Capacity and Performance Management - The Key to Successful Outsourcing

Part 4 – Performance and Capacity Acceptance Testing

Agenda

• Acceptance Testing
• Single Workload Tests
• Mixed Workload Tests
• Stress Tests
• Soak Tests
• Components Failure Tests
• Component Slow-down Tests

© Capacitas 2002-2006
Terminology

**System integration testing**

- Web server
- Application server
- Database

**Component/System testing**

Non-functional Acceptance Testing

- Is required when a 3rd party is engaged to:
  - Develop an application
  - Deliver a system integration
- Demonstrates that the outsourced product or system is able to meet the requirements
  - Requirements include both functional and non-functional
- Acceptance criteria need to be agreed in advance
- Non-functional acceptance testing shouldn’t be skipped, this is as important as functional testing
- Acceptance testing can be carried out in front of the customer, i.e. witness testing
Requirements

- Functional
  - Usually relatively straightforward, e.g. pass or fail
- Non-functional
  - Less quantifiable
  - Examples:
    - Usability
    - Performance/capacity
    - Reliability
    - Fault tolerance
    - Scalability
    - Security

Acceptance Testing

- The following parameters are closely related:
  - Performance/capacity
  - Reliability
  - Fault tolerance
  - Scalability
- A series of tests needs to be executed to understand the above non-functional requirements under a range of different load conditions
- Use a series of tests as part of acceptance test rather than a single test
Concurrenty

- Load generation tools have configurable parameters that impact performance
  - Number of user threads
    - Impact concurrency and hence response times
  - Wait times
    - Impact the arrival pattern of transactions
- Define the number of concurrent threads that the acceptance test is going to be carried out with for all the test types
- How do you define the number of concurrent threads?
  - One approach is to use a Poisson distribution to predict the number of concurrent threads
    - This approach is not quite correct because it does not take into account previous arrivals

Error Handling

- Error message response times can be different from successful transaction response times
- Error message response times can skew transaction response times
- Performance test scripts should:
  - Capture errors
  - Report percentage of errors
  - Report error type – are the errors load related?
  - Report response times based on successful transactions
Caching

- Caching can skew performance test results significantly
  - Typically provide more optimistic response time measures
- Performance test scripts use parameters
  - The way these parameters are selected within the scripts can have a significant impact on the transaction response times
  - Aim to select data for parameters randomly from a large range of data

Single Transaction Type Tests

- Single transaction type tests
  - Prioritise
  - Key transactions
    - High volume
    - End user experience
- Transaction types:
  - Queries
  - Updates
  - Creates
  - Reliable messages
  - Web services
Mixed Transaction Type Tests

- Mixed transaction type tests
  - Typically, response times will be greater than single transaction types at corresponding loads
  - Measure performance metrics at different loads

Response Times vs. Load

![Graph showing mixed transaction type tests](image)
Soak Tests

- Related to availability
- Short tests may hide
  - Memory leaks
  - Stability issues
- Carry out a ‘soak test’ as part of your acceptance testing
  - Soak tests can range from 12 hours to several days and nights
  - This will also highlight other non-functional areas such as availability, resilience etc.
- Soak test workload
  - A representative work load needs to be used, e.g. a mixture of different queries, updates and creates at appropriate volume

Stress Test

- Related to performance, stability and scalability
- Tests are run at volumes significantly higher than expected
- The system should behave in a predictable way, i.e. not become unstable even if response times climb
- System response times and resource utilisation should be similar to queuing model predictions
Component Failure Tests

- Related to resilience and stability
- Tests should be carried out with failed components
- Other components should behave in a stable and predictable manner
- Issues to look for:
  - Performance coupling between components
  - Management of timeouts
  - Retry mechanisms
  - Excessive connection or session creation between interface of working components and failed components
  - Load balancing issues
Component Slow-down Tests

- Related to integration performance and stability
- When one component of a system is outsourced it is critical that it performs to expectation with the other components
- This can be carried out before system integration testing, i.e. during component testing

- Stubs can be used to represent component
- Stubs should use configurable random delays rather than fixed delays
- Issues to look for:
  - Performance coupling between components
  - Management of timeouts
  - Retry mechanisms
  - Excessive connection or session creation between interface of working components and failed components
  - Load balancing issues
Summary

- Acceptance Testing
- Single Workload Tests
- Mixed Workload Tests
- Stress Tests
- Soak Tests
- Components Failure Tests
- Component Slow-down Tests