

Software Testing and Quality Assurance

2CSDE80

Roll No.: 22BCE501

PRACTICAL 7:

Parameters :

1. Resource Utilization: Monitor CPU, memory, disk I/O, and network bandwidth to determine how effectively the application uses system resources.
2. Throughput: The number of transactions or requests processed in a given time frame, often measured in transactions per second (TPS).
3. Latency: The delay before a transfer of data begins following an instruction for its transfer.
4. Response Time: Measure how long it takes for the application to respond to a request.
5. Scalability: Assess how well the application can handle increased load by adding resources or scaling up infrastructure.
6. Load Handling: Evaluate how the application performs under expected and peak load conditions.
7. Stability and Reliability: Test how the application behaves over extended periods, checking for crashes or memory leaks.
8. Error Rate: Measure the frequency of errors during normal operations, including server errors and application-specific errors.
9. Concurrency: Evaluate how the application handles multiple users or processes simultaneously.

10. Configuration Impact: Assess how different configurations affect performance, such as changes in server settings or database configurations.
11. Network Performance: Analyse the impact of network latency and bandwidth limitations on application performance.
12. Benchmarking: Compare performance against established standards or similar applications to gauge effectiveness.
13. End-user Experience: Collect metrics that reflect actual user experiences, such as time to first byte (TTFB) and page load times.
14. Database Performance: Measure query execution times, connection pooling, and transaction handling.
15. Integration Points: Evaluate the performance of any external services or APIs the application relies on.

Tools :

1. **Apache JMeter:** A widely used open-source tool for load testing and performance measurement. It supports various protocols and can simulate heavy loads.
2. **LoadRunner:** A comprehensive performance testing tool from Micro Focus that supports a wide range of applications and protocols. It provides detailed reporting and analysis.

3. **Gatling:** An open-source load testing framework designed for ease of use, particularly for testing web applications. It offers a high-level scripting language and detailed metrics.

4. **k6:** A modern load testing tool that uses JavaScript for scripting and is particularly suited for testing cloud-native applications and APIs.

5. **NeoLoad:** A performance testing tool that provides end-to-end testing for web and mobile applications. It integrates well with CI/CD pipelines.

6. **BlazeMeter:** A cloud-based performance testing service that supports JMeter scripts and allows for easy scaling of tests.

7. **AppDynamics:** An application performance management tool that provides real-time monitoring and analytics to understand application behaviour under load.

8. **New Relic:** Offers performance monitoring and analytics for web applications, allowing you to track response times, error rates, and resource usage.

9. **Dynatrace:** Provides full-stack monitoring with AI-driven insights for application performance, including real-user monitoring and synthetic testing.

10. **Locust:** An open-source load testing tool that uses Python for scripting. It allows you to define user behaviour in code and scale tests easily.

These tools can help you evaluate various aspects of application performance, from load handling to resource usage and end-user experience.

