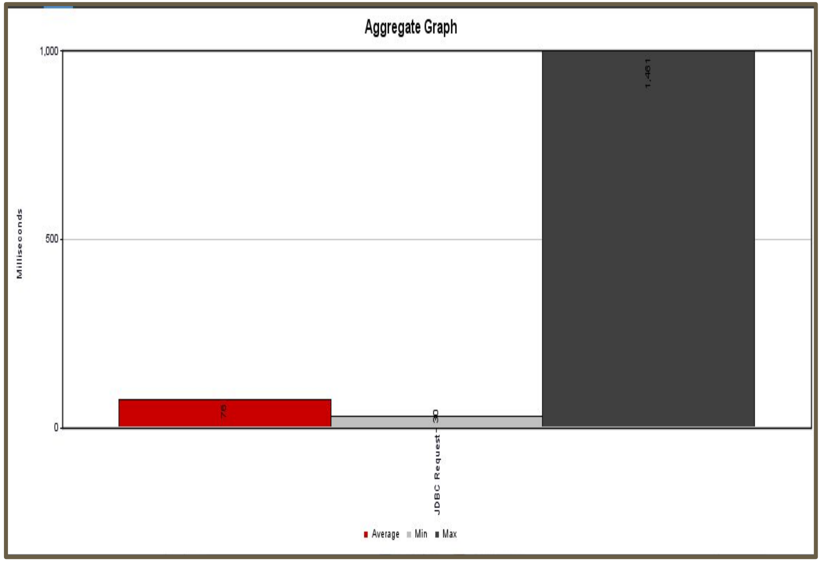
No of Samples 200
Deviation 33

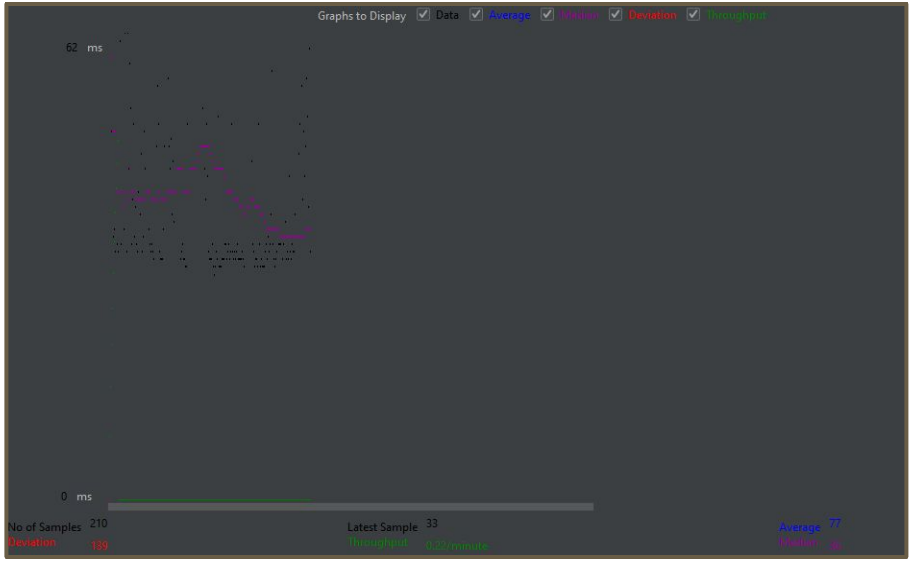
Latest Sample 14
Throughput 3,293,988/minute

Average 17
Median 12

Average Latency = 17 ms

Result: No. of Transcripts - 25 000




Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

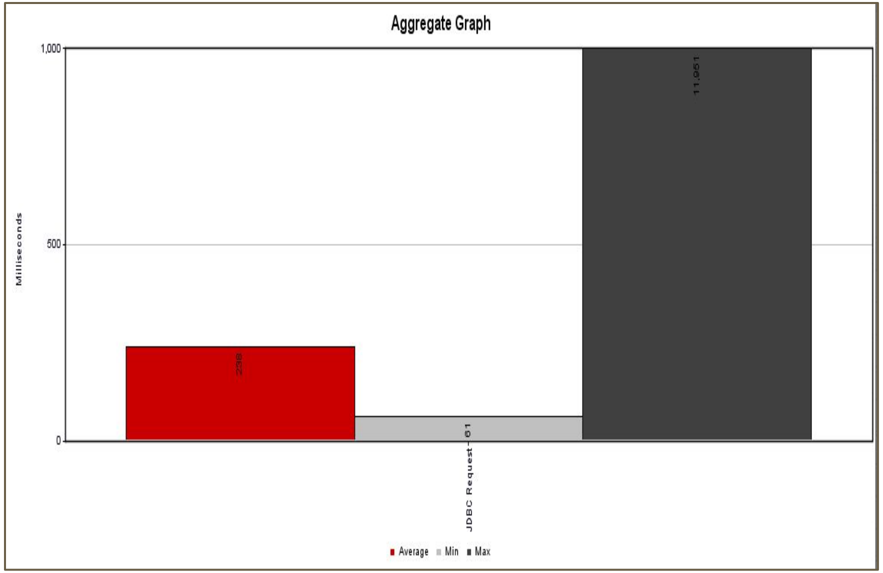
No of Samples 210
Deviation 139

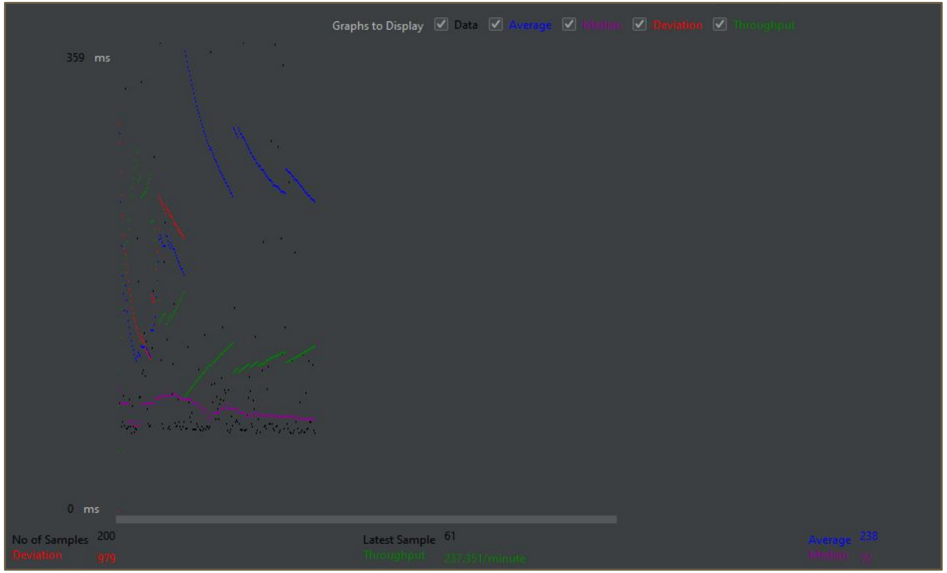
Latest Sample 33
Throughput 0.22/minute

Average 77
Median 36

Average Latency = 77 ms

Result: No. of Transcripts - 50 000




Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

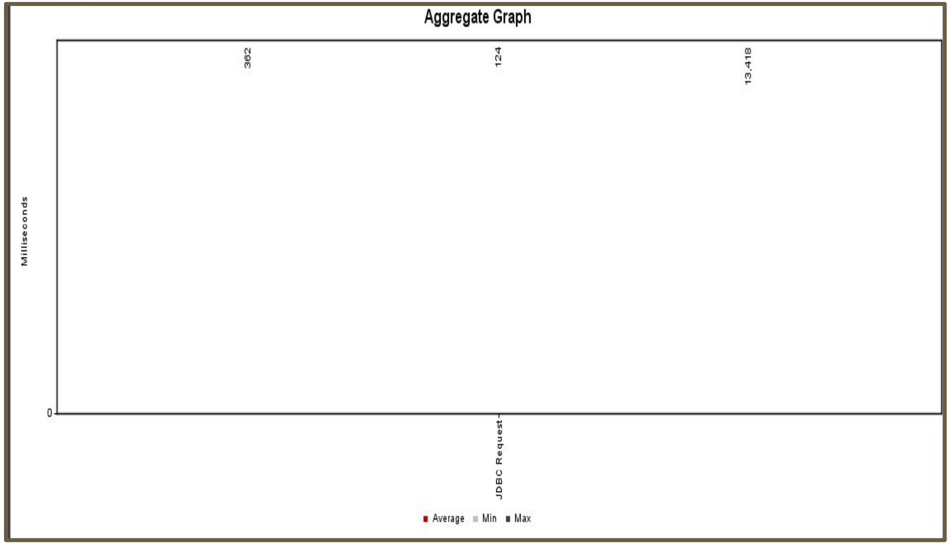
No of Samples 200
Deviation 979

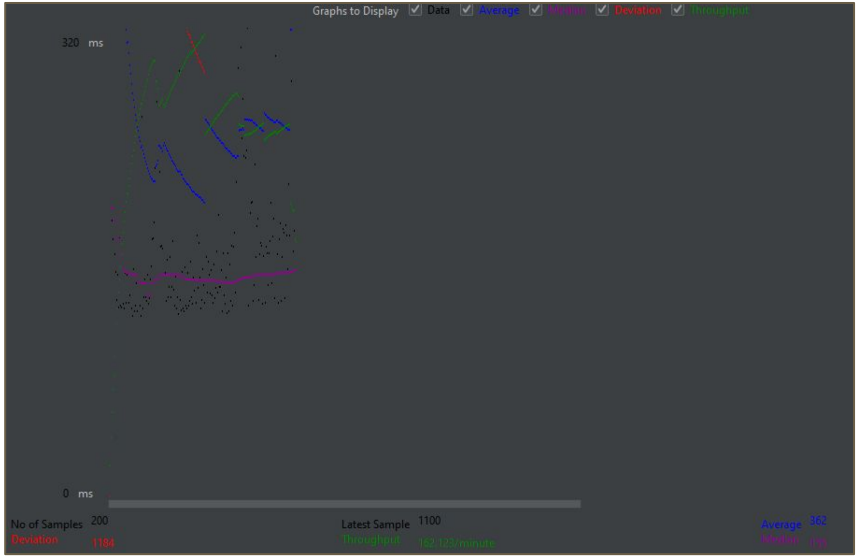
Latest Sample 61
Throughput 237,351/minute

