**INSIGHTS AND RECOMMENDATIONS ON BANKING SYSTEM PERFORMANCE ANALYSIS**

1. How fast does the system respond to transactions on average?

**Overall System Performance is Below Expectations**

* Out of 60 transactions, 32 (53.3%) were slow and the system did not respond fast enough, meaning the system is failing more than half the time to meet its performance targets. The fact that airtime transactions were always slow (100% failure rate) is a major issue, as it suggests an inherent delay with third-party integrations. Customer loan information (35% slow) and log-in (25% slow) are performing better but still have notable inefficiencies.

**Airtime Transactions are a Critical Bottleneck**

* With a 0% success rate (all 20 transactions slow), airtime purchases are the worst-performing transaction type. This suggests that third-party integration is a major cause of delay, possibly due to external system inefficiencies, network latency, or API rate limits.

**Loan Information and Log-In Transactions Show Moderate Issues**

* Customer loan information transactions had a 65% success rate, meaning they are mostly within the target threshold but still failing more than a third of the time. Log-in transactions had a 75% success rate, which is acceptable but not optimal for a critical function like authentication.

**The System is Struggling to Maintain Speed Consistency**

* The mix of fast and slow transactions suggests inconsistent system performance, possibly due to peak load periods, server resource limitations, or inefficient processing algorithms.

**Recommendations**

**Investigate Third-Party API Latency for Airtime Transactions**

* Since all airtime transactions failed the speed test, the bank should: review the contract with the third-party provider to check for SLA violations, test API latency separately to determine if it’s an external issue, consider caching or batch processing to reduce response time.

**Optimize Customer Loan Information Processing**

* Since 7 out of 20 transactions were slow, improvements could include: Database indexing to speed up loan information retrieval or Asynchronous processing to reduce front-end waiting time.

**Improve Log-In Authentication Performance**

* Since 5 out of 20 log-in attempts were slow, this could be due to: High server load affecting response time, inefficient authentication protocols (e.g., excessive encryption overhead). Session management could also be optimized to reduce processing overhead.

**Perform Load Testing & Scaling Analysis**

* Analysis shows that the system slows down as load increases which calls for scaling solutions like: Load balancing (distributing requests across multiple servers), Increasing server capacity (adding RAM, CPU power) and Optimizing query execution times (using caching, indexing).

**Set Up Monitoring & Alerts for Performance Degradation**

* Real-time tracking of transaction response times can help identify patterns of slow performance. Alerts should trigger when response time breaches thresholds to proactively address delays.

1. How far is the system’s response time from the set thresholds? By how much is it excelling or underperforming?

**Severe Airtime Transaction Delays Worsen Under High Load**

* Overall, airtime transactions are 10 seconds slower than the target 5-second threshold. Under high load, this worsens significantly to 20 seconds slower, meaning transactions take four times longer than expected during peak usage. Under medium load, airtime is still 11 seconds slower, showing a consistent performance issue

**Customer Loan Information Transactions Are Slightly Below Target**

* Overall, CLI transactions are 1.6 seconds slower than the 4-second threshold. Under high load, delays improve to only 100 milliseconds slower, suggesting that loan queries are more resilient under heavy usage than other transactions. Under medium load, they are 356 milliseconds slower, indicating slightly increased delays with medium traffic.

**Login Performance Varies by Load**

* Overall, login transactions are 61 milliseconds faster than the 1.5-second target. Under low load, login is even faster by 635 milliseconds. Under high load, logins become 1.6 seconds slower, meaning authentication struggles under peak traffic.

**High Load Negatively Impacts All Transactions**

* Airtime becomes 20 seconds slower and logins become 1.6 seconds slower, indicating the system struggles with concurrent transactions. Customer loan transactions are least affected under high load, suggesting better optimization compared to the others.

**Recommendations**

**Urgent Optimization of Airtime Transactions**

* Investigate third-party API bottlenecks—review how the external service handles transaction requests under load. Consider load balancing techniques to distribute transaction requests across multiple endpoints or servers. Enable transaction queuing or caching to prefetch commonly used data and reduce real-time query execution times.

**Strengthen High-Load Performance for Logins**

* Optimize authentication protocols to reduce processing overhead during peak times. Implement session management improvements such as persistent sessions or adaptive authentication. Scale infrastructure resources dynamically during high traffic periods to prevent slowdowns.

**Enhance Medium-Load Performance for Loan Information Transactions**

* Optimize database queries related to loan data retrieval to eliminate minor slowdowns. Implement predictive preloading to fetch loan information before the user completes input, reducing perceived delay.

**Capacity Planning**

* Identify peak transaction periods and allocate more server resources dynamically to handle surges. Optimize middleware processes to prioritize essential transactions (e.g., logins) under high load conditions.

1. How many errors does the system make in transactions?

**Customer Loan Information Transactions Fail at High Load**

* Customer Loan Information has an overall error rate of 2.7%, meaning a small but notable percentage of customer loan information requests completely fail. It has a high load error rate of 10.8%, indicating the system struggles significantly under peak demand. It has no errors under low or medium load, suggesting the issue only arises when the system is under stress.

**Airtime and Login Transactions Have 0% Failure Rate**

* No failed airtime transactions across all load categories, meaning despite slow speeds, the system always completes the transactions successfully. No failed logins, showing the authentication system is reliable in execution, even though performance drops under high load.

**System Stability Under Low and Medium Load**

* The system operates successfully at lower loads, meaning failure is not an inherent flaw but a capacity and scalability issue under stress.

**Recommendations**

**Optimize Customer Loan Information Processing for High Load**

* Analyse backend failures. Investigate whether these failures are due to database query timeouts, server crashes, or API call failures. Consider autoscaling servers or optimizing database indexing to handle high demand. Introduce transaction retry mechanisms or fallback workflows for failed requests.

**Prevent Future High-Load Failures**

* Optimize system architecture by implementing queue management and load balancing to distribute requests more efficiently.

**Maintain Airtime and Login Stability While Improving Performance**

* Airtime and login transactions are reliable but slow—prioritize performance tuning without disrupting their stability.

1. What is the range of the system’s transaction speeds?

**Airtime Transactions Have the Most Inconsistent Performance**

* The minimum response time is 2.9 seconds, which is acceptable. However, the maximum response time is 52 seconds, which is 10× higher than the set threshold (5s). This extreme variability suggests third-party system delays or server queuing issues.

**Customer Loan Information Transactions Are Unpredictable**

* The response time ranges from 94 milliseconds to 30 seconds. Even at low load, the response time varies significantly (426 ms to 17.7 seconds). Medium load has better consistency (7.79 ms to 15.7 seconds), but high-load spikes mirror the worst-case scenario. This suggests database query inefficiencies, potential bottlenecks in loan validation, or processing queue delays.

**Login Transactions Are More Stable but Still Have Issues**

* The fastest response time is 114 ms, which is excellent. However, the slowest response time is 5.4 seconds, exceeding the threshold of 1.5 seconds. Under high load, login times reach 9.4 seconds, which is 6× slower than the target. This suggests authentication server congestion or inefficient session management during peak hours.

**Recommendations**

**Reduce Airtime Transaction Latency and Variability**

* Investigate third-party integration. If possible, batch airtime requests to reduce individual transaction load. Check for network bottlenecks affecting response time spikes.

**Improve Customer Loan Information Processing**

* Optimize database queries: Index frequently used tables and optimize query execution plans. Use background processing or caching for less critical loan information to improve real-time performance. Implement dynamic resource allocation for loan-related queries.

**Optimize Login System Performance**

* Improve session caching: Use server-side caching (Redis, Memcached) to reduce repeated authentication delays.
* Upgrade authentication infrastructure: If high load causes slowdowns, consider dedicated authentication servers or autoscaling.
* Load balancing: Implement a load balancer to distribute authentication requests more evenly across servers.

1. Is the system processing enough transactions per second?

**High Load Performance is Poor – System is Bottlenecked**

* Out of 16 high-load transactions, 10 (62.5%) did not meet the throughput threshold. A healthy system should have increased throughput as the load increases, but this system’s throughput does not increase as expected at higher loads, suggesting resource saturation or processing limitations.

**Airtime Transactions Have the Worst Throughput Performance**

* 100% unmet throughput target in high load (5/5 transactions below threshold). 100% unmet throughput target in medium load (4/4 transactions below threshold). Even in low load, more transactions did not meet targets (6) than did (5).
* Total airtime unmet throughput target: 15/20 (75%). This suggests inefficient processing or dependency on external systems.

**Customer Loan Information Transactions Struggle Under High Load**

* 100% unmet throughput target in high load (5/5 transactions below threshold). 50% unmet throughput target in medium load (2/4 transactions below threshold, 2 above). Low load is efficient (1/11 transactions below threshold, 10/11 above). Total unmet throughput target: 8/20 (40%). Unlike airtime, it improves at medium and low loads, meaning the failure may be due to resource contention at high load.

**Log-in Transactions Have Excellent Throughput**

* 0 unmet throughput target s across all load categories. This suggests optimized authentication processing, likely using caching or lightweight request handling. Other transactions could benefit from similar efficiency improvements.

**Overall System is Underperforming at High Loads**

* Across all transactions: 23 unmet throughput targets, 37 above target. Unmet throughput targets are concentrated in high and medium loads, where the system should be scaling up but isn’t.

**Recommendations**

**Identify and Address Bottlenecks for High Load Performance**

* Monitor server utilization (CPU, RAM, disk I/O) during peak loads to find bottlenecks. Optimize database queries for transactions (especially airtime and customer loans). Consider autoscaling infrastructure to dynamically allocate resources during peak times.

**Fix Airtime Transaction Performance Issues**

* Investigate third-party API response times (if applicable). If delays come from external providers, consider switching or optimizing API calls. Implement queuing mechanisms (Kafka, RabbitMQ) to prevent request congestion. Reduce synchronous dependencies – offload less urgent processing tasks to background jobs.

**Improve Customer Loan Processing Under High Load**

* Use caching for frequent queries (e.g., storing common customer data in Redis). Batch loan verification requests to process multiple at once instead of sequential handling. Ensure loan transaction prioritization (e.g., fast-track critical transactions when the system is under heavy load).

**Maintain and Apply Log-in Efficiency to Other Transactions**

* Study why log-in transactions maintain good throughput. If caching is used, consider applying caching to airtime and customer loan transactions. If log-in benefits from distributed processing, implement similar parallelization techniques for transactions.