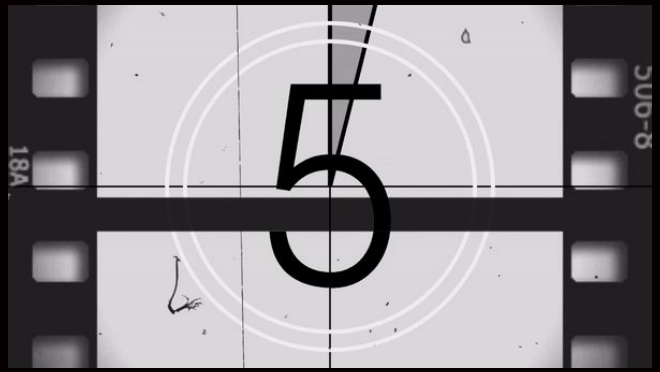
MCO2

Distributed Database

IMDb

Cua, Lander Peter E.
Gaba, Jacob Bryan B.
Gonzales, Mark Edward M.
Lee, Hylene Jules G.



01

DISTRIBUTED DATABASE SETUP

Cloud platform, data sharding, remote database access

02

ALGORITHM

Flow, happy paths and unhappy paths

03

GLOBAL CONCURRENCY

System handling of concurrent transactions

04

GLOBAL RECOVERY

System response to crashing nodes

05

EVALUATION AND RESULTS

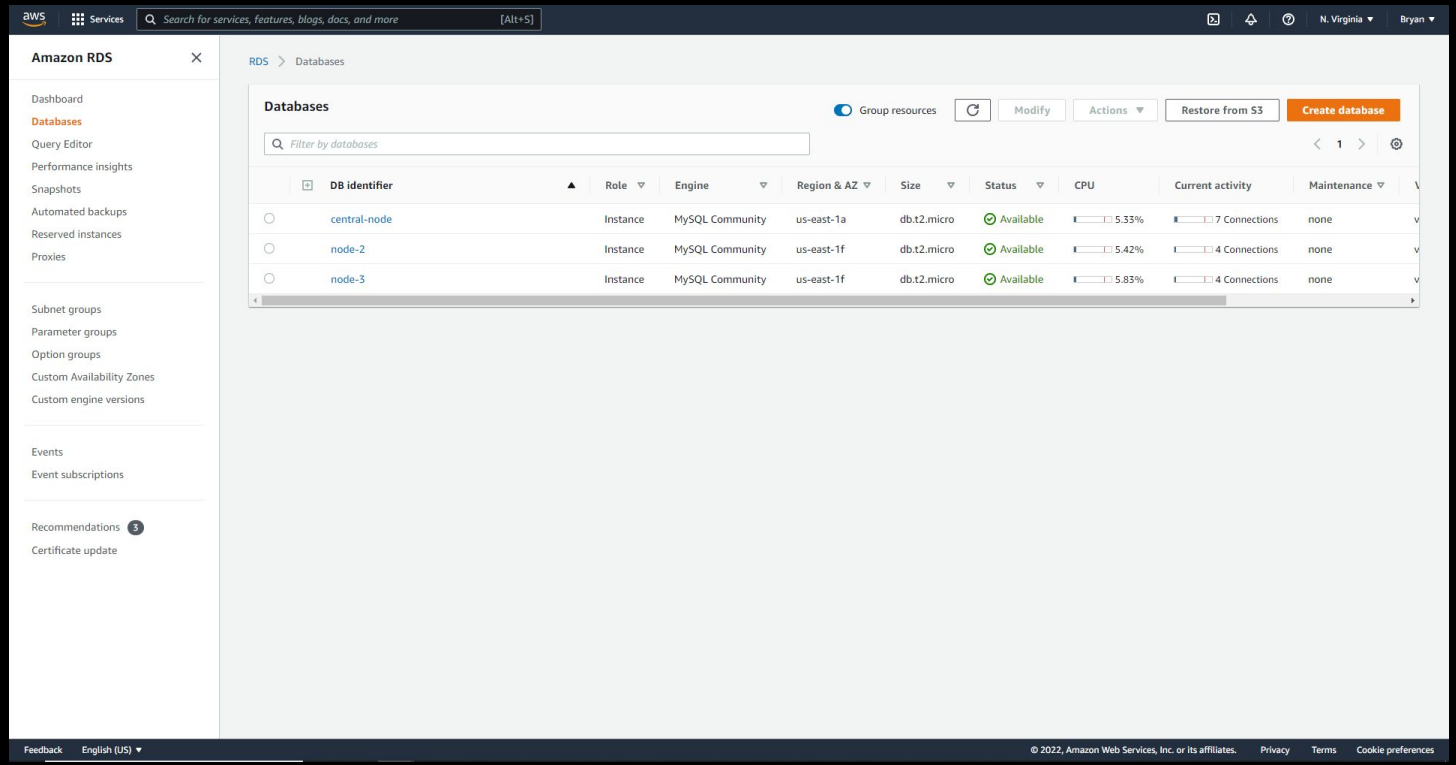
Concurrency and failure recovery validation

06

INSIGHTS AND CONCLUSION

Summary of findings and analyses

Cloud Platform AWS RDS



Database Access

```
const centralPool = mySQL.createPool({
    connectionLimit: 20,
    host: process.env.CENTRAL_URL,
    port: process.env.DB_PORT,
    user: process.env.CENTRAL_USERNAME,
    password: process.env.CENTRAL_PASSWORD,
    database: DATABASE,
    connectTimeout: 30000
});

const node2Pool = mySQL.createPool({
    connectionLimit: 20,
    host: process.env.NODE2_URL,
    port: process.env.DB_PORT,
    user: process.env.NODE2_USERNAME,
    password: process.env.NODE2_PASSWORD,
    database: DATABASE,
    connectTimeout: 30000
});

const node3Pool = mySQL.createPool({
    connectionLimit: 20,
    host: process.env.NODE3_URL,
    port: process.env.DB_PORT,
    user: process.env.NODE3_USERNAME,
    password: process.env.NODE3_PASSWORD,
    database: DATABASE,
    connectTimeout: 30000
});
```

```
PORT = 3000
DB_PORT = 3306
HOSTNAME = localhost
CENTRAL_URL = central-node.cb7q6f4hhftm.us-east-1.rds.amazonaws.com
CENTRAL_USERNAME = centralAdmin
CENTRAL_PASSWORD = centralPassword
NODE2_URL = node-2.cb7q6f4hhftm.us-east-1.rds.amazonaws.com
NODE2_USERNAME = node2Admin
NODE2_PASSWORD = node2Password
NODE3_URL = node-3.cb7q6f4hhftm.us-east-1.rds.amazonaws.com
NODE3_USERNAME = node3Admin
NODE3_PASSWORD = node3Password
```

CONNECTION POOL

Database Schema Movies Table

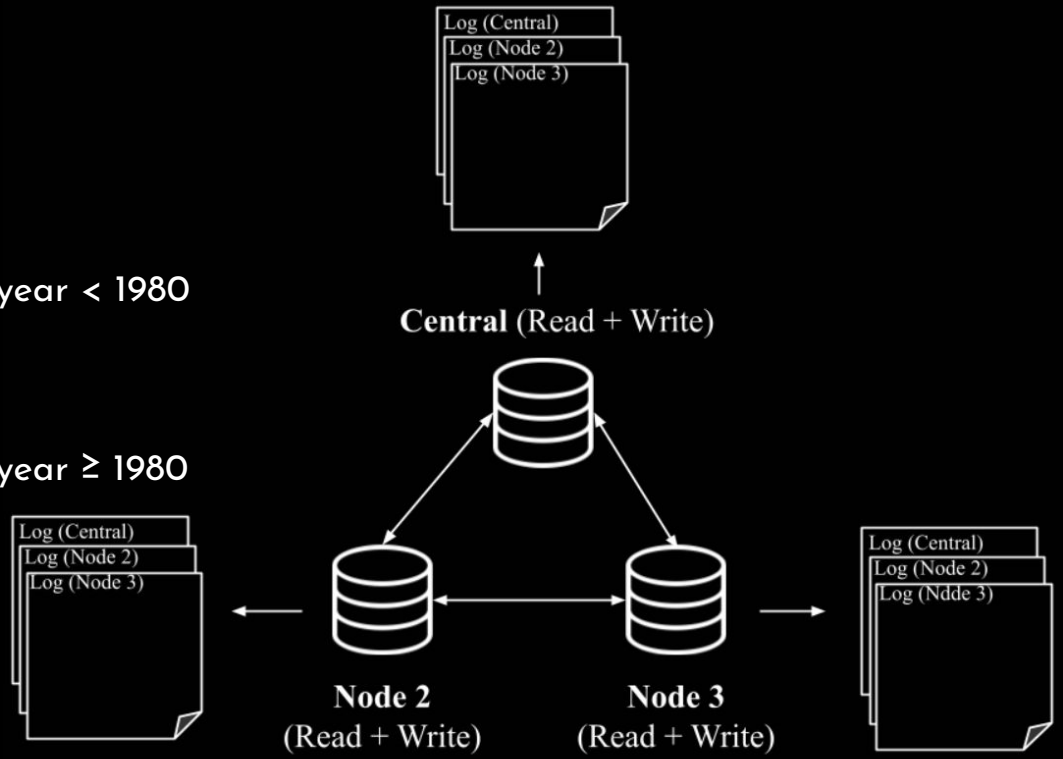
Attribute	Data Type	Description
ID	int	ID of the movie (primary key)
title	varchar	Title of the movie
year	int	Year of release of the movie
genre	varchar	Genre of the movie
rank	float	Rating of the movie
director	varchar	Director of the movie
actor1	varchar	First actor in the movie
actor2	varchar	Second actor in the movie

Distributed Database Topology

CENTRAL NODE
- Contains all movies
- 173394 entries

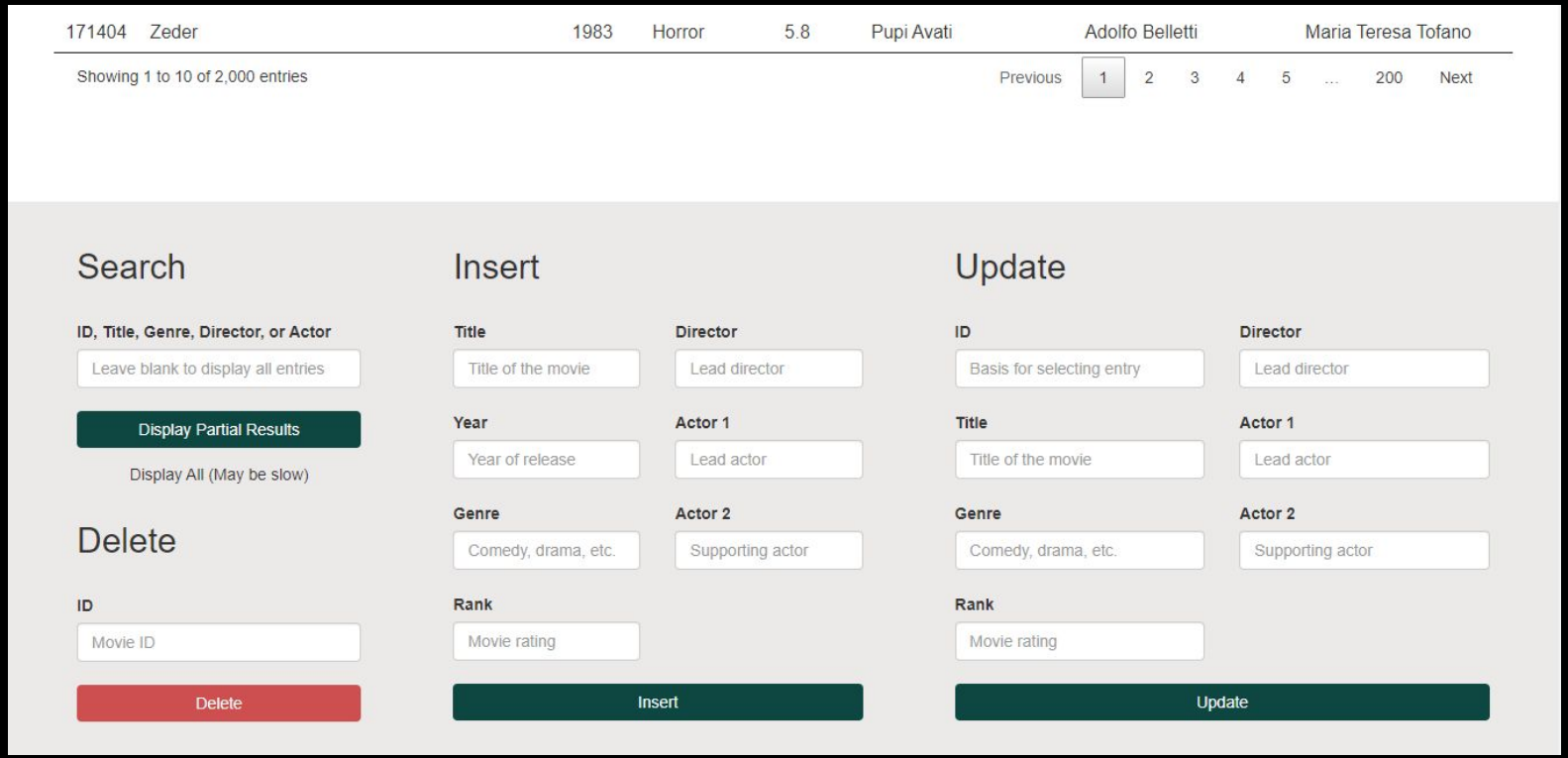
NODE 2
- Contains all movies with year < 1980
- 92003 entries

NODE 3
- Contains all movies with year ≥ 1980
- 81391 entries



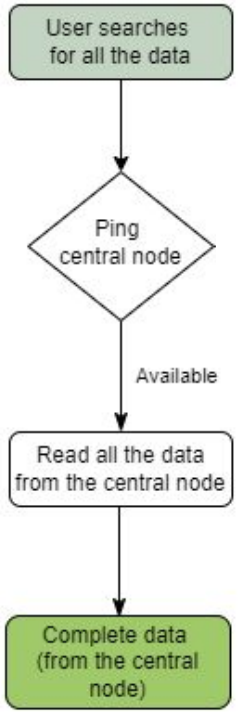
Web Application

https://cggl-distributed-db.herokuapp.com/



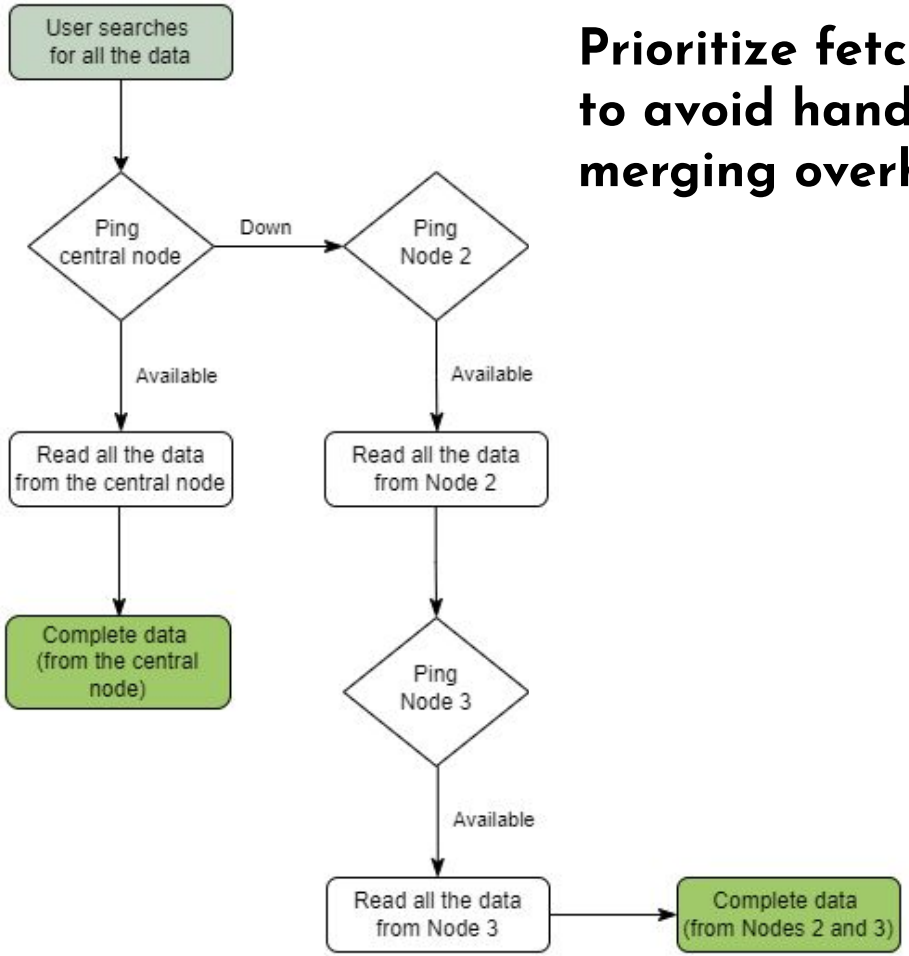
Web Application

Retrieve All

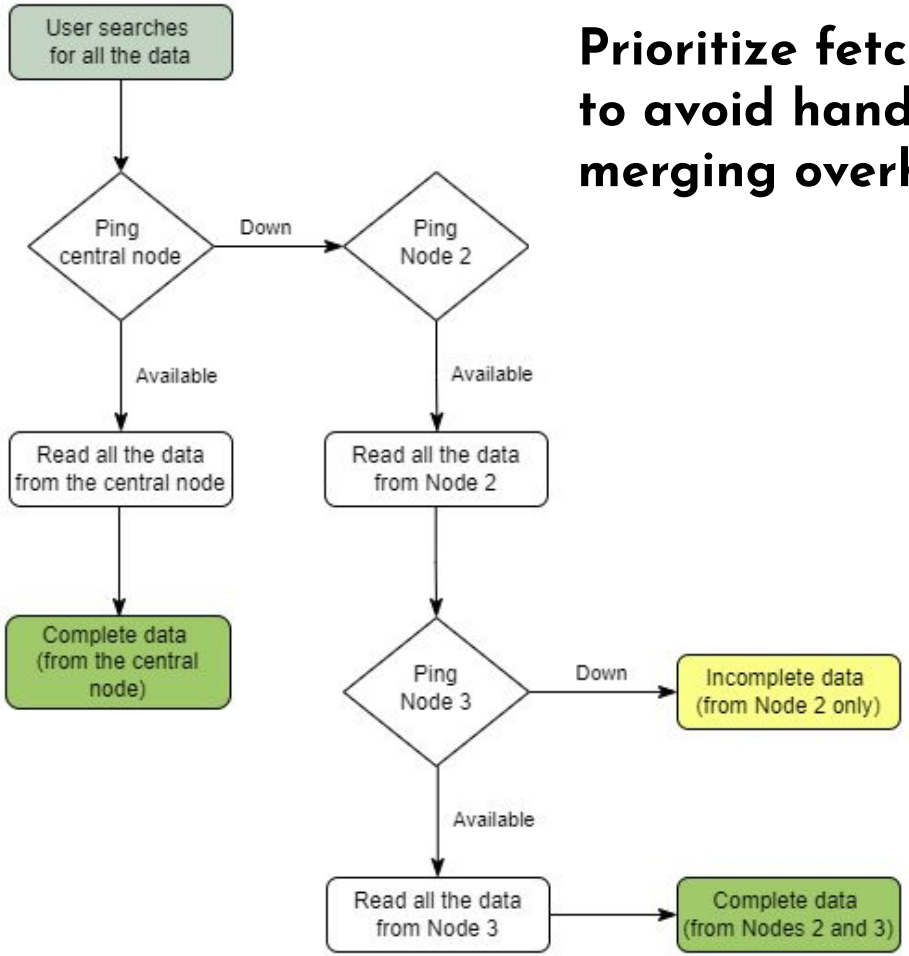


Prioritize fetching from central node to avoid handshaking and result set merging overhead

ALGORITHM (RETRIEVE ALL)