Average 238
Median [illegible]

Average Latency = 238 ms

Result: No. of Transcripts - 100 000





Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 200

Deviation 1184

Latest Sample 1100

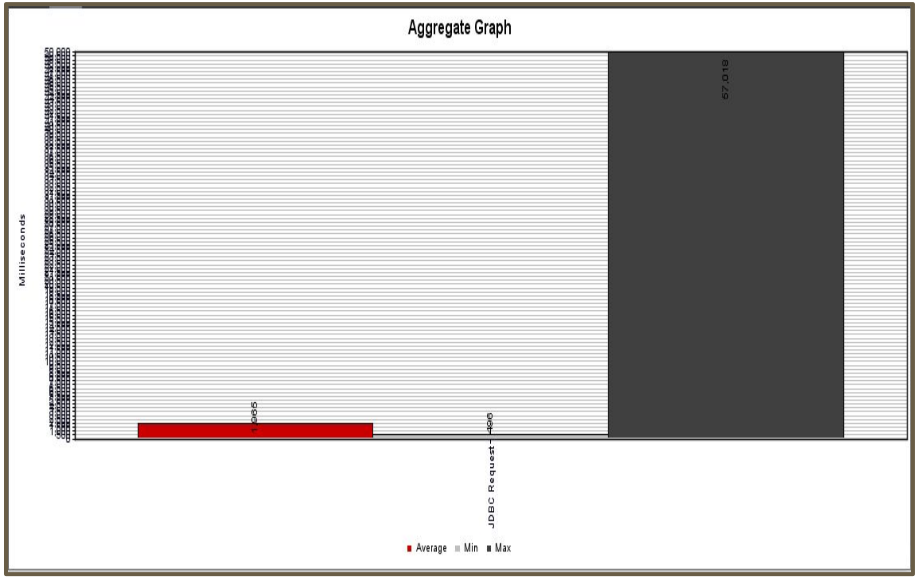
Throughput 162,123/minute

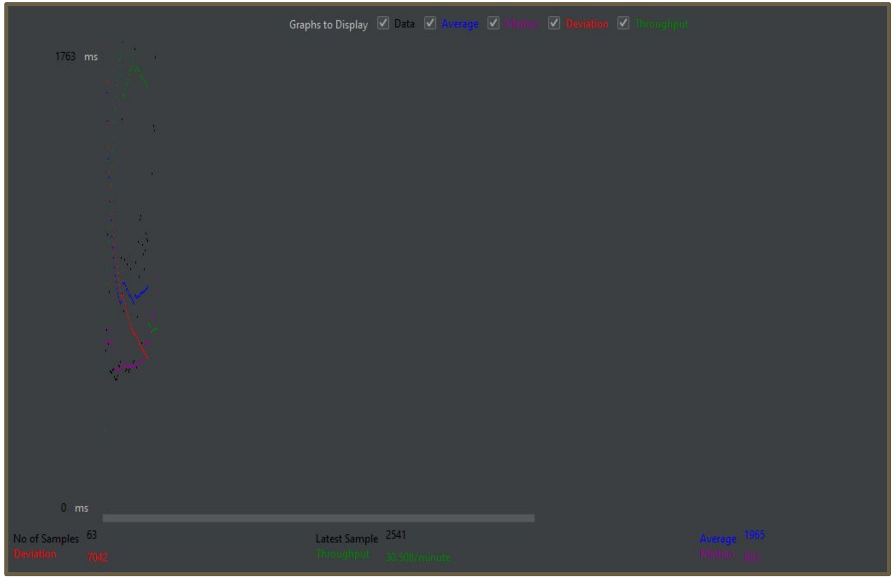
Average 362

Median 135

Average Latency = 362 ms

Result: No. of Transcripts - 300 000



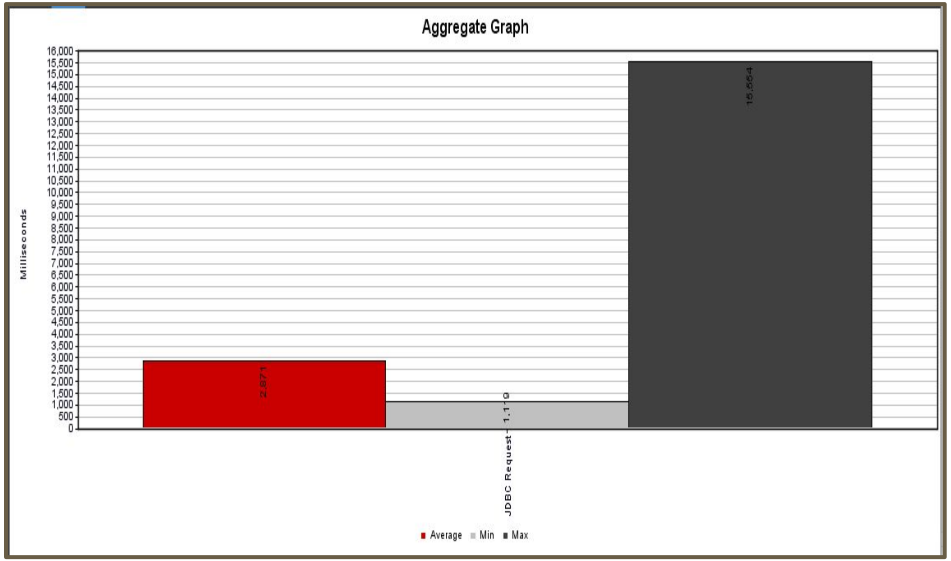


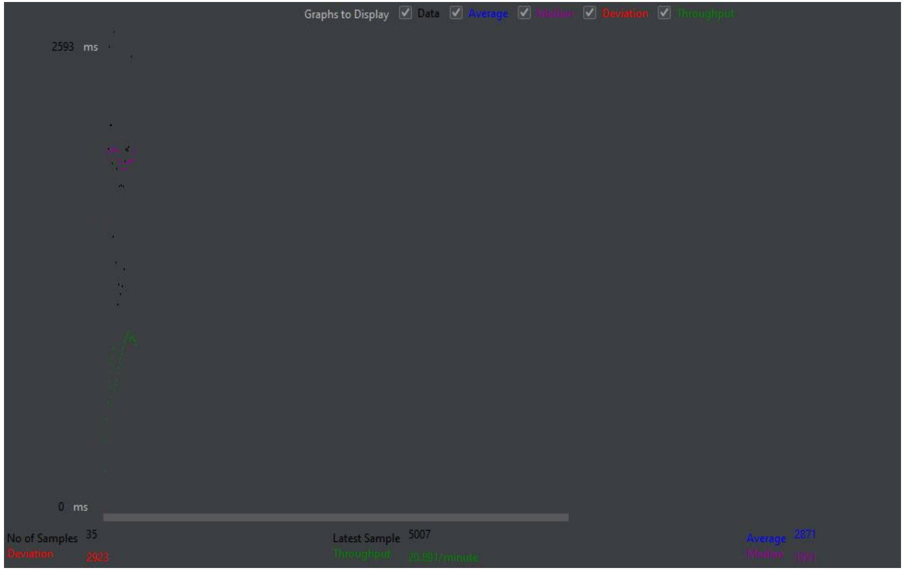
Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 63
Deviation 7042
Latest Sample 2541
Throughput 30,508/minute
Average 1965
Median 883

Average Latency = 1965 ms

Result: No. of Transcripts - 500 000





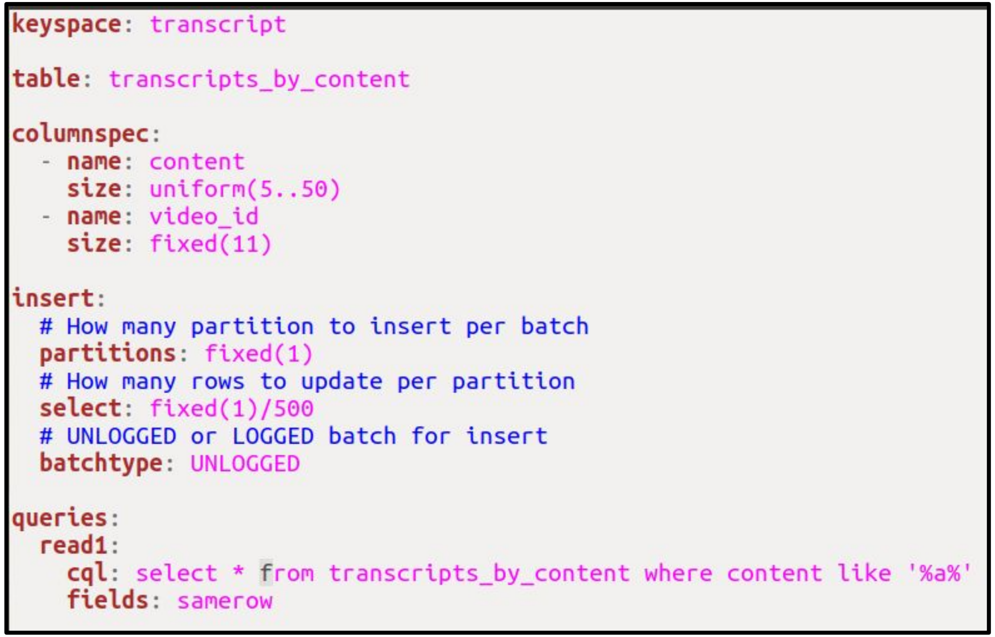
Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 35
Deviation 2923
Latest Sample 5007
Throughput 20,891/minute
Average 2871
Median 1991

