Payment Routing Module using Kafka Streams

K V Sarath Kumar[1], Dr Anala M R[2]

[1,2]Department of Information Science and Engineering, R V College of Engineering, Bangalore, India

---***---

Abstract - *Online payment systems are gaining popularity in this digital age. Payment systems are necessary for the efficient operation of all financial systems in the globe. A payment system that is efficient, secure, and dependable lowers the cost of exchanging goods and services. Since a payment system has a significant impact on firms' revenue, improving its efficiency is crucial. A payment system consists of single or multiple payment gateways, which are associated with payment terminals. The payment routing module does the job of selecting the best route for a transaction. Here, a payment routing module is introduced which works on the performance and availability of the acquirer banks based on the previous data. An informed decision is taken and sent to the payment gateway to route the transaction to the mentioned acquirer.*

Key Words: Payment Gateway, Payment Routing, Payment System, Payment Transaction, Payment Switch, Payment Terminal, Payment Routing Module, Kafka Streams, Audit Data, Stream Processing, tumbling window, Merchant app, Acquirer bank, Machine Learning.

1. INTRODUCTION

A payment gateway is a piece of technology that merchants use to accept debit or credit card payments from customers. The payment gateway takes the decision of routing the transaction to a specific acquirer. The acquirer is the financial institution that maintains a merchant's account so that credit cards can be accepted. Each bank is associated with single or multiple acquirers. Fig 1 shows the important components of a payment transaction. The following are the fundamental elements of a typical payment flow:

i) Customer: A person who begins an online transaction . A customer buys a product from a merchant and does the payment for it.

ii) Merchant app: The platform through which the customer initiates a transaction. This is the application where a customer adds required items to their carts and does the payment activity.

iii) Payments app: A backend application which handles the payments initiated through the merchant app. The data related to the payment activity is sent from the merchant app to the payments app. The proposed routing module is added as a separate module into the payments app.

iv) Payment gateway: A payment gateway is a network that customers use to send money to merchants. The point-of-sale terminals used at the majority of physical stores are remarkably similar to payment gateways. Customers and businesses must collaborate while using a payment gateway in order to complete a transaction.

v) Acquirer Bank: A bank or financial institution that handles credit or debit card transactions on behalf of a merchant is referred to as an acquiring bank. The proposed routing module deals with the failures of the acquirer banks.

vi) Bank: After the acquirer banks process the transaction, the transaction amount is sent to the merchant account(bank). Generally, each bank is linked with a certain set of acquirer banks.

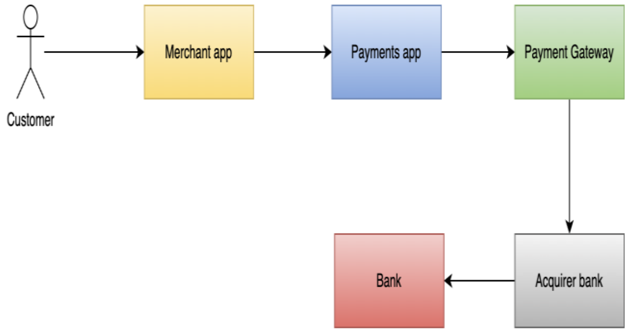

Fig -1: Flow of a payment transaction.

A payment transaction is either successful or a failure. Payment failures can be broadly divided into two categories: those related to customers and those not related to customers. Wrong card verification value, payment timeouts, and wrong one time password belong to the category of customer related failure. Non customer related failures generally happen at the gateways for one of the following reasons:

1. The Gateway can see a rapid drop in success rates due to being overloaded with more capacity than it can handle.

2. The acquirer bank or bank server system could fall down or require repair, which would result in the gateway completely failing.

The proposed solution deals with the acquirer bank's server system failure or unavailability.

If the decision taken by the payment gateway is not leading to successful transactions or the routing logic used by the payment gateway is not up to the mark then the proposed payment routing module can be introduced at the merchant side. This routing module involves stream processing, filtering, storing and querying the data.

The objective behind this work is to increase the efficiency of payment systems by introducing a payment routing module which works based on the previous data. An informed decision is taken and is sent to the payment gateway. In this way the probability of a transaction getting successful is increased.

2. RELATED WORKS

The subject of payment systems is a very intriguing subject of study and is highly useful. In this section, several relevant and well-known papers are studied to understand what the present systems offer and how they operate.

An architecture consisting of two main modules: a static module and a dynamic module. The static module retrieves the list of available terminals and filters out terminals that aren't relevant by applying business-based rules. The dynamic module receives the filtered input terminals and assigns a success chance to each terminal. This routing solution does real-time processing of millions of transactions that significantly raises the success percentage for payments. It shows how ML models can be integrated with existing architectures to improve business performance. For each ML model precision and ROC-AUC scores are computed and analyzed[1]. A pipeline that combines a static module and a dynamic module makes up the approach. To narrow down the list of potential terminals for a specific payment transaction, the static module uses rules and basic machine learning algorithms. The dynamic module forecasts each terminal's chance of success using specially created, dynamically updated characteristics. The author stated that the proposed solution increases the effectiveness of a payment system's whole lifespan. By raising success rates and thus their revenues, it has a significant impact on enterprises[2].An approach which makes use of reinforcement learning. The rewards and punishments for the algorithm are the success and failure of transactions. They are maximizing the expected rewards of a transaction over time based on these values.[3]

Kafka streams usage for complex activities[4]. The description of various inbuilt functionalities of kafka streams[5]. Comparison of stream processing frameworks like Apache Flink, Apache Kafka Streams, Apache Spark: Spark Streaming, 4.4 Apache Spark: Structured Streaming is done for different workloads. The results of each of the workloads has been summarized in a table per workload and for each additional decision factor. These decision factors influence the decision for a framework based on job requirements or pipeline complexity[6]. An architecture of the real time data warehouse which is capable of handling heavy workloads is proposed[8].

The proposed routing module system performs stream processing using kafka streams as it provides various inbuilt functionalities.

3. METHODOLOGY

In the proposed system the payment routing module gets the audit data which contains the transaction details. In payment routing module stream processing is carried out using kafka streams. Only the records related to the transactions are filtered out and for each minute the no of successful and failed transactions for each acquirer is calculated. These results are pushed into another data store. kafka streams provides a functionality called tumbling window using which one can process data in windows and these windows never overlap, which makes sure that a record will always belong to one and only one window. In the proposed system the window time is set for one minute. As the no of successful and failed transactions count is available in the datastore for each minute when a new transaction is carried out the data store is queried for last x minutes of data.

The obtained data is aggregated and the success and failed transactions of each acquirer is shown in the form of percentages. If the failure percentage for an acquirer exceeds a certain threshold then the transaction is routed to another acquirer where the failure percentage does not exceed the threshold value. The value of x and threshold percentage need to be formulated based on the frequency of transactions and the past data available. Fig 2 shows the architecture of the proposed payment routing module. The major components present are audit datastore, payment routing module, Data store, payments app, payment Gateway and acquirer bank.

The payment routing module mainly performs these tasks:-

1. Filtering payment transactions

The audit data not only contains payment transactions it might also have details about the users, cards and some other tasks performed by the users but only payment transactions are required to carry out performance analysis. The payment routing module does the filtering of past payment transactions present in the audit data with the help of kafka streams. Using kafka streams each record is processed and a condition is applied to check for a payment transaction.

2. Aggregating the count for each acquirer

After filtering the audit data for payment transactions each payment transaction is processed and the respective count of each acquirer is increased accordingly. For example, a payment transaction is associated with a particular acquirer and it is successful then the number of successful transactions for that acquirer is increased by one. Similarly, if a payment transaction is failed then the number of failed transactions for that particular acquirer associated to the transaction is increased by one. In this way for each minute the count of number of successful and failed transactions for each acquirer is calculated.

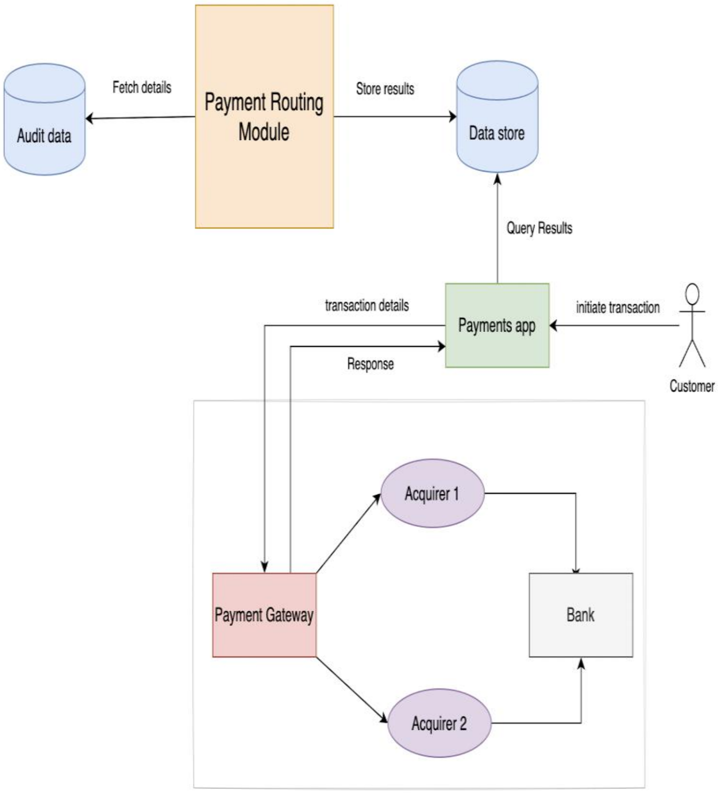

Fig -2: Architecture of the proposed system

3. Storing the results

The computed data in the previous step for each minute is stored in another data store. This data is stored so that queries can be done on this data for past x minutes and the routing decision can be taken. The data store should be selected in such a way that the performance of time series based queries on the data store should perform with minimum resource usage to reduce costs.

4. Querying the results

On data store time series based queries are performed and the query results are aggregated. Whenever a new payment transaction occurs then a query is executed on the data store for last x minutes. For last x minutes percentages are obtained which gives an indication on performance of each acquirer. For each acquirer, percentage of successful and failure payment transactions is present.

5. Routing Decision

When a payment transaction is initiated then the data store is queried for the last x minutes and the query results are obtained in the form of percentages. If the failure rate of an acquirer is more than the threshold set then the transaction is routed through another acquirer. In this way when a new payment transaction is initiated the payment routing module takes a decision which reduces the probability of failure of that transaction. Figure 3 shows the tasks involved in the payment routing module. It gives details about the sequence of tasks/steps needed to be carried out by the payment routing module.

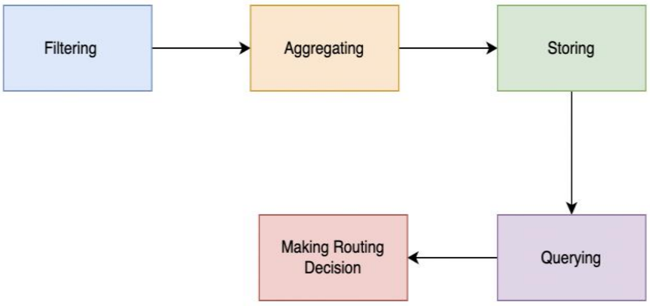

Figure 3: - Sequential steps in Payment routing module

4. RESULTS AND ANALYSIS

Introducing a simple payment routing module into the system impacts the transaction success rate of transactions drastically. The proposed payment routing module efficiently reduces the probability of failures by routing the payment transactions to a particular acquirer which increases its success probability. The proposed routing module is mainly based on the performance of the acquirers recently. The payment module is efficiently able to take the routing decision for a payment transaction. Instead of payment gateway taking a decision on the acquirer the payment routing module takes the decision and sends it to the payment gateway. An informed decision is taken to route the payment transaction to an available acquirer. In this way a payment routing module can be introduced with minimal effort and routing decisions can be made instead of giving more power to a third party payment gateway.

5. CONCLUSION AND FUTURE SCOPE

A payment routing module solution is proposed which is based on the performance of the acquirers in the last x minutes time frame. Stream processing of the audit data is carried out first considering one minute time frame and the count of successful and unsuccessful transactions for each acquirer is calculated and pushed into another data store. Later these results are queried for last x minutes when a new transaction is initiated and a decision is made for routing the transaction to a specific acquirer. The proposed routing module is only based on previous data; it doesn't involve parameters like bin, type of card used for payment etc. Telemetry can be introduced to view and monitor the count metrics and can easily find the performance of the acquirers. Machine Learning can be introduced to make the module more robust including all the necessary parameters for routing.

ACKNOWLEDGEMENT

I would like to thank the Department of Information Science and Engineering, R. V. College Of Engineering for the constant support in the field of Research and Development. I am indebted to my mentor, Dr Anala M R, who helped in the preparation of this thesis, for her hearty support, suggestions and invaluable advice throughout our project work.

REFERENCES

[1] R. Bygari, A. Gupta, S. Raghuvanshi, A. Bapna and B. Sahu, "An AI-powered Smart Routing Solution for Payment Systems," 2021 IEEE International Conference on Big Data (Big Data), 2021.

[2] S. Langhi, R. Tommasini and E. D. Valle, "Extending Kafka Streams for Complex Event Recognition," 2020 IEEE International Conference on Big Data (Big Data), 2020.

[3] G. van Dongen and D. Van den Poel, "Evaluation of Stream Processing Frameworks," in IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 8, pp. 1845-1858, 1 Aug. 2020.

[4] R. Shree, T. Choudhury, S. C. Gupta and P. Kumar, "KAFKA: The modern platform for data management and analysis in big data domain," 2019 2nd International Conference on Telecommunication and Networks (TEL-NET), 2019.

[5] P. Trivedi and A. Singh, "Stochastic multi-path routing problem with non-stationary rewards: Building payu's dynamic routing", Companion Proceedings of the The Web Conference 2018, pp. 1707-1712, 2018.

[6] F. Majeed, Muhammad Sohaib Mahmood and M. Iqbal, "Efficient data streams processing in the real time data warehouse," 2020 3rd International Conference on Computer Science and Information Technology, 2020.

[7] M. A. Lopez, A. G. P. Lobato and O. C. M. B. Duarte, "A Performance Comparison of Open-Source Stream Processing Platforms," 2019 IEEE Global Communications Conference (GLOBECOM), 2019.

[8] T. M. Evans, Y. Peng and H. A. Mantooth, "Placement and Routing for Power Module Layout," 2021 IEEE Design Methodologies Conference (DMC), 2021.

[9] s. J. Pon, S. S. Ramya, A. V. Christal and k. Mythili, "Secured payment gateway for authorizing E-commerce websites and transactions using Machine Learning Algorithm," 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020.

[10] H. -R. Lee, W. -J. Kim, K. -H. Park, H. -J. Cho and C. -H. Lin, "Development of an easy payment system based on IoT gateway," 2018 International Conference on Electronics, Information, and Communication (ICEIC), 2018.

[11] K. Guntupally, R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data (Big Data), 2018.

[12] H. Wang and B. Jia, "Research Based on Web Development of Spring Integration Framework," 2010 International Forum on Information Technology and Applications, 2010.

[13] Chunsheng Zhao, Mai Jiang and Zhiyong He, "The design of E-commerce system architecture based on Struts2, Spring and Hibernate," The 2nd International Conference on Information Science and Engineering, 2010.

[14] M. Klymash, I. Tchaikovskyi, O. Hordiichuk-Bublivska and Y. Pyrih, "Research of Microservices Features in Information Systems Using Spring Boot," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020.

[15] E. H. Kim, J. C. Na and S. M. Ryoo, "Implementing an Effective Test Automation Framework," 2009 33rd Annual IEEE International Computer Software and Applications Conference, 2009.