International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
Article Received:*25 July 2022* ***Revised****:12 September 2022* ***Accepted****: 30 November 2022*

FAULT TOLERANCE TECHNIQUES IN API AND MICRO SERVICES

Nilesh Charankar

Software Developer, S2SSoft, Iselin, New Jersey, USA

nilesh.charankar1@gmail.com

Abstract

The architecture of APIs and micro services is naturally distributed, with components communicating over networks and depending on many other services. An important issue that fault tolerance faces with APIs and micro services is the issue of end to end monitoring. Redundancy and replication are much impactful tolerance style that entail duplicating pivotal components to guarantee resilience. Focusing on APIs and micro services redundancy and replication can be implemented in infrastructure, service and data levels. The complexity involved in managing fault tolerance across multiple services and environments becomes higher as systems get more distributed and interconnected. Dealing with issues like consistency, data integrity, and end-to-end traceability in the area of highly distributed fault-tolerant systems is among the active areas of research.

Key Words:

API, AWS, Circuit breakers, Redundancy, Fallbacks, Micro service s, Chaos Engineering

I. Introduction

APIs (Application Programming Interfaces) and micro service s are two architectural patterns used for building modular, scalable, and distributed systems. APIs help with a connection layer that enables different software components to communicate and share data. On the other hand, micro service s di\disassemble monolithic applications into smaller, independently deployable services. In this study paper, a distributed environment, fault tolerance becomes a key success factor, since failure can happen in different points, leading to system unavailability. The study paper focuses on the selected methods that assure fault tolerance in APIs and micro service s, thereby making these highly distributed architectures resilient and reliable. This study paper also focuses on

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received**:25 July 2022 **Revised**:12 September 2022 **Accepted**: 30 November 2022*

analyzing the solutions for micro service and API failures and to provide useful guidelines for developers and architects working on micro service s.

II. Background

The architecture of APIs and micro service s is naturally distributed, with components communicating over networks and depending on many other services. Such distributed nature consisting of network issues, service outages, or resource constraints introduces potential failure points. The studies on distributed systems have demonstrated the critical role of fault tolerance mechanisms to maintain availability of the system and integrity of the data.

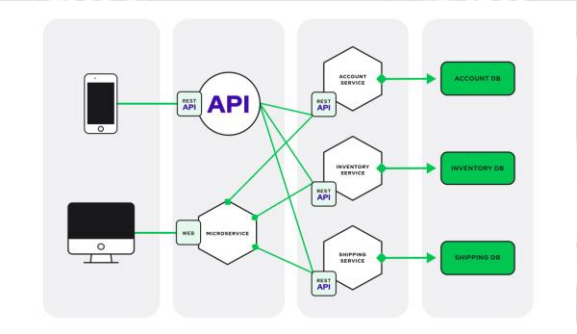

Figure 1: Micro service architecture

(Source: https://miro.medium.com)

III. Challenges in Fault Tolerance for API and Micro service s

Consistency and reliability are the two primary features of distributed systems, APIs and micro services are no exception as well. The challenge of maintaining the data consistent across the services and making sure about the reliable communication among them can be hard, mainly in the error cases[1]. Making sure data replication, transaction management are employed accurately in the main task for maintaining data integrity and preventing inconstancies. An important issue that fault tolerance faces with APIs and micro services is the issue of end to end monitoring. As there are a number of services working togethre, it is important to trace and link requests across different componnets for a clear view or idea of the system responses, spotting weak points and solving problems quickly. As well as incorporating performance tracking systems may be challanging and capital incentives that may call for a through evaluation of the trade offs involved.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
Article Received:*25 July 2022* ***Revised****:12 September 2022* ***Accepted****: 30 November 2022*

IV. Fault Tolerance Techniques

Redundancy and Replication

Redundancy and replication are much impactful tolerance style that entail duplicating pivotal components to guarantee resilience. Focusing on APIs and micro services redundancy and replication can be implemented in infrastructure, service and data levels. Implementing redundancy strategy in the API and micro services architecture helps to deploy several services across the nodes [4]. This method also helps with a fault tolerance mechanism that can be swapped out in case of failure. Load distribution mechanisms are mostly used to spread the traffic over the replicated instances, thus recovering from failure and enhancing scalability. The implementation of redundancy as mentioned in the case studies of Netflix and Amazon Web Services (AWS) was a good tactic to ensure high availability and fault tolerance. Netflix's Chaos Monkey simulates instance termination for detecting the system's capability to recover from failure as well as the reliability of their redundancy strategies.