Average Latency = 2871 ms

Test Scenario 2: Cassandra

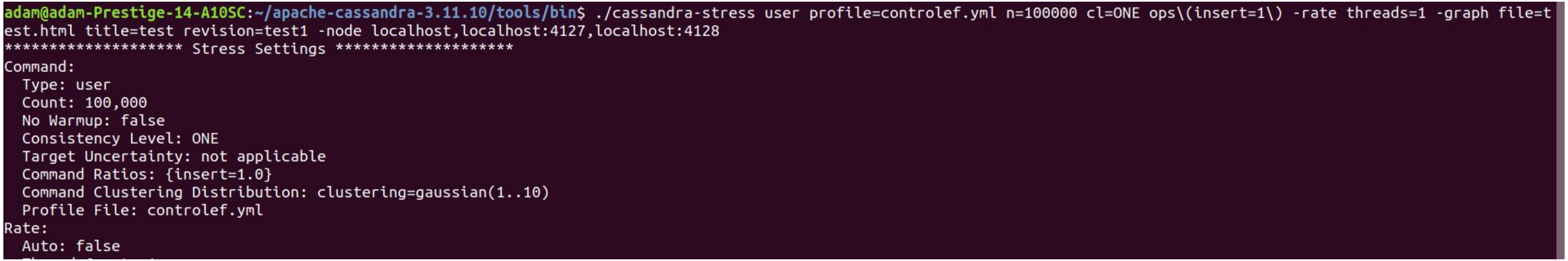
- Use YAML file for config to populate and read data



```yaml
keyspace: transcript

table: transcripts_by_content

columnspec:
  - name: content
    size: uniform(5..50)
  - name: video_id
    size: fixed(11)

insert:
  # How many partition to insert per batch
  partitions: fixed(1)
  # How many rows to update per partition
  select: fixed(1)/500
  # UNLOGGED or LOGGED batch for insert
  batchtype: UNLOGGED

queries:
  read1:
    cql: select * from transcripts_by_content where content like '%a%'
    fields: samerow
```

Test Scenario 2: Cassandra contd.

- Run stress tool to populate the database.

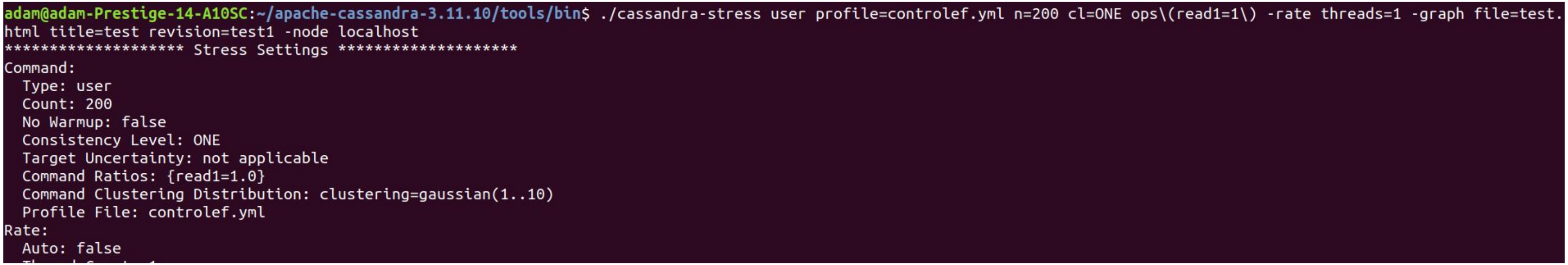


```
adam@adam-Prestige-14-A10SC:~/apache-cassandra-3.11.10/tools/bin$ ./cassandra-stress user profile=controlef.yml n=100000 cl=ONE ops\(insert=1\) -rate threads=1 -graph file=t
est.html title=test revision=test1 -node localhost,localhost:4127,localhost:4128
******************** Stress Settings ********************
Command:
  Type: user
  Count: 100,000
  No Warmup: false
  Consistency Level: ONE
  Target Uncertainty: not applicable
  Command Ratios: {insert=1.0}
  Command Clustering Distribution: clustering=gaussian(1..10)
  Profile File: controlef.yml
Rate:
  Auto: false
```

- `n` here refers to batches - basically the number of times the insert will be run.

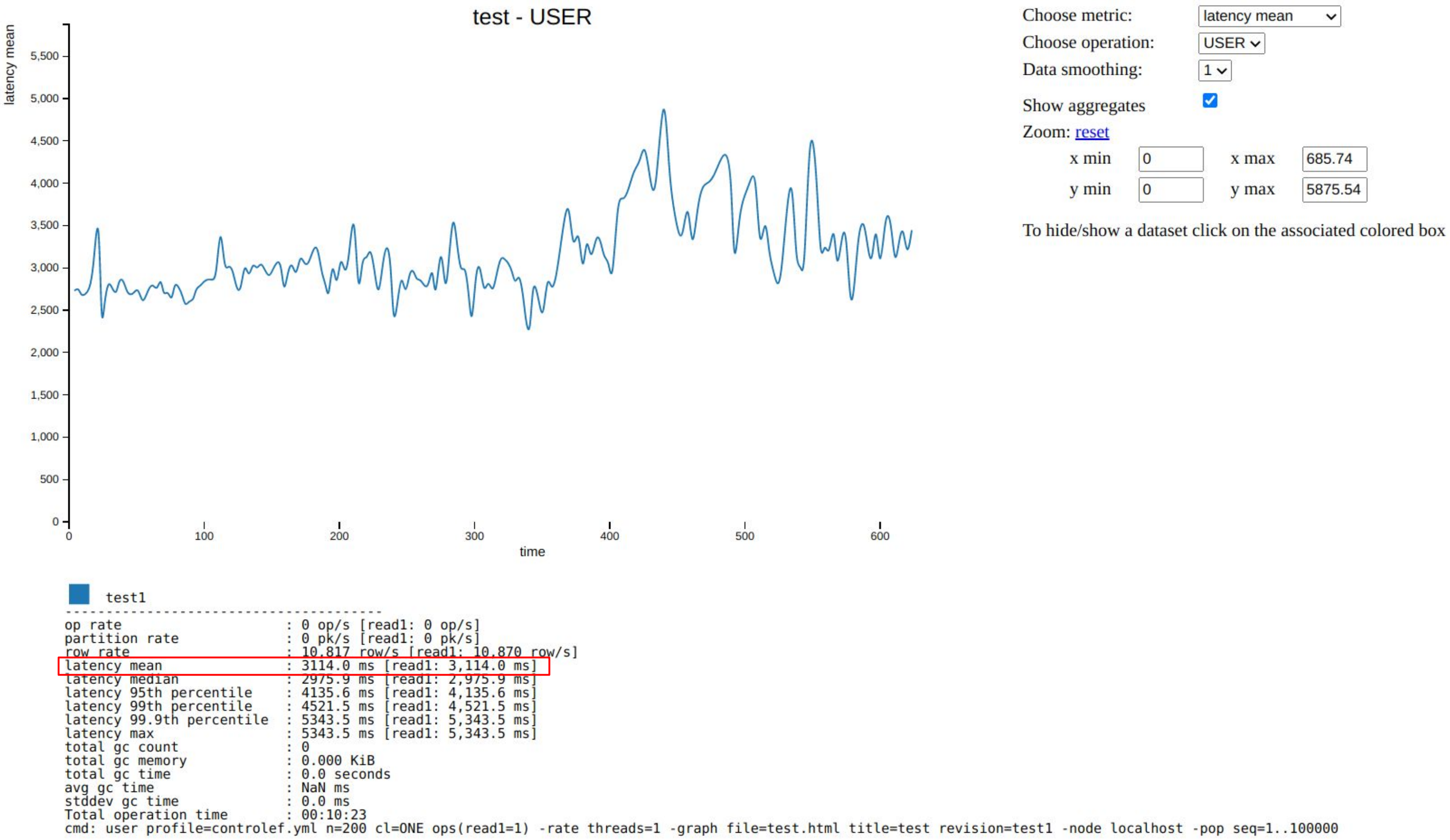
Test Scenario 2: Cassandra contd.

