

CSE 565 Assignment 4

Member Name: Gautham Vijayaraj

ASU ID: 1229599464

Date: 10/30/2023

Part 1

A web review about endurance testing:

Definition of Endurance Testing

Endurance testing is a type of performance testing that assesses how a **system handles prolonged stress**. It evaluates the stability and reliability of a system over an extended period under sustained load.

It is a subset of performance testing, aimed at identifying issues that might develop over prolonged usage, such as memory leaks, database connection failures, and system slowdowns. By running tests continuously, endurance testing **ensures that an application can maintain performance standards even when subjected to lengthy operational loads**, mimicking real-world scenarios like high-traffic days for web applications.

Types of Errors Targeted

Endurance testing seeks to uncover issues that may not appear in shorter tests but can impair long-term functionality, including:

1. **Memory Leaks:** Failure to release memory no longer in use, which can slow down or crash applications.
2. **Response Time Degradation:** Gradual reduction in system responsiveness due to prolonged processing demands.
3. **Resource Leaks and Database Failures:** Inefficient resource allocation or unclosed connections that can cause database inconsistencies or data loss over time.
4. **Log and File Handling Issues:** Problems with log cycling and file handling that could disrupt operations.

Tools for Endurance Testing

1. **Apache JMeter:** JMeter is a popular open-source tool used for testing application performance. It can simulate various load scenarios and integrates with Selenium for automated testing. Its versatility and ability to run long-duration tests make it ideal for endurance testing.
 - JMeter's ability to maintain long-duration tests and its powerful analytics allow testers to identify issues like memory leaks, slowdowns, and resource bottlenecks over extended periods, making it ideal for systems that require stability under consistent load.
 - Its architecture, developed in Java, allows for wide adaptability across platforms, making it a reliable choice for performance and endurance testing.
2. **LoadRunner by Micro Focus:** LoadRunner is widely used in performance testing and supports endurance testing by allowing for sustained test simulations under different loads. It offers real-time monitoring and analytics, making it useful for identifying and resolving memory leaks or performance degradation issues as they occur.

- LoadRunner is particularly useful for endurance testing due to its ability to monitor system health continuously and adjust load parameters as needed.
- Its extensive reporting capabilities also allow testers to spot performance degradation trends and resolve issues like memory leaks or high response times that could cause system failures under sustained usage.

Part 2

Description of 2 types of systems that require endurance testing:

1. E-commerce Platforms

- **Potential Problems:** High-traffic days (e.g., Black Friday sales) place significant, prolonged load on e-commerce platforms, which can lead to slow loading times, payment gateway failures, and potential crashes. Prolonged stress can expose hidden issues like memory leaks, degraded server response times, and overloaded databases. During high-traffic events, applications can accumulate unused memory if not managed properly, leading to slower performance or even crashes. Frequent queries to retrieve products, update stock, or process payments can overload databases if not optimized, causing disruptions in service.
- **Endurance Testing Strategy:** For e-commerce platforms, test scenarios should replicate peak traffic conditions over a continuous period. This includes testing sustained login sessions, shopping cart usage, and payment processing under heavy load. Monitoring memory usage, response times, and database connection stability helps identify potential bottlenecks and system breakdowns before they affect users. A detailed description of their testing strategy would be:
 - Simulate Realistic High-Load Scenarios:** Create load tests simulating peak user actions over a sustained period, focusing on critical actions like browsing, adding to cart, and checking out.
 - Monitor Key Performance Metrics:** Track metrics such as memory usage, database performance, and response times. Identifying trends in memory usage can help pinpoint potential leaks or inefficiencies.
 - Optimize and Test Database Connections:** Use tools to test and ensure that connections are efficiently managed, avoiding overload on database queries.
 - Establish Load Thresholds and Breakpoints:** Define realistic thresholds for key actions (e.g., number of simultaneous checkouts) and stress test to measure system behavior under extended peak conditions

2. Banking Systems

- **Potential Problems:** Banking systems require stability for continuous online transactions and ATM processing. Prolonged testing may reveal issues like transaction failures, unclosed database connections, or response time degradation, which could cause service outages and user dissatisfaction. Consistently high loads can increase response times, delaying transaction confirmations and negatively affecting user confidence. Banking systems handle sensitive data and require secure connections; prolonged loads can reveal vulnerabilities or connectivity issues, impacting data security.
- **Endurance Testing Strategy:** For banking systems, endurance testing should simulate sustained transactional loads over days to weeks. Testing should cover activities like online transfers, account queries, and loan applications. Continuous monitoring of memory and database connectivity

ensures these systems can handle prolonged loads without affecting transaction accuracy. A detailed description of their testing strategy would be:

- i. **Run Long-Term Simulations with Realistic Transaction Loads:** Mimic regular transactional patterns over extended periods, including common user actions such as logging in, checking balances, and transferring funds.
- ii. **Implement Continuous Monitoring for Security and Connectivity:** Monitor for slowdowns, especially in multi-layered systems, ensuring all layers from the UI to the database can handle prolonged load without performance degradation.
- iii. **Stress-Test Under Peak Period Conditions:** Ensure testing includes peak usage scenarios to evaluate system resilience. For example, simulate activities during end-of-month transactions when user activity is higher.
- iv. **Analyze and Address Identified Vulnerabilities:** Post-test, assess any detected weak points in code or system architecture, optimizing for long-term stability under heavy, sustained loads

References

ChatGPT Prompt: <https://chatgpt.com/share/6722db30-8e7c-8013-b938-0e0c7b6405a8>

Definition: <https://artoftesting.com/endurance-testing>

Endurance Testing Guide: <https://www.softwaretestingmaterial.com/endurance-testing/>

QA Testing Instights: <https://blog.qasource.com/a-complete-guide-to-endurance-testing>

Endurance Testing for System Stability: <https://www.loadview-testing.com/learn/endurance-testing/>