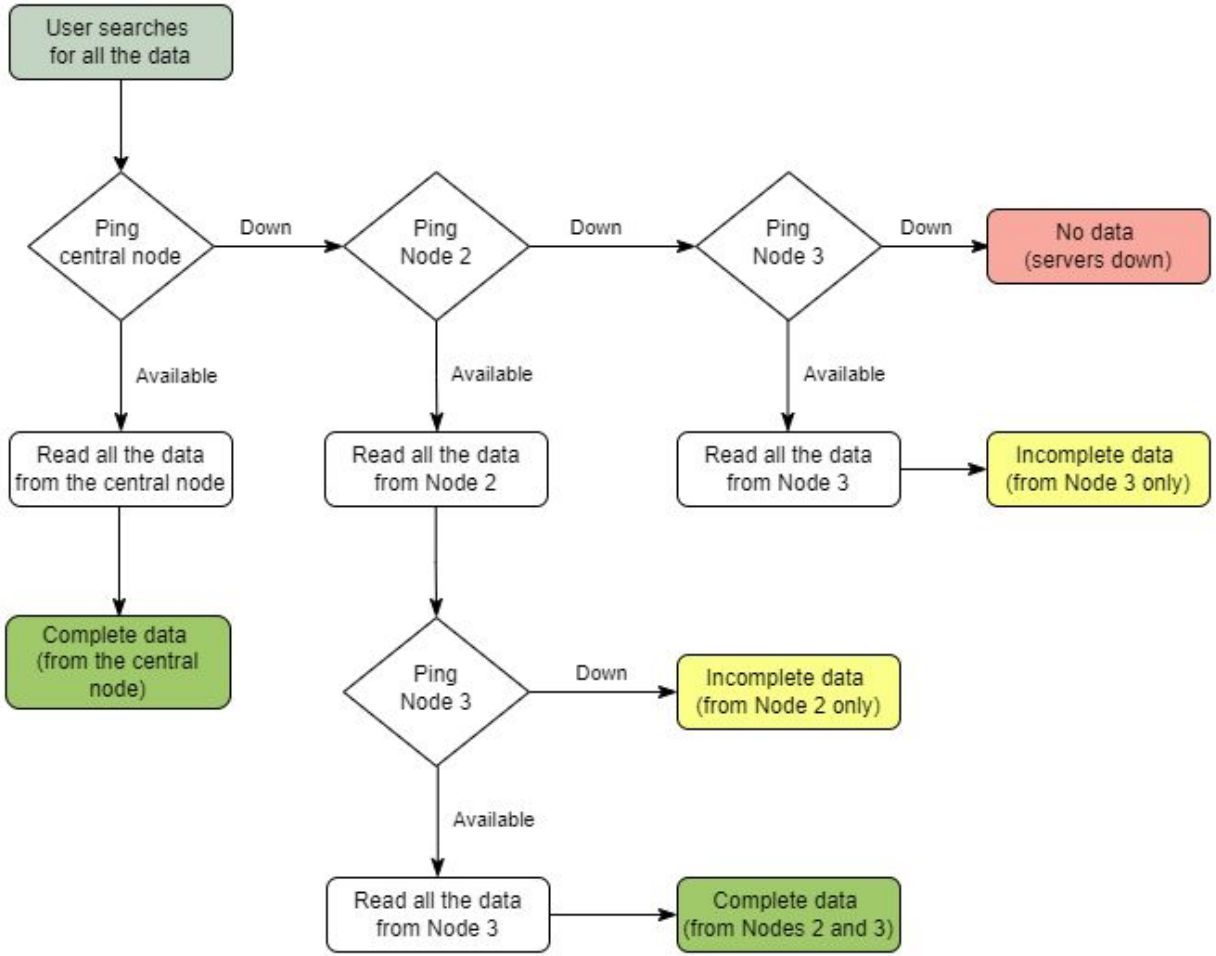


Prioritize fetching from central node to avoid handshaking and result set merging overhead



Prioritize fetching from central node to avoid handshaking and result set merging overhead

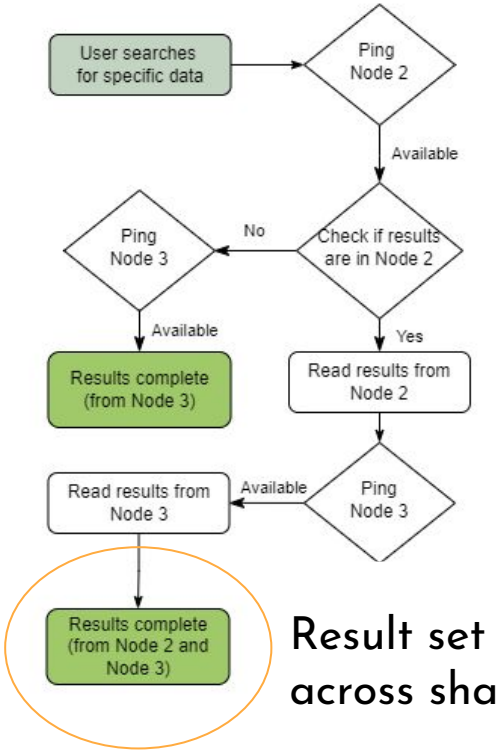
ALGORITHM (RETRIEVE ALL)



Web Application

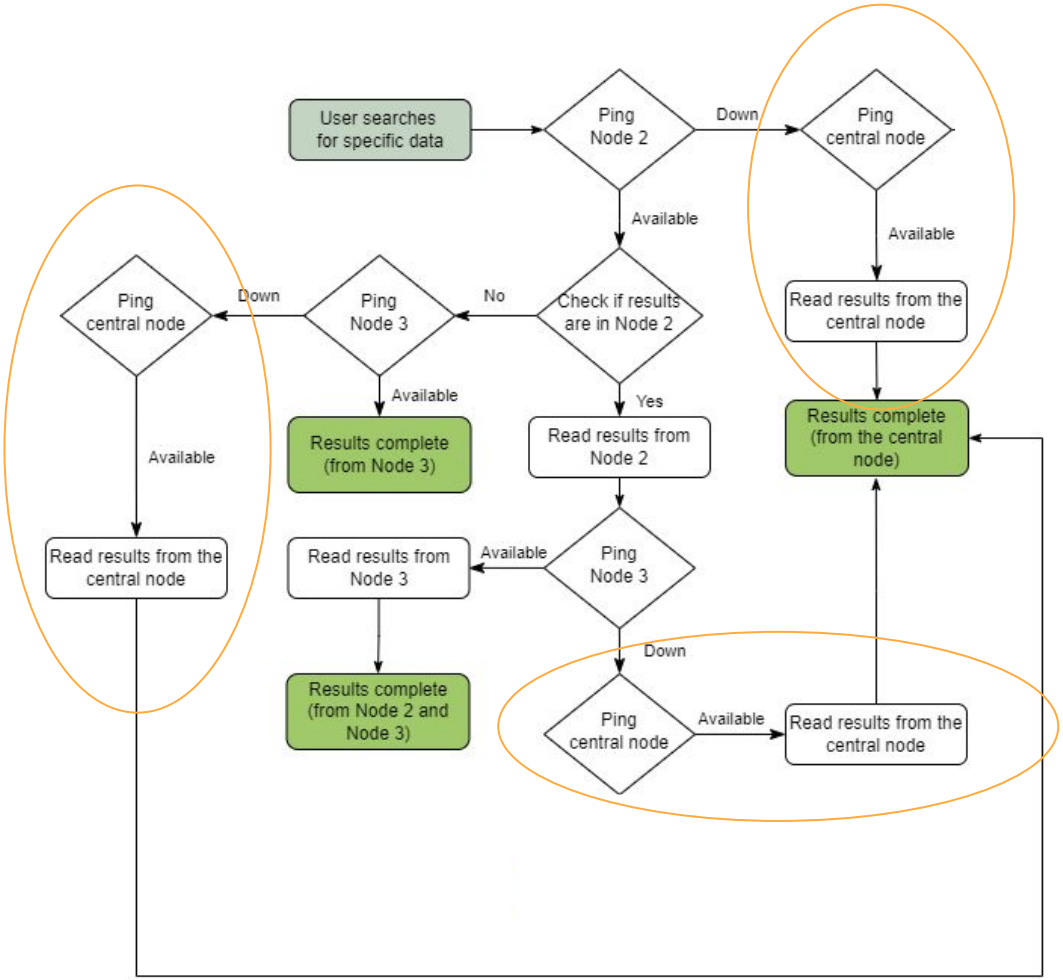
Retrieve All

Retrieve Based
on Search Criterion



Prioritize fetching from partial replicas to avoid overloading the central node

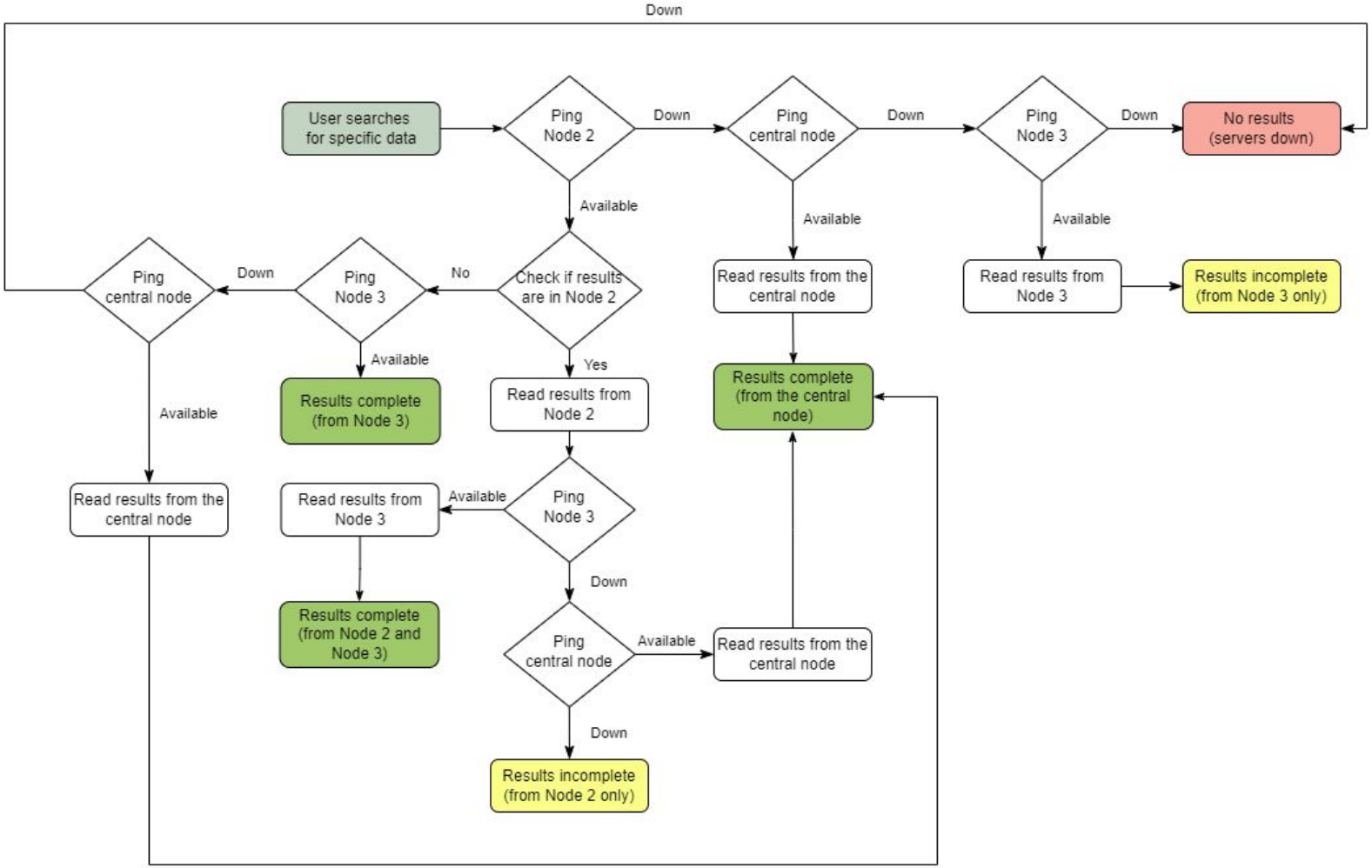
Result set may span across shards



If one of the partial replicas is down, fetch from central

ALGORITHM (RETRIEVE SPECIFIC)

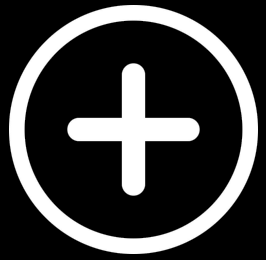
ALGORITHM (RETRIEVE SPECIFIC)



Web Application

Retrieve All

Retrieve Based
on Search Criterion



Insert

Update

Delete

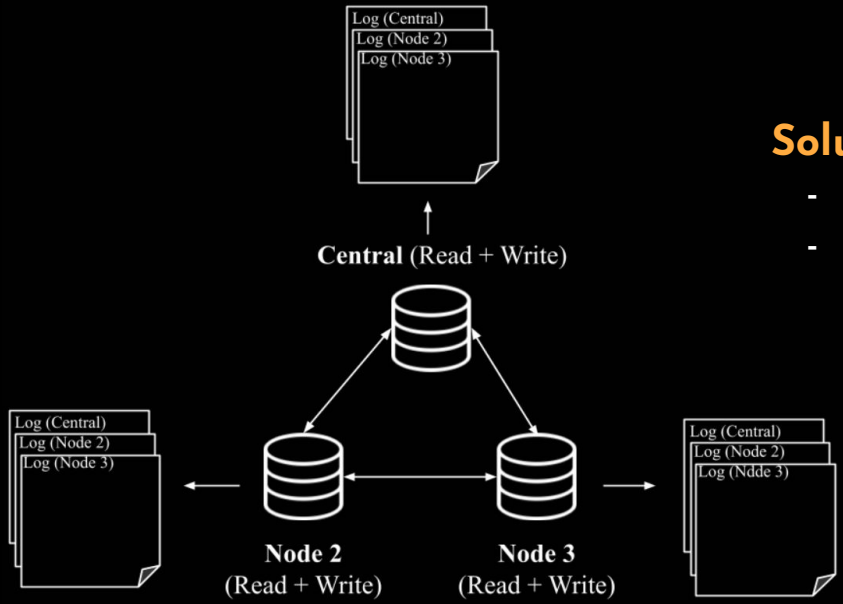
ALGORITHM

Log Table

Partly based on MySQL's GTID logging technique

Attribute	Description
`node_id`	Numerical identifier of the node (1 for the central node, 2 for Node 2, and 3 for Node 3)
`lock_status`	Binary value indicating whether the node is locked (0 for not locked, 1 for locked)
`next_trans_record`	Identifier of the next transaction to be recorded in the log
`next_trans_commit`	Identifier of the next transaction to be committed
`id_new_entry`	ID (primary key) of the most recently inserted entry
`statements`	SQL queries executed (or to be executed) in the node, prefixed by a zero-based auto-incrementing identifier and delimited by three vertical pipes

Log Table

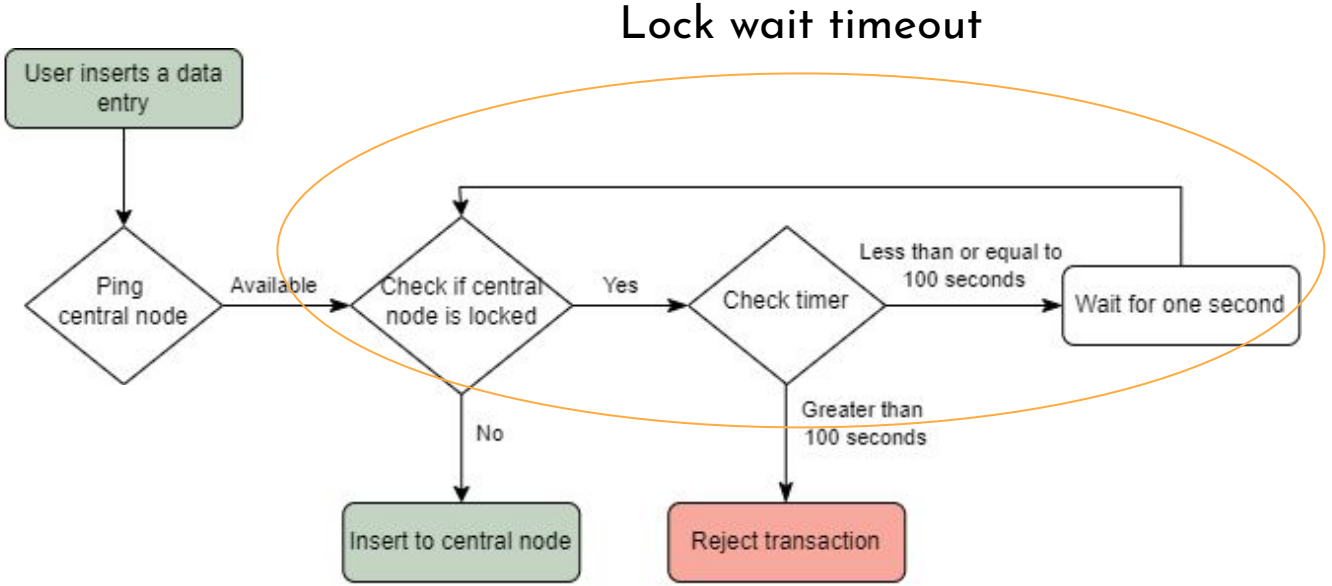


Scenario

- Central node fails and the user inserts a movie entry before 1980
- Committed in Node 2 but not in central node
- Central node recovers, but Node 2 is now down
- **Problem:** Central node cannot refer to Node 2's logs to catch up

Solution

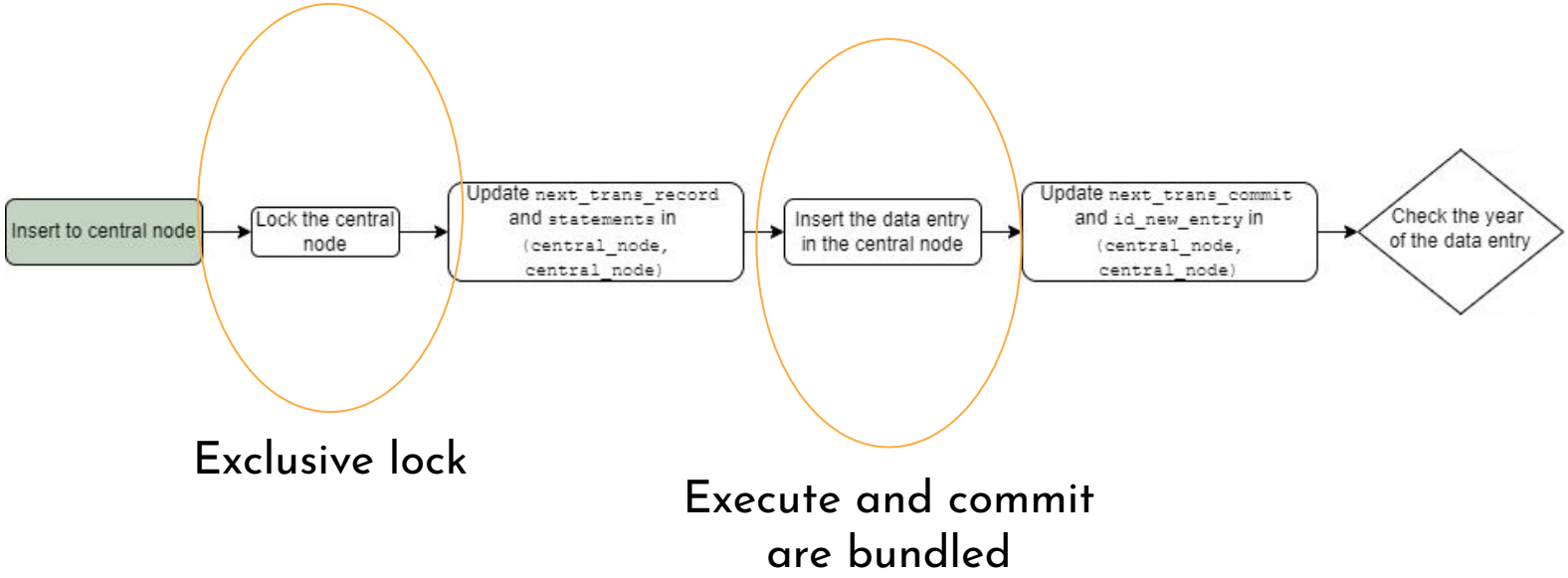
- Each node has a copy of all the nodes' logs
- Motivated by decentralized strategy of blockchain (Crosby et al., 2015)



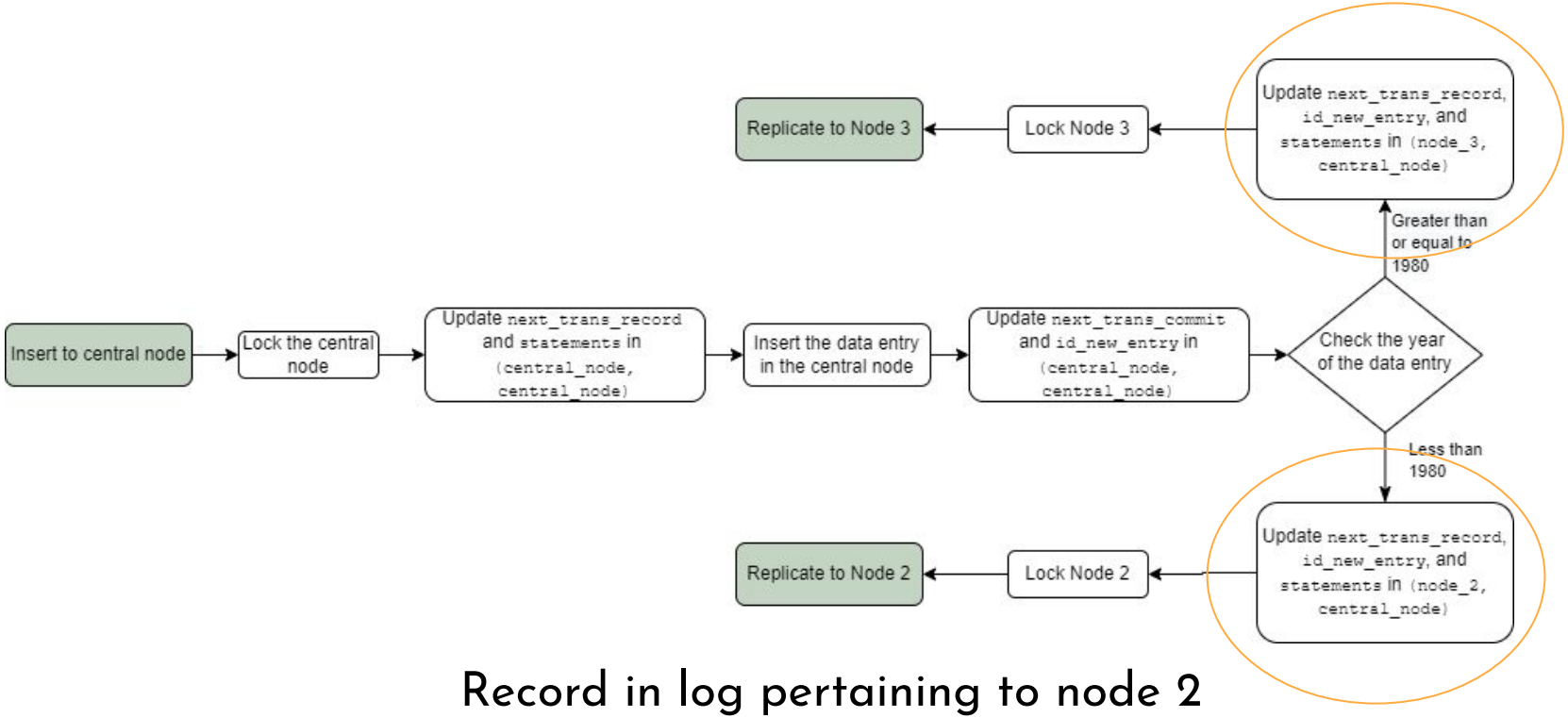
Prioritize inserting to the central node