- Run stress tool to read the database.
- Generates an html with a graph of the results



```
adam@adam-Prestige-14-A10SC:~/apache-cassandra-3.11.10/tools/bin$ ./cassandra-stress user profile=controlef.yml n=200 cl=ONE ops\(read1=1\) -rate threads=1 -graph file=test.
html title=test revision=test1 -node localhost
******************** Stress Settings ********************
Command:
  Type: user
  Count: 200
  No Warmup: false
  Consistency Level: ONE
  Target Uncertainty: not applicable
  Command Ratios: {read1=1.0}
  Command Clustering Distribution: clustering=gaussian(1..10)
  Profile File: controlef.yml
Rate:
  Auto: false
  Thread Count: 1
```

Test Scenario 2: Cassandra contd. - Read Results



Choose metric: latency mean

Choose operation: USER

Data smoothing: 1

Show aggregates ☑

Zoom: reset

x min 0 x max 685.74

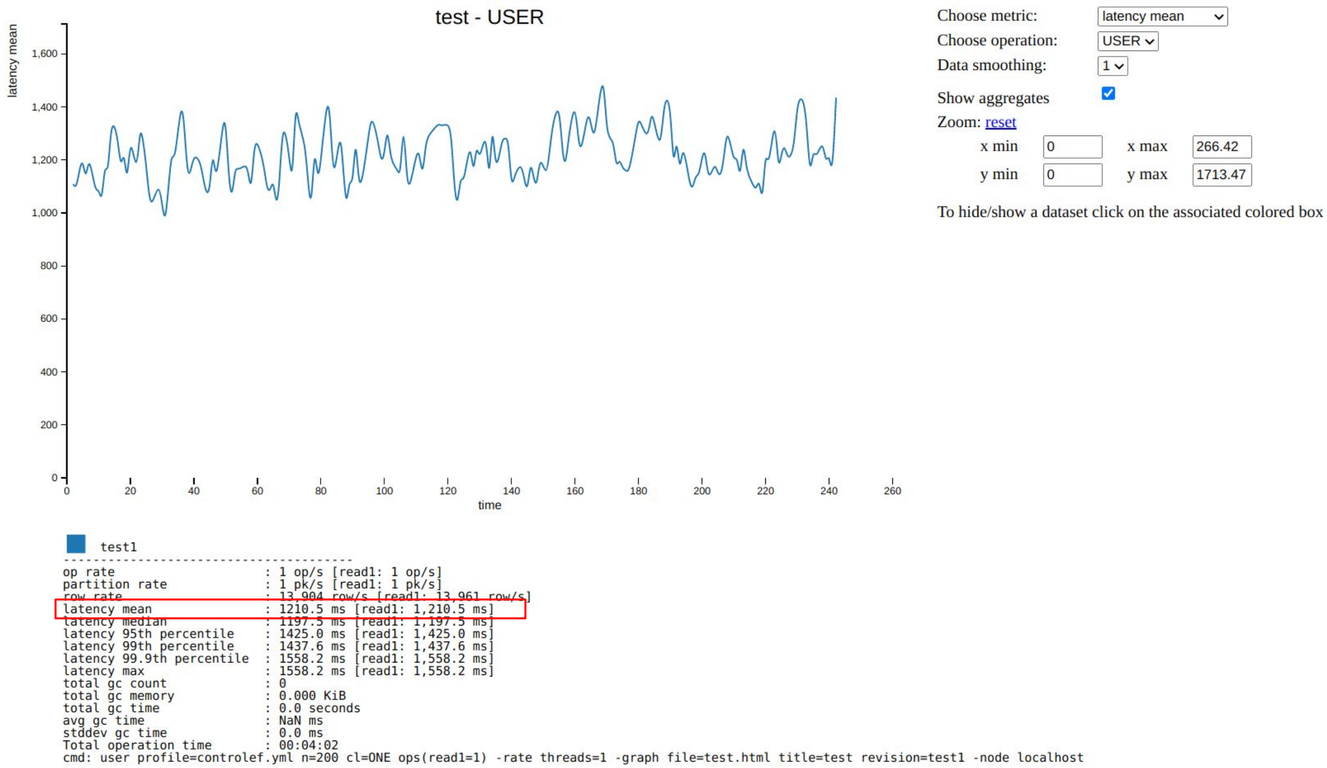
y min 0 y max 5875.54

To hide/show a dataset click on the associated colored box

```
🟦 test1
---------------------------------------
op rate                   : 0 op/s [read1: 0 op/s]
partition rate            : 0 pk/s [read1: 0 pk/s]
row rate                  : 10,817 row/s [read1: 10,870 row/s]
latency mean              : 3114.0 ms [read1: 3,114.0 ms]
latency median            : 2975.9 ms [read1: 2,975.9 ms]
latency 95th percentile   : 4135.6 ms [read1: 4,135.6 ms]
latency 99th percentile   : 4521.5 ms [read1: 4,521.5 ms]
latency 99.9th percentile : 5343.5 ms [read1: 5,343.5 ms]
latency max               : 5343.5 ms [read1: 5,343.5 ms]
total gc count            : 0
total gc memory           : 0.000 KiB
total gc time             : 0.0 seconds
avg gc time               : NaN ms
stddev gc time            : 0.0 ms
Total operation time      : 00:10:23
cmd: user profile=controlef.yml n=200 cl=ONE ops(read1=1) -rate threads=1 -graph file=test.html title=test revision=test1 -node localhost -pop seq=1..100000
```

Test Scenario 2: Cassandra contd.

- Repeat for other data record levels (2k, 5k, 10k, **50k (shown below)** etc.)



Choose metric: latency mean

Choose operation: USER

Data smoothing: 1

Show aggregates ☑

Zoom: reset

x min 0 x max 266.42

y min 0 y max 1713.47

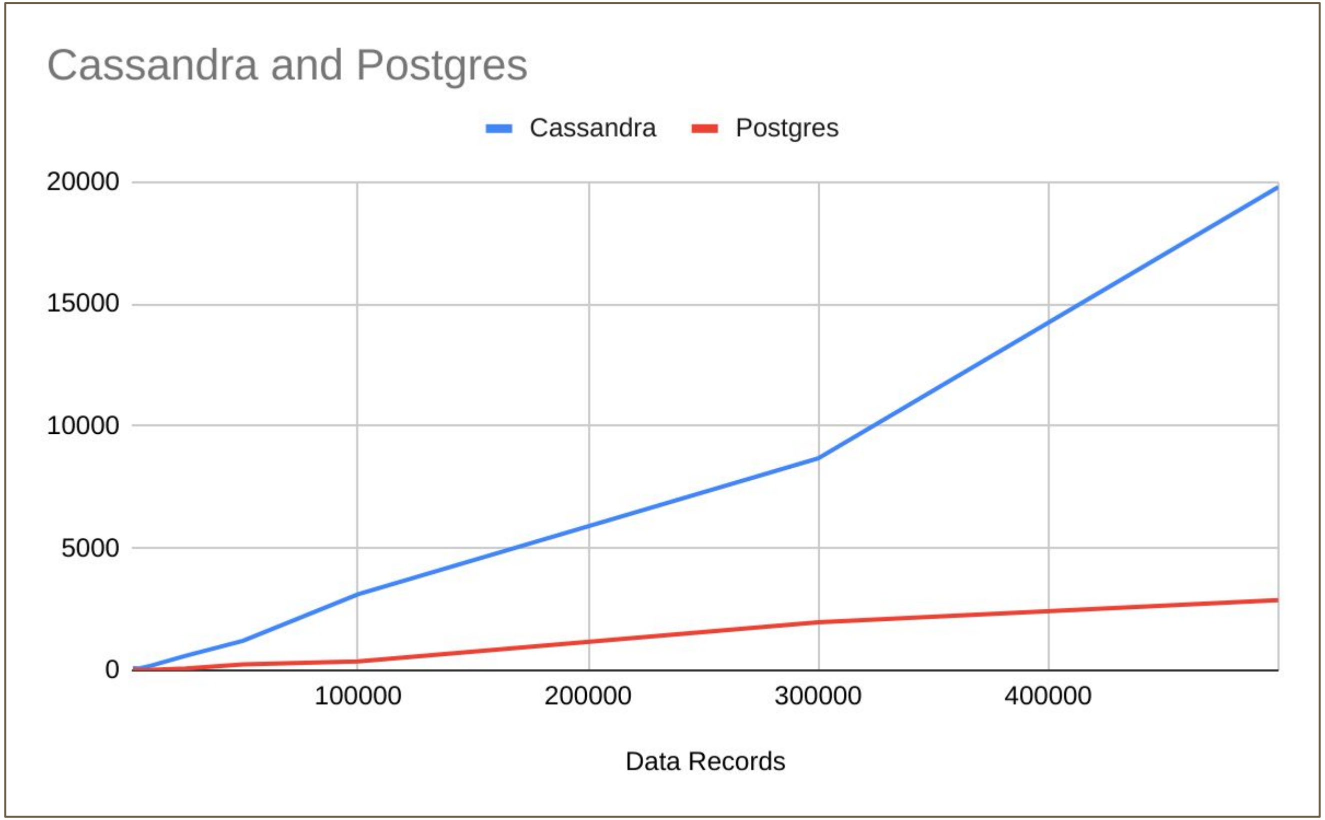
To hide/show a dataset click on the associated colored box

```
■  test1
---------------------------------------
op rate                   : 1 op/s [read1: 1 op/s]
partition rate            : 1 pk/s [read1: 1 pk/s]
row rate                  : 13,904 row/s [read1: 13,961 row/s]
latency mean              : 1210.5 ms [read1: 1,210.5 ms]
latency median            : 1197.5 ms [read1: 1,197.5 ms]
latency 95th percentile   : 1425.0 ms [read1: 1,425.0 ms]
latency 99th percentile   : 1437.6 ms [read1: 1,437.6 ms]
latency 99.9th percentile : 1558.2 ms [read1: 1,558.2 ms]
latency max               : 1558.2 ms [read1: 1,558.2 ms]
total gc count            : 0
total gc memory           : 0.000 KiB
total gc time             : 0.0 seconds
avg gc time               : NaN ms
stddev gc time            : 0.0 ms
Total operation time      : 00:04:02
cmd: user profile=controlef.yml n=200 cl=ONE ops(read1=1) -rate threads=1 -graph file=test.html title=test revision=test1 -node localhost
```

