Why & How : Performance Test

For: QAAC
By: Deepak
• “Investigate Performance Early; Validate Performance Last”
  - by Scott Barber is the Chief Technologist, CEO, and President of PerfTestPlus, Inc. Reference article: Performance Investigator in Better Software Magazine (March 06 issue).

• “To derive a good testing strategy out of scalability and performance testing, there needs to be testable and measurable requirements” - by Srinivasan Desikan is the director of quality engineering at Siebel Systems, Inc. Reference article: Scalability and Performance Testing of Server Software (Website http://www.sticky minds.com published July 05).
Table of Contents

• Performance Test description and its need
• Performance Test Objectives
• Performance Testing Engagements
• PVS Test Terminology
• PVS Test Requirements
• PVS Test Process
• PVS Tool-set
• Transaction Response Time
• Response Times Vs Service-level agreements (SLAs)
• Monitor Application and Database Server Usage
• Performance Testing Best Practices
• Benefits of Performance Testing
• Summary and Conclusion
What is Performance Testing?

The process of exercising an application by emulating actual users. This can be used to monitor how the database, application code and hardware will act under load.

Performance testing verifies that the application meets specified requirements under normal conditions.
The Need For Performance Testing

No matter how rich application is functionally, if it fails to meet the performance expectations (SLA) of customer, the application will be branded a failure.
Performance Test Objectives

- Determine business transaction response time
- Identify system bottlenecks
  - Server resource usage e.g. CPU, Memory and Disk Space
  - Network Bandwidth (throughput) and Latency (delay)
    e.g. Limited network throughput speeds serve to introduce a latency (delay) when transmitting larger amounts of data at specific location
- Determine system (hardware/software) optimal configuration
- Verify current system capacity and scalability for future growth
  - Determine how many users the system can support
  - Determine if the application will meet its SLA
Performance Test Engagements

• Performance Risk Assessment
  - Determine if application needs to be PVS tested by completing performance risk assessment matrix
• Performance Volume Stress (PVS) Testing
  - Test Strategy with Action plan
  - Script development
  - Test execution
  - Resource monitoring
  - Analysis & Recommendations
• Load-balancing and Fail-over Test
• Network Analysis
PVS Test Terminology

**Types of Tests**

- **Load Test**: To determine the impact on an application’s service level for a specific number of users and user types.

- **Volume Test**: Inject heavy workload (e.g. future volumes) to show if the system can handle the peak volume of data. Tests server response time under heavy load and varying user levels, measures round-trip response time and determines maximum number of users.

- **Stress Test**: Test the application or server to determine a load point at which the application suffers severe degradation or malfunction.

- **Network Application Analysis Test**: To capture the network characteristics of an application and to proactively seek inefficiencies to forecast potential performance and network risk.

- **Latency**: The amount of time it takes a packet to travel from source to destination. Together, latency and bandwidth define the speed and capacity of a network.
Other Terminology

• Workload: The total user load exercised in a production environment.

• Workload Model: A subset of actual workload used for performance testing (80/20 rule: 80% of the workload is generated by 20% of the transactions mix).

• Test Cycle: A group of testing scenarios designed to accomplish a specific threshold or condition (production level conditions, future production conditions, end-state conditions).

• Scenario: A logical unit of work or a transaction sequence to exercise a type of performance test that achieves a performance test objective.

• Script: Automated set of instructions to drive the performance test during execution.

• Virtual User: A software process that simulates real user interactions with the application.

• Response Time: The time it takes between initial request and complete download of response (rendering of entire web page).
PVS Test Requirements/Prerequisites

• Technical design and architecture, Non-functional business requirements and Data flow diagrams.

• Hardware/software configuration and access to monitor resource usage.

• Establish and document SLAs (e.g. scope and objectives, response time, availability, business scenarios, acceptance criteria and project timelines etc).

• Stable test environment, test data, user IDs or any other unique data required for the Performance test types.

• Use-cases: Test scripts reflecting how the end-users are using the system.
Performance Testing Processes

- Performance Testing Questionnaire
- Test Cases

- Performance Test Strategy
- Effort Estimation Guideline

- Performance Testing Best Practices
- Scripting Standards and Naming Conventions Guidelines
- Workload calculation

- Performance Test Report
PVS Test Process

• Gather and Review Checklist Requirements

• Test Strategy/Approach Document
  - Scope and Objectives
  - Identify needed test resources
  - Milestones

• Script Development
  - Design test
  - Develop workload model
  - Build test bed
  - Create script test data

• Test Execution
  - Performance/Load test
  - Review initial output
  - Volume and Endurance test
  - Stress test

• Result Analysis and Final Report
  - Executive summary
  - Recommendations to improve application performance
  - Results and Analysis
Performance Engineering Methodology

1. Understand System (Sys. Architecture, Workload, Performance Goals)
2. Develop Test Assets (Test Plan, Test Scripts, Monitoring Strategy etc…)
3. Execute Tests
4. Collect monitoring metrics
5. Analyze Results and monitoring metrics
6. Performance criteria are met
7. Complete Testing
8. Re-test after tuning
   - Criteria not met
   - TUNE
PVS Tool-set

• Mercury-HP’s LoadRunner, Borland-Segue’s SilkPerformer, Compuware’s QA Load, IBM’s RPT, and Empirix’s eLoad
  ▪ Load injection and scenario design mirrors production
  ▪ Round trip transaction times
  ▪ Monitor application and architectural components

• Perfmon/Sitescope
  ▪ Performance logs and alerts supports detailed monitoring of the utilization of operating system resources.