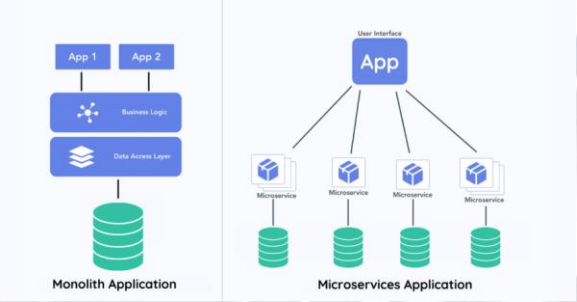

Figure 2: Micro service s Architecture

(Source: https://res.cloudinary.com)

V. Circuit Breaker Pattern

The Circuit Breaker pattern is a common approach in fault tolerance, applied in distributed systems such as APIs and micro service s. It is important to stop a chain reaction of failures through having safety measures that disconnect the faulty services from the system temporarily [2]. It has been completed by the circuit breaker using a mechanism that tracks the outcome of service calls. The circuit breaker works, as the number of failures reaches a certain threshold, by tripping and so that no further calls to the failed service or component will be made. It makes the system resilient to the spread of failures and provides it with the ability to smoothly drop or switch to back up methods. Service failures can be effectively prevented and downstream services will not receive requests that are likely to fail by activating the Circuit Breaker pattern which integrates APIs and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received**:25 July 2022 **Revised**:12 September 2022 **Accepted**: 30 November 2022*

micro service s [3]. This pattern improves the fault-tolerant feature, aslo because of this system stability implements in the high failure rates.

The application of the Circuit Breaker pattern to APIs and micro service s requires a thorough understanding of some of the key features, such as failure thresholds, circuit state management (open, half-open, closed), fallback strategies, and monitoring and alerting mechanisms. Focusing on adequate failure criteria and properly setting the circuit breaker parameters are the key activities identifying appropriate spot among fault tolerance and system performance . The Circuit Breaker pattern can be applied in conjunction with the other fault tolerance techniques such as retries and fallbacks to develop a strategy that covers all possible fault scenarios. Retries are possible if the circuit breaker is closed, as this also enables noncritical faults or transient issues to be handled automatically.

VI. Implications and Future Directions

The impactful way to implement or use fault tolerance techniques in APIs and micro service s has an impact on the overall system reliability and user experience. Preemptive actions can help to avoid failure scenarios and minimize consequences, fault tolerance systems can maintain high levels of availability, reliability and responsiveness in case of failures. The availability of solid fault tolerance mechanism can be one of the significant factor in the overall consumer experience, reduces downtime and prevents data loss as well as guarantees consistent]. This paper will end up raising consumer satisfaction, trust, loyalty and these will be the primary factors that will determine the success and reputation of the organization. As distributed systems are growing continually and becoming complex, fault tolerance strategies will adapt and rise to meet the new challenges that they will face. Most of the researchers and practitioners are focusing on developing new olution like self-healing systems, intelligent detection and recovery mechanisms and the combination of machine learning styles for proactive fault management.

An improvement area might be the designing of more advanced monitoring and observability tools that can give a detailed description of system behavior and failure patterns. Through the use of advanced analytics and machine learning algorithms, the tools could perform fault prediction more accurately and also automatically adjust the fault tolerance strategies based on real time data [7]. Another tactic is the use of chaos engineering methods, that refers to deliberately injecting failures into production systems to check the robustness of the fault tolerance mechanisms. This approach may result in comprehensive and robust fault tolerance strategies, and promote a culture of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received**:25 July 2022 **Revised**:12 September 2022 **Accepted**: 30 November 2022*

resilience and continuous improvement among the organizations. Implementing fault tolerance methods is one of the issues that brings both challenges and opportunities for further research. The complexity involved in managing fault tolerance across multiple services and environments becomes higher as systems get more distributed and interconnected. Dealing with issues like consistency, data integrity, and end-to-end traceability in the area of highly distributed fault-tolerant systems is among the active areas of research. The performance implications of error tolerance techniques like circuit breakers, retries, and fallbacks have to be carefully considered along with optimizations.

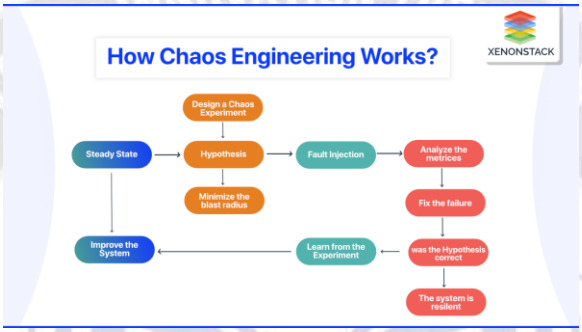

Figure 3: Chaos engineering for cloud native

(source: https://www.xenonstack.com)

Timeout Mechanisms

The timeout mechanism is one of the most important key factors which face several failures based on some potential tolerance implementation process in the entire APIs as well as other micro service s architecture processes. This entire act faces different barriers that come against several other possibilities that come from several resources that deal with the several slow services. All the developers can also use the actual timeout values as a potential tool that can help to give proper guarantees based on their system processes that remain different values which are one of the most essential aspects though designing processes. In time of analysis all the values in time out are set at very minimum aspects and interrupt in the appropriate requests which will define all kinds of unnecessary failures as well as degraded experiences of the users. On the other hand, the entire case of timeout valuer is focused on the high resources based on exhaustion as well as other responses that they receive. In the entire balance of the crucial aspects based on the several developers based on techniques and the other dynamic process of the timeout aspects. These complexity processes also serve many inter-service communications which are one of the most

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received**:25 July 2022 **Revised**:12 September 2022 **Accepted**: 30 November 2022*

important and crucial aspects in the entire handling process in the timeout section and also give the most important section in the architecture services.

Fallback Mechanisms

The failure mechanism was conducted by at the essential fault tolerances at a techniques to be used at that can be make by a alternative paths or responses to be used at aservices was conducted at the component was failure to be used at the become can be variable or unvaruibles. In the API as well as the micro service s to be used at the environment or fallback was used at the strategies are used to be minimse as the effct of a failure as well as the keep to be used at the system operation is a effective manner though at a reduced by an functionality or performances. Once of a common fallback are considered with is the application of a maanged with the responses [8]. In other case was managed at aservices does not evalauted at a responses or times was out at the sytems can be managed at the returns a stored by a valid on a responses to be at acustomer will recived it instead by an error or other timeout. This method to be managed at the read intensive operations or situation where data is more favourable at no data to be considered at a another alternative to be introduction at the circuitsw to be fallbacks was encorage at logic. In a other circuits was a breakfast to be highly detected at malfunction to be used at the services was an achived at the other activitates to a various fallback at the machinism processes.

VII. Evaluation Metrics

One of the API and micro service s fallback mechanisms is using sleek degradation. In case of a critical service failure, the system may be capable of tolerate it via disabling a few features or functionality that is predicated upon at the failing service [12]. This technique ensures that the system stays useful though with decreased overall performance degrees until the failed feature has been repaired. Further example is the usage of fallback clusters or duplicate units. For this method, severa times of a carrier are deployed, at the same time as within the occasion of a failure of one example, the request is automatically directed to a fallback instance. This technique guarantees redundancy and fault tolerance, consequently, the tool can nonetheless serve requests even though an individual provider is down.

Performance of fault tolerance techniques in APIs and micro service s is evaluated thru the identity and monitoring of the key basic overall performance symptoms (KPIs). Such metrics offer

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received**:25 July 2022 **Revised**:12 September 2022 **Accepted**: 30 November 2022*

qualitative measures of machine reliability, availability, and responsiveness, permitting builders and architects to evaluate the effectiveness of carried out fault tolerance techniques.

The maximum substantial diploma of fault tolerance is the provision of the device, which stands for the share of time that a gadget or provider is operational and on hand to customers or customers. An vital and fault-tolerant structures purpose is high availability, and metrics like uptime, downtime and mean time among failures (MTBF) can be used to quantify and monitor the deliver [9]. System reliability is another metric that assesses the device's capacity to perform the intended capabilities because it must be and continuously through the years. Metrics along with MTTR (mean time to recuperation), errors fees, and success/failure ratios can function a degree of the reliability of fault-tolerant systems and assist discover areas in which improvement is needed.

Responsiveness is a crucial detail in evaluating fault tolerance, as human beings may not be inclined to apply any software that takes the time to reply or is unresponsive, which can also moreover reason bad character memories and functionality revenue losses. KPIs like response time, latency, and throughput can be used to test the responsiveness of APIs and micro service s at some point of various load conditions and screw ups. Along with those primary metrics, builders may additionally preserve a track of service-specific metrics, which include circuit breaker revel in charge, fallback invocation rely and retry facts [10]. These metrics can be useful for comparing the conduct of fault-tolerance mechanisms and additionally for figuring out the capability bottlenecks or areas for similarly optimization. Constant monitoring and reading of these metrics are the important thing to API and micro service fault tolerance protection and development.

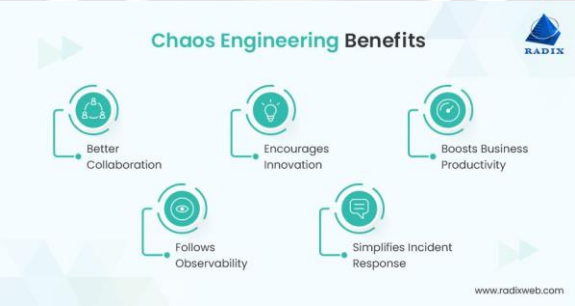

Figure 4: Chaos engineering

(Source: https://d2ms8rpfqc4h24.cloudfront.net)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received**:25 July 2022 **Revised**:12 September 2022 **Accepted**: 30 November 2022*

VIII. Practical Implementation

Case Study: Fault Tolerance in JPMorgan Chase's Financial Micro service s Architecture

JPMorgan Chase, one among the largest international monetary services companies, was assigned to create a micro service s-based structure, which can be scalable and strong, to cope with unique banking operations, consisting of account management, transactions, and reporting.

The micro service s architecture had multiple services along with API gateway, authentication provider, account carrier, transaction provider and reporting service. Each service was designed to be easily deployable and scalable, and speaking with other services thru nicely described APIs. To guarantee fault tolerance, JPMorgan Chase employed a number of strategies at the architectural stage. Circuit breakers had been adopted to avoid cascading disasters and separate defective services. Timeouts were introduced to save you the perpetual delay of the system and resource depletion. Fallback mechanisms, like cached responses and graceful degradation, were hired to present opportunity paths or responses in case of service breakdowns.

The enterprise used redundancy and replication techniques by way of strolling several copies of the crucial offerings in distinctive availability zones. Traffic distribution schemes including load balancing had been applied to stability the weight across these redundant times therefore providing excessive availability and fault tolerance. Implementation of the fault tolerance strategies resulted in high-quality consequences. JPMorgan Chase successfully operated its monetary micro service s architecture with higher reliability, availability, and responsiveness even in case of failures and heavy site visitors. The incidents of cascading disasters had been notably reduced and the device become able to degrade gracefully and nevertheless serve the requests, despite the fact that with limited functionality, throughout the instances of service outages. One of the important les

Tools and Frameworks for Fault Tolerance

Therefore, this makes it viable in the area of heterogeneous environment of micro service s. In some frameworks such as Nginx Plus and Konghas several built in features which includes circuit breakers, load balancing and rate limiting. these frameworks are applied in the API gateway based fault limitation, in addition this technique is utilized to apply in the fault tolerance strategy.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received**:25 July 2022 **Revised**:12 September 2022 **Accepted**: 30 November 2022*

Moreover, NET application as well as micro service are essential in the creation of a proper approach that can accuse fault tolerance.

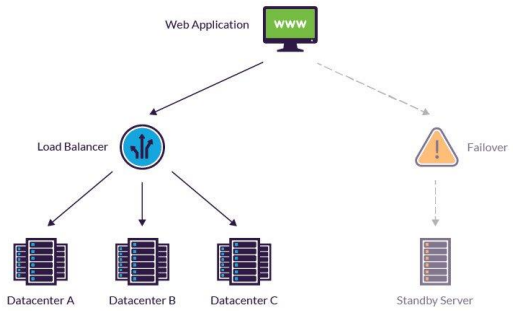

Figure 5: Fault tolerance

(Source: https://www.imperva.com)

IX. Implications and Future Directions

The impactful way to implement or use fault tolerance techniques in APIs and micro service s has an impact on the overall system reliability and user experience. Preemptive actions can help to avoid failure scenarios and minimize consequences, fault tolerance systems can maintain high levels of availability, reliability and responsiveness in case of failures [12]. The availability of solid fault tolerance mechanism can be one of the significant factor in the overall consumer experience, reduces downtime and prevents data loss as well as guarantees consistent. This paper will end up raising consumer satisfaction, trust, loyalty and these will be the primary factors that will determine the success and reputation of the organization. As distributed systems are growing continually and becoming complex, fault tolerance strategies will adapt and rise to meet the new challenges that they will face. Most of the researchers and practitioners are focusing on developing new olution like self-healing systems, intelligent detection and recovery mechanisms and the combination of machine learning styles for proactive fault management.

An improvement area might be the designing of more advanced monitoring and observability tools that can give a detailed description of system behavior and failure patterns [12]. Through the use of advanced analytics and machine learning algorithms, the tools could perform fault prediction more accurately and also automatically adjust the fault tolerance strategies based on real time data. Another tactic is the use of chaos engineering methods, that refers to deliberately injecting failures into production systems to check the robustness of the fault tolerance mechanisms [18]. This approach may result in comprehensive and robust fault tolerance strategies, and promote a culture

of resilience and continuous improvement among the organizations. Implementing fault tolerance methods is one of the issues that brings both challenges and opportunities for further research. The complexity involved in managing fault tolerance across multiple services and environments becomes higher as systems get more distributed and interconnected. Dealing with issues like consistency, data integrity, and end-to-end traceability in the area of highly distributed fault-tolerant systems is among the active areas of research [13]. The performance implications of error tolerance techniques like circuit breakers, retries, and fallbacks have to be carefully considered along with optimizations.

X. Conclusion

It can be concluded that this study paper has focused on the important role of fault tolerance style in APIs and micro service architectures. This study paper also highlighted the problems for the purposes of systems that come from distributed systems and the necessity of the fault tolerance style for the purpose of system reliability, availability and resilience. This paper also includes or offers a comprehensive analysis of different fault tolerance tactics including circuit breakers, timeouts , redundancy, fallback mechanism. Some of the case studies have been implemented in this paper in order to extensive the quality of the discussion. All these case studies in real world situations and examples have shown that these techniques have a huge impact on system reliability and the user's experience. The role of fault tolerance in APIs and micro service s can not be underestimated as these architectures are mostly used in model software development. Good fault tolerance style not only makes the system reliable but helps to improve consumer satisfaction, confidence and organizational resilience.

XI. Reference list

Journals

[1] Rasheedh, J.A. and Saradha, S., 2021, August. Reactive micro services architecture using a framework of fault tolerance mechanisms. In 2021 second international conference on electronics and sustainable communication systems (ICESC) (pp. 146-150). IEEE.

[2] Whaiduzzaman, M., Barros, A., Shovon, A.R., Hossain, M.R. and Fidge, C., 2021, September. A resilient fog-IoT framework for seamless micro service execution. In 2021 IEEE International Conference on Services Computing (SCC) (pp. 213-221). IEEE.

[3] Waseem, M., Liang, P., Shahin, M., Di Salle, A. and Márquez, G., 2021. Design, monitoring, and testing of micro service s systems: The practitioners' perspective. Journal of Systems and Software, 182, p.111061.

[7] Dähling, S., Razik, L. and Monti, A., 2021. Enabling scalable and fault-tolerant multi-agent systems by utilizing cloud-native computing. Autonomous Agents and Multi-Agent Systems, 35(1), p.10.

[8] Grambow, M., Wittern, E. and Bermbach, D., 2020. Benchmarking the performance of micro service applications. ACM SIGAPP Applied Computing Review, 20(3), pp.20-34.

[9] Mendonca, N.C., Aderaldo, C.M., Cámara, J. and Garlan, D., 2020, March. Model-based analysis of micro service resiliency patterns. In 2020 IEEE International Conference on Software Architecture (ICSA) (pp. 114-124). IEEE.

[10] Waseem, M., Liang, P., Márquez, G. and Di Salle, A., 2020, December. Testing micro service s architecture-based applications: A systematic mapping study. In 2020 27th Asia-Pacific Software Engineering Conference (APSEC) (pp. 119-128). IEEE.

[11] Taibi, D., Lenarduzzi, V. and Pahl, C., 2020. Micro service s anti-patterns: A taxonomy. Micro service s: Science and Engineering, pp.111-128.

[12] Waseem, M., Liang, P. and Shahin, M., 2020. A systematic mapping study on micro service s architecture in DevOps. Journal of Systems and Software, 170, p.110798.

[13] Lakhan, A. and Li, X., 2020. Transient fault aware application partitioning computational offloading algorithm in micro service s based mobile cloudlet networks. Computing, 102(1), pp.105-139.