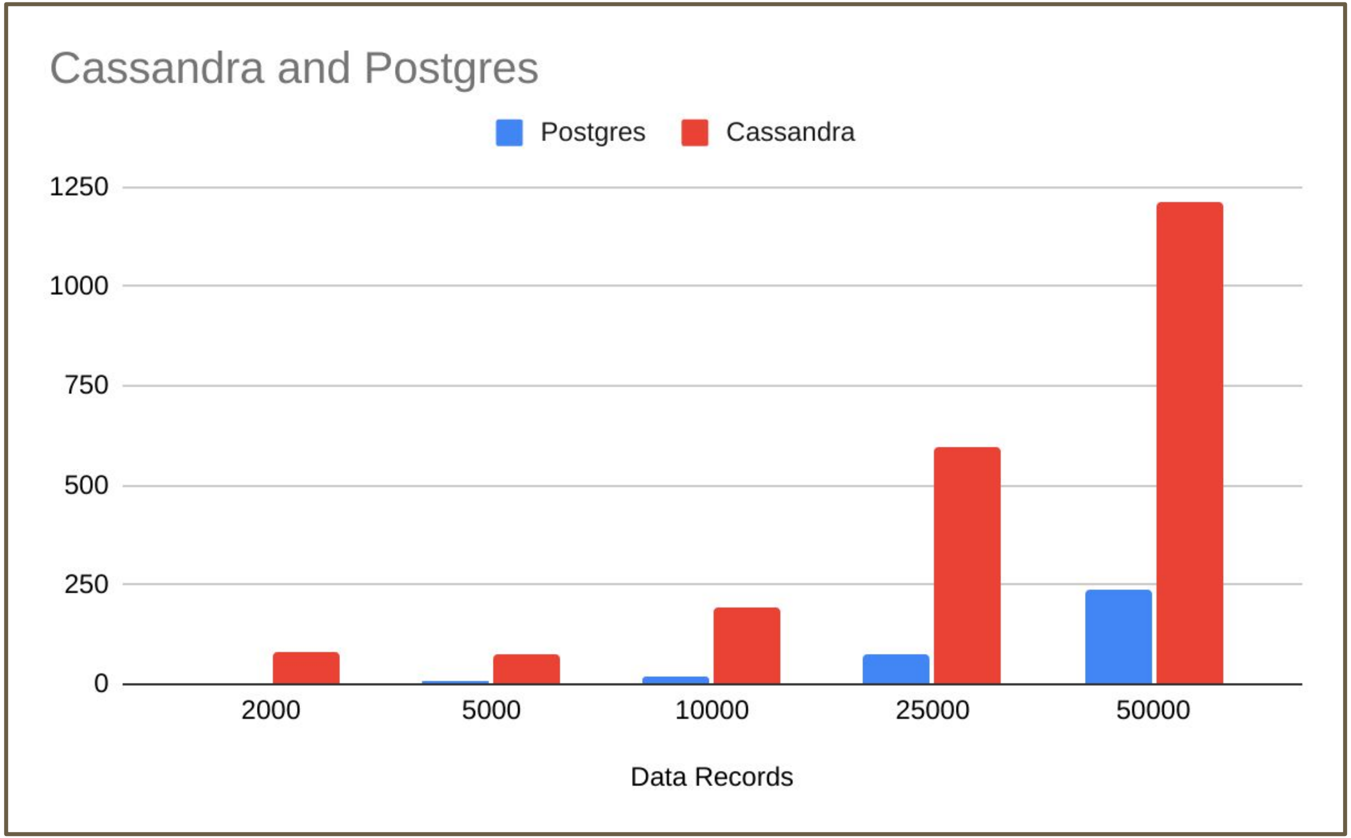
Test Scenario 2 - Cassandra contd.

See this link in the Github Repository for instructions on how to run cassandra-stress

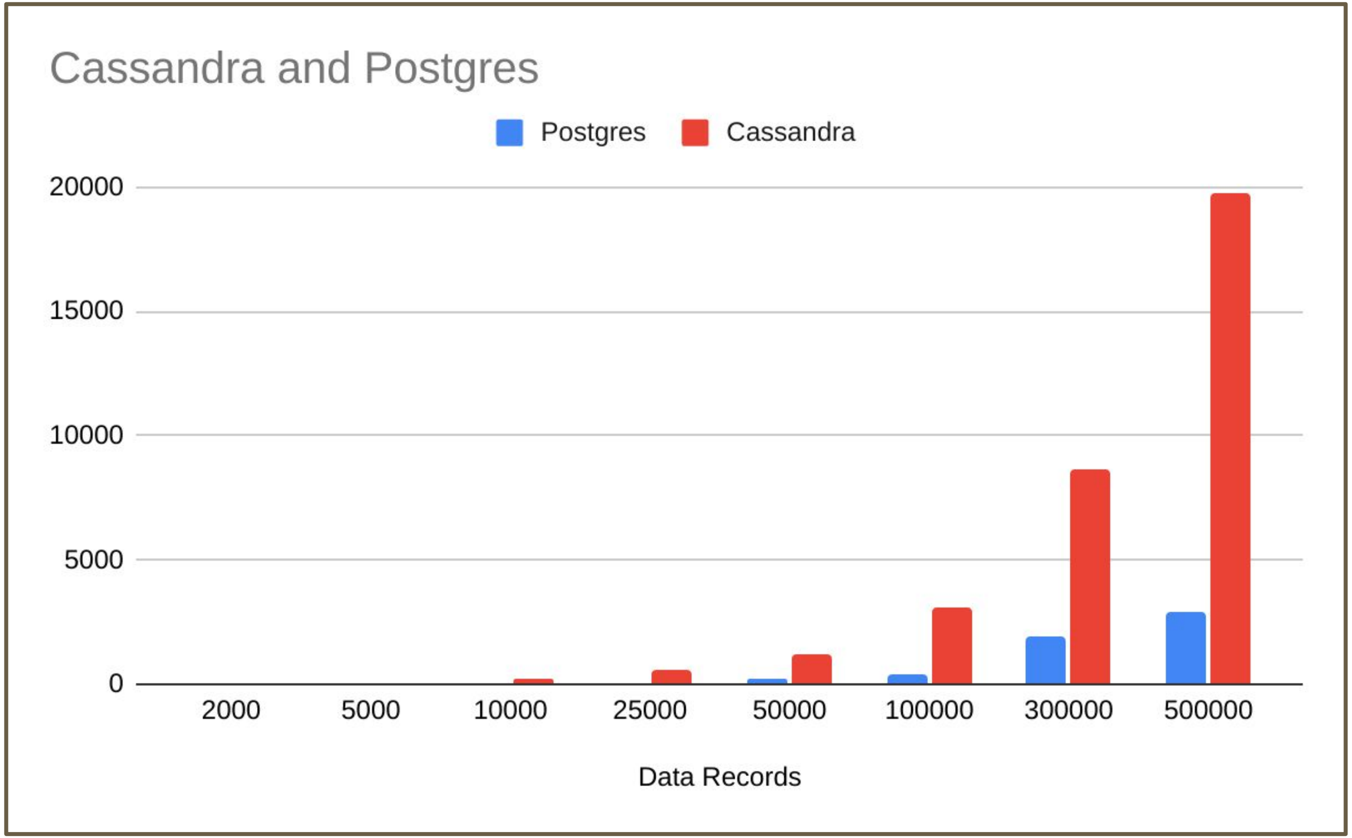
Final Result - Comparing Postgres and Cassandra



Final Result - Comparing Postgres and Cassandra

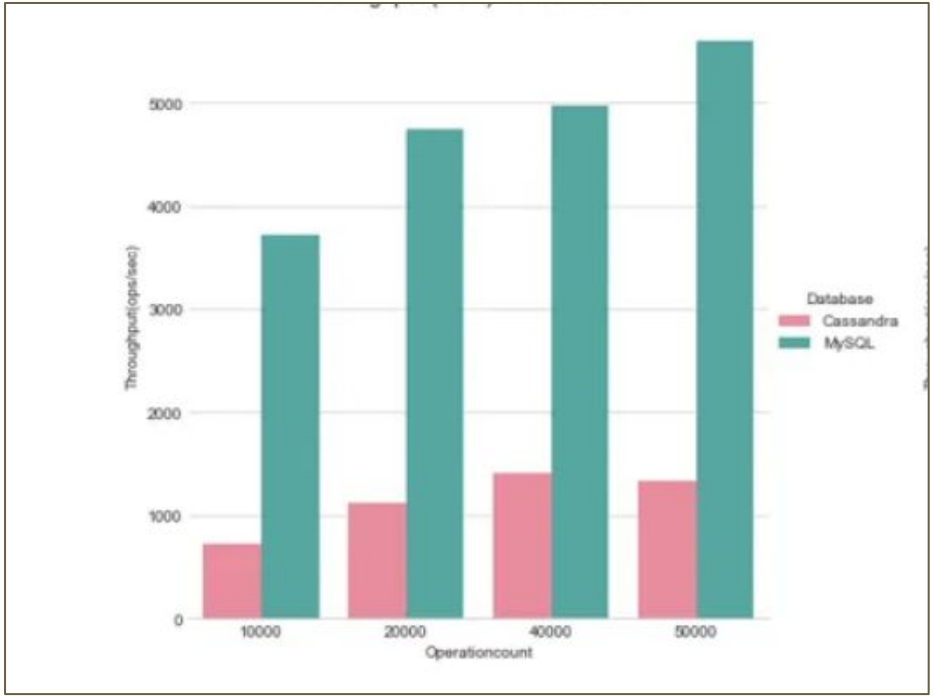


Final Result - Comparing Postgres and Cassandra



Relation to other findings

- Workload C (Read-only)
- https://adataanalyst.com/data-analysis-resources/a-comparison-between-cassandra-and-mysql/



Relation to other findings (contd.)

- Mahmood, K. (2016). Performance Comparison of NOSQL Database Cassandra and SQL Server for Large Databases. *Journal of Independent Studies and Research (JISR)*, *14*(2).
- But questionable?

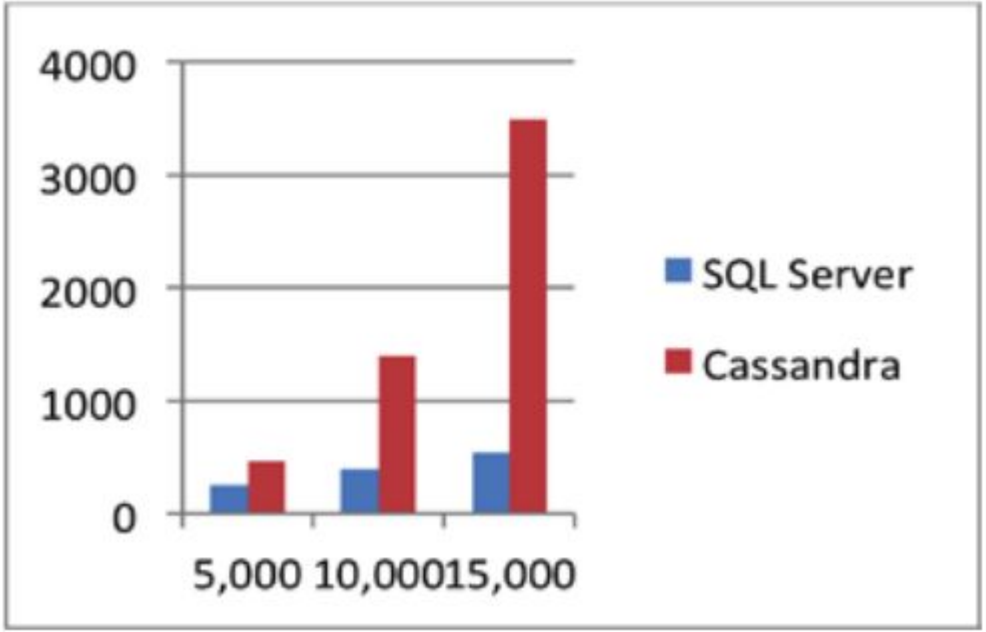
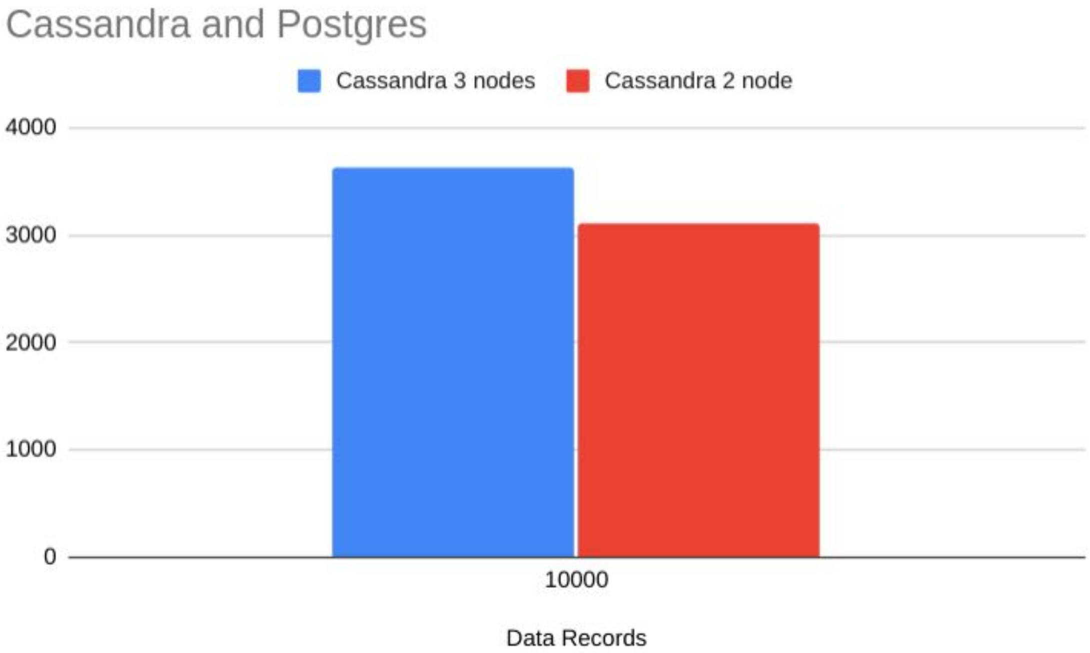


Fig. (3). Select /Read Records by SQL Server and Cassandra

Bonus Ablation Study (2 node vs 3 node)

- 3 node takes longer than 2 nodes
- However, difference cannot be said as significant
- Need more samples



Quality Attribute Analysis - Conclusion

Availability	**H**	Cassandra has higher availability than Postgres because of replication ability. If nodes are down in cluster, then other nodes have backup.
Performance	**H**	Postgres seems to be better in terms of read performance according to our testing.
Scalability	**H**	Cassandra will definitely be more scalable. Postgres only supports vertical scalability whereas Cassandra supports horizontal scalability. We can easily add more nodes whether in same local machine or on a remote machine. See README for info.

Conclusion and Analysis

- We couldn’t do more than 500k - took too long time
- Reason for inferior Cassandra Performance may be the lack of FKs and joins in query
- If we were having joins in our queries, then Cassandra would be superior.
- Perhaps indexing and in-memory databases may be better
- Possible limitations in the use of Docker containers
- Possible config issues

Future work:

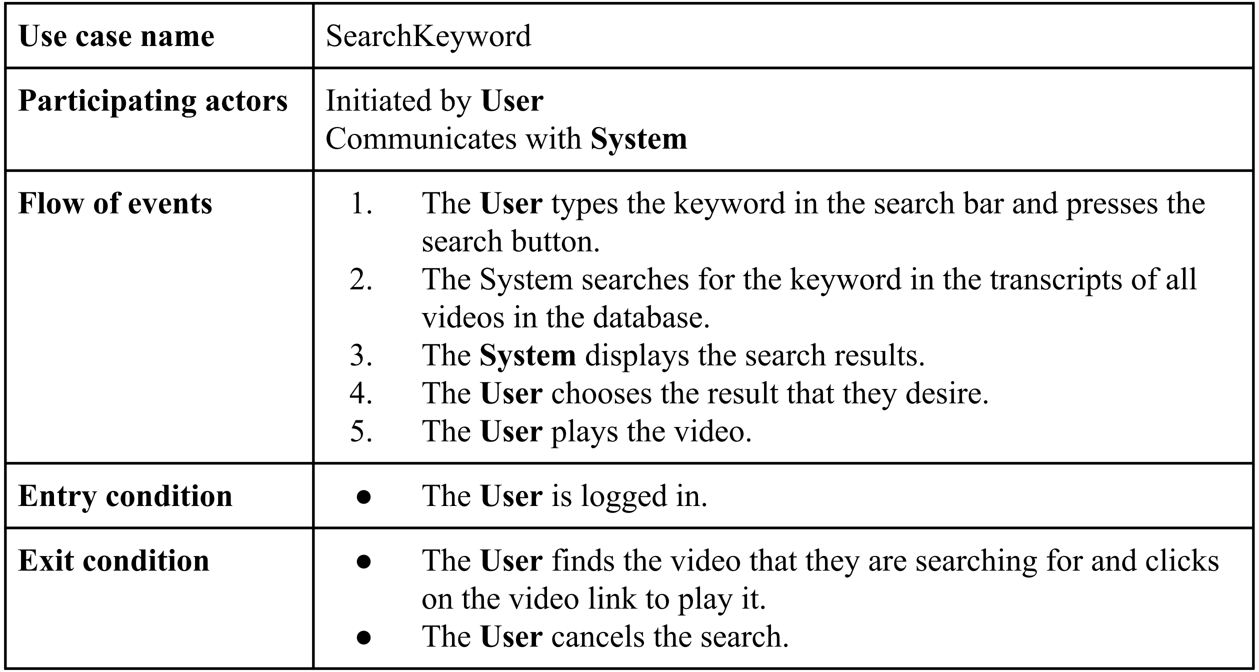
- Youtube Integration will be added later as we wanted to focus on finding a suitable architecture first before continuing development

USE CASES

&

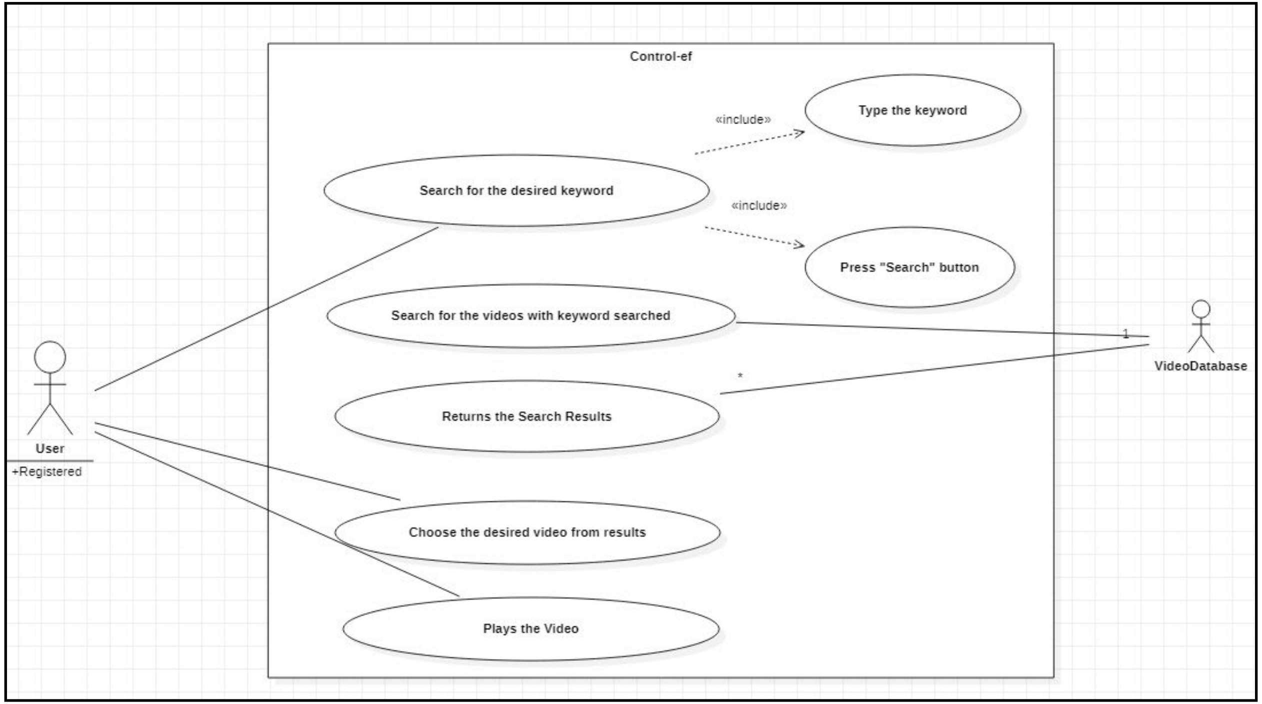
UI DESIGN

Use Case: SearchKeyword



Use case name	SearchKeyword
Participating actors	Initiated by **User** Communicates with **System**
Flow of events	1. The **User** types the keyword in the search bar and presses the search button. 2. The System searches for the keyword in the transcripts of all videos in the database. 3. The **System** displays the search results. 4. The **User** chooses the result that they desire. 5. The **User** plays the video.
Entry condition	● The **User** is logged in.
Exit condition	● The **User** finds the video that they are searching for and clicks on the video link to play it. ● The **User** cancels the search.