ALGORITHM (INSERT)

For reference, MySQL follows an execute-log-commit pipeline

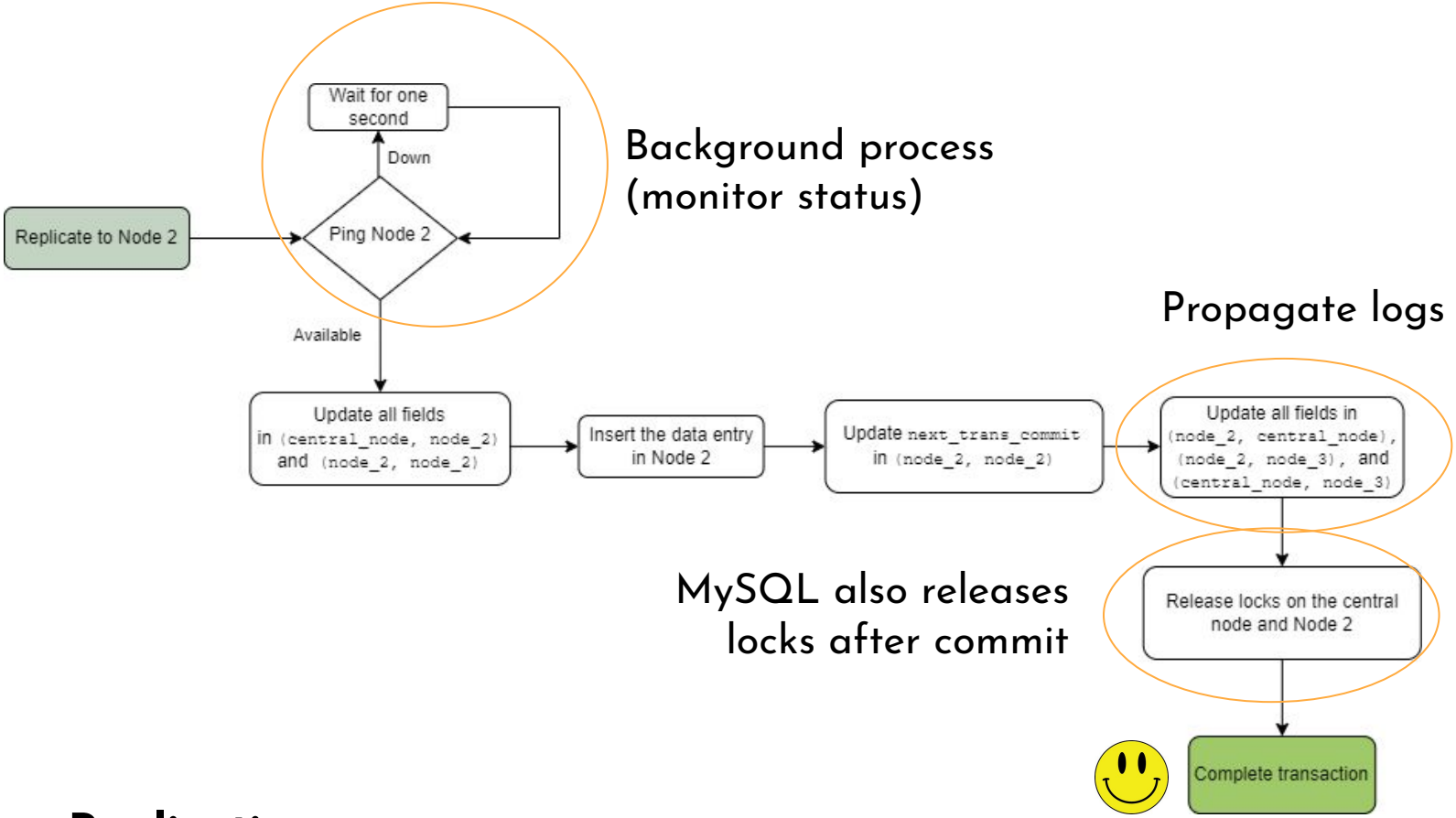


In our system, logging takes precedence



Record in log pertaining to node 2
(stored in the central node)

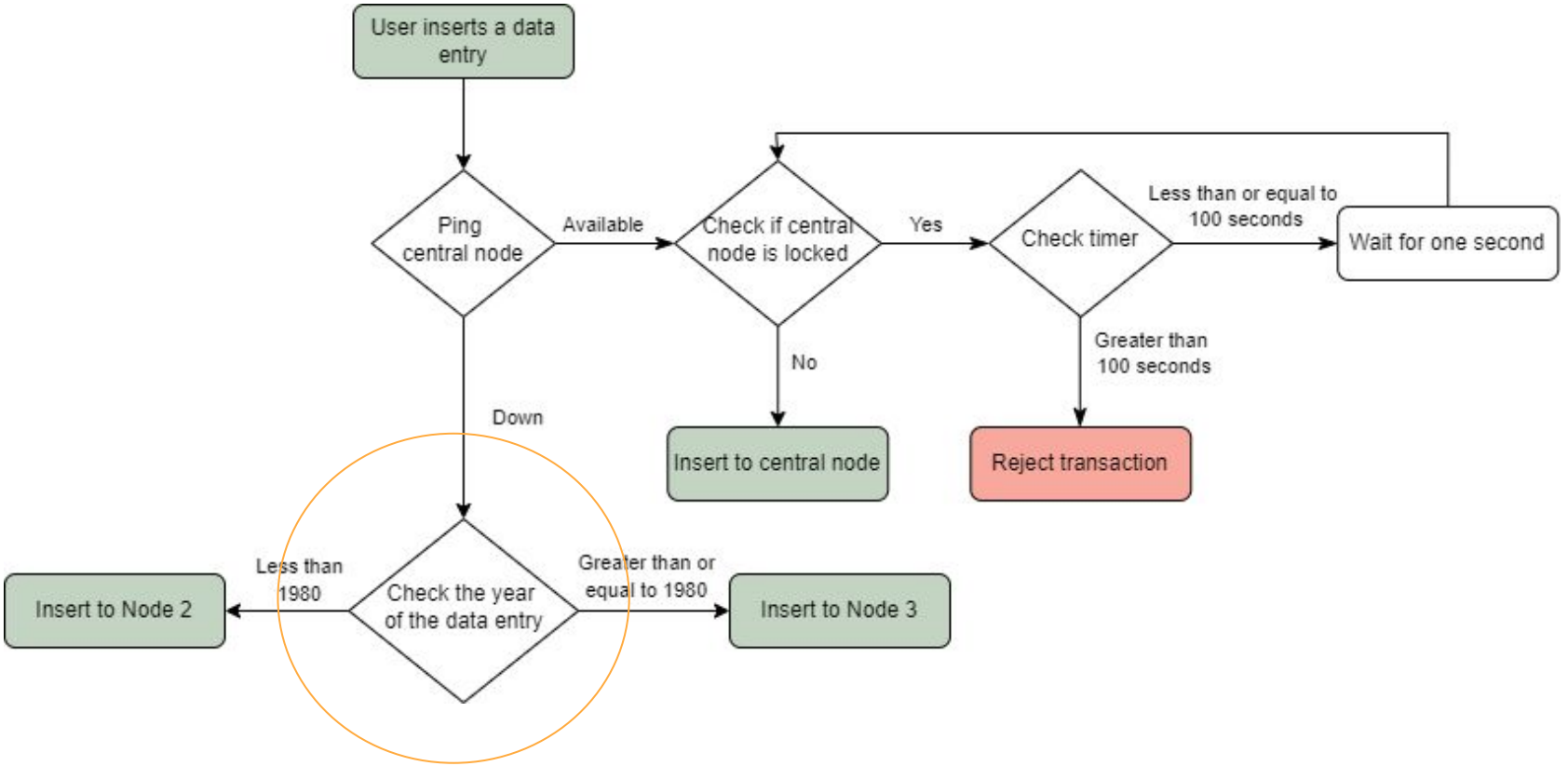
ALGORITHM (INSERT)



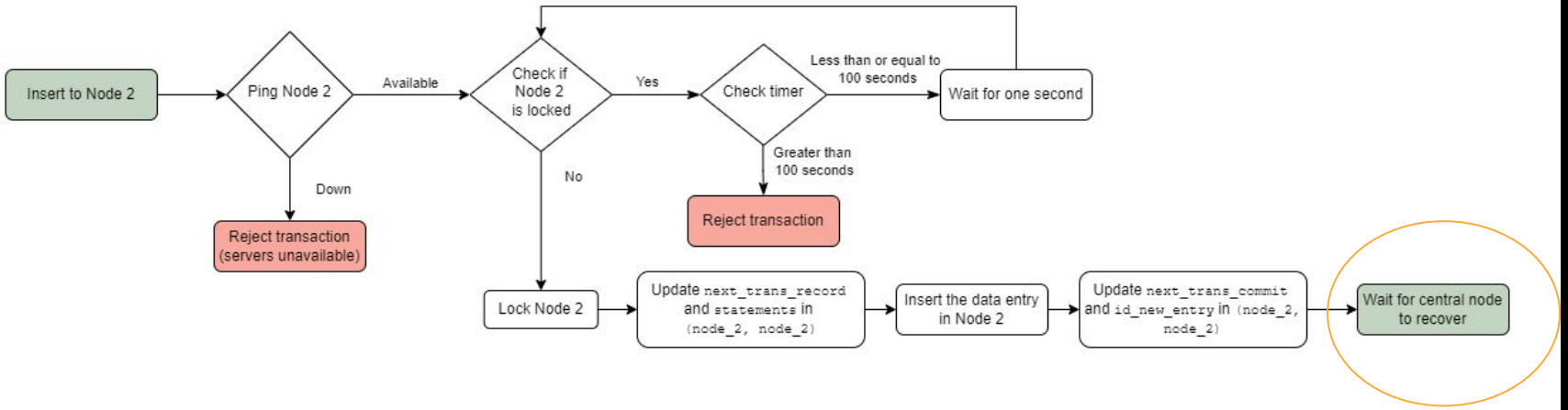
Replication

ALGORITHM (INSERT)

If central node is down, redirect write to replica



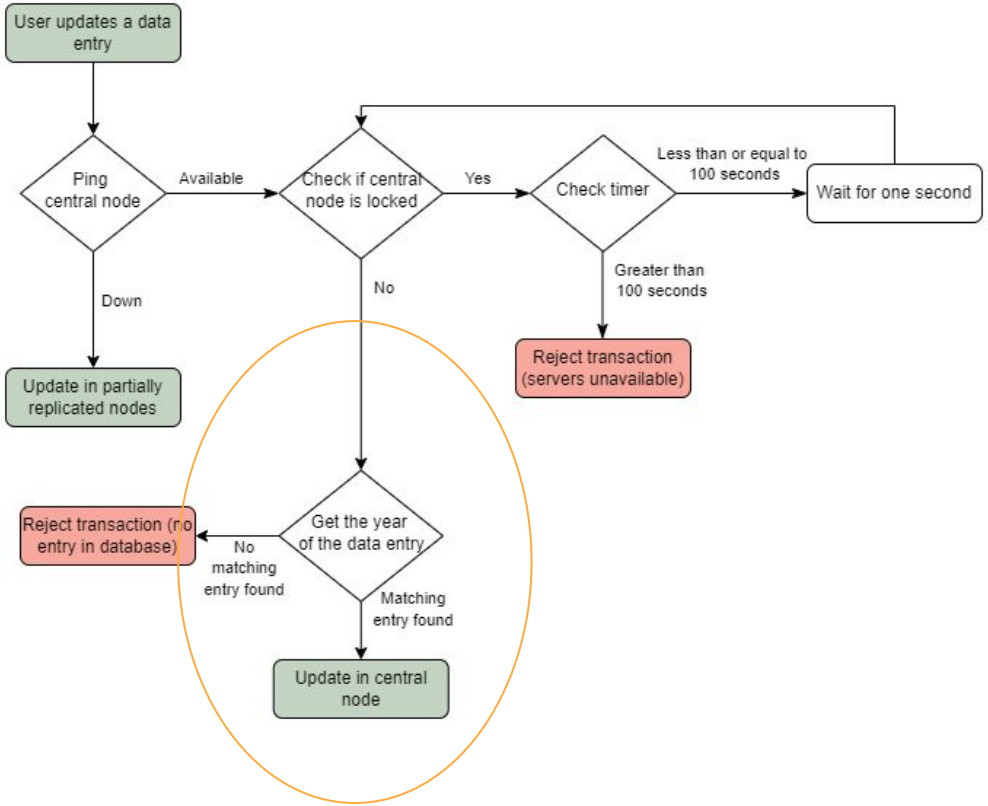
ALGORITHM (INSERT)

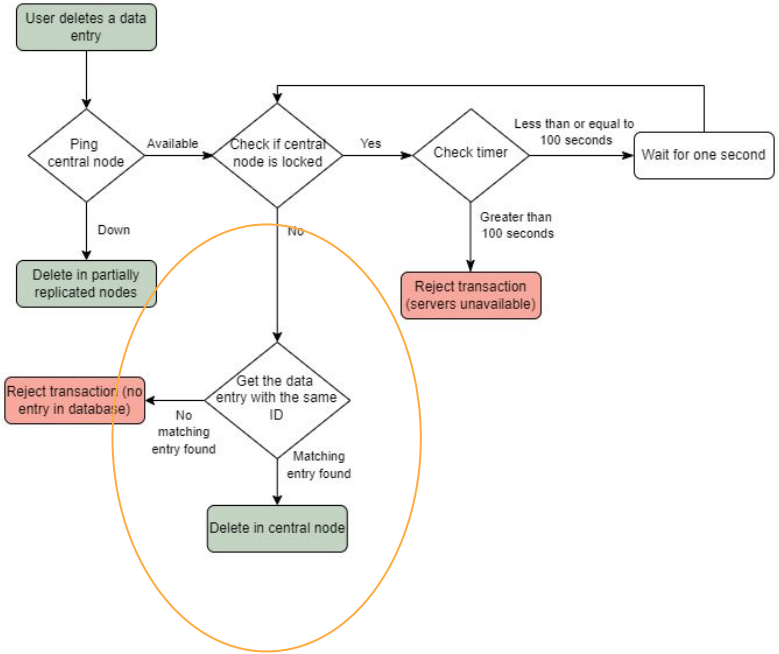


Background process

If central node is down, redirect write to replica

ALGORITHM (INSERT)





ALGORITHM (UPDATE AND DELETE)

Challenges & Considerations

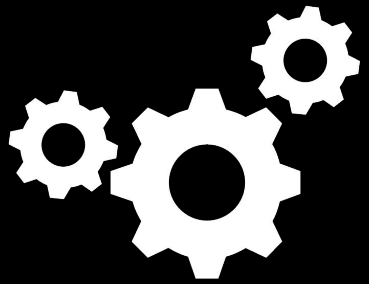
Callback Hell

```
pool.getConnection(func(err, c) {
  c.ping(func(err) {
    c.beginTransaction(func(err) {
      log(func(query, param, result) {
        c.query(func(query, param, result){
          conn.commit(func(err) {
            log(func(query, param, result) {
});});});});});});});});
```

String Parsing

Log files have to undergo string processing, especially since SQL queries have been parameterized to prevent SQL injection.

Challenges & Considerations



Background Process

```
function monitor() {
  if (c != undefined)
     c.ping(func(err) {
       t = setTimeout(function(monitor, 1000);
       if (!err)
         connected();
     });

  else
     setTimeout(function(monitor, 1000));
}
```

Global Concurrency

Case 1: All Transactions are Reading

User 1	User 2	User 3
START TRANSACTION; SELECT * FROM movies WHERE id = 174202;		
DO SLEEP(20); COMMIT;	START TRANSACTION; SELECT * FROM movies WHERE id = 174202; DO SLEEP(20); COMMIT;	START TRANSACTION; SELECT * FROM movies WHERE id = 174202; DO SLEEP(20); COMMIT;

CONCURRENCY

Global Concurrency

Case 1: All Transactions are Reading

Reads are **non locking**. They **do not request locks** as well.

Concurrent reads are **allowed**, and they **do not change** the database state.

Global Concurrency

Case 2: 1 Transaction is Writing, 2 Transactions are Reading

The application layer ensures that a transaction contains only a single `SELECT` statement.

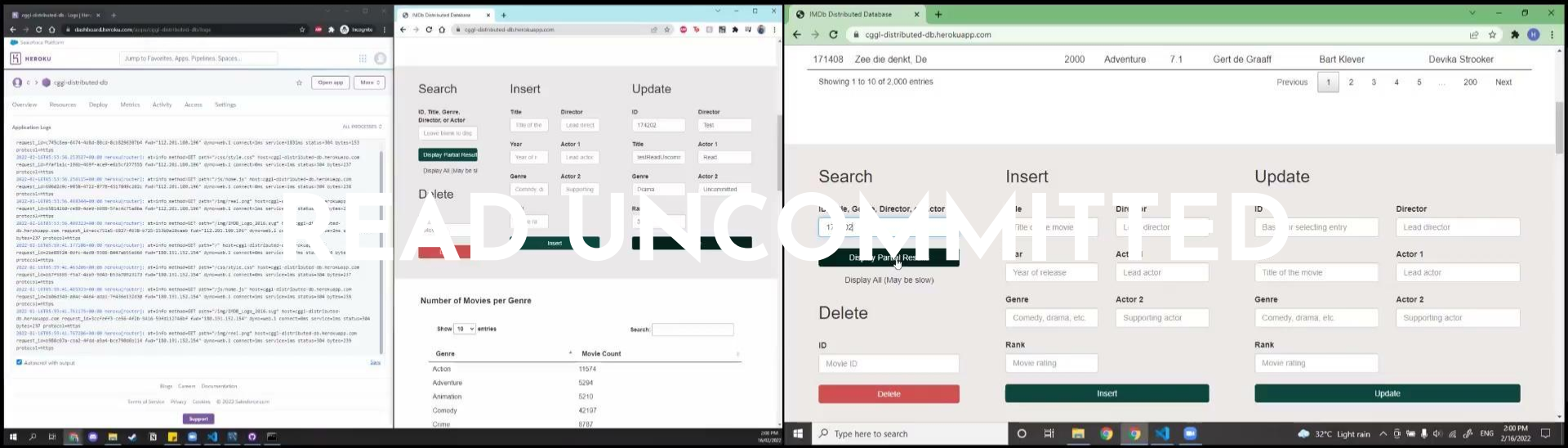
This prevents the possibility of non-repeatable and phantom reads.

