Volume 9, Issue 10, October – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT760

MDM System Performance Test Recommendations

Sai Reddy Anugu

Abstract:- The paper provides detailed guidance for a Performance Test Process and Recommendations for a Master Data Management System, focusing on the IBM Infosphere MDM Server. And outlines the Importance of early involvement of the Business and systems owners in setting performance benchmarks, and the stakeholders' collaboration, during the test and the steps to establish monitoring, change control, and document the test results and the performance test processes and system scalability Testing recommendations.

Keywords:- *MDM(Master Data Management); Transactions Per Second (TPS); Infosphere MDM Server; Customizations; Scalability; Performance Metrics, Monitoring Tools.*

I. INTRODUCTION

This paper focuses on the importance of performance testing for the success of MDM Project implementation. MDM Server is designed with high performance and easily scalable, however, we need to ensure customizations made for client-specific business requirements do not impact the overall system performance. Customizations should also be done through a rigorous performance testing process while implementing the MDM project. Performance must be forethought and not an afterthought. Need to utilize all the opportunities to improve performance, by following the best practices in all the phases of the project implementation. Document the performance objectives and performance test data collection for key usage scenarios. Plan performance testing early in the development cycle to reduce costly remedies for performance issues identified later cycle.

II. UNDERSTAND AND PLAN PERFORMANCE TESTING

It has been proven many times that early involvement of the Business and system owners, will help in setting up the service performance goals like the number of Transactions Per Second (TPS), and acceptable resource utilization (CPU, RAM, and I/O). Establish a consistent and repeatable performance testing process that includes proper and sufficient performance data collection, a dedicated or controlled performance testing environment, and good source code and change control.

Include regression test suits to test the functional failures, after each change to the system.

Major code changes should be tested and evaluated against a baseline to determine their performance impact during the journey to get the system into production. A repeatable process and the right set of data to cover the production workflow scenarios, properly evaluate the impact of the changes, and accurately quantify and characterize the system's level of performance. Source control will allow the team to revert to the performance tests for a particular version of the implementation, thereby facilitating the performance problem investigation process.

Implement monitoring tools to cover all the critical components in the System, for Example: Operating system, I/O, CPU, Memory, Network, and DB components.

III. RECOMMENDATIONS FOR A PERFORMANCE TEST PROCESS

At a very high level, the following strategies help to achieve comparable, consistent, and effective Performance testing results for MDM Systems:

- Allocate a dedicated or well-controlled performance test environment with similar resource allocation to production.
- Maintain a consistent system state by restoring the Database and restarting the application server before starting the next test.
- To establish comparability and consistency before evaluating any changes, the same test should be done using the same input data, running the same test cases, going through the same steps (e.g. warm up, and steady phase) with the same data collection traces or logging enabled Once such repeatability is established, the system is good to evaluate the targeted changes with all the other parameters and variables being kept the same.
- Use production-like data, and volume, and cover all the production workflow scenarios to reduce the risk of performance issues in production.
- Ensure a sufficient warm-up period of around 20 minutes before taking measurements, the warm-up time depends on the size of the system. To find a suitable warm-up period, do a set of long runs and look for a repeatable inflection point where throughput has risen to a flat or nearly flat line.
- Performance metrics need to be measured during a steady state where they become constant. The number of transactions or data to drive the test should last long enough for stable and consistent performance.
- It is strongly recommended that you run the same test multiple times, without any changes, and then compare the key performance metrics.
- Set up the source control so that you can revert to a particular version of the application code and the performance test. So, both the application and the test harness need to be managed

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT760

- Maintain a README for each test to record key run information, including the time, run, the purpose of the run, software settings and hardware configurations, application version being used, configuration differences, and performance metrics (e.g. Average Response Time, Transaction Per Second).
- Evaluate bottlenecks with stakeholders, implement the fix, and conduct a retest to ensure that the performance has improved.

IV. CONCLUSION

In conclusion, paying close attention to early planning, setup and execution of performance tests for customizations of MDM Systems is key to building reliable, high-performing scalable applications. Allow organizations to adopt Agile culture and improve customer experience.

REFERENCES

[1]. Performance Considerations for IBM Infosphere Master Data Management Advanced Edition
[2]. Prepare for the Database Performance Test (informatica.com)
[3]. TIBCO® MDM Performance Tuning Guide
[4]. Reltio Developer portal