



Implementation of High Availability Based on Load Balancing and Failover Method in E-commerce Using MySQL Database

Bima Prasetya Adi Pratama^{1*}, Bambang Gunawan Hardianto², Dharmayanti³

Universitas Gunadarma, Indonesia

Emails : bimapap@gmail.com¹, bambang_gunawan@staff.gunadarma.ac.id²,
dharmayanti77@gmail.com³

ABSTRACT

The rapid growth of e-commerce has brought about challenges in managing traffic loads and ensuring the availability of database servers. One critical issue is what happens when the main database server fails, potentially leading to downtime or data loss. This research aims to address these problems by optimizing server resource distribution and implementing data replication across multiple database servers. The goal is to enhance system efficiency, prevent server overloads, ensure data availability, and minimize downtime or data loss when server failures occur. This research utilizes a multi-stage methodology. First, data collection was conducted to gather relevant information. In the second stage, a needs analysis was performed to identify system requirements. The third stage involved designing the system architecture, including the creation of entity-relationship diagrams (ERD), flowcharts, and network topologies. In the final stage, the system was implemented through database replication and the application of failover methods. The results demonstrate that the failover mechanism operates effectively, ensuring that data is successfully replicated across all database servers. This confirms that the system achieves high availability and minimizes the risk of data loss or service downtime. The implications of this research provide a reliable solution for improving the resilience and performance of e-commerce systems, particularly in managing database server failures.

Keywords: High Availability, Load Balancing, Failover, Orchestrator, Proxysql, MYSQL.

INTRODUCTION

In recent years, the growth of the development of the internet and its users has been very rapid and has become an important part of daily life (Greengard, 2021). Electronic commerce or e-commerce is the result of information technology that is currently developing so rapidly towards the purchase of goods, services and information through computer networks (Faqih & Wahyudi, 2022).



Figure 1. Transaction Category Data Trends E-commerce

Source: (Nurdian, 2022)

Figure 1. explains the trend of transaction category data through e-commerce including electronics, fashion, furniture, toys/hobbies, personal needs, food, beverages and physical media (DVD/CD).

According to (Hartman et al., 2001) in his book entitled "Net Ready-Strategies for Success in the E-Economy" defines e-commerce as a type of electronic business mechanism that focuses on individual-based business transactions by using the internet as a medium for the exchange of goods or services both between two institutions (B-to-B) and between institutions and direct consumers (B-to-C).

E-commerce has the need for good data availability at all times is a complicated problem (Sudrajat, 2020). This is related to system performance, the number of data accessors, and data network traffic which is getting congested all the time and this triggers software-based development requiring the addition of servers that will later function to help the load contained in the main server.

Load balancing works by distributing the workload evenly across multiple servers that act as the back-end server, so that the workload can be lighter (Syahrir, 2023). By implementing load balancing, it can shorten the access time to the web server and have high service availability (Kahanwal & Singh, 2013).

Failover, commonly known as high availability, generally aims to increase the availability of services provided by a server (Supendar & Handrianto, 2019). The failover element has redundant servers that will be used to provide services when one of the components fails (Kahanwal & Singh, 2013).

By combining the two technologies, a collection of servers acts as a single entity in providing resources and services to the network with the aim of maintaining the availability of resources for clients who access in the event of a system or hardware failure on the server.

Previous studies conducted by (Sumarna et al., 2019) this research applies N-clustering high availability web server with load balancing and failover. The results of the research concluded that the application of the n-clustering structure in the web server cluster can reduce the traffic load because the load sharing is carried out in two steps of load balancing before being given to the web server back-end (Antonelli et al., 2024). The availability of services in a web server cluster becomes higher due to the process of taking over requests to inactive web servers. The access time perceived by the client becomes faster and can increase the maximum number of clients without having to perform a thorough hardware replacement.

The next research was conducted by (Subekti et al., 2019) This research designed web server and database infrastructure using replication mirror and failover clustering methods. The results of the research were obtained by replication mirror and with the help of failover clustering that has been designed, it can be the best solution choice, because it can replace a dead server to another server and the client can still access the web server.

Based on the background description above, this research aims to determine and analyze the Implementation of High Availability Based on Load Balancing and Failover Method in E-commerce Using MySQL Database. So the benefit in this research is that this research is to provide solutions to improve service availability in e-commerce systems through the application of load balancing and failover, which allows the distribution of workload evenly to several servers and provides redundancy in the face of system failure. By using this method, it is expected that the e-commerce system can operate more stably, the access time to the server is shorter, and the service can continue to be available even if there is a disruption in one of the server components. This is very important considering the growing number of e-commerce users who continue to increase and increasingly demand fast and reliable performance..