• Web Analytics/Trends/Topaz/Gomez
  ▪ Interactive web analytics reporting (monitor production usage pattern)

• Vantage/Sniffer/Netsense
  ▪ Network Management - Analysis for bandwidth usage and latency issues
Multi-Tier Architecture

For instance, it's very important to performance test a complex architecture which support highly scalable and robust load-balancing mechanisms at all tiers of the architecture using a combination of hardware and software load-balancing.

All tiers can be scaled individually by building clusters of servers for load-balancing and high redundancy.
End-to-End Interface Architecture Diagram

- PWM
- JSP
- MQ
- CICS [Mainframe]
- MQ Wrapper program
  - Request [CobolCopybook]
  - Response [CobolCopybook]
- Policy Request
- Admin Server
- SeeBeyond
  - HTTP call [XML]

Agent
Insured
Home office employee
Monitor Server Usage

- **Timings**
  - Response time
  - Execution time
  - Wait and connection times

- **Resource Consumption**
  - CPU, Disk Space and Memory usage
  - Number of connections, active threads, timeouts and Servlets requested, loaded and active
  - Web page counts, size and timings

- **Database Performance**
  - Physical reads & writes
  - Access paths and connections
  - Locking performance
  - I/O, buffer and wait counts and timing

- **Additional application specific metrics**
Measurement of Application Business Transaction Response Time

**Average Transaction Response Time**

Graph data in Excel format

**Description:** Displays the average time taken to perform transactions during each second of the load test. This graph helps you determine whether the performance of the server is within acceptable minimum and maximum transaction performance time ranges defined for your system.
Response Times Vs Service-level agreements (SLAs)

Average Transaction Response Time

<table>
<thead>
<tr>
<th>Transaction Type</th>
<th>Response Time</th>
<th>Cycle 1-1300 users</th>
<th>Cycle 2-1500 users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claims Login</td>
<td>0.632</td>
<td>0.609</td>
<td></td>
</tr>
<tr>
<td>Corres. Login</td>
<td>0.587</td>
<td>0.892</td>
<td></td>
</tr>
<tr>
<td>Claims Enter ID</td>
<td>0.07</td>
<td>0.068</td>
<td></td>
</tr>
<tr>
<td>Corres. Enter ID</td>
<td>0.066</td>
<td>0.063</td>
<td></td>
</tr>
<tr>
<td>Claims View Image</td>
<td>0.59</td>
<td>0.533</td>
<td></td>
</tr>
<tr>
<td>Corres. View Image</td>
<td>0.658</td>
<td>0.536</td>
<td></td>
</tr>
<tr>
<td>Claims Item Close</td>
<td>0.594</td>
<td>0.581</td>
<td></td>
</tr>
<tr>
<td>Corres. Item Close</td>
<td>0.622</td>
<td>0.611</td>
<td></td>
</tr>
<tr>
<td>Claims Logout</td>
<td>0.595</td>
<td>0.586</td>
<td></td>
</tr>
<tr>
<td>Corres. Logout</td>
<td>0.594</td>
<td>0.586</td>
<td></td>
</tr>
</tbody>
</table>
Monitor Database Server

![Analysis Summary Report](image)

**SQL Server**

### Resource Usage

<table>
<thead>
<tr>
<th>Resource</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Op Time (PhysicalDisk, Total op time)</td>
<td>0</td>
<td>0.856</td>
<td>21.264</td>
<td>1.542</td>
</tr>
<tr>
<td>Processor Time (Processor, Total cpu wait)</td>
<td>0</td>
<td>0.391</td>
<td>37.982</td>
<td>0.929</td>
</tr>
<tr>
<td>Current Disk Queue Length (PhysicalDisk, Total qps)</td>
<td>0</td>
<td>0.052</td>
<td>14</td>
<td>0.604</td>
</tr>
<tr>
<td>Memory Usage (System, qps)</td>
<td>28.346</td>
<td>520.803</td>
<td>803.051</td>
<td>87.643</td>
</tr>
<tr>
<td>Page Faults/sec (Memory, qps)</td>
<td>0</td>
<td>0.757</td>
<td>26.857</td>
<td>155.019</td>
</tr>
<tr>
<td>Page Faults/sec (Memory, qps)</td>
<td>25.906</td>
<td>192.341</td>
<td>552.151</td>
<td>322.737</td>
</tr>
<tr>
<td>Log File Operations/sec (Log)</td>
<td>0</td>
<td>0.150</td>
<td>0.557</td>
<td>0.428</td>
</tr>
<tr>
<td>SQL Server Log File Operations/sec (Log)</td>
<td>1.382488</td>
<td>3.983803</td>
<td>3.397452</td>
<td>1.264952</td>
</tr>
<tr>
<td>Process Queue Length (System, qps)</td>
<td>0</td>
<td>0.256</td>
<td>35.667</td>
<td>4.961</td>
</tr>
<tr>
<td>Thread (Objects, qps)</td>
<td>566</td>
<td>820.467</td>
<td>791</td>
<td>43.739</td>
</tr>
</tbody>
</table>

**Description**: Displays a summary of SQL Server Resources.
Monitor Database Server on Mainframe

### Database Server Timings – Application AAA

<table>
<thead>
<tr>
<th>Batch Job / Metric</th>
<th>Cycle: 1B 3.5mil records 09/15 – 09/22</th>
<th>Cycle: 2 5.4mil records 09/22 – 90/30</th>
<th>Cycle: 3 11.6mil records 10/01 – 10/10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GXX00000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Total Elapsed Time</td>
<td>01:44:03.25</td>
<td>00:55:29.78</td>
<td>06:10:12.24</td>
</tr>
<tr>
<td>SQL CPU Time</td>
<td>00:04:27.76</td>
<td>00:01:53.88</td>
<td>00:31:02.91</td>
</tr>
<tr>
<td>SQL I/O Elapsed Time</td>
<td>01:15:38.35</td>
<td>00:38:32.57</td>
<td>03:38:11.97</td>
</tr>
<tr>
<td>SQL Wait for Read</td>
<td>00:00:13.71</td>
<td>00:00:07.06</td>
<td>00:00:08.20</td>
</tr>
<tr>
<td>SQL Lock/Latch Time</td>
<td>00:00:00.00</td>
<td>00:00:00.09</td>
<td>00:00:00.57</td>
</tr>
<tr>
<td><strong>Total SQL Elapsed Time</strong></td>
<td><strong>01:57:23.10</strong></td>
<td><strong>01:12:36.65</strong></td>
<td><strong>06:26:08.99</strong></td>
</tr>
</tbody>
</table>

### DML Comparisons – Application AAA

<table>
<thead>
<tr>
<th>Batch Job / Metric</th>
<th>Cycle: 1B 3.5mil records 09/15 – 09/22</th>
<th>Cycle: 2 5.4mil records 09/22 – 90/30</th>
<th>Cycle: 3 11.6mil records 10/01 – 10/10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GXX00000V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Cursor Open / Closes</td>
<td>3,235</td>
<td>1,336</td>
<td>8,462</td>
</tr>
<tr>
<td>Total Fetch</td>
<td>7,035</td>
<td>3,323</td>
<td>25,443</td>
</tr>
<tr>
<td>Total Select</td>
<td>441,554</td>
<td>170,307</td>
<td>2,284,024</td>
</tr>
<tr>
<td>Total Insert, Prepare, Delete</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Locks Suspended / Timeout</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max Page Locks Held</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total Waits for Lock/Latch</td>
<td>132</td>
<td>620</td>
<td>752</td>
</tr>
<tr>
<td>Locks Requested</td>
<td>112,028</td>
<td>5,594</td>
<td>2,149,614</td>
</tr>
</tbody>
</table>
Performance Testing Best Practices

• Performance testing requires a different mind set and skill-set to that of functional testing.
  - Best started early in the development life cycle whenever possible.

• Understanding business requirements and expectations.
  - User activities and behaviors is key to designing suitable tests.
  - Ensure tests represent realistic usage of the application.

• Automated test tools coupled with fast backup and restore mechanisms are essential due to the need to repeat tests multiple times.

• System bottlenecks can rapidly become very technical in nature and consume considerable resources and efforts to diagnose.
  - Resolution may require considerable re-work and even redesign.
Benefits of Performance Testing

• The ultimate goal of all performance testing is to optimize performance while minimizing costs.
• Does the application perform as required?. Will it continue to perform as required?.
• Will it meet current and future performance requirements for the least cost?.
• Improve new or upgraded applications and third-party services quality, workflow and reporting.
• Reduce system downtime by isolating and resolving infrastructure bottlenecks, if any.
• Improve quality from a user’s prospective.
• Early identification of major application defects and architectural issues Reduce cost of change, system costs and increase profits.

<table>
<thead>
<tr>
<th>Cost per Defect</th>
<th>Cost per Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Test</td>
</tr>
<tr>
<td>$1x</td>
<td>$10x</td>
</tr>
<tr>
<td>Production</td>
<td></td>
</tr>
<tr>
<td>$100x</td>
<td></td>
</tr>
</tbody>
</table>

Cost Distribution - x1000
Summary and Conclusion

In this presentation we have covered…

• Performance Testing and its need to start early in the development process.
• Performance Testing Objectives and Engagements.
• Performance Testing Best Practices.
• Benefits of Performance Testing.
Questions?

For More Information contact:
Deepak Sharma
Email: Deepak.Sharma@Phooenixwm.com
PH: (860)4032195(O), (919)9397091(C)