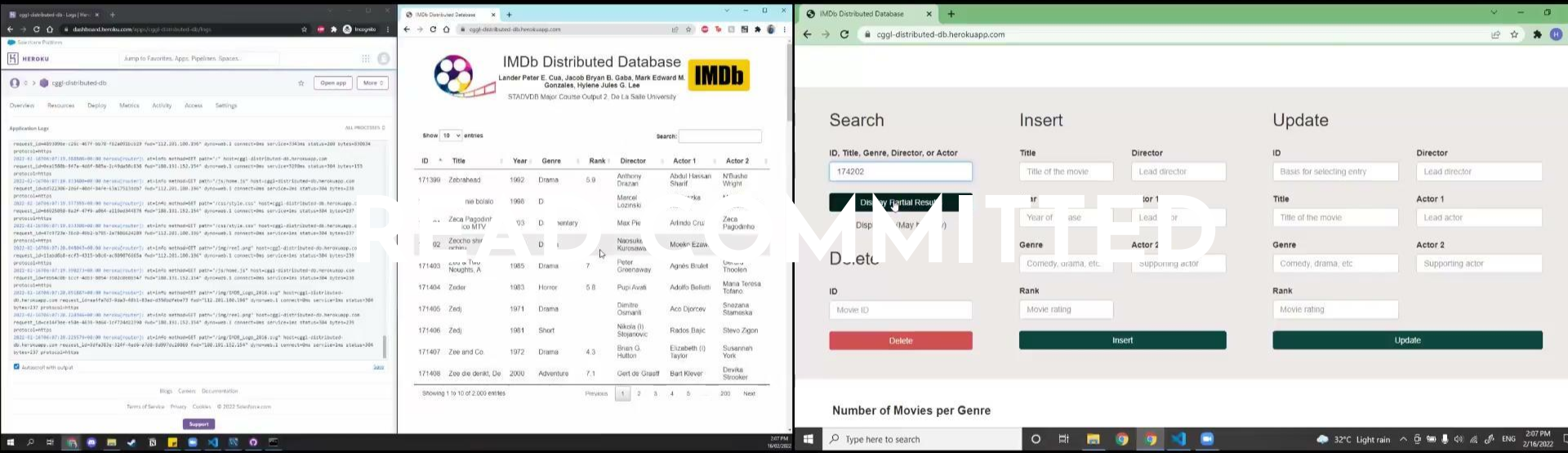
Global Concurrency

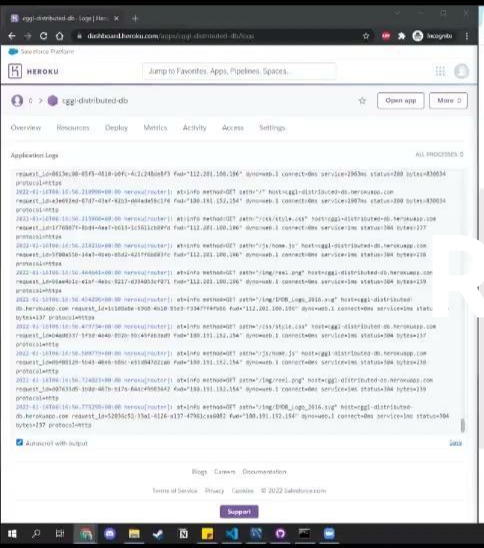
Case 2: 1 Transaction is Writing, 2 Transactions are Reading

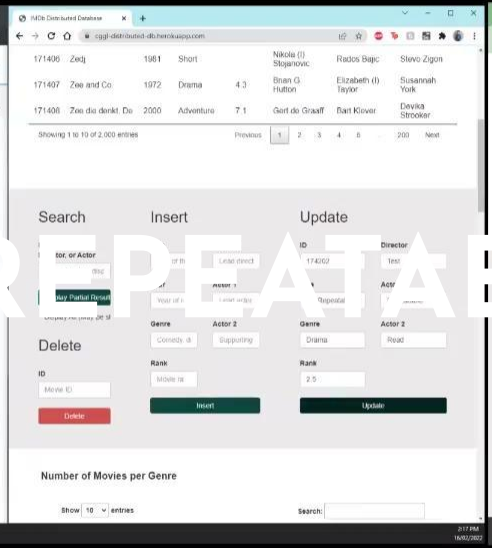
User 1	User 2	User 3
START TRANSACTION; UPDATE movies SET title = 'testReadCommitted', genre = 'Drama', `rank` = 2.5, director = 'Test', actor1 = 'Read', actor2 = 'Committed' WHERE id = 174202;		
DO SLEEP(20);	START TRANSACTION; SELECT * FROM movies WHERE id = 174202; COMMIT;	START TRANSACTION; SELECT * FROM movies WHERE id = 174202; COMMIT;
COMMIT;		

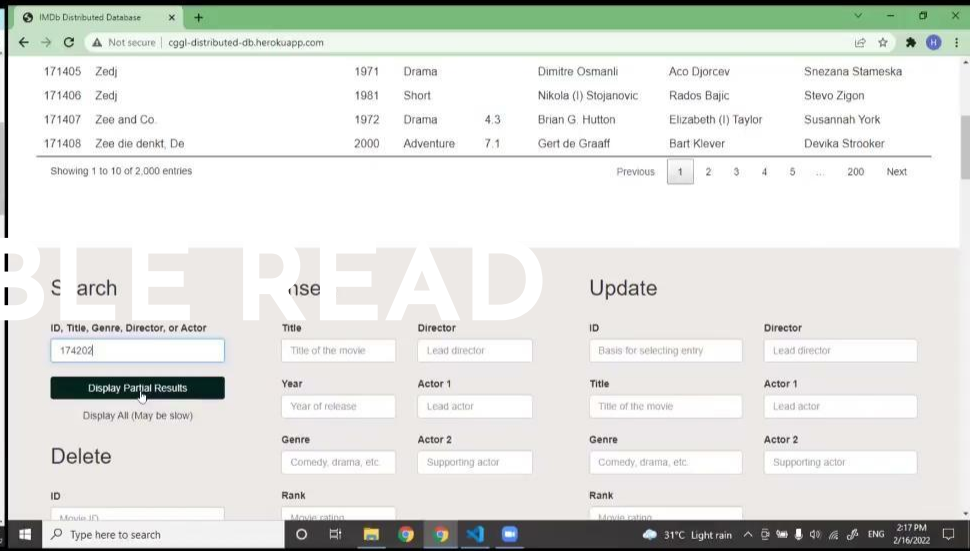
CONCURRENCY

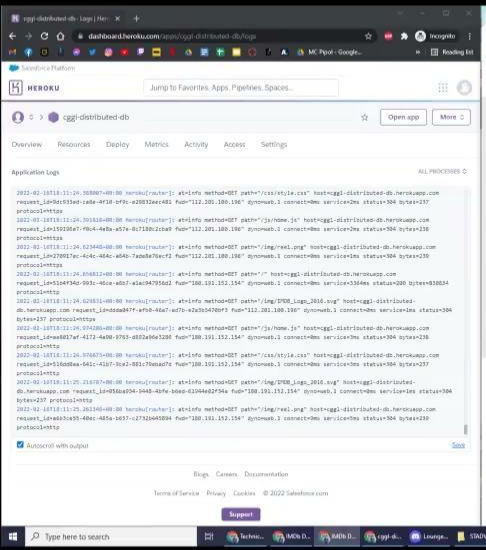


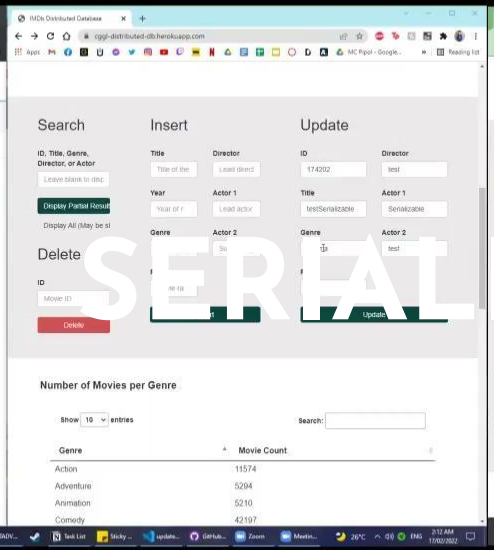

IMDb Distributed Database
Lander Peter E. Cua, Jacob Bryan B. Gaba, Mark Edward M. Gonzales, Hylene Jules G. Lee
STADVDB Major Course Output 2, De La Salle University
Search
ID, Title, Genre, Director, or Actor
174202
Insert
Title
Director
Update
ID
Director
Delete
Number of Movies per Genre

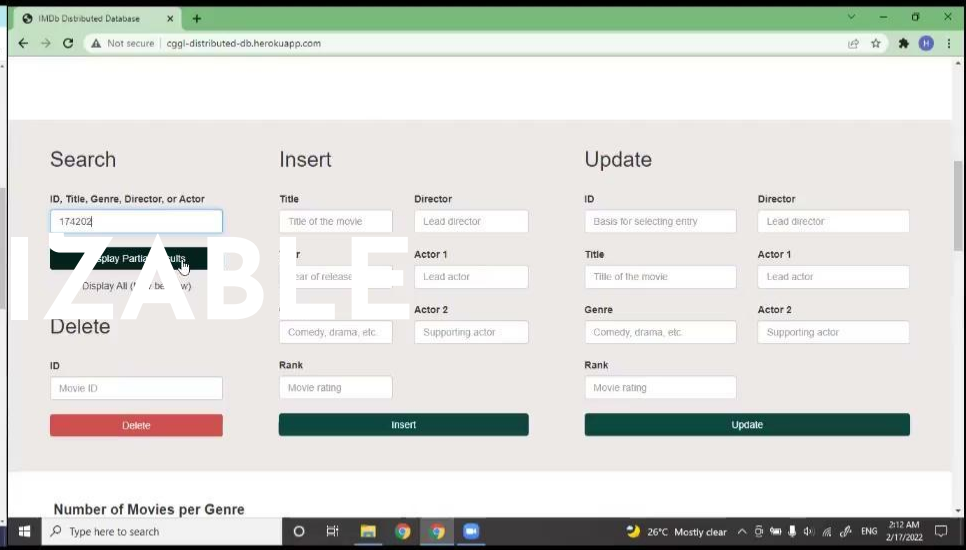












Global Concurrency

Case 2: 1 Transaction is Writing, 2 Transactions are Reading

Concurrency is **supported**.

Dirty reads are blocked by using the **READ COMMITTED** isolation level.

The other read anomalies are prevented by the **design of the application layer**.

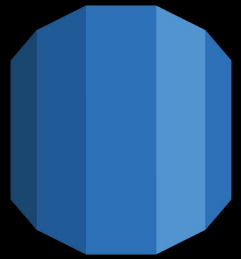
READ COMMITTED is also the default in major DBMS, such as Microsoft SQL Server (Microsoft, 2021) and PostgreSQL (PostgreSQL, n.d.).

Global Concurrency

Case 2: 1 Transaction is Writing, 2 Transactions are Reading

Challenge:
Changing the isolation level globally

Create a parameter group in AWS RDS



Disable `autocommit` attribute

Global Concurrency

Case 3: All Transactions are Writing

User 1	User 2	User 3
START TRANSACTION; INSERT INTO movies (title, year, genre, `rank`, director, actor1, actor2) VALUES ('First entry', 1976, 'Comedy', 6.6, 'Cua', 'Gaba', 'Gonzales');		
DO SLEEP(20); COMMIT;	START TRANSACTION; INSERT INTO movies (title, year, genre, `rank`, director, actor1, actor2) VALUES ('Second entry', 1980, 'Drama', 7.7, 'Gaba', 'Gonzales', 'Lee'); DO SLEEP(20); COMMIT;	START TRANSACTION; INSERT INTO movies (title, year, genre, `rank`, director, actor1, actor2) VALUES ('Third entry', 1985, 'Action', 8.8, 'Gonzales', 'Lee', 'Cua'); DO SLEEP(20); COMMIT;

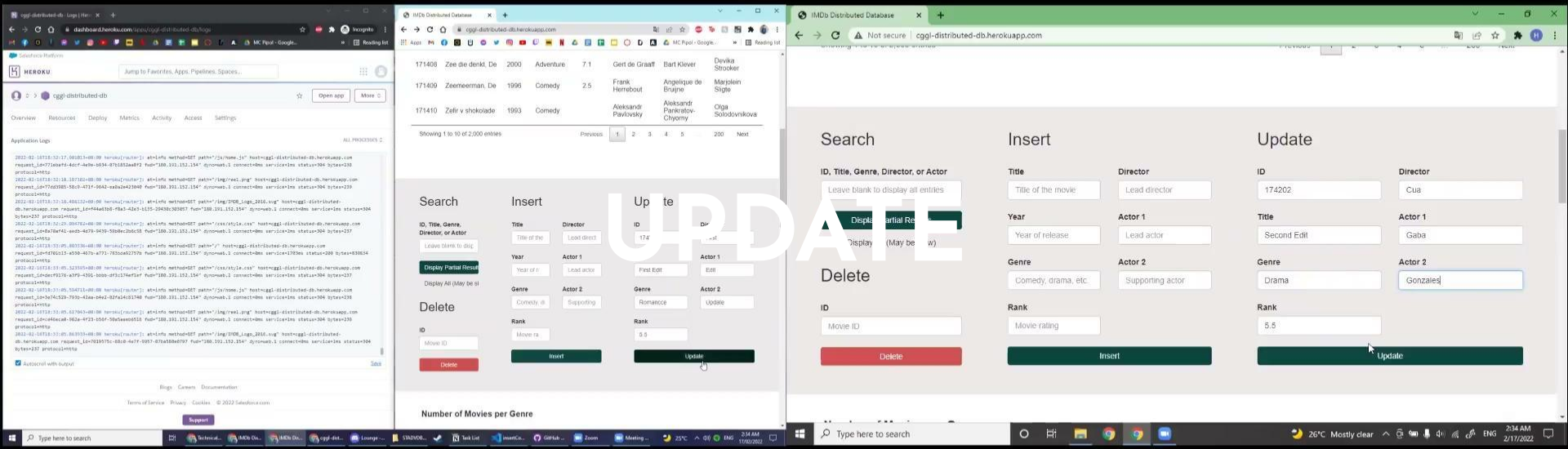
CONCURRENCY

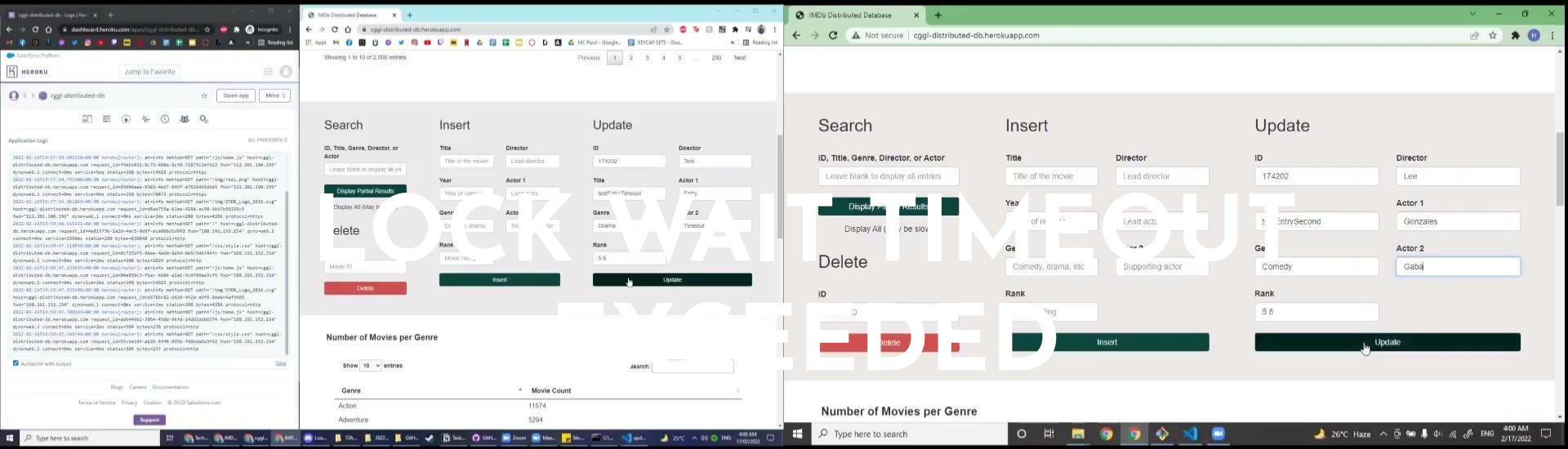
Global Concurrency

Case 3: All Transactions are Writing

User 1	User 2	User 3
START TRANSACTION; UPDATE movies SET title = 'First Edit' genre = 'Comedy' `rank` = 6.6 director = 'Gaba' actor1 = 'Gonzales' actor2 = 'Lee'; WHERE id = 174202;		
DO SLEEP(20); COMMIT;	START TRANSACTION; UPDATE movies SET title = 'Second Edit' genre = 'Comedy' `rank` = 6.6 director = 'Gaba' actor1 = 'Gonzales' actor2 = 'Lee'; WHERE id = 174202; DO SLEEP(20); COMMIT;	START TRANSACTION; UPDATE movies SET title = 'Third Edit' genre = 'Comedy' `rank` = 6.6 director = 'Gaba' actor1 = 'Gonzales' actor2 = 'Lee'; WHERE id = 174202; DO SLEEP(20); COMMIT;

CONCURRENCY


IMDb Distributed Database
cggl-distributed-db.herokuapp.com
Not secure
Search
ID, Title, Genre, Director, or Actor
Leave blank to display all entries
Display Partial Result
Display All (May be slow)
Delete
ID
Movie ID
Delete
Insert
Title
Title of the movie
Director
Lead director
Year
Year of release
Actor 1
Lead actor
Genre
Comedy, drama, etc.
Actor 2
Supporting actor
Rank
Movie rating
Insert
Update
ID
174202
Director
Cua
Title
Second Edit
Actor 1
Gaba
Genre
Drama
Actor 2
Gonzales
Rank
5.5
Update
UPDATE
Number of Movies per Genre
Type here to search
26°C Mostly clear
ENG
2:34 AM
2/17/2022


Search
Insert
Update
ID, Title, Genre, Director, or Actor
Leave blank to display all entries
Display Partial Results
Display All (May be slow)
Delete
Movie ID
Title
Title of the movie
Director
Lead director
Year
Actor 1
Genre
Actor 2
Rank
5.6
Insert
Update
ID
174202
Test
Lee
Drama
Gonzales
Comedy, drama, etc.
Supporting actor
Comedy
Gaba
Number of Movies per Genre
Show 10 entries
Search:
Genre
Movie Count
Action
11574
Adventure
5294
LOCK WAIT TIMEOUT
EXCEEDED

Global Concurrency

Case 3: All Transactions are Writing

Write transactions are handled via **locking**.

A lock wait timeout mechanism was implemented to prevent **lock contention issues**.

Global Recovery

Recovery occurs by establishing a new connection that is directed towards the crashed node.

For the experiments, three down-time windows were tested: **5, 60, and 300 seconds**.

Global Recovery

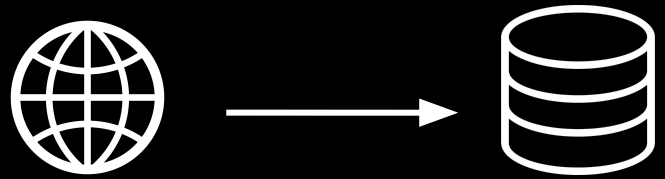
Case 1: Failure in replicating from Node 2 or Node 3 to Central Node

SIMULATING CASE 1

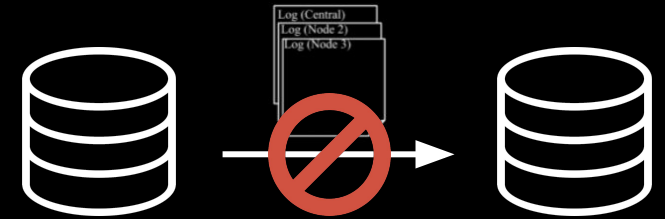
- Force Central Node crash before transaction to route algorithm to Node 2 / 3
- Central Node is crashed again during replication from Node 2 / 3

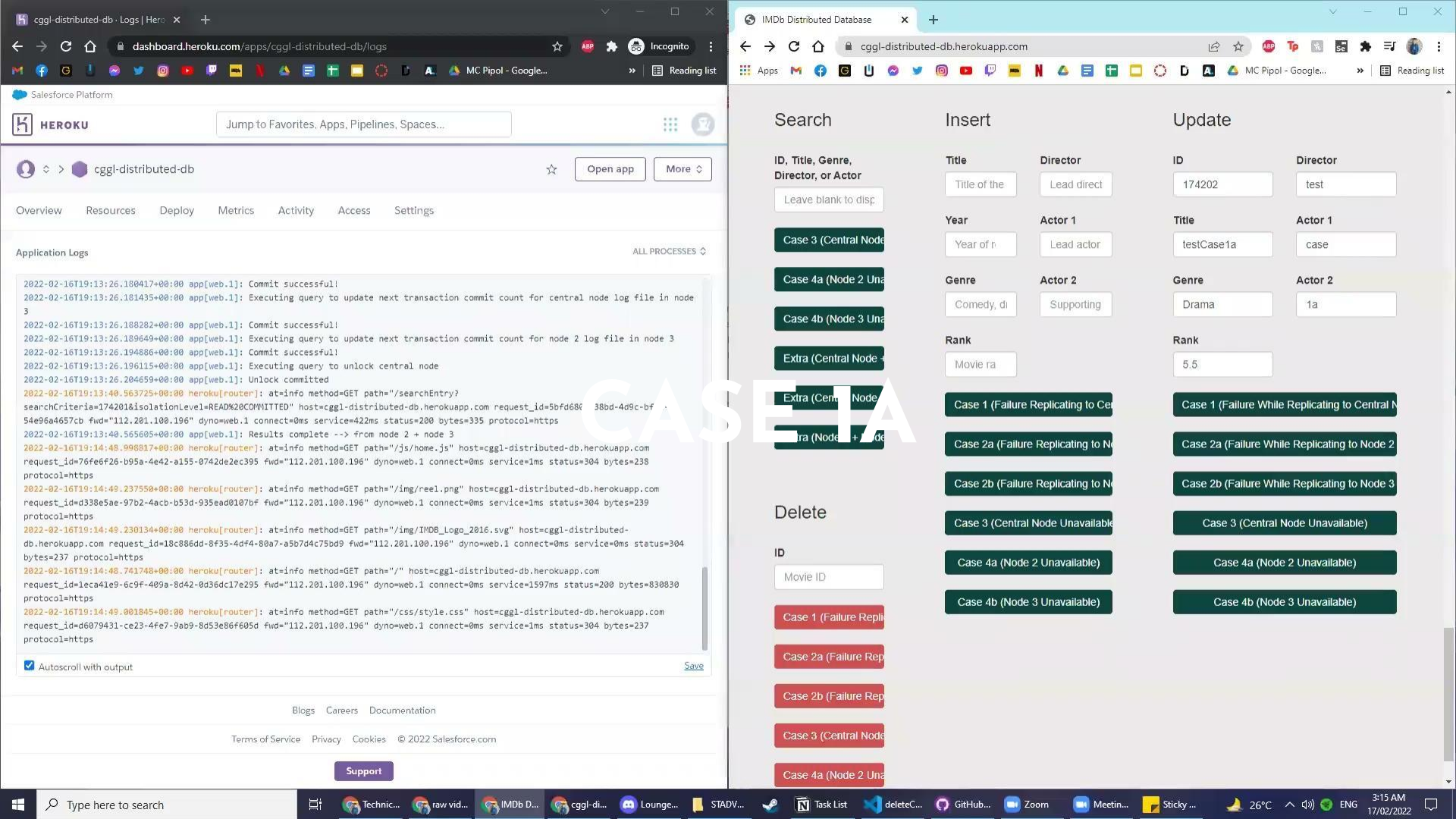
1. Initial Transaction

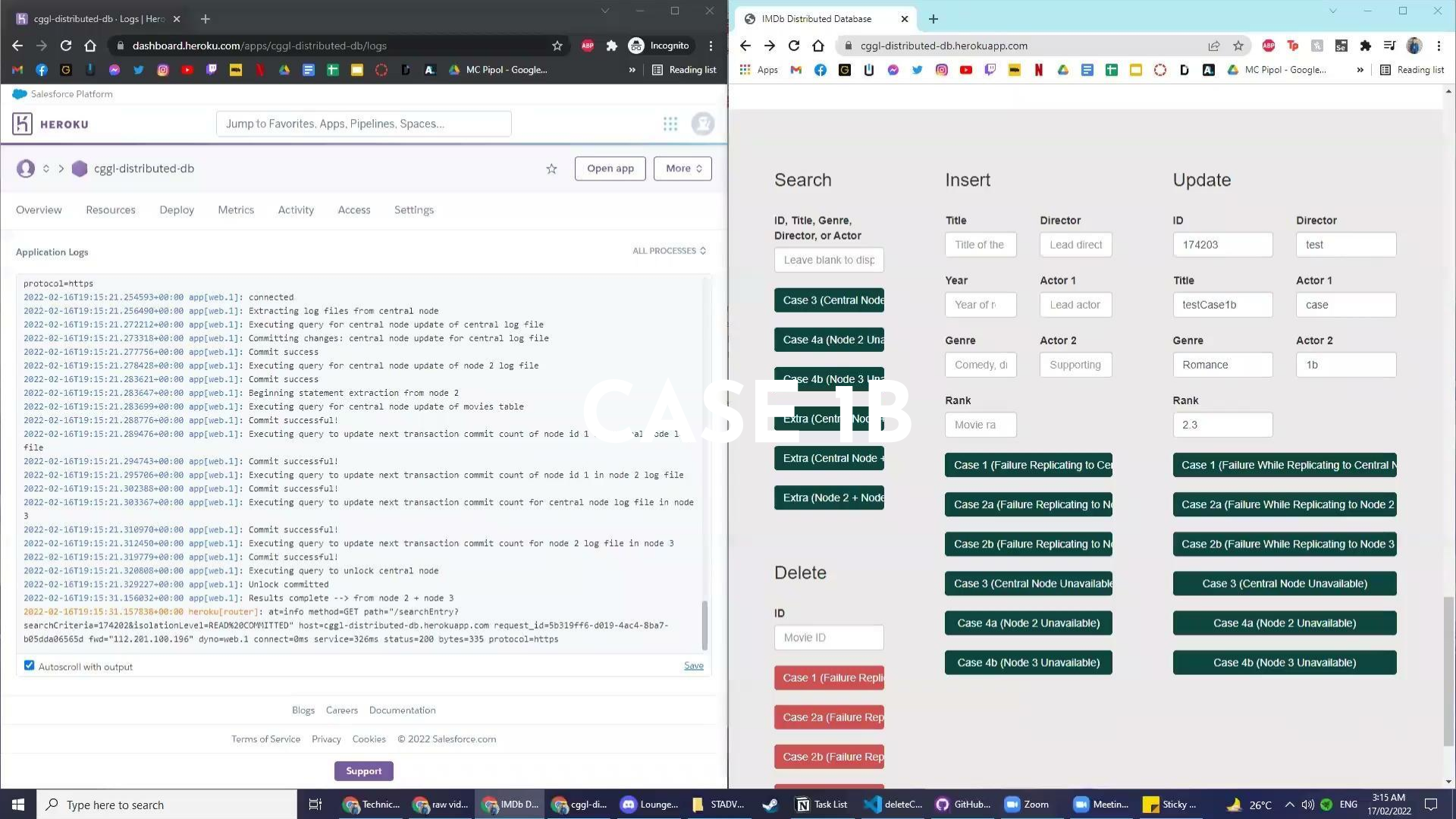
2. Redirection



3. Replication




cggl-distributed-db · Logs | Hero
dashboard.heroku.com/apps/cggl-distributed-db/logs
Incognito
MC Pipol - Google...
Reading list
Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
cggl-distributed-db
Open app
More
Overview
Resources
Deploy
Metrics
Activity
Access
Settings
Application Logs
ALL PROCESSES
2022-02-16T19:13:26.180417+00:00 app[web.1]: Commit successful!
2022-02-16T19:13:26.181435+00:00 app[web.1]: Executing query to update next transaction commit count for central node log file in node 3
2022-02-16T19:13:26.188282+00:00 app[web.1]: Commit successful!
2022-02-16T19:13:26.189649+00:00 app[web.1]: Executing query to update next transaction commit count for node 2 log file in node 3
2022-02-16T19:13:26.194886+00:00 app[web.1]: Commit successful!
2022-02-16T19:13:26.196115+00:00 app[web.1]: Executing query to unlock central node
2022-02-16T19:13:26.204659+00:00 app[web.1]: Unlock committed
2022-02-16T19:13:40.563725+00:00 heroku[router]: at=info method=GET path="/searchEntry?searchCriteria=174201&isolationLevel=READ%20COMMITTED" host=cggl-distributed-db.herokuapp.com request_id=5bfd68
38bd-4d9c-bf
54e96a4657cb fwd="112.201.100.196" dyno=web.1 connect=0ms service=422ms status=200 bytes=335 protocol=https
2022-02-16T19:13:40.565605+00:00 app[web.1]: Results complete --> from node 2 + node 3
2022-02-16T19:14:48.998817+00:00 heroku[router]: at=info method=GET path="/js/home.js" host=cggl-distributed-db.herokuapp.com request_id=76fe6f26-b95a-4e42-a155-0742de2ec395 fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=238 protocol=https
2022-02-16T19:14:49.237550+00:00 heroku[router]: at=info method=GET path="/img/reel.png" host=cggl-distributed-db.herokuapp.com request_id=d338e5ae-97b2-4acb-b53d-935ead0107bf fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=239 protocol=https
2022-02-16T19:14:49.230134+00:00 heroku[router]: at=info method=GET path="/img/IMDB_Logo_2016.svg" host=cggl-distributed-db.herokuapp.com request_id=18c886dd-8f35-4df4-80a7-a5b7d4c75bd9 fwd="112.201.100.196" dyno=web.1 connect=0ms service=0ms status=304 bytes=237 protocol=https
2022-02-16T19:14:48.741748+00:00 heroku[router]: at=info method=GET path="/" host=cggl-distributed-db.herokuapp.com request_id=1eca41e9-6c9f-409a-8d42-0d36dc17e295 fwd="112.201.100.196" dyno=web.1 connect=0ms service=1597ms status=200 bytes=830830 protocol=https
2022-02-16T19:14:49.001845+00:00 heroku[router]: at=info method=GET path="/css/style.css" host=cggl-distributed-db.herokuapp.com request_id=d6079431-ce23-4fe7-9ab9-8d53e86f605d fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
Autoscroll with output
Save
Blogs
Careers
Documentation
Terms of Service
Privacy
Cookies
© 2022 Salesforce.com
Support
IMDb Distributed Database
cggl-distributed-db.herokuapp.com
Apps
MC Pipol - Google...
Reading list
Search
ID, Title, Genre, Director, or Actor
Leave blank to disp
Case 3 (Central Node
Case 4a (Node 2 Una
Case 4b (Node 3 Una
Extra (Central Node +
Extra (Cen Node
Extra (Node + Node
Delete
ID
Movie ID
Case 1 (Failure Repli
Case 2a (Failure Rep
Case 2b (Failure Rep
Case 3 (Central Node
Case 4a (Node 2 Una
Insert
Title
Title of the
Director
Lead direct
Year
Year of r
Actor 1
Lead actor
Genre
Comedy, di
Actor 2
Supporting
Rank
Movie ra
Case 1 (Failure Replicating to Ce
Case 2a (Failure Replicating to N
Case 2b (Failure Replicating to N
Case 3 (Central Node Unavailable
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
Update
ID
174202
Director
test
Title
testCase1a
Actor 1
case
Genre
Drama
Actor 2
1a
Rank
5.5
Case 1 (Failure While Replicating to Central N
Case 2a (Failure While Replicating to Node 2
Case 2b (Failure While Replicating to Node 3
Case 3 (Central Node Unavailable)
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
CASE 1A
Type here to search
Technic...
raw vid...
IMDb D...
cggl-di...
Lounge...
STADV...
Task List
deleteC...
GitHub...
Zoom
Meetin...
Sticky ...
26°C
ENG
3:15 AM
17/02/2022


cggl-distributed-db · Logs | Hero
dashboard.heroku.com/apps/cggl-distributed-db/logs
Incognito
MC Pipol - Google...
Reading list
Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
cggl-distributed-db
Open app
More
Overview
Resources
Deploy
Metrics
Activity
Access
Settings
Application Logs
ALL PROCESSES
protocol=https
2022-02-16T19:15:21.254593+00:00 app[web.1]: connected
2022-02-16T19:15:21.256490+00:00 app[web.1]: Extracting log files from central node
2022-02-16T19:15:21.272212+00:00 app[web.1]: Executing query for central node update of central log file
2022-02-16T19:15:21.273318+00:00 app[web.1]: Committing changes: central node update for central log file
2022-02-16T19:15:21.277756+00:00 app[web.1]: Commit success
2022-02-16T19:15:21.278428+00:00 app[web.1]: Executing query for central node update of node 2 log file
2022-02-16T19:15:21.283621+00:00 app[web.1]: Commit success
2022-02-16T19:15:21.283647+00:00 app[web.1]: Beginning statement extraction from node 2
2022-02-16T19:15:21.283699+00:00 app[web.1]: Executing query for central node update of movies table
2022-02-16T19:15:21.288776+00:00 app[web.1]: Commit successful!
2022-02-16T19:15:21.289476+00:00 app[web.1]: Executing query to update next transaction commit count of node id 1
file
2022-02-16T19:15:21.294743+00:00 app[web.1]: Commit successful!
2022-02-16T19:15:21.295706+00:00 app[web.1]: Executing query to update next transaction commit count of node id 1 in node 2 log file
2022-02-16T19:15:21.302388+00:00 app[web.1]: Commit successful!
2022-02-16T19:15:21.303367+00:00 app[web.1]: Executing query to update next transaction commit count for central node log file in node 3
2022-02-16T19:15:21.310970+00:00 app[web.1]: Commit successful!
2022-02-16T19:15:21.312450+00:00 app[web.1]: Executing query to update next transaction commit count for node 2 log file in node 3
2022-02-16T19:15:21.319779+00:00 app[web.1]: Commit successful!
2022-02-16T19:15:21.320808+00:00 app[web.1]: Executing query to unlock central node
2022-02-16T19:15:21.329227+00:00 app[web.1]: Unlock committed
2022-02-16T19:15:31.156032+00:00 app[web.1]: Results complete --> from node 2 + node 3
2022-02-16T19:15:31.157838+00:00 heroku[router]: at=info method=GET path="/searchEntry?searchCriteria=174202&isolationLevel=READ%20COMMITTED" host=cggl-distributed-db.herokuapp.com request_id=5b319ff6-d019-4ac4-8ba7-b05dda06565d fwd="112.201.100.196" dyno=web.1 connect=0ms service=326ms status=200 bytes=335 protocol=https
Autoscroll with output
Save
Blogs
Careers
Documentation
Terms of Service
Privacy
Cookies
© 2022 Salesforce.com
Support
IMDb Distributed Database
cggl-distributed-db.herokuapp.com
Apps
MC Pipol - Google...
Reading list
CASE 1B
Search
ID, Title, Genre, Director, or Actor
Leave blank to disp
Case 3 (Central Node
Case 4a (Node 2 Una
Case 4b (Node 3 Una
Extra (Centr Nod
Extra (Central Node +
Extra (Node 2 + Node
Delete
ID
Movie ID
Case 1 (Failure Repli
Case 2a (Failure Rep
Case 2b (Failure Rep
Insert
Title
Title of the
Director
Lead direct
Year
Year of r
Actor 1
Lead actor
Genre
Comedy, di
Actor 2
Supporting
Rank
Movie ra
Case 1 (Failure Replicating to Ce
Case 2a (Failure Replicating to N
Case 2b (Failure Replicating to N
Case 3 (Central Node Unavailable
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
Update
ID
174203
Director
test
Title
testCase1b
Actor 1
case
Genre
Romance
Actor 2
1b
Rank
2.3
Case 1 (Failure While Replicating to Central N
Case 2a (Failure While Replicating to Node 2
Case 2b (Failure While Replicating to Node 3
Case 3 (Central Node Unavailable)
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
Type here to search
Technic...
raw vid...
IMDb D...
cggl-di...
Lounge...
STADV...
Task List
deleteC...
GitHub...
Zoom
Meetin...
Sticky ...
26°C
ENG
3:15 AM
17/02/2022

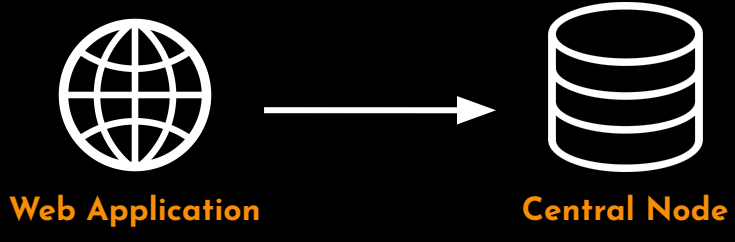
Global Recovery

Case 2: Failure in replicating from Central Node to Node 2 or Node 3

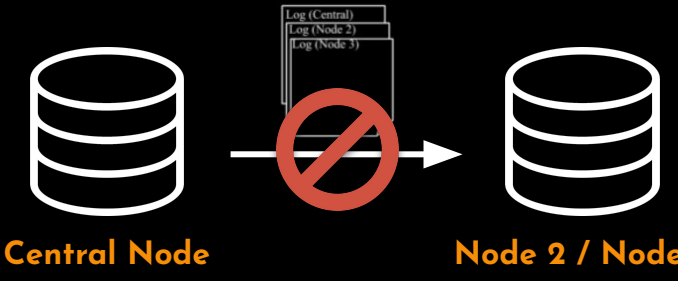
SIMULATING CASE 2

- Crash Node 2 / 3 during replication by destroying connection to Web App

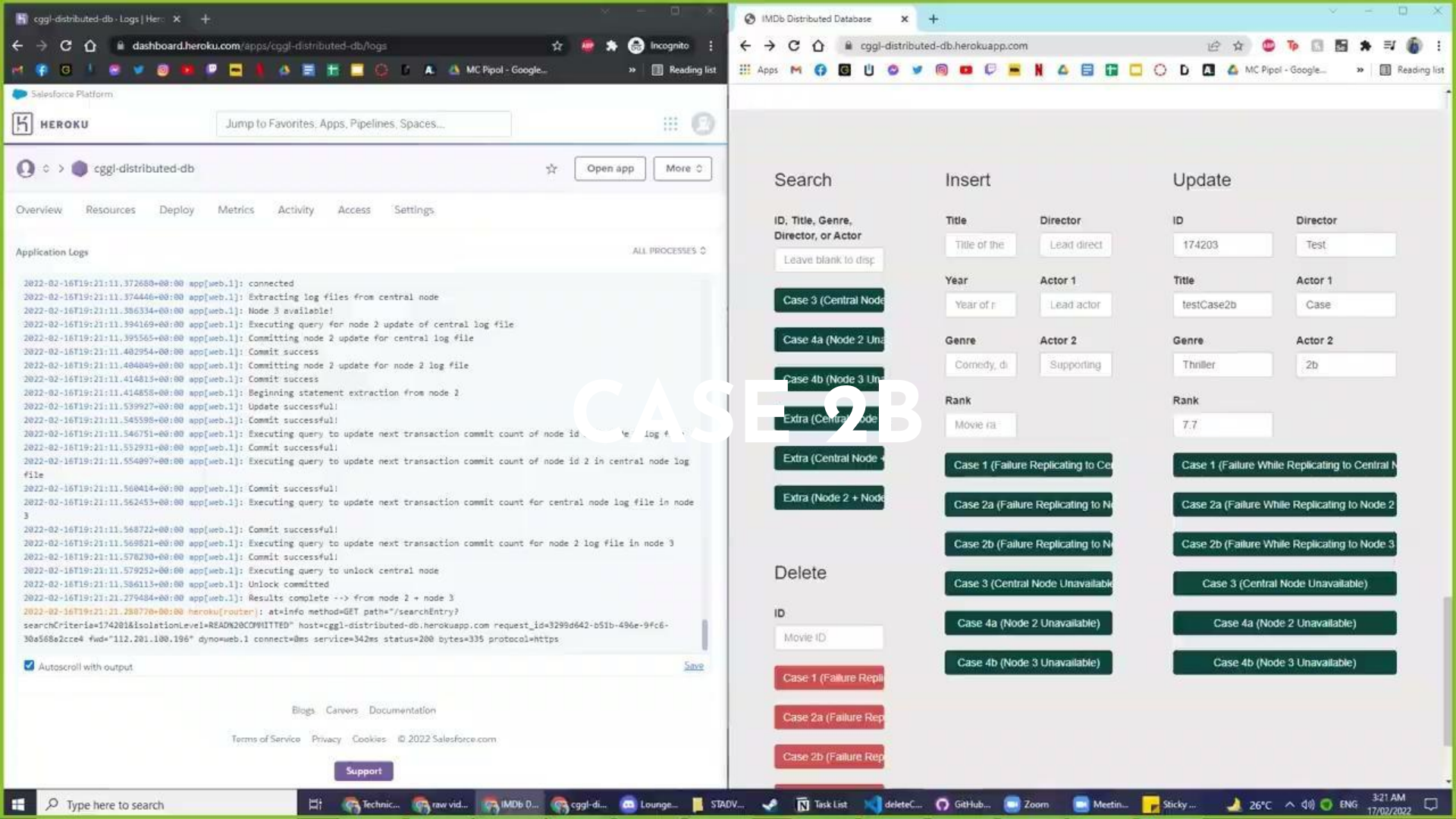
1. Initial Transaction



2. Replication



CASE 2A


cggl-distributed-db - Logs | Her
dashboard.heroku.com/apps/cggl-distributed-db/logs
Incognito
MC Pipol - Google...
Reading list
Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
cggl-distributed-db
Open app
More
Overview
Resources
Deploy
Metrics
Activity
Access
Settings
Application Logs
ALL PROCESSES
2022-02-16T19:21:11.372680+00:00 app[web.1]: connected
2022-02-16T19:21:11.374446+00:00 app[web.1]: Extracting log files from central node
2022-02-16T19:21:11.386334+00:00 app[web.1]: Node 3 available!
2022-02-16T19:21:11.394169+00:00 app[web.1]: Executing query for node 2 update of central log file
2022-02-16T19:21:11.395565+00:00 app[web.1]: Committing node 2 update for central log file
2022-02-16T19:21:11.402954+00:00 app[web.1]: Commit success
2022-02-16T19:21:11.404049+00:00 app[web.1]: Committing node 2 update for node 2 log file
2022-02-16T19:21:11.414815+00:00 app[web.1]: Commit success
2022-02-16T19:21:11.414858+00:00 app[web.1]: Beginning statement extraction from node 2
2022-02-16T19:21:11.539927+00:00 app[web.1]: Update successful!
2022-02-16T19:21:11.545598+00:00 app[web.1]: Commit successful!
2022-02-16T19:21:11.546751+00:00 app[web.1]: Executing query to update next transaction commit count of node id
2022-02-16T19:21:11.552931+00:00 app[web.1]: Commit successful!
2022-02-16T19:21:11.554097+00:00 app[web.1]: Executing query to update next transaction commit count of node id 2 in central node log file
2022-02-16T19:21:11.560414+00:00 app[web.1]: Commit successful!
2022-02-16T19:21:11.562453+00:00 app[web.1]: Executing query to update next transaction commit count for central node log file in node 3
2022-02-16T19:21:11.568722+00:00 app[web.1]: Commit successful!
2022-02-16T19:21:11.569821+00:00 app[web.1]: Executing query to update next transaction commit count for node 2 log file in node 3
2022-02-16T19:21:11.578230+00:00 app[web.1]: Commit successful!
2022-02-16T19:21:11.579252+00:00 app[web.1]: Executing query to unlock central node
2022-02-16T19:21:11.586113+00:00 app[web.1]: Unlock committed
2022-02-16T19:21:21.279484+00:00 app[web.1]: Results complete --> from node 2 + node 3
2022-02-16T19:21:21.280720+00:00 heroku[router]: at=info method=GET path="/searchEntry?searchCriteria=174201&isolationLevel=READ%20COMMITTED" host=cggl-distributed-db.herokuapp.com request_id=3299d642-b51b-496e-9fc6-30a568a2cce4 fwd="112.201.100.196" dyno=web.1 connect=0ms service=342ms status=200 bytes=335 protocol=https
Autoscroll with output
Save
Blogs
Careers
Documentation
Terms of Service
Privacy
Cookies
© 2022 Salesforce.com
Support
IMDb Distributed Database
cggl-distributed-db.herokuapp.com
Apps
MC Pipol - Google...
Reading list
Search
ID, Title, Genre, Director, or Actor
Leave blank to disp
Case 3 (Central Node
Case 4a (Node 2 Una
Case 4b (Node 3 Un
Extra (Central Node
Extra (Central Node
Extra (Node 2 + Node
Delete
ID
Movie ID
Case 1 (Failure Repli
Case 2a (Failure Rep
Case 2b (Failure Rep
Insert
Title
Director
Title of the
Lead direct
Year
Actor 1
Year of r
Lead actor
Genre
Actor 2
Comedy, da
Supporting
Rank
Movie ra
Case 1 (Failure Replicating to Ce
Case 2a (Failure Replicating to N
Case 2b (Failure Replicating to N
Case 3 (Central Node Unavailable
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
Update
ID
Director
174203
Test
Title
Actor 1
testCase2b
Case
Genre
Actor 2
Thriller
2b
Rank
7.7
Case 1 (Failure While Replicating to Central N
Case 2a (Failure While Replicating to Node 2
Case 2b (Failure While Replicating to Node 3
Case 3 (Central Node Unavailable)
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
CASE 2B
Type here to search
26°C
ENG
3:21 AM
17/02/2022

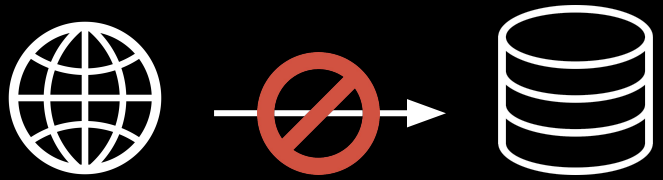
Global Recovery

Case 3: Central Node is not available

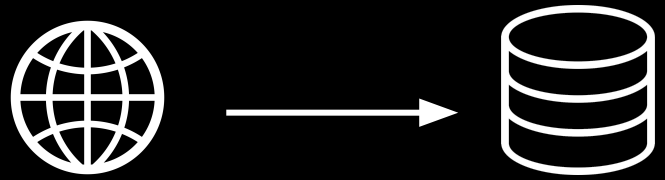
SIMULATING CASE 3

- Central Node is crashed before initial transaction by destroying connection of Web App

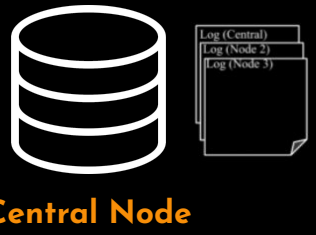
1. Initial Transaction

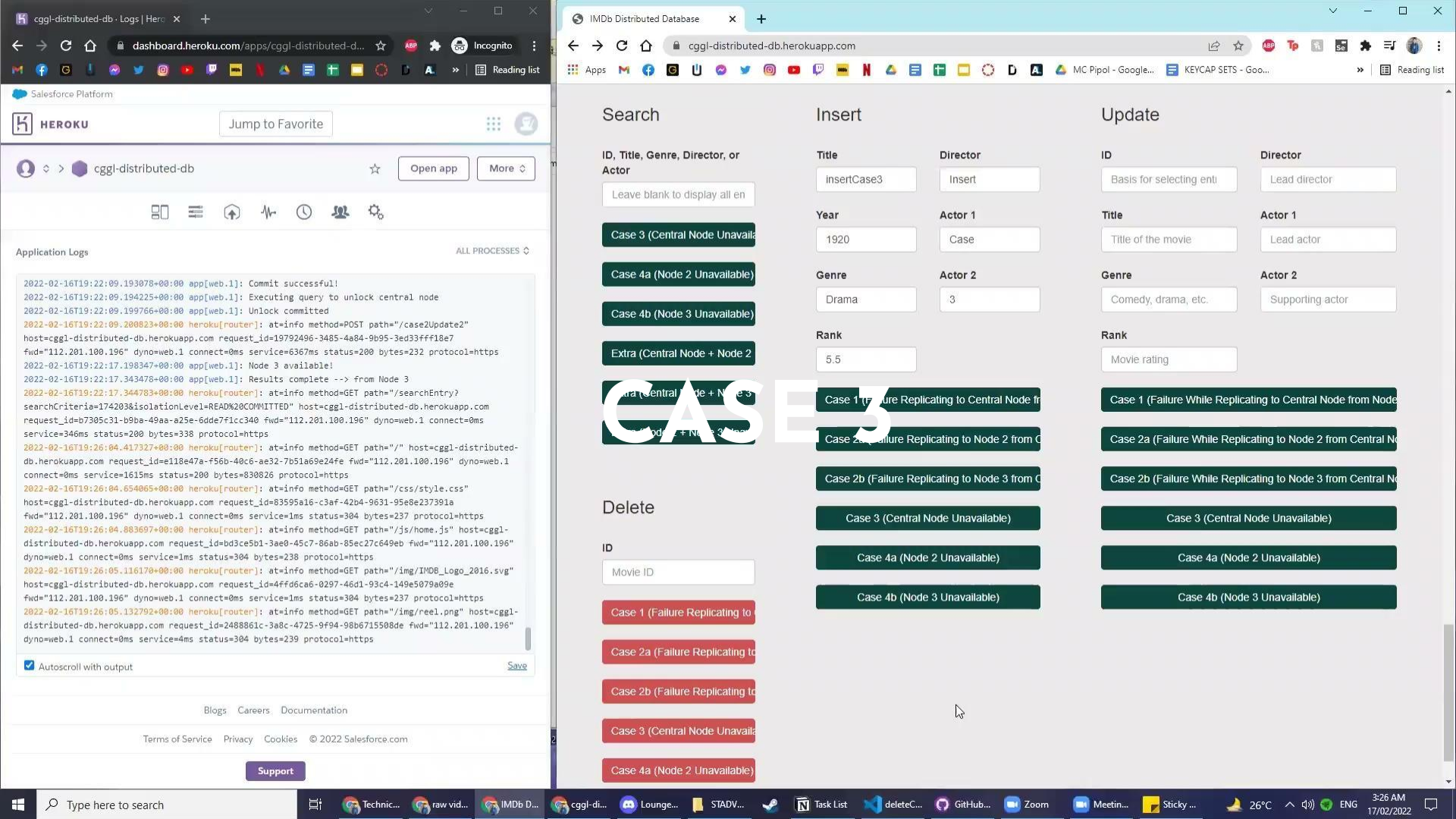


2. Redirection



3. Catching Up




cggl-distributed-db · Logs | Hero
dashboard.heroku.com/apps/cggl-distributed-d...
Incognito
Reading list
Salesforce Platform
HEROKU
Jump to Favorite
cggl-distributed-db
Open app
More
Application Logs
ALL PROCESSES
2022-02-16T19:22:09.193078+00:00 app[web.1]: Commit successful!
2022-02-16T19:22:09.194225+00:00 app[web.1]: Executing query to unlock central node
2022-02-16T19:22:09.199766+00:00 app[web.1]: Unlock committed
2022-02-16T19:22:09.200823+00:00 heroku[router]: at=info method=POST path="/case2Update2" host=cggl-distributed-db.herokuapp.com request_id=19792496-3485-4a84-9b95-3ed33fff18e7 fwd="112.201.100.196" dyno=web.1 connect=0ms service=6367ms status=200 bytes=232 protocol=https
2022-02-16T19:22:17.198347+00:00 app[web.1]: Node 3 available!
2022-02-16T19:22:17.343478+00:00 app[web.1]: Results complete --> from Node 3
2022-02-16T19:22:17.344783+00:00 heroku[router]: at=info method=GET path="/searchEntry?searchCriteria=174203&isolationLevel=READ%20COMMITTED" host=cggl-distributed-db.herokuapp.com request_id=b7305c31-b9ba-49aa-a25e-6dde7f1cc340 fwd="112.201.100.196" dyno=web.1 connect=0ms service=346ms status=200 bytes=338 protocol=https
2022-02-16T19:26:04.417327+00:00 heroku[router]: at=info method=GET path="/" host=cggl-distributed-db.herokuapp.com request_id=e118e47a-f56b-40c6-ae32-7b51a69e24fe fwd="112.201.100.196" dyno=web.1 connect=0ms service=1615ms status=200 bytes=830826 protocol=https
2022-02-16T19:26:04.654065+00:00 heroku[router]: at=info method=GET path="/css/style.css" host=cggl-distributed-db.herokuapp.com request_id=83595a16-c3af-42b4-9631-95e8e237391a fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:26:04.883697+00:00 heroku[router]: at=info method=GET path="/js/home.js" host=cggl-distributed-db.herokuapp.com request_id=bd3ce5b1-3ae0-45c7-86ab-85ec27c649eb fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=238 protocol=https
2022-02-16T19:26:05.116170+00:00 heroku[router]: at=info method=GET path="/img/IMDB_Logo_2016.svg" host=cggl-distributed-db.herokuapp.com request_id=4ffd6ca6-0297-46d1-93c4-149e5079a09e fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:26:05.132792+00:00 heroku[router]: at=info method=GET path="/img/reel.png" host=cggl-distributed-db.herokuapp.com request_id=2488861c-3a8c-4725-9f94-98b6715508de fwd="112.201.100.196" dyno=web.1 connect=0ms service=4ms status=304 bytes=239 protocol=https
Autoscroll with output
Save
Blogs
Careers
Documentation
Terms of Service
Privacy
Cookies
© 2022 Salesforce.com
Support
IMDb Distributed Database
cggl-distributed-db.herokuapp.com
Apps
MC Pipol - Google...
KEYCAP SETS - Goo...
Reading list
Search
ID, Title, Genre, Director, or Actor
Leave blank to display all en
Case 3 (Central Node Unavaila
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
Extra (Central Node + Node 2
Delete
ID
Movie ID
Case 1 (Failure Replicating to
Case 2a (Failure Replicating to
Case 2b (Failure Replicating to
Case 3 (Central Node Unavaila
Case 4a (Node 2 Unavailable)
Insert
Title
insertCase3
Director
Insert
Year
1920
Actor 1
Case
Genre
Drama
Actor 2
3
Rank
5.5
Case 1 (Failure Replicating to Central Node fr
Case 2a (Failure Replicating to Node 2 from C
Case 2b (Failure Replicating to Node 3 from C
Case 3 (Central Node Unavailable)
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
Update
ID
Basis for selecting enti
Director
Lead director
Title
Title of the movie
Actor 1
Lead actor
Genre
Comedy, drama, etc.
Actor 2
Supporting actor
Rank
Movie rating
Case 1 (Failure While Replicating to Central Node from Node
Case 2a (Failure While Replicating to Node 2 from Central No
Case 2b (Failure While Replicating to Node 3 from Central No
Case 3 (Central Node Unavailable)
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
CASE 3
Type here to search
26°C
ENG
3:26 AM
17/02/2022

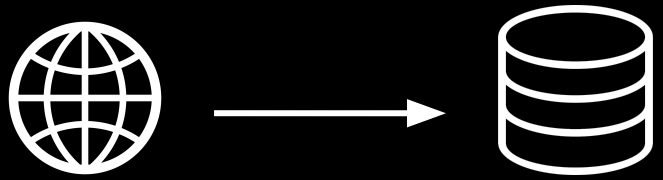
Global Recovery

Case 4: Node 2 or Node 3 is not available

SIMULATING CASE 4

- Crash Node 2 / 3 before the start of the transaction by destroying connection to Web App

1. Initial Transaction

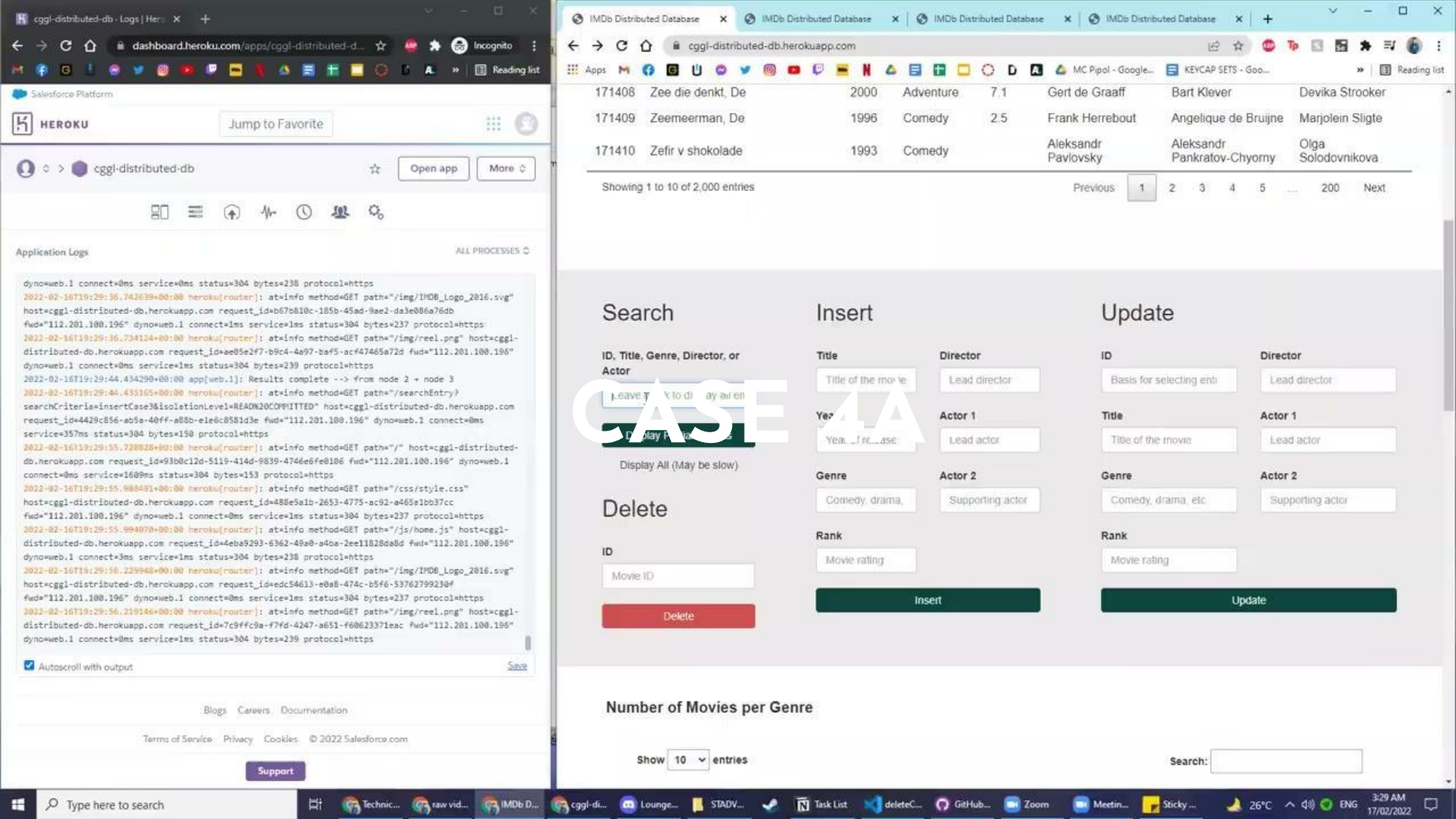


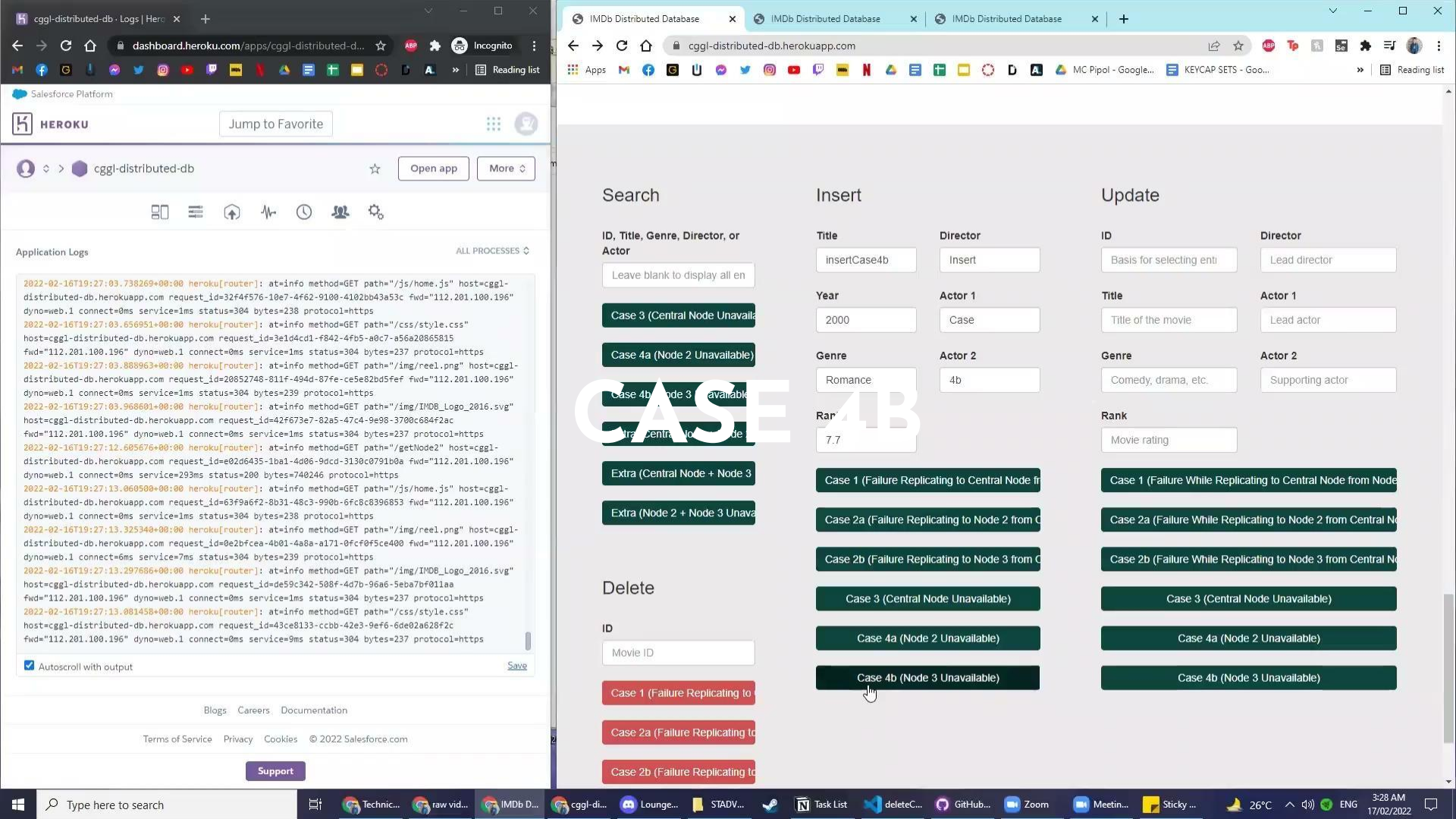
Web Application

Central Node

2. Catching Up

Node 2 / Node 3


cggl-distributed-db · Logs | Hero
dashboard.heroku.com/apps/cggl-distributed-d...
Incognito
Reading list
Salesforce Platform
HEROKU
Jump to Favorite
cggl-distributed-db
Open app
More
Application Logs
ALL PROCESSES
dyno=web.1 connect=0ms service=0ms status=304 bytes=238 protocol=https
2022-02-16T19:29:36.742639+00:00 heroku[router]: at=info method=GET path="/img/IMDB_Logo_2016.svg" host=cggl-distributed-db.herokuapp.com request_id=b67b810c-185b-45ad-9ae2-da3e086a76db fwd="112.201.100.196" dyno=web.1 connect=1ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:29:36.734124+00:00 heroku[router]: at=info method=GET path="/img/reel.png" host=cggl-distributed-db.herokuapp.com request_id=ae05e2f7-b9c4-4a97-baf5-acf47465a72d fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=239 protocol=https
2022-02-16T19:29:44.434290+00:00 app[web.1]: Results complete --> from node 2 + node 3
2022-02-16T19:29:44.435165+00:00 heroku[router]: at=info method=GET path="/searchEntry?searchCriteria=insertCase3&isolationLevel=READ%20COMMITTED" host=cggl-distributed-db.herokuapp.com request_id=4429c856-ab5a-40ff-a88b-e1e6c8581d3e fwd="112.201.100.196" dyno=web.1 connect=0ms service=357ms status=304 bytes=150 protocol=https
2022-02-16T19:29:55.728828+00:00 heroku[router]: at=info method=GET path="/" host=cggl-distributed-db.herokuapp.com request_id=93b0c12d-5119-414d-9839-474de6fe0106 fwd="112.201.100.196" dyno=web.1 connect=0ms service=1609ms status=304 bytes=153 protocol=https
2022-02-16T19:29:55.988491+00:00 heroku[router]: at=info method=GET path="/css/style.css" host=cggl-distributed-db.herokuapp.com request_id=488e5a1b-2653-4775-ac92-a465e1bb37cc fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:29:55.994070+00:00 heroku[router]: at=info method=GET path="/js/home.js" host=cggl-distributed-db.herokuapp.com request_id=4eba9293-6362-49a0-a4ba-2ee11828da8d fwd="112.201.100.196" dyno=web.1 connect=3ms service=1ms status=304 bytes=238 protocol=https
2022-02-16T19:29:56.229948+00:00 heroku[router]: at=info method=GET path="/img/IMDB_Logo_2016.svg" host=cggl-distributed-db.herokuapp.com request_id=edc54613-e0a8-474c-b5f6-53762799230f fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:29:56.219146+00:00 heroku[router]: at=info method=GET path="/img/reel.png" host=cggl-distributed-db.herokuapp.com request_id=7c9ffc9a-f7fd-4247-a651-f60623371eac fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=239 protocol=https
Autoscroll with output
Save
Blogs Careers Documentation
Terms of Service Privacy Cookies © 2022 Salesforce.com
Support
IMDb Distributed Database
cggl-distributed-db.herokuapp.com
Apps
Reading list
171408 Zee die denkt, De 2000 Adventure 7.1 Gert de Graaff Bart Klever Devika Strooker
171409 Zeemeerman, De 1996 Comedy 2.5 Frank Herrebout Angelique de Bruijne Marjolein Sligte
171410 Zefir v shokolade 1993 Comedy Aleksandr Pavlovsky Aleksandr Pankratov-Chyorny Olga Solodovnikova
Showing 1 to 10 of 2,000 entries
Previous 1 2 3 4 5 ... 200 Next
Search
ID, Title, Genre, Director, or Actor
Display All (May be slow)
Delete
ID
Movie ID
Delete
Insert
Title
Title of the movie
Director
Lead director
Year
Actor 1
Lead actor
Genre
Comedy, drama,
Actor 2
Supporting actor
Rank
Movie rating
Insert
Update
ID
Basis for selecting entr
Director
Lead director
Title
Title of the movie
Actor 1
Lead actor
Genre
Comedy, drama, etc
Actor 2
Supporting actor
Rank
Movie rating
Update
CASE 4A
Number of Movies per Genre
Show 10 entries
Search:
Type here to search
26°C
ENG
3:29 AM
17/02/2022


cggl-distributed-db · Logs | Hero
dashboard.heroku.com/apps/cggl-distributed-d...
Incognito
Reading list
Salesforce Platform
HEROKU
Jump to Favorite
cggl-distributed-db
Open app
More
Application Logs
ALL PROCESSES
2022-02-16T19:27:03.738269+00:00 heroku[router]: at=info method=GET path="/js/home.js" host=cggl-distributed-db.herokuapp.com request_id=32f4f576-10e7-4f62-9100-4102bb43a53c fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=238 protocol=https
2022-02-16T19:27:03.656951+00:00 heroku[router]: at=info method=GET path="/css/style.css" host=cggl-distributed-db.herokuapp.com request_id=3e1d4cd1-f842-4fb5-a0c7-a56a20865815 fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:27:03.888963+00:00 heroku[router]: at=info method=GET path="/img/reel.png" host=cggl-distributed-db.herokuapp.com request_id=20852748-811f-494d-87fe-ce5e82bd5fef fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=239 protocol=https
2022-02-16T19:27:03.968601+00:00 heroku[router]: at=info method=GET path="/img/IMDB_Logo_2016.svg" host=cggl-distributed-db.herokuapp.com request_id=42f673e7-82a5-47c4-9e98-3700c684f2ac fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:27:12.605676+00:00 heroku[router]: at=info method=GET path="/getNode2" host=cggl-distributed-db.herokuapp.com request_id=e02d6435-1ba1-4d06-9dcd-3130c0791b0a fwd="112.201.100.196" dyno=web.1 connect=0ms service=293ms status=200 bytes=740246 protocol=https
2022-02-16T19:27:13.060500+00:00 heroku[router]: at=info method=GET path="/js/home.js" host=cggl-distributed-db.herokuapp.com request_id=63f9a6f2-8b31-48c3-990b-6fc8c8396853 fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=238 protocol=https
2022-02-16T19:27:13.325340+00:00 heroku[router]: at=info method=GET path="/img/reel.png" host=cggl-distributed-db.herokuapp.com request_id=0e2bfcea-4b01-4a8a-a171-0fcf0f5ce400 fwd="112.201.100.196" dyno=web.1 connect=6ms service=7ms status=304 bytes=239 protocol=https
2022-02-16T19:27:13.297686+00:00 heroku[router]: at=info method=GET path="/img/IMDB_Logo_2016.svg" host=cggl-distributed-db.herokuapp.com request_id=de59c342-508f-4d7b-96a6-5eba7bf011aa fwd="112.201.100.196" dyno=web.1 connect=0ms service=1ms status=304 bytes=237 protocol=https
2022-02-16T19:27:13.081458+00:00 heroku[router]: at=info method=GET path="/css/style.css" host=cggl-distributed-db.herokuapp.com request_id=43ce8133-ccbb-42e3-9ef6-6de02a628f2c fwd="112.201.100.196" dyno=web.1 connect=0ms service=9ms status=304 bytes=237 protocol=https
Autoscroll with output
Save
Blogs
Careers
Documentation
Terms of Service
Privacy
Cookies
© 2022 Salesforce.com
Support
IMDb Distributed Database
cggl-distributed-db.herokuapp.com
Apps
MC Pipol - Google...
KEYCAP SETS - Goo...
Reading list
Search
ID, Title, Genre, Director, or Actor
Leave blank to display all en
Case 3 (Central Node Unavaila
Case 4a (Node 2 Unavailable)
Extra (Central Node + Node 3
Extra (Node 2 + Node 3 Unava
Delete
ID
Movie ID
Case 1 (Failure Replicating to
Case 2a (Failure Replicating to
Case 2b (Failure Replicating to
Insert
Title
insertCase4b
Director
Insert
Year
2000
Actor 1
Case
Genre
Romance
Actor 2
4b
Rank
7.7
Case 1 (Failure Replicating to Central Node fr
Case 2a (Failure Replicating to Node 2 from C
Case 2b (Failure Replicating to Node 3 from C
Case 3 (Central Node Unavailable)
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
Update
ID
Basis for selecting enti
Director
Lead director
Title
Title of the movie
Actor 1
Lead actor
Genre
Comedy, drama, etc.
Actor 2
Supporting actor
Rank
Movie rating
Case 1 (Failure While Replicating to Central Node from Node
Case 2a (Failure While Replicating to Node 2 from Central No
Case 2b (Failure While Replicating to Node 3 from Central No
Case 3 (Central Node Unavailable)
Case 4a (Node 2 Unavailable)
Case 4b (Node 3 Unavailable)
CASE 4B
Type here to search
26°C
ENG
3:28 AM
17/02/2022

Concurrency

Results of the transactions were **recorded** and compared against the **expected results** as enumerated in the **test scripts**.

EVALUATION METHOD

Concurrency

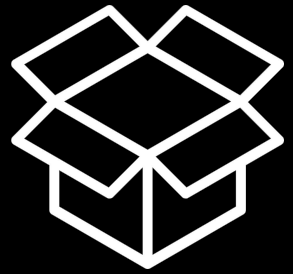
Case	Scenario	Behavior
1 - All Reads		Reads returned the same data
2 - One Write Two Reads	Read Uncommitted	Dirty reads occurred
	Read Committed	Reads return the original transaction values before committing
	Repeatable Read	Reads return the original transaction values before committing
	Serializable	Reads return the updated transaction values after committing
3 - All Writes	Lock wait timeout not exceeded	Transactions are executed serially
	Lock wait timeout exceeded	First transaction is executed, other transactions are aborted

EVALUATION RESULTS

Global Failure Recovery

White-box examinations of the **logs** were performed.

Black-box tests were performed on **attributes related to the expected output** or the **database states** at each of the nodes.



EVALUATION METHOD

Global Failure Recovery

Read Transactions

Transaction Type	Attributes
Select (returns single entry)	Attribute values of the entry
Select (returns multiple entries)	Cardinality and attributes of the results
	Attribute values of the entry with the lowest ID
	Attribute values of the entry with the highest ID

Global Failure Recovery

Write Transactions

Transaction Type	Attributes
Insert	Attribute values of the added entry
	Cardinality of the movies table in each node
	Mutual exclusivity of data in Nodes 2 and 3
Update	Attribute values of the updated entry
	Presence of updated entry in the nodes
Delete	Absence of deleted entry in the application layer
	Absence of deleted entry in the nodes

Global Failure Recovery

Case 1: Failure in Replicating from Node 2 or Node 3 to the Central Node

The transaction is directed to the proper partial replica based on the **year**.

The transaction is **executed** and **committed**, and its **logs are updated**.

A record of the transaction persists in the **(central node, Node x)** log.

While the central node is down, **next_trans_commit is behind next_trans_record**.

Global Failure Recovery

Case 2: Failure in Replicating from the Central Node to Node 2 or Node 3

The transaction is directed to the **central node**.

The transaction is **executed** and **committed**, and its **logs are updated**.

A record of the transaction persists in the **(Node x, central node)** log.

While Node x is down, **next_trans_commit is behind next_trans_record**.

Global Failure Recovery

Case 3: Central Node is Not Available

The same recovery strategy as **Case 1** is applied.

The **id_new_entry** log field informs the partial replicas of the **ID of the next entry** to be inserted.

Users can still **perform transactions** on **Nodes 2 and 3**.

After the central node recovers it **performs a catch up** based on the **logs**.

Global Failure Recovery

Case 4: Node 2 or Node 3 is Not Available

The same recovery strategy as **Case 2** is applied.

Users can still **perform transactions** on the **central node**.

After a partial replica recovers it **performs a catch up** based on the **logs**.

Global Failure Recovery

Unavailable Node	Cardinality of Result Set	Description
Central Node	11574	Full result set
Node 2	11574	Full result set
Node 3	11574	Full result set
Central Node and Node 2	7900	Partial result set
Central Node and Node 3	3674	Partial result set
Nodes 2 and 3	11574	Full result set
Central Node, Node 2, and Node 3	0	Empty result set

INSIGHTS AND CONCLUSIONS

Implementation independent of DBMS for concurrency and recover control protocols highlighted complexities of distributed database system maintenance

Decentralized statement-based logs facilitated robust concurrency and failure management

Combination of the `READ COMMITTED` isolation level and write locks prevents transaction anomalies

INSIGHTS AND CONCLUSIONS

Reliable point of access for the IMDb dataset and proof of concept for distributed database management techniques

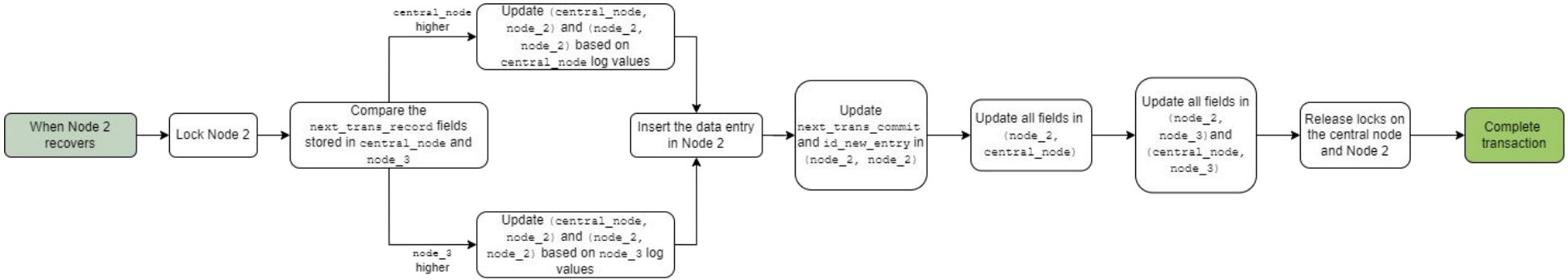
Maintaining replicated logs allows system operation to continue even when the central node is down

Use of `async` and `await` functions and threading can improve the quality of the implementation

THANK YOU!

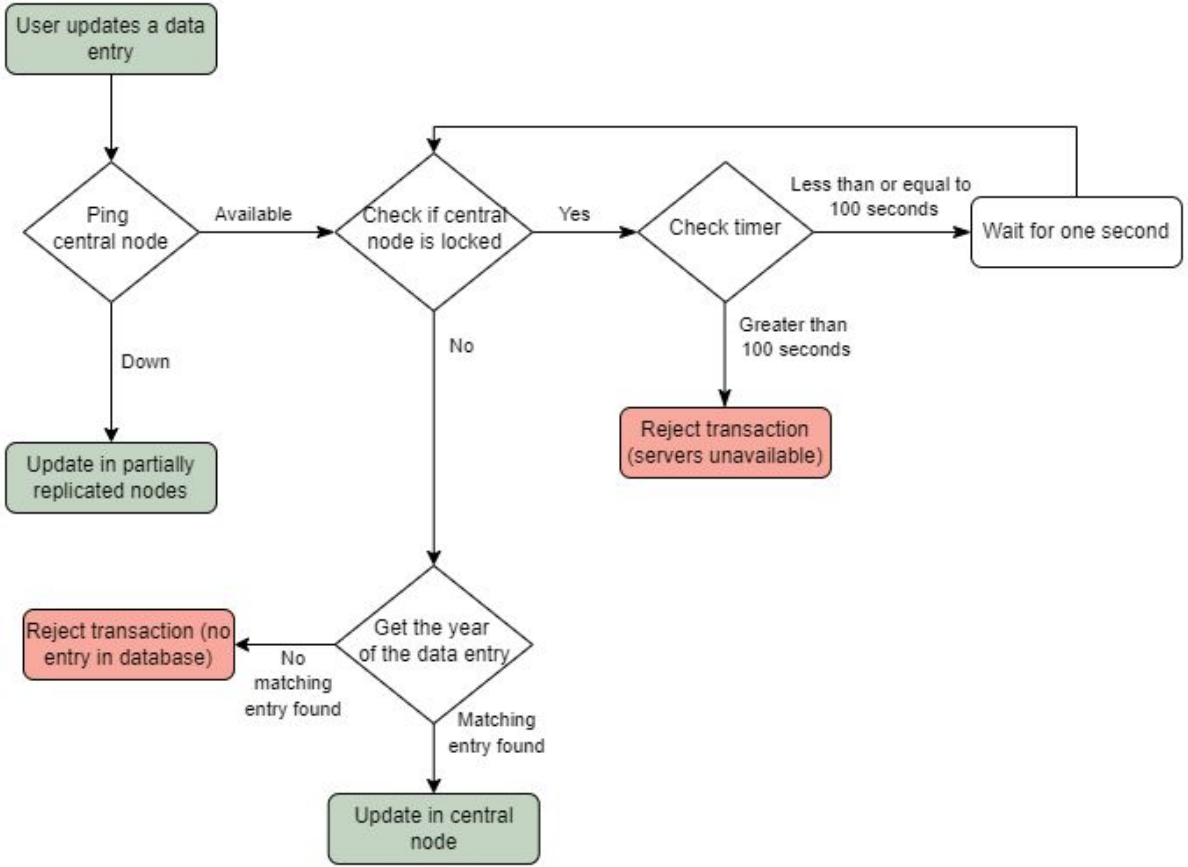
Group 13

Cua, Lander Peter E
Gaba, Jacob Bryan B.
Gonzales, Mark Edward M.
Lee, Hylene Jules G.


When Node 2 recovers
Lock Node 2
Compare the next_trans_record fields stored in central_node and node_3
central_node higher
Update (central_node, node_2) and (node_2, node_2) based on central_node log values
node_3 higher
Update (central_node, node_2) and (node_2, node_2) based on node_3 log values
Insert the data entry in Node 2
Update next_trans_commit and id_new_entry in (node_2, node_2)
Update all fields in (node_2, central_node)
Update all fields in (node_2, node_3) and (central_node, node_3)
Release locks on the central node and Node 2
Complete transaction

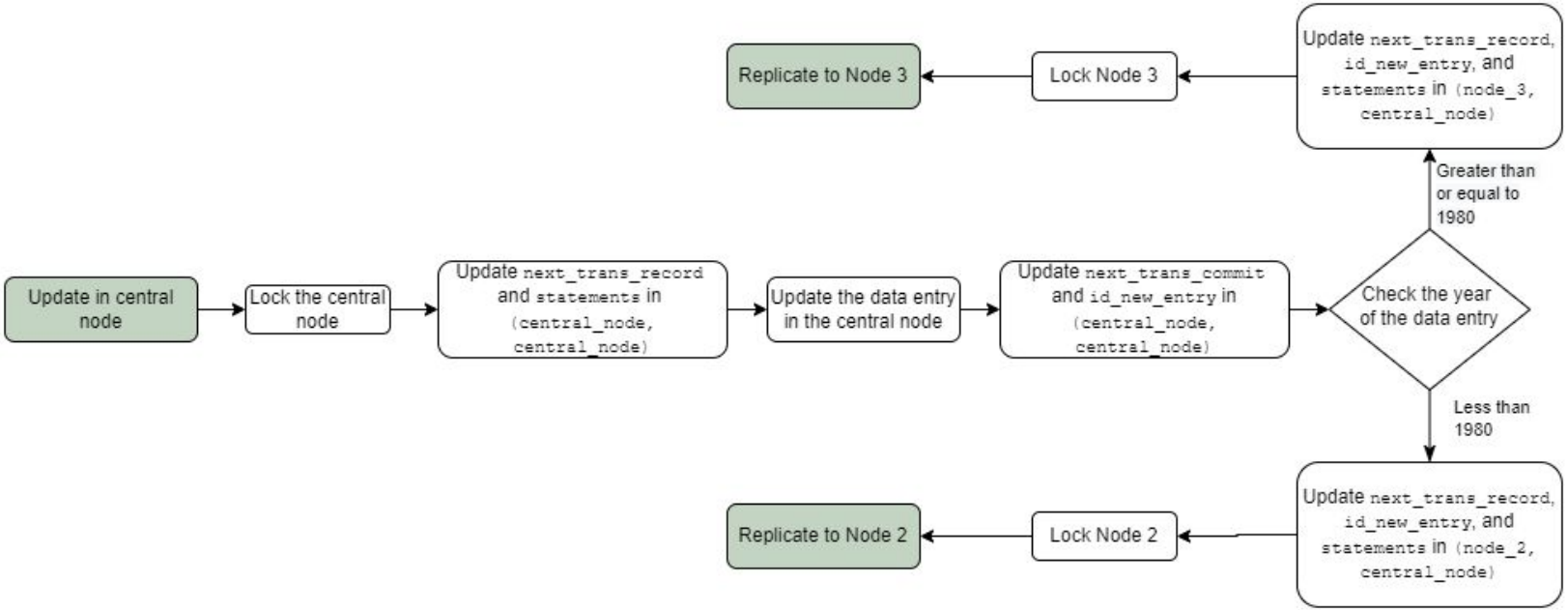
CATCHING UP

ALGORITHM (UPDATE)

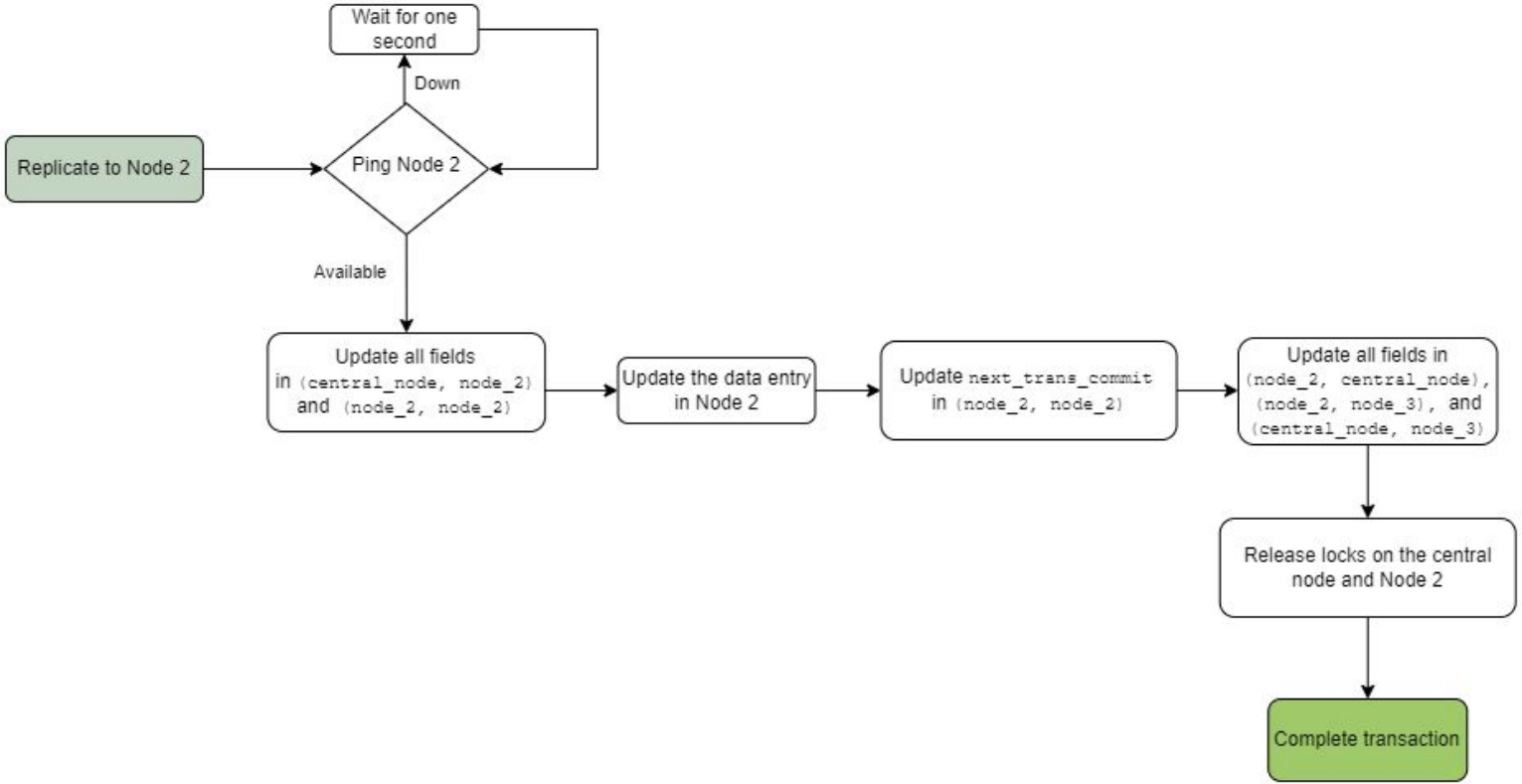


ALGORITHM (UPDATE)

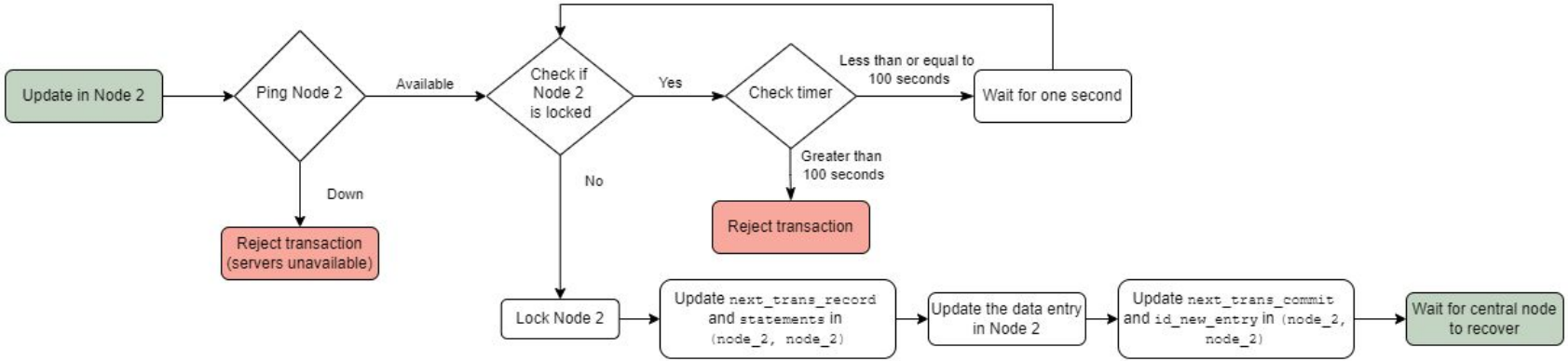
ALGORITHM (UPDATE)



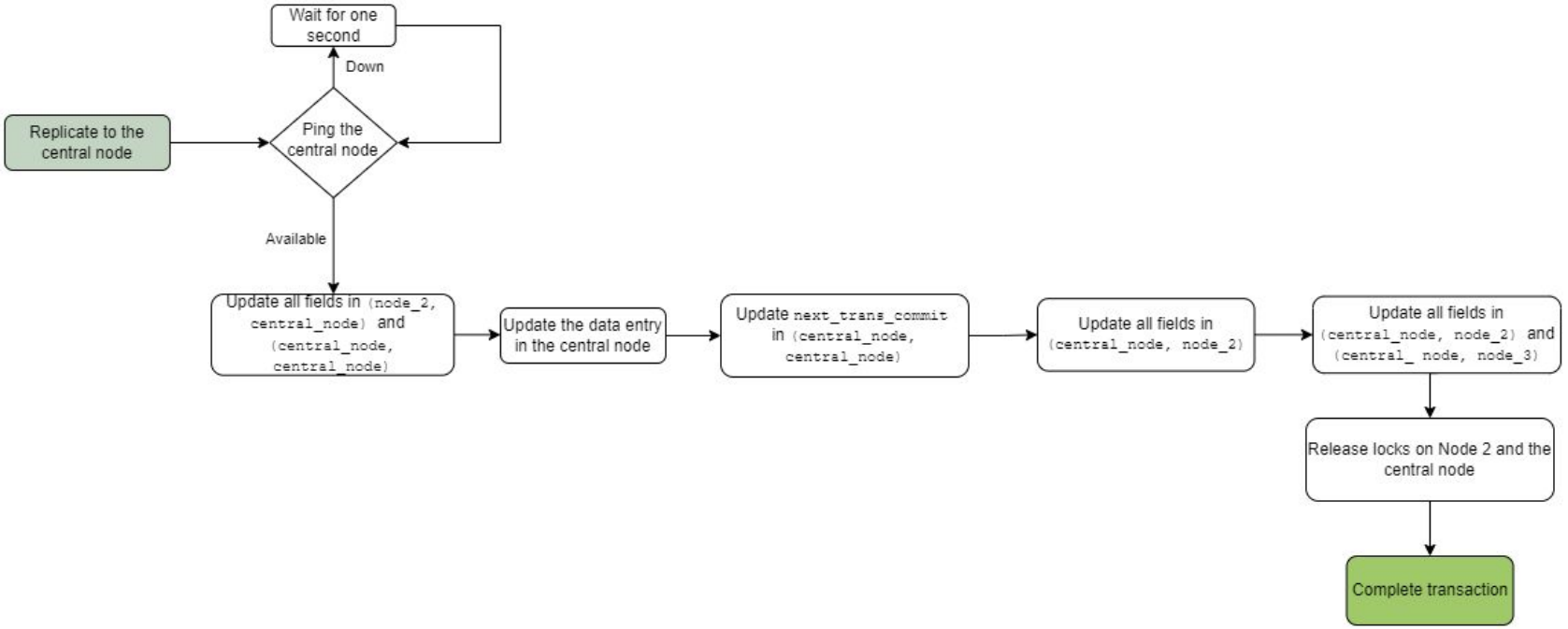
ALGORITHM (UPDATE)



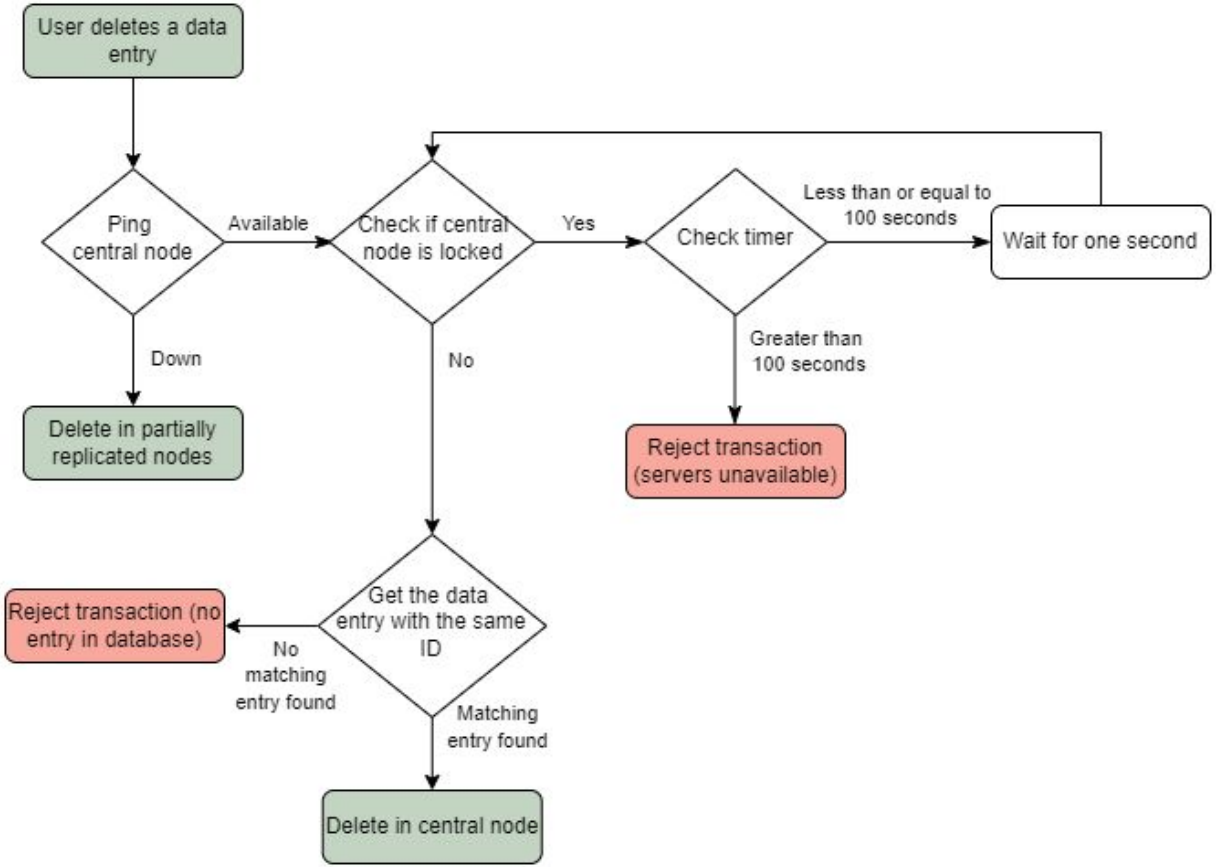
ALGORITHM (UPDATE)



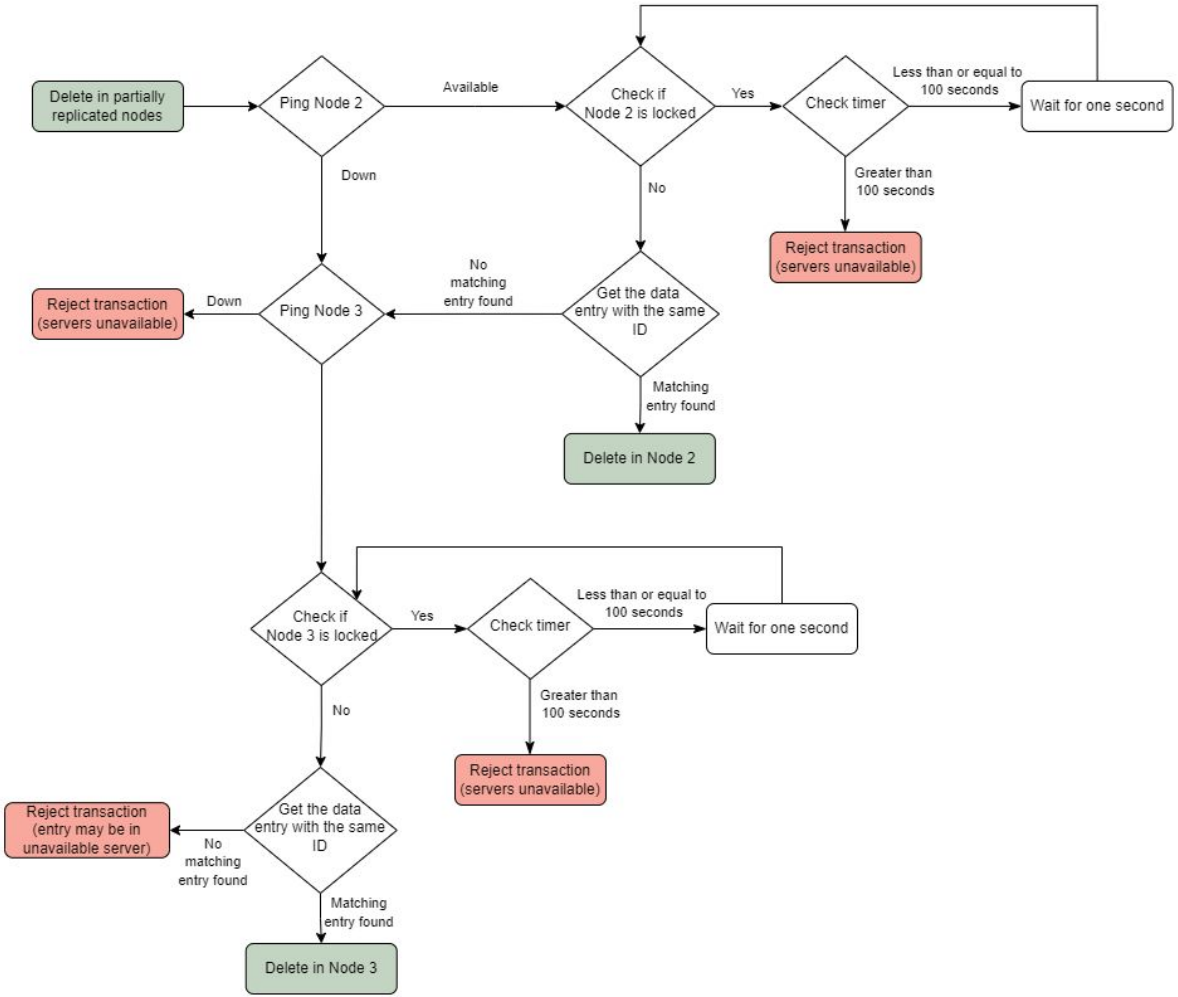
ALGORITHM (UPDATE)



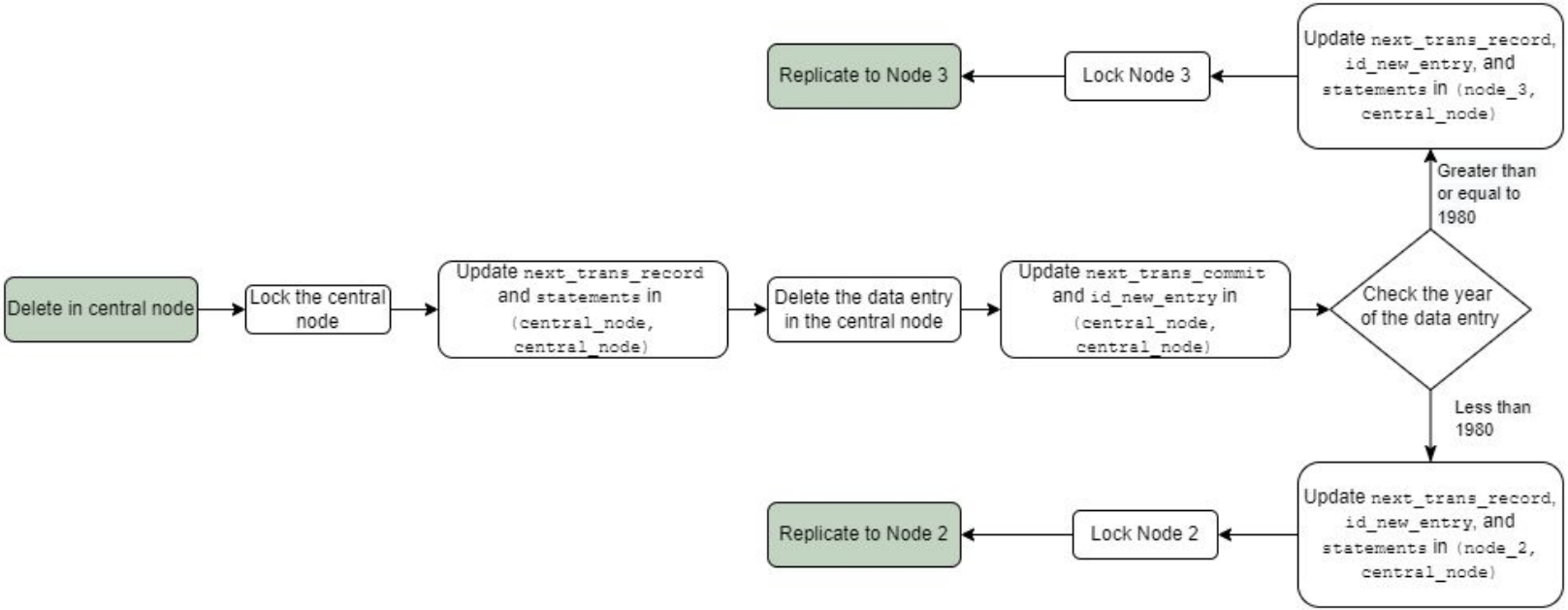
ALGORITHM (DELETE)



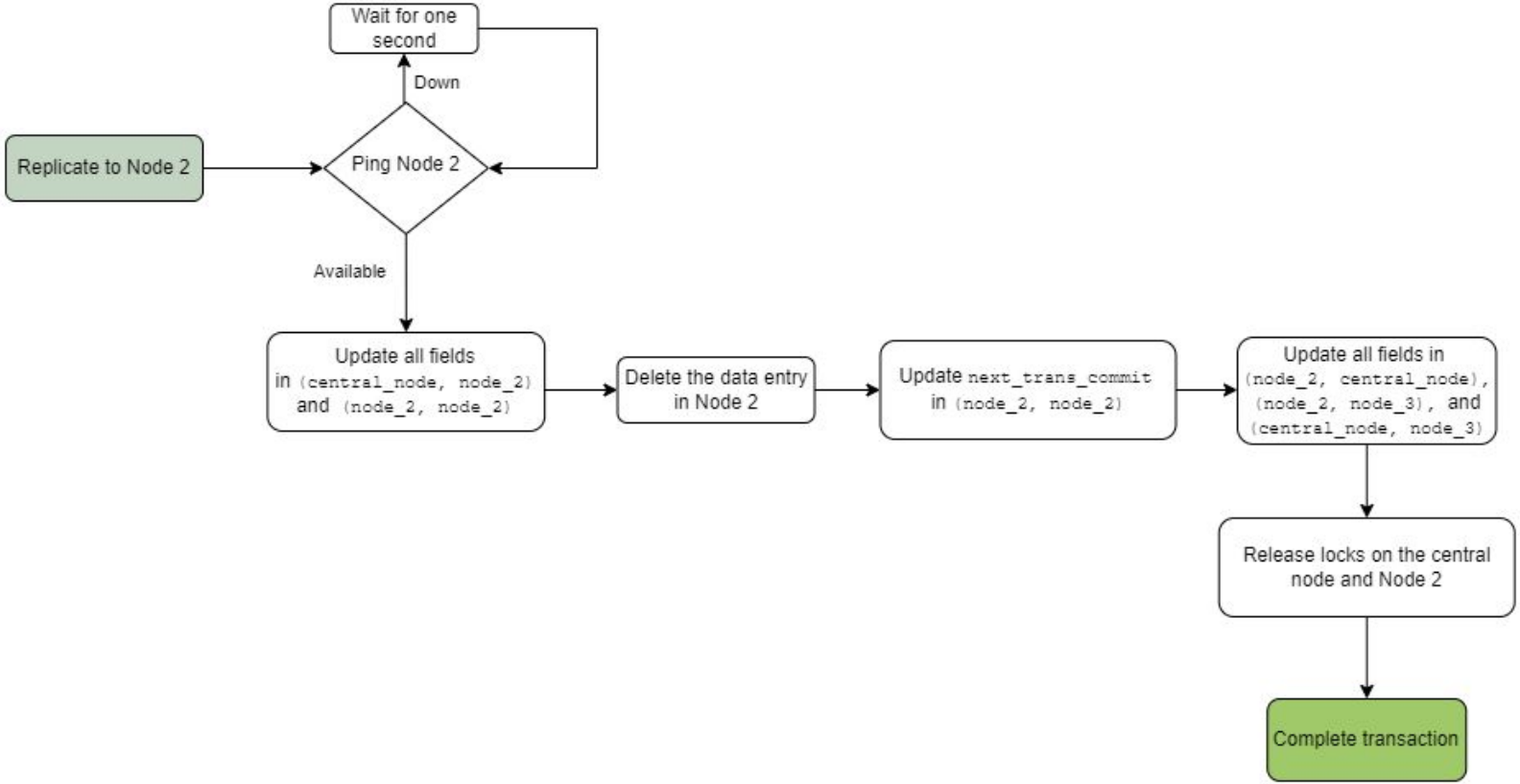
ALGORITHM (DELETE)



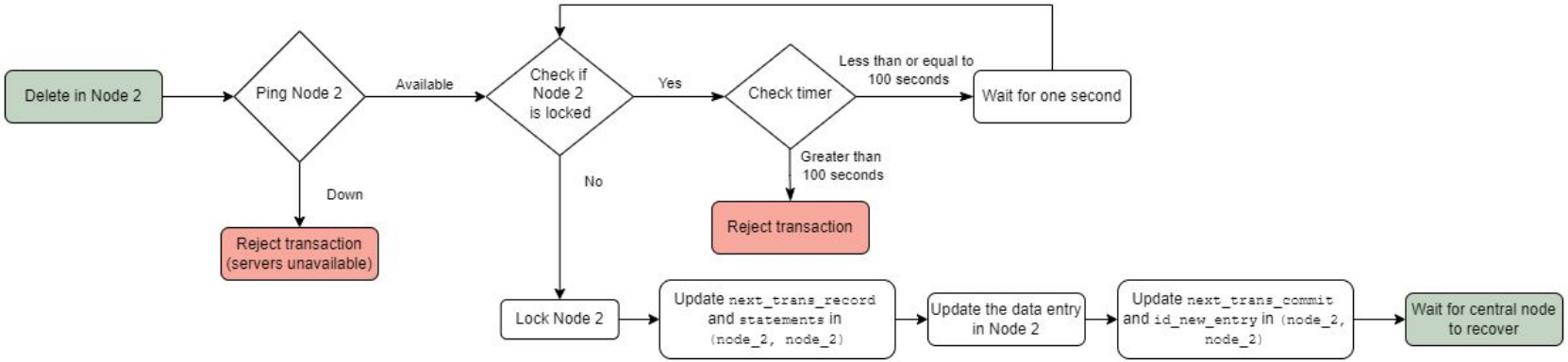
ALGORITHM (DELETE)



ALGORITHM (DELETE)



ALGORITHM (DELETE)



ALGORITHM (DELETE)