RESEARCH METHOD

This research is divided into several stages to implement. For more details, see figure 2.

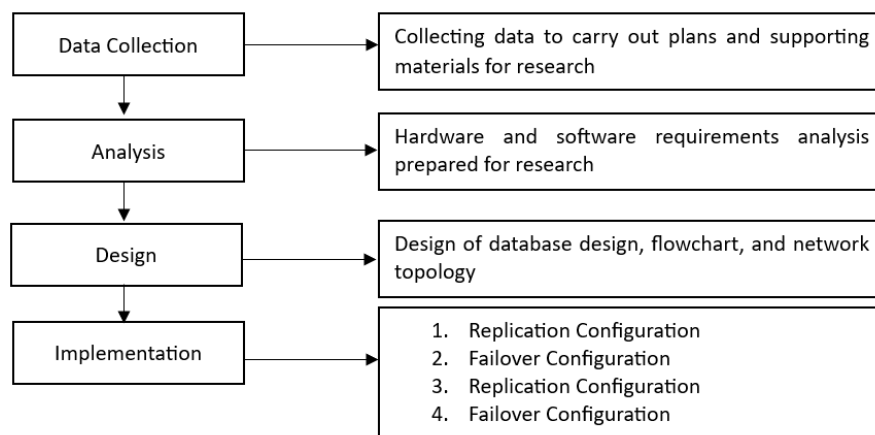


Figure 2. Research Stage

Figure 2 explains the research stage, namely the first stage of data collection, the second stage of analysis includes requirements analysis activities, the third stage of design includes activities of designing entity relationship diagrams (ERD), designing flowcharts, and designing network topology, and the last stage of implementation is activities of making database replication and creating methods failover.

In the early stages of the research is the collection of data to carry out plans and supporting materials for the research. The analysis stage in the research includes the hardware and software requirements prepared for the research. At the design stage, database design, flowchart and replication network topology and failover method are designed.

RESULT AND DISCUSSION

Analysis Results

At the analysis stage, there are two results, namely hardware needs and software needs.

a) Hardware Requirements

The hardware used in this research can be seen in table 1.

Table 1. Hardware Requirements

Name	Type	Size
Motherboard	MSI H370	-
Processor	Intel i5-8400	2.80 GHz
Memory	DDR4	16 GB
Hard disk	SSD	256 GB

Based on Table 1, the hardware used in this research includes an MSI H370 motherboard, an Intel i5-8400 processor, 16 GB DDR4 RAM memory, and a 256 GB SSD hard disk.

b) Software Requirements

The software used in this research can be seen in Table 2.

Table 2. Software requirements.

Name	Information
Ubuntu Operating System	Operating system that is one of the linux distros
Oracle VM VirtualBox	Software to create virtual machines
MySQL server	Software to manage databases
Orchestrator	Software to manage high availability
ProxySQL	Software to manage between server backend and server database
MobaXterm	Software for remote ssh to server
MySQL Workbench	Software to establish a connection to a database
Google Chrome	Software to open monitoring applications

Based on Table 2, the software used in this research uses Oracle VM VirtualBox, MySQL server, Orchestrator, ProxySQL, MobaXterm, MySQL Workbench and Google Chrome.

Design Results

The design results in this paper are divided into three, namely Entity Relationship Diagram (ERD), flowchart and network topology.

a) Entity Relationship Diagram (ERD)

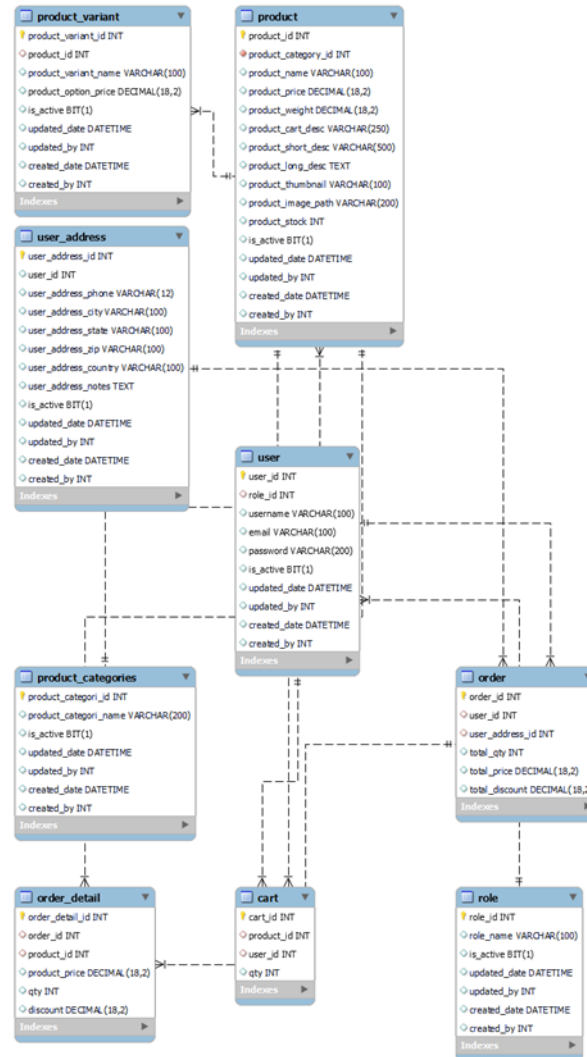


Figure 3. Results of entity relationship diagram (ERD)

Gambar 2 menjelaskan tentang desain entity relationship diagram (ERD) dari database yang digunakan sebagai pada penelitian ini, terdapat sembilan table dengan nama product, product variant, product categories, user, role, user address, order, order detail dan cart.

b) Flowchart

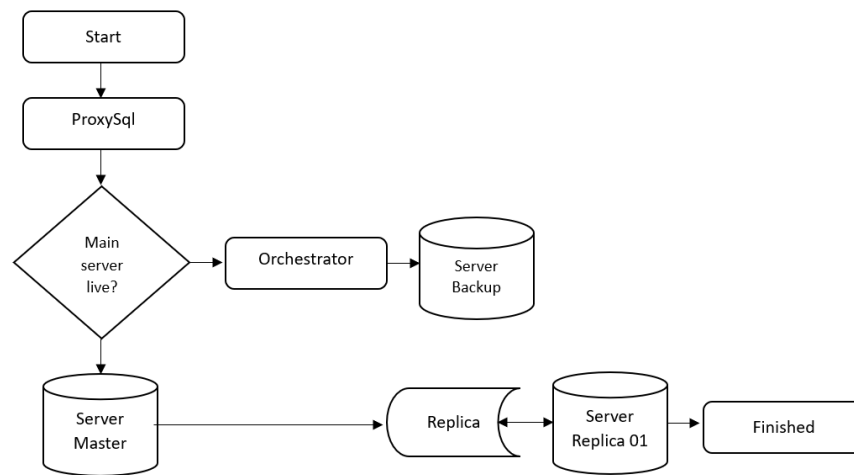


Figure 4. Flowchart results

B Figure 4 explains the flowchart used in this research. When connecting to ProxySQL there is a check whether the master server is up, if it is alive using the master server but if not, it is directed to the backup database through the Orchestrator application, and if there is an addition or change of data duplicated to the backup database or replica.

c) Network Topology

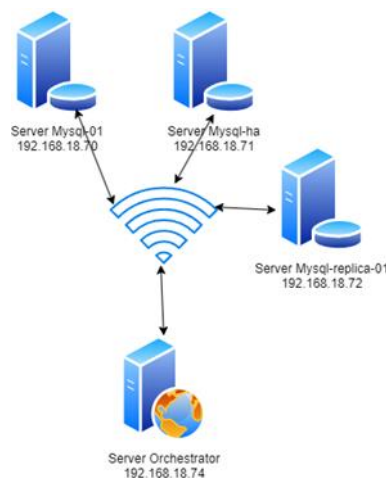


Figure 5. Network topology results

Figure 5 explains the network topology used as a simulation for this research. For more details, see table 3.

Table 3. virtual server networking specifications in this simulation

Server	OS	IP	Description
mysql-01	Ubuntu 22.04.1	192.168.18.70	Master
Mysql-ha	Ubuntu 22.04.1	192.168.18.71	Backup
replica-01	Ubuntu 22.04.1	192.168.18.72	Read server

Server	OS	IP	Description
Orchestrator	Ubuntu 22.04.1	192.168.18.74	Orchestrator and ProxySQL web servers

Based on Table 6, this research uses four virtual machines, one master database server with hostname Mysql-01 and IP 192.168.18.70, one backup database server with hostname Mysql-ha and IP 192.168.18.71, one replica database server with hostname replica-01 and IP 192.168.18.72 and finally one web server Orchestrator to install Orchestrator and ProxySQL with the hostname Orchestrator IP 192.168.18.74.

Implementation Results

a) Replication Configuration

```
mysql> show master status\G
***** 1. row *****
      File: mysql-bin.000004
      Position: 197
      Binlog_Do_DB:
      Binlog_Ignore_DB:
      Executed_Gtid_Set: f83b1238-1975-11ed-bde2-080027d33ca5:1-80
1 row in set (0,00 sec)
```

Figure 6. Status of Mysql-01 Master Server

Figure 6 describes a query to view the master state on the Mysql-01 server.

```
mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for source to send event
      Master_Host: 192.168.18.70
      Master_User: rep_user
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000014
      Read_Master_Log_Pos: 197
      Relay_Log_File: mysql-ha-relay-bin.000018
      Relay_Log_Pos: 413
      Relay_Master_Log_File: mysql-bin.000014
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
```

Figure 7. Mysql-ha server slave status

Figure 7 explains the query to see the slave status, if slave_io_state has been waiting for source to send event, then the slave has successfully connected to the master server.

```
mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for source to send event
      Master_Host: 192.168.18.70
      Master_User: rep_user
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000014
      Read_Master_Log_Pos: 197
      Relay_Log_File: mysql-ha-relay-bin.000018
      Relay_Log_Pos: 413
      Relay_Master_Log_File: mysql-bin.000014
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
```

Figure 8. Slave server replica-01 status

Figure 8 explains the query to see the status of the slave, if slave_io_state has been waiting for source to send event, then the slave has successfully connected to the master server.

b) Failover Configuration

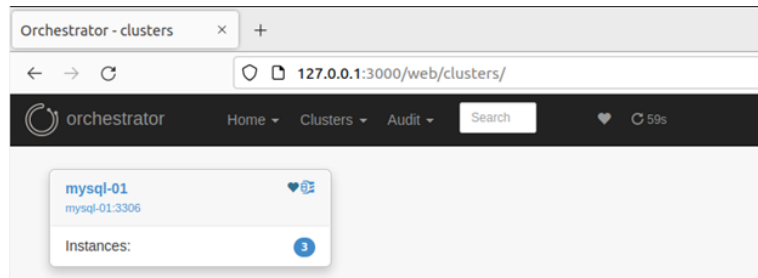


Figure 9. Dashboard menu on Orchestrator

Figure 9 explains the display of the dashboard menu, the content of which is the master server that has been registered in the discover menu.



Figure 10. Server topology on Orchestrator

Figure 10 describes the topology of a connected database server.

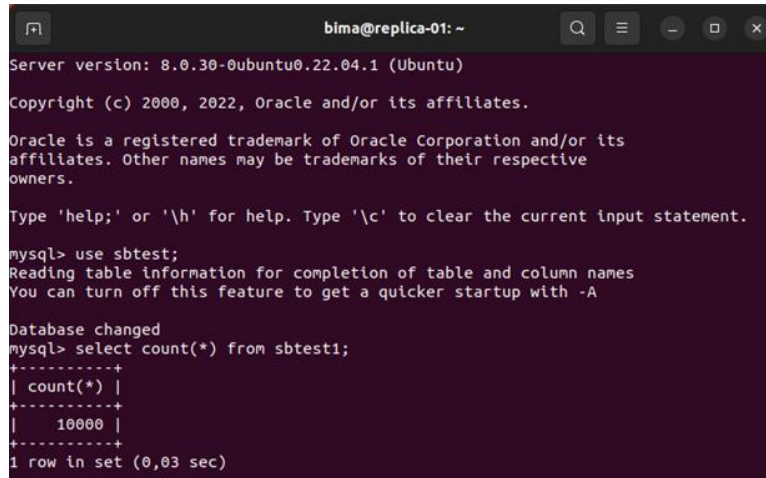
c) Replication Testing

Based on replication testing using the Sysbench application, the results of the data were successfully replicated to the Mysql-ha and replica-01 servers.

```
bima@mysql-ha: ~  
Server version: 8.0.30-0ubuntu0.22.04.1 (Ubuntu)  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> use sbtest;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed  
mysql> select count(*) from sbtest1;  
+-----+  
| count(*) |  
+-----+  
| 10000 |  
+-----+  
1 row in set (0.03 sec)
```

Figure 11. Replication results with Sysbench on a Mysql-ha server

Figure 11 describes the query to view the replication results on the sbtest1 table by calculating the amount of data, which is 10000, on the Mysql-ha server.

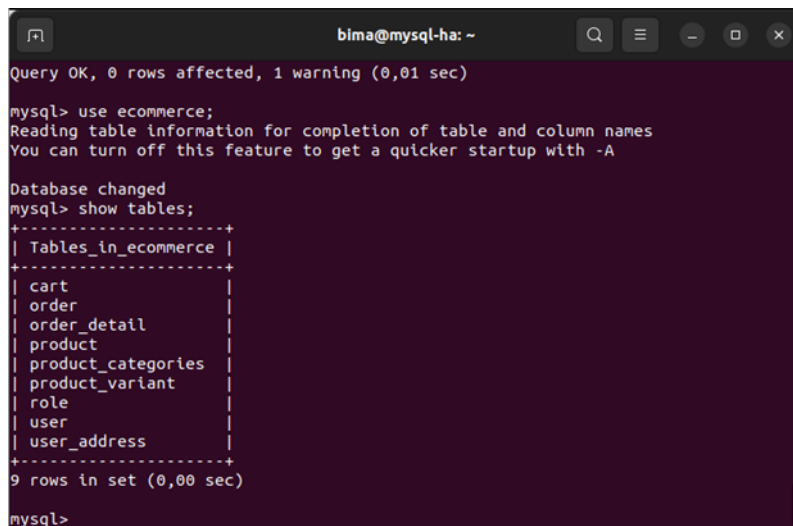


```
bima@replica-01: ~  
Server version: 8.0.30-0ubuntu0.22.04.1 (Ubuntu)  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> use sbtest;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> select count(*) from sbtest1;  
+-----+  
| count(*) |  
+-----+  
|    10000 |  
+-----+  
1 row in set (0,03 sec)
```

Figure 12. Replication results with Sysbench on the replica-01 server

Figure 12 describes the query to see the replication results on the sbtest database and check the data on the sbtest1 table by calculating the number of data, which is 10000, on the replica-01 server.

Based on the replication test with the e-commerce database restore, the data was successfully replicated to the Mysql-ha and replica-01 servers.



```
bima@mysql-ha: ~  
Query OK, 0 rows affected, 1 warning (0,01 sec)  
  
mysql> use ecommerce;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_ecommerce |  
+-----+  
| cart                 |  
| order               |  
| order_detail         |  
| product              |  
| product_categories   |  
| product_variant      |  
| role                 |  
| user                 |  
| user_address         |  
+-----+  
9 rows in set (0,00 sec)  
  
mysql>
```

Figure 13. Database restore replication results on Mysql-ha server

Figure 13 explains about the query to see the replication results on the e-commerce database, there are nine tables that have been successfully replicated on the Mysql-ha server.

```
bima@replica-01: ~  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> use ecommerce;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_ecommerce |  
+-----+  
| cart                  |  
| order                 |  
| order_detail          |  
| product               |  
| product_categories    |  
| product_variant       |  
| role                  |  
| user                  |  
| user_address          |  
+-----+  
9 rows in set (0.00 sec)  
  
mysql>
```

Figure 14. Database restore replication results on replica-01 serve

Figure 14 explains the query to view the replication results on the e-commerce database, there are nine tables successfully replicated on the replica-01 server.

d) Failover Testing

There are two results from testing the failover method, through the Orchestrator and querying ProxySQL.

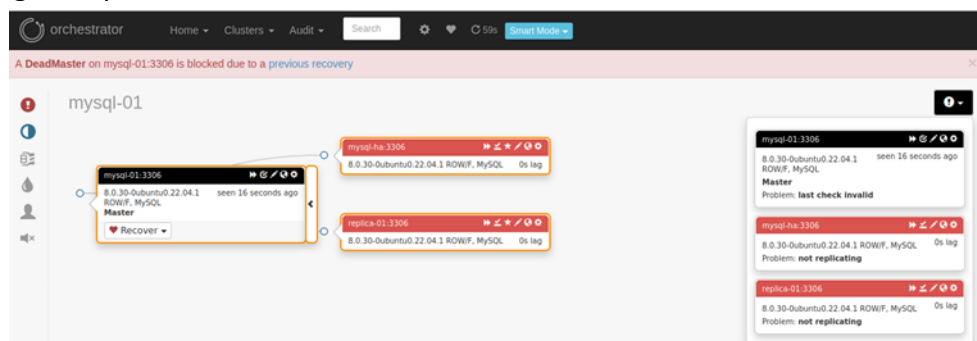


Figure 15. Orchestrator detects Mysql-01 server

Figure 15 describes the Orchestrator application detecting the Mysql-01 server is inaccessible.

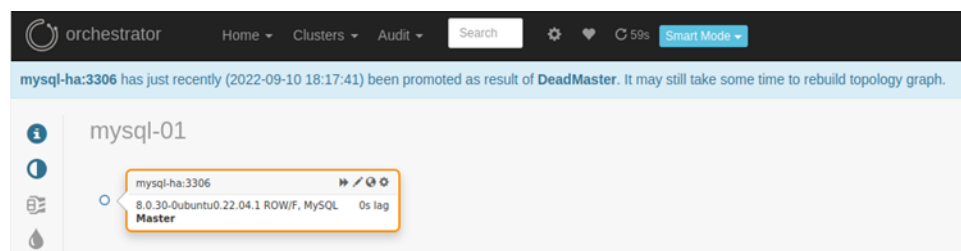


Figure 16. Orchestrator changing the master server

Figure 16 explains that the Orchestrator application successfully changed the master server to a Mysql-ha server.

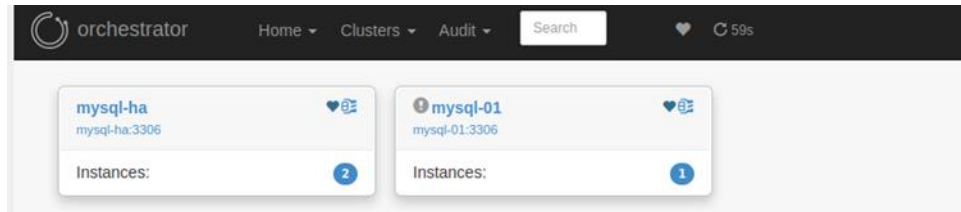


Figure 17. Orchestrator dashboard at failover

Figure 17 explains the dashboard on the Orchestrator application when a failover occurs, a new cluster is created with the Mysql-ha master.

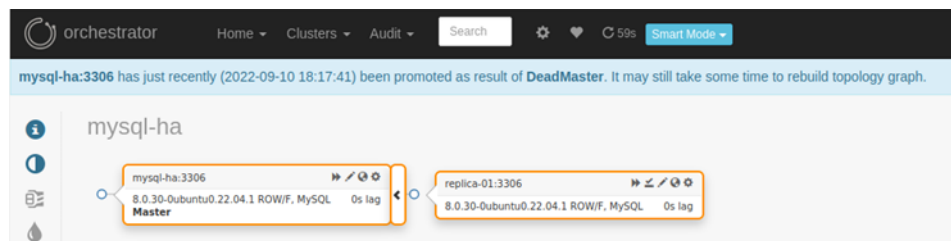


Figure 18. Topology of Mysql-ha server on Orchestrator

Figure 18 explains the topology on the Mysql-ha server when it is made the master.

Table 4. Failover Test Results

Description	Recovery Time (Seconds)
Test 1	2
Test 2	1
Test 3	1

Based on Table 4, the test was carried out three times, the first test took two seconds to recover, the second test took one second and the third test took one second.

To see the results of this test requires the MySQL workbench application to establish a connection into ProxySQL.

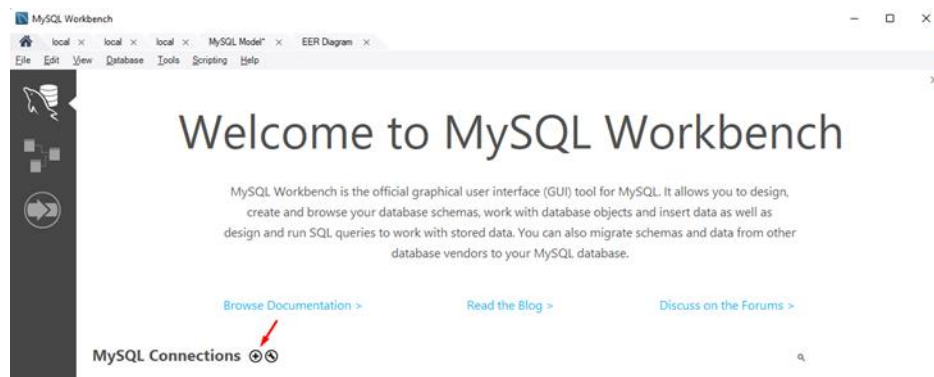


Figure 19. Adding a connection to the MySQL workbench

Figure 19 describes the location to add a connection to the MySQL workbench.

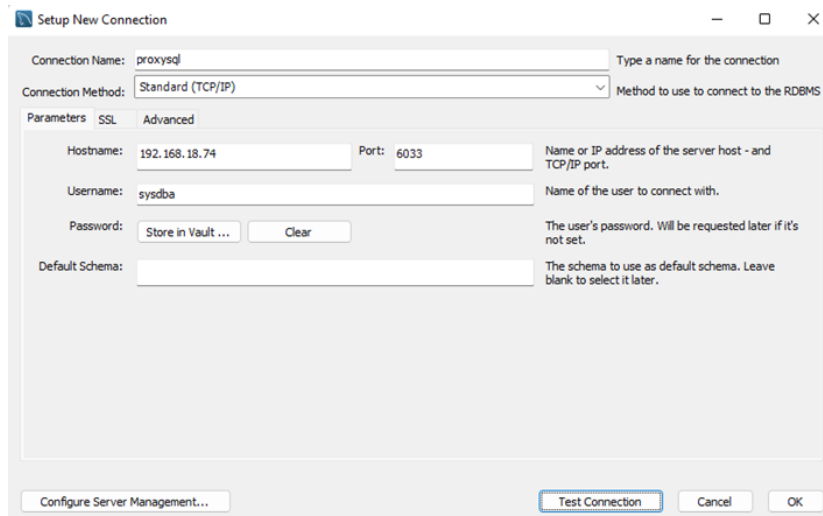


Figure 20. Connection configuration on MySQL workbench

Figure 20 describes the connection configuration with the name ProxySQL, hostname 192.168.18.74, port 6033, sysdba username and password DB@DMIN.

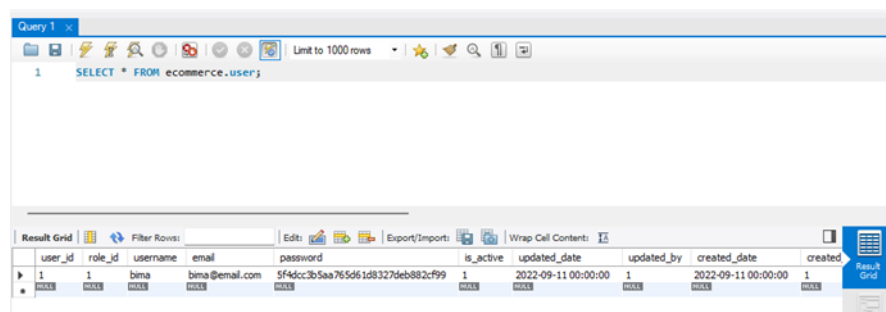


Figure 21. Query results on ProxySQL

Figure 21 explains how the query uses a ProxySQL connection and the result can still be selected into the user table, even if the Mysql-01 server is down.

Previous research by (Aprilliandi, 2019) examined the use of N-clustering on high availability web servers with load balancing and failover. Their results show that the use of this technology is able to reduce traffic load and increase server service availability. This is consistent with the findings of this research, where the implementation of replication and failover is also able to evenly distribute the workload and maintain system availability in the event of server failure (Ardiansyah et al., 2017). In addition, the failover test results in this research show a very fast recovery time (only one to two seconds), which supports the claim of increased efficiency of e-commerce systems.

In addition, research by (Saleh, 2021) who designed a web server and database infrastructure using mirror replication and failover clustering methods also provided similar

results. They found that these technologies are very effective in replacing failed servers without stopping user access to web services. This research supports the findings of our research, which also shows the successful replication and redirection of the master server when a failure occurs through Orchestrator, ensuring system availability for users.

Research by (Sofyan & Kusuma, 2022) on improving server availability in web-based applications using load balancing and failover also found that these methods can improve system performance and reduce downtime. This research is in line with our results, where the failover process runs smoothly and data is replicated successfully to the backup server.

In this research, failover and load balancing are proven to provide an efficient solution to improve service availability in MySQL-based e-commerce systems. The replication method ensures that data is always available even in the event of a main server failure, which is reinforced by empirical evidence from previous studies (Bahri et al., 2023).

CONCLUSION

The conclusion of the replication and failover method testing on the e-commerce database system shows success in the data replication process and server workload sharing. High availability testing proved that the data created on the three servers was identical, while read and write load sharing was effective with good replication support. In addition, the failover method was also successful, where the MySQL-01 server was successfully replaced by the MySQL-ha server after MySQL was shut down, demonstrating the effectiveness of the failover mechanism in maintaining system operational continuity. Future research can further develop this replication and failover method by examining its effectiveness on a larger scale, for example on systems with a larger number of servers or higher data volumes. In addition, research can expand the focus to the security aspects of replication and failover systems, as well as the integration of new technologies such as blockchain to improve data integrity and auditability in e-commerce systems. The use of containerization technologies such as Docker and Kubernetes in database replication management can also be an interesting research area to maximize efficiency in automatic failover scenarios and more dynamic management of server resources.

REFERENCES

- Antonelli, D., Cascella, R., Schiano, A., Perrone, G., & Romano, S. Pietro. (2024). "Dirclustering": a semantic clustering approach to optimize website structure discovery during penetration testing. *Journal of Computer Virology and Hacking Techniques*, 1–13.
- Aprilliandi, R. A. (2019). *Perancangan dan Implementasi Load Balancing Web Server Menggunakan Haproxy (High Availability Proxy): Studi Kasus di SMK Telekomunikasi Tunas Harapan Kab. Semarang*. Program Studi Teknik Informatika FTI-UKSW.
- Ardiansyah, S., Nur, J., & Mukmin, M. (2017). Rancang Bangun Load balancing Pada Database Cluster Menggunakan Haproxy. *Jurnal Informatika*, 6(2).

- Bahri, R. A. N., Ruuhwan, R., & Hartono, R. (2023). Pengembangan IP Failover & Replikasi Database Menggunakan Heartbeat Pada Komputer Server. *Informatics and Digital Expert (INDEX)*, 5(2), 51–59.
- Faqih, A. S., & Wahyudi, A. D. (2022). Rancang Bangun Sistem Informasi Penjualan Berbasis Web (Studi Kasus: Matchmaker). *Jurnal Teknologi Dan Sistem Informasi*, 3(2).
- Greengard, S. (2021). *The internet of things*. MIT press.
- Hartman, A., Kador, J., & Sifonis, J. G. (2001). *Net Ready: Strategies for Success in the E-economy*. McGraw-Hill, Inc.
- Kahanwal, D. B., & Singh, D. T. P. (2013). The distributed computing paradigms: P2P, grid, cluster, cloud, and jungle. *ArXiv Preprint ArXiv:1311.3070*.
- Nurdian, G. (2022). *Data E-commerce Indonesia 2022 (2 Tahun Pandemi)*. Grahanurdian.Com. <https://grahanurdian.com/data-e-commerce-indonesia-2022/>
- Saleh, S. B. (2021). *Perancangan Infrastruktur Web Server Dan Database Menggunakan Metode Replication Mirror Dan Failover Clustering*.
- Sofyan, A. R., & Kusuma, S. D. Y. (2022). Implementasi load balancing web server menggunakan Haproxy pada virtual server Direktorat SMK Kemendikbudristek. *Jurnal Pendidikan Tambusai*, 6(2), 9669–9682.
- Subekti, Z. M., Subandri, S., & Rakasiwi, G. (2019). Perancangan Infrastruktur Web Server dan Database Menggunakan Metode Replication Mirror dan Failover Clustering. *Jurnal Cendikia*, 18(1), 359–370.
- Sudrajat, A. (2020). Pajak E-Commerce, Pemecahan dan Solusinya. *Jurnal Pajak Vokasi (JUPASI)*, 2(1), 22–36.
- Sumarna, S., Nurdin, H., & Handono, F. W. (2019). Perancangan N-Clustering High Availability Web Server Dengan Load Balancing Dan Failover. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, 4(2), 149–154.
- Supendar, H., & Handrianto, Y. (2019). Teknik Availability Manajemen Server Berbasis Clustering. *Bina Insani ICT Journal*, 6(1), 1–10.
- Syahrir, M. (2023). *Metode Load Balancing Haproxy pada OpenNebula*. Politeknik Negeri ujung Pandang.

Copyright holder:

Bima Prasetya Adi Pratama, Bambang Gunawan Hardianto, Dharmayanti (2024)

First publication right:

Asian Journal of Engineering, Social and Health (AJESH)

This article is licensed under:

