Performance testing can help you identify your website or application’s bottlenecks. Follow these steps to ensure that it performs well under pressure.
STEP 1
IDENTIFY
TEST
OBJECTIVES
Analyze key performance indicators based on historical data to determine the extent of your performance testing.

<table>
<thead>
<tr>
<th>KPI</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response Time:</strong> The amount of time the application or website takes to respond to a request.</td>
<td>A product search shouldn’t take more than 5 seconds. Make sure your system responds in a timely manner.</td>
</tr>
<tr>
<td><strong>Throughput:</strong> This can be measured as the number of transactions done by the AUT in a given time period, say, per second, and it depends upon the load.</td>
<td>Make sure your application server has the capacity to entertain 500 transactions per second.</td>
</tr>
<tr>
<td><strong>Resource Utilization:</strong> The rate at which resources such as CPU, I/O, memory, and databases are utilized by a typical application.</td>
<td>All the resources such as processor and memory utilization, network Input output, disk input output, etc. should be at less than 90% of their maximum capacity.</td>
</tr>
<tr>
<td><strong>Maximum User Load:</strong> This measures how many concurrent tasks or users the application can handle at a given time.</td>
<td>The system should be able to entertain 10,000 concurrent users by fulfilling all of the above defined objectives.</td>
</tr>
<tr>
<td><strong>Behavior Under Stress Condition:</strong> This test is to ensure users are able to interact with systems normally under a high load condition for a specific time span or duration.</td>
<td></td>
</tr>
</tbody>
</table>
STEP 2
IDENTIFY KEY SCENARIOS
Select scenarios based on how critical they are to the overall application performance.

Select the most frequently accessed scenarios.

Select resource intensive scenarios.

**EXAMPLE SCENARIOS**

- Browsing product catalog
- Creating a user account
- Searching for a product
- Logging in to application
- Placing an order
STEP 3

IDENTIFY WORKLOAD
- Determine navigation paths of key scenarios.
- Identify unique test data.
- Identify relative load distribution across identified scenarios.
- Identify target load levels.
  - Time period
  - Normal load requests
  - Peak load requests
  - Peak load duration
- Other miscellaneous options:
  - Think time
  - Pause time
  - Location of request
STEP 4
IDENTIFY METRICS
Metrics provide information about how well or poorly your application is performing compared to your performance objectives.

Metrics can help you identify problem areas and bottlenecks within your application.

Example performance metrics:

- Memory available/memory utilization/processor utilization/disk utilization/network utilization
- Transactions succeeded/transactions failed/transactions times
- Deadlocks/crash
Step 5: Create effective test cases.
**Order Placement Scenario**

- Workload: 10,000 simultaneous users.
- Think time: Use a random think time between 1 and 10 seconds in the test script after each operation.
- Test duration: Run the test for two days.

**Expected results**

- Response time should not be greater than 10 seconds for 95% of total transactions completed.
- Order transactions should not fail during test execution. Database entries should match the “transactions succeeded” count.
- Throughput should not fall below 100 requests per second.
- Application hosting process should not recycle because of deadlock or memory consumption.
- “Server busy” errors should not be more than 10% of the total response because of contention-related issues.
Step 6: Simulate load.
• Validate that the test environment matches the configuration that you were expecting and/or designed your test for.

• Ensure that both the test and the test environment are correctly configured for metrics collection.

• Before running the test, execute a quick “smoke test” to make sure that the test script and remote performance counters are working correctly.
STEP 7

ANALYZE

RESULTS
Identify potential application issues that surface under extreme conditions.

Such conditions range from exhaustion of system resources such as memory, processor cycles, network bandwidth, and disk capacity to excessive load due to unpredictable usage patterns.
Want more information on performance testing?

Visit LogiGear.com