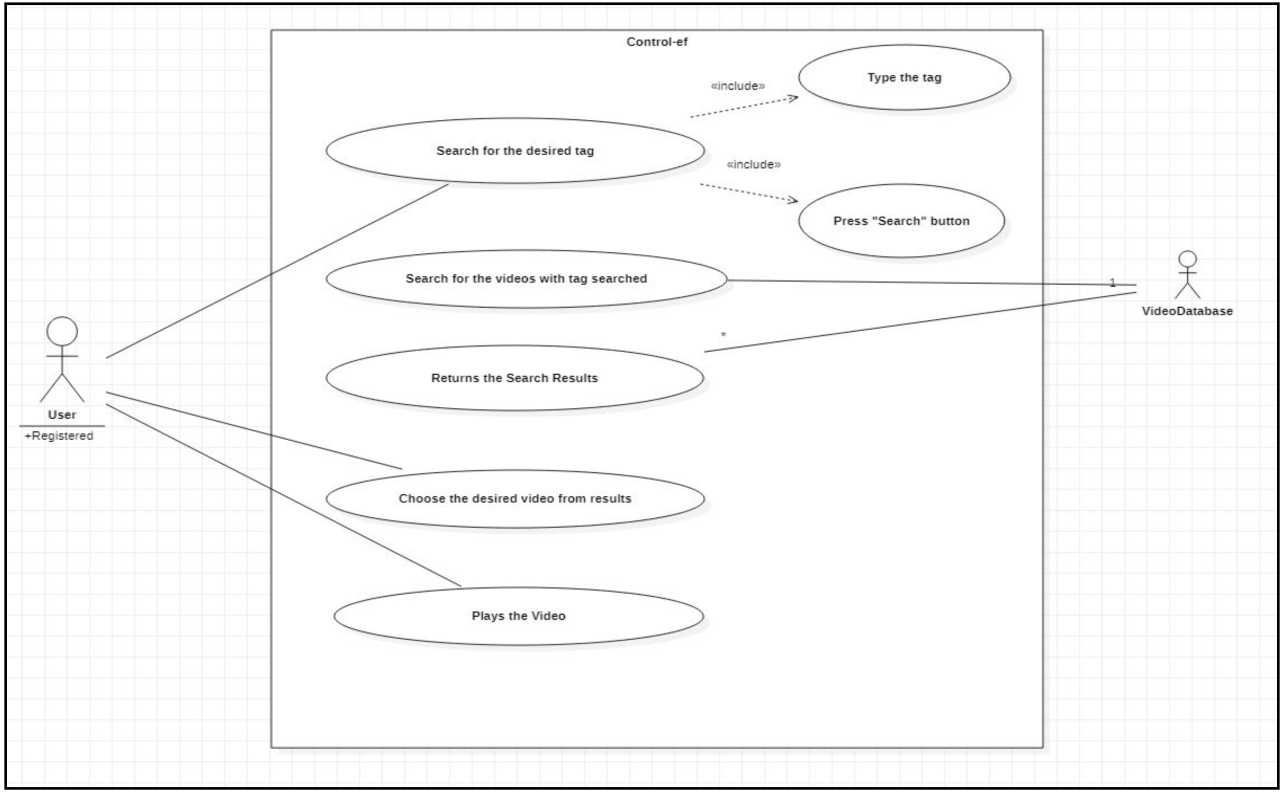
Use Case: SearchKeyword


Control-ef
«include»
Type the keyword
Search for the desired keyword
«include»
Press "Search" button
Search for the videos with keyword searched
1
VideoDatabase
*
Returns the Search Results
User
+Registered
Choose the desired video from results
Plays the Video

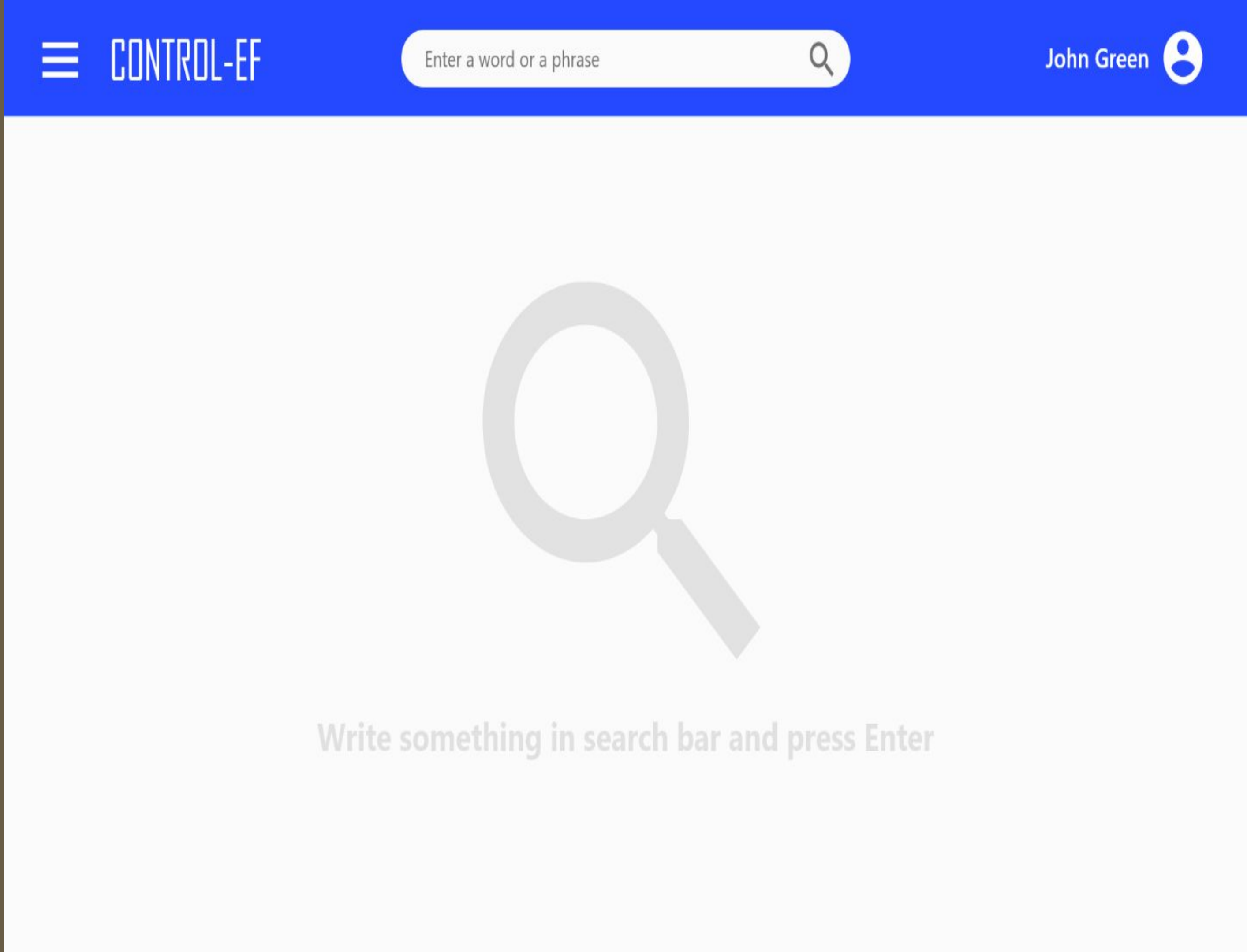
Use Case: SearchTag

Use case name	SearchTag
Participating actors	Initiated by **User** Communicates with **System**
Flow of events	1. The **User** types the tag in the search bar and presses the search button. 2. The **System** searches for all videos under that predefined tag in the database. 3. The **System** displays the search results. 4. The **User** chooses the result that they desire. 5. The **User** plays the video.
Entry condition	● The **User** is logged in.
Exit condition	● The **User** finds the video that they are searching for and clicks on the video link to play it. ● The **User** cancels the search.

Use Case: SearchTag

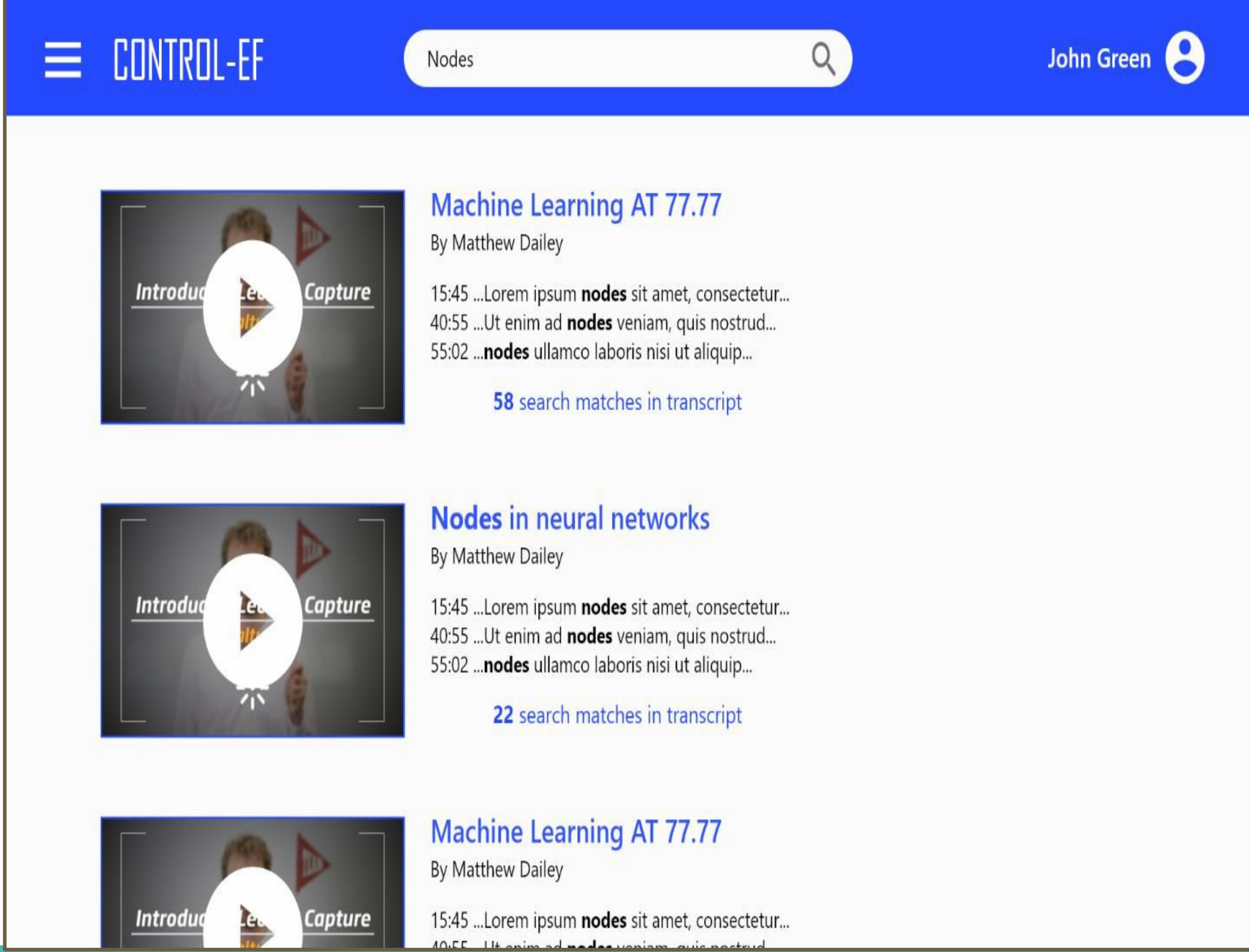

Control-ef
«include»
Type the tag
Search for the desired tag
«include»
Press "Search" button
Search for the videos with tag searched
1
VideoDatabase
*
Returns the Search Results
User
+Registered
Choose the desired video from results
Plays the Video

UI Design



Write something in search bar and press Enter

UI Design



Machine Learning AT 77.77

By Matthew Dailey

15:45 ...Lorem ipsum **nodes** sit amet, consectetur...
40:55 ...Ut enim ad **nodes** veniam, quis nostrud...
55:02 ...**nodes** ullamco laboris nisi ut aliquip...

58 search matches in transcript

Nodes in neural networks

By Matthew Dailey

15:45 ...Lorem ipsum **nodes** sit amet, consectetur...
40:55 ...Ut enim ad **nodes** veniam, quis nostrud...
55:02 ...**nodes** ullamco laboris nisi ut aliquip...

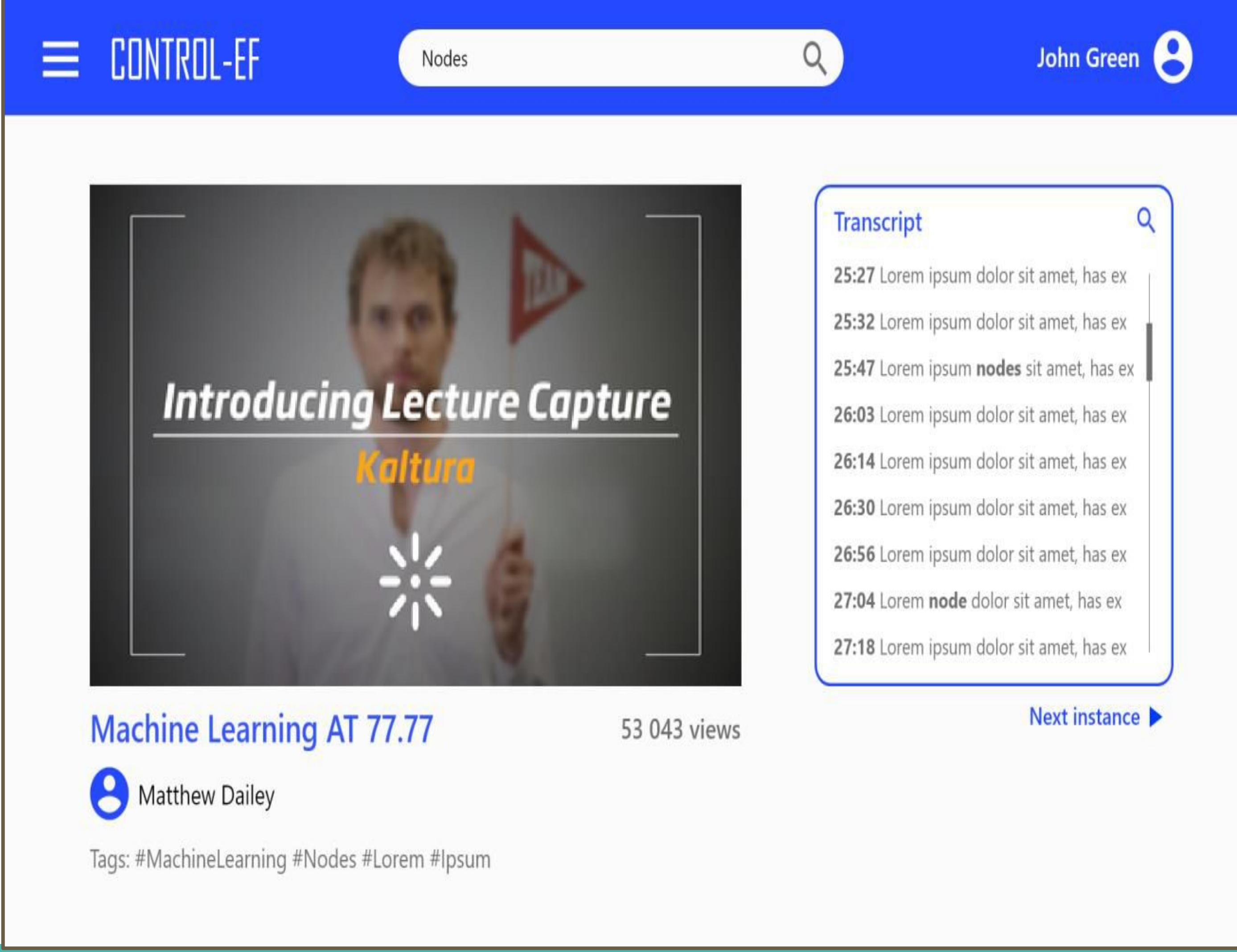
22 search matches in transcript

Machine Learning AT 77.77

By Matthew Dailey

15:45 ...Lorem ipsum **nodes** sit amet, consectetur...
40:55 ...Ut enim ad **nodes** veniam, quis nostrud...

UI Design



Nodes

25:27 Lorem ipsum dolor sit amet, has ex

25:32 Lorem ipsum dolor sit amet, has ex

25:47 Lorem ipsum **nodes** sit amet, has ex

26:03 Lorem ipsum dolor sit amet, has ex

26:14 Lorem ipsum dolor sit amet, has ex

26:30 Lorem ipsum dolor sit amet, has ex

26:56 Lorem ipsum dolor sit amet, has ex

27:04 Lorem **node** dolor sit amet, has ex

27:18 Lorem ipsum dolor sit amet, has ex

Machine Learning AT 77.77

53 043 views

Matthew Dailey

Tags: #MachineLearning #Nodes #Lorem #Ipsum

Project Demo

Conclusion

- New problems always come up after coding
- Scope creep/inflation when we try something new
- Finding a suitable architecture or architectures
- Development/production environment discrepancy

рахмат
danke
謝謝
ngiyabonga
spas
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
malondo
paldies
grazzi
gracias
kiitos
dankie
nanni
naudi
dhanyavad
bedankt
bayarlalaa
gracie
hvala
maururu
köszönöm
thank you
dankon
akun
хвала
asante
manana
obrigada
tinki
murakoze
djere dieuf
tau
mochchakkeram
enkosi
dziękuję
дякую
mamnun
chnorakaloutioun
gratias ago
gràcies
sulpay
taiku
go raibh maith agat
choukrane
sobodi
dėkuji
sagolun
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
obrigado
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
terima kasih
rahmet
ευχαριστώ
diolch
dhanyavadagalu
shukriya
merce
мерси
kam sah hamnida
rahmat
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci