Enhancing Business Continuity: Strategies and Innovations in Database Resilience and Failure Testing

Sriramaraju Sagi

NetApp

Abstract: ***The paper discusses the importance of ensuring database resilience and conducting failure tests in business environments. It emphasizes the need for databases to withstand disruptions and ensure company operations. Through experiments and analysis, the article highlights how specific database systems can remain resilient, to failures without affecting network or storage traffic. It stresses the significance of integrating resilience into database infrastructure design as a strategic business choice. The study underlines the importance of improving resilience practices in alignment with technologies to uphold reliable and efficient database systems. The research outcomes offer insights and practical guidance for companies aiming to enhance their database resilience for data access and business functions in a data driven world. It emphasizes the importance of updating and adjusting database infrastructure to mitigate risks and maintain operations. By giving priority to resilience, in database design organizations can enhance protection against disruptions while ensuring high performance levels and reliability are maintained.***

Keywords: Resilience, Disaster Recovery, Database, Testing

1. Introduction

Resilience, in databases refers to their ability to quickly and effectively bounce back from failures or interruptions ensuring access to data for businesses. To achieve a level of resilience organizations, need to implement strategies and practices that align with their specific needs and goals. These include backup and recovery mechanisms as customized disaster recovery plans. The purpose of these strategies is to minimize system downtime prevent data loss and ensure the availability and accuracy of business information. Having databases is crucial for enterprises as it reduces periods of inactivity guaranteeing access to valuable data. This capability enables organizations to maintain their operations when faced with events like system failures or natural disasters. By implementing backup and recovery systems businesses can safeguard themselves against data breaches, cyberattacks or human errors that might result in permanent data loss. The ability to swiftly recover from disruptions plays a role in preserving the accessibility and reliability of corporate information. Resilient databases ensure the availability and accuracy of data thereby facilitating business processes while mitigating potential financial losses or damage, to reputation. By integrating disaster recovery plans into their database strategies organizations can effectively address events.

The plans contain instructions, for restoring databases minimizing disruptions to operations and speeding up the return to normalcy. By investing in measures that improve the resilience of databases organizations demonstrate their commitment to safeguarding data assets and customer information. This also reinforces compliance with industry regulations. Builds trust among stakeholders such as clients, partners and regulators. It showcases a dedication to maintaining operations and reducing downtime in the face of disasters or system failures.

Regularly conducting failure tests is crucial for ensuring the resilience of databases. These tests simulate events and evaluate the effectiveness of an organization's strategies and plans. By identifying and addressing weaknesses or vulnerabilities through these tests' organizations can enhance their disaster recovery capabilities. Additionally, failure tests provide insights into areas that can be improved upon continuously allowing organizations to progressively strengthen their measures for database resilience.

In the healthcare industry it is essential to perform failure tests in order to uphold database resilience. These tests enable the replication of system failures. Assess how well existing strategies and plans work. Proactively identifying weaknesses or vulnerabilities through failure testing empowers healthcare organizations to enhance their disaster recovery capabilities. Continuous improvement based on insights gained from failure tests gradually strengthens database resilience measures, in the healthcare sector.

These findings can be applied to improve and update disaster recovery plans ensuring they are up, to date and aligned with the evolving needs of the healthcare industry. Additionally conducting failure tests provides healthcare organizations an opportunity to train their staff on responses, in case of system failures thereby strengthening their recovery capabilities.

2. Trends and Evolution of Database Resiliency and Failure Testing

Lately there has been a growing trend, towards adopting proactive approaches to ensure the durability and test the failure of databases. Companies are realizing the importance of testing and validating their database systems to ensure their ability to withstand failures. The increasing complexity and size of databases along with the frequency of cyber risks and data breaches are driving this change in mindset. As a result organizations are dedicating resources to acquire tools and technologies that enable them to simulate failure scenarios like system crashes, network outages or hardware failures. These assessments help companies evaluate the effectiveness of their resilience measures and identify any weaknesses or areas that need improvement. Failure testing, also known as chaos engineering involves causing failures in a controlled environment to uncover vulnerabilities and enhance a systems recovery capability. This approach is crucial for ensuring that databases can withstand challenges such as hardware failures, software glitches or external vulnerabilities. By addressing vulnerabilities through simulating real world failure situations organizations can boost their confidence in the reliability of their databases.

Database systems play a role, in supporting businesses by storing, managing and retrieving critical data.

Maintaining the longevity and reliability of these databases is crucial, for businesses as any downtime or data loss can have consequences. Lately there has been a growing emphasis, on adopting methods and strategies to enhance the resilience of databases.

2.1 Trends, in Database Resiliency

Microservices Design; Nowadays databases are adopting a microservices design approach, which involves dividing programs into services. This architectural style enhances the resilience of the database by containing failures within components thereby preventing disruptions that could affect the system. For example, Netflix relies on microservices to ensure streaming services in case of breakdowns.

Multi Cloud and Hybrid Cloud Techniques; Enterprises are now employing multi hybrid cloud strategies to distribute data across data centers and cloud providers. This approach boosts resiliency by minimizing the chances of a point of failure. Leading cloud service providers like Amazon Web Services (AWS) Microsoft Azure and Google Cloud Platform (GCP) offer flexible cloud solutions to support this emerging trend.

Data. Sharding Solutions; Ensuring data availability and resilience is increasingly achieved through data replication and sharding techniques. Organizations can mitigate the impact of hardware failures by duplicating data across servers located in locations and splitting information into smaller fragments. Prominent examples include MongoDBs capabilities and MySQLs replication mechanisms.

Automated Failover and Recovery; Automation plays a role in maintaining database resilience. Automated failover solutions swiftly transition, to backup systems when primary systems fail thus minimizing downtime significantly. Notable instances of automated mechanisms are Amazon RDS and Microsoft SQL Server Always On Availability Groups.

The use of containerization and orchestration has led to a revolution, in how databasesre deployed and managed. Technologies, like Docker and Kubernetes have played a role in this transformation. With container orchestration the distribution and scaling of database instances are efficiently handled based on the demand ensuring availability and scalability. CockroachDB, which is a distributed SQL database is designed to perform in containerized systems.

2.2 Recent Advancements in Database Evolution

Recent advancements, in the evolution of databases have brought about changes in how intricate data linkagesre managed. Graph databases like Neo4j and Amazon Neptune have gained increasing importance in applications such as fraud detection, social networks and recommendation systems.

When it comes to managing and analyzing time ordered data time series databases like InfluxDB and Prometheus play a role. They find usage across fields including IoT, financial analytics and system monitoring.

For those looking for auto scaling capabilities and a pay as you go pricing structure serverless databases such as AWS Aurora Serverless and Google Cloud Firestore prove to be options. These databases help minimize complexities and expenses.

Blockchain databases like Hyperledger Fabric and Ethereum are finding applications in sectors like supply chain management and banking. Their main purpose is to ensure record keeping that cannot be tampered with or changed.

In today’s dynamic business landscape database robustness is of importance. To achieve this robustness several tactics are being employed including microservices architecture, multi cloud deployments, data replication, automation and containerization. To ensure the effectiveness of these tactics against failures failure testing plays a role, in identifying vulnerabilities and strengthening systems.

Furthermore, the surge, in databases, like graph databases time series databases, serverless databases and blockchain databases showcases how organizations are seeking tailored and efficient data management solutions. To maintain an edge in the era it is vital for businesses to stay updated on the recent database usage trends as they play a crucial role in critical operational functions.

3. Literature Review

Extensive investigation has been conducted on the resilience and failover of databases, covering multiple facets of this crucial domain. Chiesa (2016) conducted a study on the ability of rapid failover to recover quickly from failures. The study proposed new strategies that are more effective than current methods. Fast Reroute (FRR) and other types of fast failover have been traditionally employed to recover from specific types of failures without activating the network control plane. Although the number of techniques in this collection is increasing, our understanding of the level of resilience to failures that this approach can offer is insufficient. His research focuses on conducting a methodical and algorithmic investigation into the ability of immediate failover to withstand and recover from various scenarios, considering factors such as packet marking and duplication. The research findings are utilized to develop novel strategies for immediate failover and demonstrate, through both theoretical analysis and experimental testing, that these strategies surpass current methods.

Gotter (2020) introduced a new technique for integrating data in NoSQL databases to improve their capacity to remain operational even when dispersed servers experience outages. These studies emphasize the significance of resilience and failover in database systems, as well as the possibility of using new methods to enhance their performance. The main objective of this paper is to introduce a new approach to combining data from different source systems. This approach includes a step - by - step process that allows users to integrate the data without needing any programming skills. The suggested technique relies on middleware as a crucial component to ensure data availability in the event of failures, using checkpointing. The experimental findings demonstrate that the suggested methodology yields effective outcomes, particularly when encountering problems in distributed NoSQL servers.

Soares (2011) suggests robust Asynchronous Commit (RAsC) as an enhancement to shared - nothing architecture, providing a more favorable balance between performance and cost for cloud - based robust database servers. These studies emphasize the importance of employing failover procedures and implementing data management strategies that are aware of resiliency in order to ensure the resilience of databases. Currently, many people depend on public cloud infrastructure - as - a - service to host their database servers. They achieve this by utilizing pooling and replication software to effectively manage massive virtual server clusters that do not share resources. However, it remains uncertain if this is still the optimal architectural decision, particularly when cloud infrastructure offers effortless virtual shared storage and charges clients based on the real utilization of disk space.

This study proposes a solution to the difficulty by introducing Resilient Asynchronous Commit (RAsC), which enhances a well - established shared - nothing approach. RAsC operates under the assumption that a significantly greater number of servers is necessary for scalability compared to resilience. Next, we evaluate this approach against alternative database server architectures using a specialized analytical model that emphasizes maximum throughput. Our analysis leads us to the conclusion that this proposal offers the most favorable balance between performance and cost, while also effectively resolving various fault scenarios.

Kaplan (2012) proposed a method to safeguard extensive in - memory data structures from failures by utilizing software erasure - correcting codes. The concept was implemented as a prototype library on the Cray XMT and can be easily adapted to other global shared memory architectures, such as the Cray XE. As the size of massively parallel processing (MPP) computers and their related applications increases, it becomes crucial to focus on resiliency in order to ensure that these programs can function for extended periods of time despite the anticipated rates of component failure. This study presents a method for safeguarding extensive read - mostly data structures stored in memory from several types of errors by utilizing software erasure - correcting codes. An initial version of a library for this approach was developed and tested on the Cray XMT supercomputer, and then used on a specific application as a demonstration. Furthermore, it can be easily adapted to other global shared memory architectures that fulfill specific criteria, such as the Cray XE.

In his 2018 publication, Wibowo conducted research on the architecture of a multi - region availability database system. The study focused on various aspects like regular backup, disaster recovery, performance, and the integration of management viewpoints to improve cost efficiency. The purpose of this research is to provide a method for implementing a database system that integrates both backup management and performance management considerations. The backup management should encompass the aspect of business continuity, while the performance management should take into account its scalability factor. The methodology employed consists of conducting literature reviews and field studies. The integration of two management views facilitates the consolidation of costs borne by both, resulting in improved cost efficiency.

This paper presents our investigation into the resilience and failure of the database, with the aim of verifying that the reference architecture can resist frequent failures resulting from unanticipated crashes, hardware malfunctions, or human errors. We performed various hardware (power disconnection), software (process termination), and operating system - specific failures to emulate real - world events under stressful conditions. During the process of destructive testing, we will showcase the distinctive failover capabilities of Cisco UCS components, NetApp Storage controllers link and Oracle RAC server node failure employed in this particular solution. The purpose of this destructive testing is to assess the system's response and resilience to these failures, ultimately ensuring a robust and reliable infrastructure.

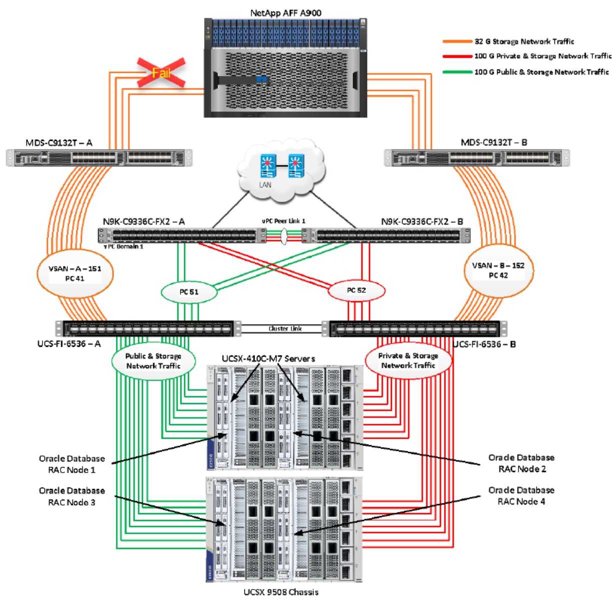

Figure 1: Storage control link failure example in reference architecture [6]

4. Results

We conducted tests to simulate failure scenarios that could arise from unexpected crashes or hardware issues. Failure scenario 1 specifically involves failures in the Chassis IFM linkages. To test this we detached one of the server port link cables, from the Cisco UCS Chassis. The goal was to replicate a situation where a server port link cable is disconnected from the chassis potentially causing network disruptions.

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Surprisingly we didn't observe any interruptions in network and storage traffic. The database continued to function despite multiple IFM link failures in the Chassis due to the Cisco UCS Port Channel Feature. The screenshot below showcases how the performance of the database workload from the storage array was unaffected during this failure, in the chassis IFM module.

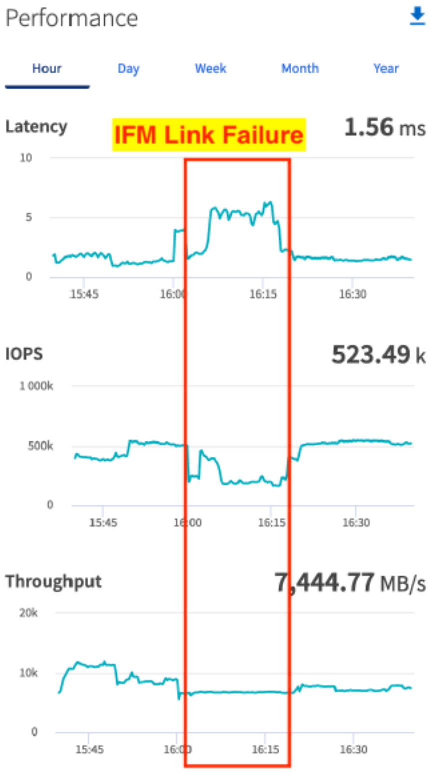

Figure 2: Performance of the database workload from the storage array [6]

We didn't experience any disruptions, in the network or storage traffic. The database continued to function despite several IFM link failures. We purposely deactivated the chassis IFM links for 15 20 minutes before reconnecting the failed links. Throughout this process we closely monitored the network traffic and database operations. Found no indications of any interruptions or disturbances. In Scenario 2 we conducted a test to simulate a failure in the storage controller's connection by disconnecting two of the Fibre Channel (FC) 32G links from the NetApp Array from one of the storage controllers. After conducting this test we observed that a storage connection failure did not cause any disruptions to Public and Storage Network Traffic. Once we reconnected the FC links to the storage controller both MDS Switch and Storage array connections were restored. As a result, all paths were reactivated at an operating system level through multipath configuration leading to optimal database performance being reinstated. Additionally, data transmissions returned to their speed and effectiveness.

In scenario 3 we initiated workload tests using Swing Bench on all four RAC nodes. During these tests we intentionally shut down one node, from the RAC cluster to assess system performance.

Losing one node, from the system did not have any impact on the performance of the database, in terms of IOPS, latency and throughput. We conducted a test to simulate a failure scenario. Found that this reference design does not have any single point of failure.

5. Conclusion

In this research we thoroughly explored the field of database resilience and failure testing, in business settings. Our insights and conclusions emphasize the importance of resilient database systems that can withstand various interruptions ensuring uninterrupted access to data and maintaining business continuity. We conducted tests to simulate potential failure scenarios. Despite encountering failures both the network and storage traffic remained unaffected with the database functioning normally according to the results. Additionally, even if there was a disruption in the connection of the storage controller it would not impact the public or storage network traffic. The Swing Bench workload test carried out on all four RAC nodes did not show any impact on database performance. Another scenario tested confirmed that this reference design lacks a point of failure. To sum up this study underscores the importance of integrating resilience into database infrastructure design.

To recapitulate our diverse research findings and practical experiments, provide guidance for organizations seeking to fortify their database systems, against interruptions. It is crucial to enhance resilience measures while incorporating technologies as they emerge in order to maintain strong and reliable database systems. The process of improvement is crucial not, from a technical perspective but also as a strategic business decision. It helps maintain stability and provides an edge to businesses, in a data driven world.

Acknowledgments
This article was supported by Tushar Patel, Cisco, Hardik, Vyas, Cisco and Bobby Oommen, NetApp – we sincerely thank them for their contribution and valuable input.

References

[1] The quest for resilient (static) forwarding tables, Marco Chiesa, Ilya Nikolaevskiy, Slobodan Mitrovic, Aurojit Panda, A. Gurtov, A. Madry, Michael Schapira, S. Shenker, IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications
[2] Enhancing High Availability for NoSQL Database Systems Using Failover Techniques, Priyanka Gotter, Kiranbir Kaur, Published 2020, Computer Science, Engineering
[3] Improving the scalability of cloud - based resilient database servers, Luís Soares, José Pereira, Published in IFIP International Conference… 6 June 2011Computer Science, Engineering
[4] Resilience to Various Failures for Read - mostly In - memory Data Structures, L. Kaplan, P. Briggs, Miles Ohlrich, W. Leslie, 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum
[5] Building Scalable and Resilient Database System to Mitigate Disaster and Performance Risks, Aryan Wibowo, Diana, Mohammad Subekti, Hendro, Published

2018, Environmental Science, Engineering, Computer Science

[6] FlexPod Datacenter with Oracle 21c RAC on Cisco UCS X - series M7 and NetAPP AFF900 with NVME/FC, Dec 2023, Cisco. com