Automatic Detection of Performance Deviations in Load Testing of Large Scale Systems

Haroon Malik
Software Analysis and Intelligence Lab (SAIL)
Queen’s University, Kingston, Canada
Large scale systems need to satisfy performance constraints.
PERFORMANCE PROBLEMS

• System not responding fast enough
• Taking too much of an important resource(s)
• Hanging and/or crashing under heavy load

Symptoms Include:
• High response time
• Increased Latency &
• Low throughput under load
Performance Analysts use load testing to detect early performance problems in the system before they become critical field problems.
LOAD TESTING STEPS

1. Environment Setup
2. Load Test Execution
3. Load Test Analysis
4. Report Generation
LOAD TESTING STEPS

1. Environment Setup
2. Load Test Execution
3. Load Test Analysis
4. Report Generation
LOAD TESTING STEPS

1. Environment Setup
2. Load Test Execution
3. Load Test Analysis
4. Report Generation
2. LOAD TEST EXECUTION

LOAD GENERATOR - 1

LOAD GENERATOR - 2

MONITORING TOOL

SYSTEM

PERFORMANCE REPOSITORY
LOAD TESTING STEPS

1. Environment Setup
2. Load Test Execution
3. Load Test Analysis
4. Report Generation
LOAD TESTING STEPS

1. Environment Setup
2. Load Test Execution
3. Load Test Analysis
4. Report Generation
LOAD TESTING STEPS

1. Environment Setup
2. Load Test Execution
3. Load Test Analysis
4. Report Generation
CHALLENGES WITH LOAD TEST ANALYSIS

1. Large Number of Counters
2. Limited Time
3. Limited Knowledge
CHALLENGES WITH LOAD TEST ANALYSIS

1. Large Number of Counters
2. Limited Time
3. Limited Knowledge
CHALLENGES WITH LOAD TEST ANALYSIS

1. Large Number of Counters

2. Limited Time

3. Limited Knowledge
I Propose 4 Methodologies

3 Unsupervised
1 Supervised

To Automatically Analyze the Load Test Results
Use Performance Counters to Construct Performance Signature
PERFORMANCE COUNTERS ARE HIGHLY CORRELATED

- CPU
- DISK (IOPS)
- MEMORY
- NETWORK
- TRANSACTIONS/SEC
HIGH LEVEL OVERVIEW OF OUR METHODOLOGIES

Input Load Test

Baseline Test

New Test

Data Preparation

Sanitization Standardization

Signature Generation

Deviation Detection

Performance Report
UNSUPERVISED SIGNATURE GENERATION

Random Sampling Methodology

- Load Test
- Random Sampling
- Signature

Clustering Methodology

- Load Test
- Data Reduction
  - Clustering
- Extracting Centroids
- Signature

- Analyst tunes weight parameter
- Dimension Reduction (PCA)
- Identifying Top k Performance Counters
  - Mapping
  - Ranking
- Signature
SUPERVISED SIGNATURE GENERATION

WRAPPER Methodology

Prepared Load Test

Labeling (only for baseline)

Partitioning the Data

Attribute Selection

Genetic Search

OneR

Identifying Top k Performance Counters

i. Count

ii. % Frequency

SPC1

SPC2

SPC10

Signature

<table>
<thead>
<tr>
<th>No.</th>
<th>Performance Counter Variables</th>
<th>10-Fold Selection</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>% CPU Time</td>
<td>10</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>2</td>
<td>% Disk Time</td>
<td>1</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>3</td>
<td>Disk sec/Write</td>
<td>2</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>4</td>
<td>Available Bytes</td>
<td>5</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>5</td>
<td>Pages/sec</td>
<td>4</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>6</td>
<td>Database Cache Request/Sec</td>
<td>5</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>7</td>
<td>Network Interface Bytes</td>
<td></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>8</td>
<td>% CPU Idle Time</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Datagram Rec/sec</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Avg. Disk Read Queue Length</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DEVIAION DETECTION TECHNIQUES

Using Control Chart

- Upper Control Limit
- Average
- Lower Control Limit

For Clustering and Random Sampling Methodologies

Using Methodology-Specific Techniques

For PCA and WRAPPER Methodologies
The Upper/Lower Control Limits (U/LCL) are the upper/lower limit of the range of a counter under the normal behavior of the system.
DEVIATION DETECTION

Clustering and Random Sampling

PCA Approach

WRAPPER Approach
CASE STUDY

How effective are our signature-based approaches in detecting performance deviations in load tests?
How effective are our signature-based approaches in detecting performance deviations in load tests?

**Evaluation** Using: Precision, Recall and F-measure

An **Ideal approach** should predict a minimal and correct set of performance deviations.
SUBJECT OF STUDY

System: **Industrial System**
Domain: Telecom

**Type of data:**
1. Load Test Repository
2. Data From Our Experiments on the Company’s Testing Platform

System: **Open Source**
Domain: Ecommerce

**Type of data:**
1. Data From Our Experiments with an Open Source Benchmark Application
# FAULT INJECTION

<table>
<thead>
<tr>
<th>Category</th>
<th>Faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Failure</td>
<td>CPU Stress</td>
</tr>
<tr>
<td></td>
<td>Memory Stress</td>
</tr>
<tr>
<td></td>
<td>Abnormal Workload</td>
</tr>
<tr>
<td>Operator Errors</td>
<td>Interfering Workload</td>
</tr>
<tr>
<td></td>
<td>Unscheduled Replication</td>
</tr>
</tbody>
</table>
CASE STUDY FINDINGS

Effectiveness
Precision/Recall/F-measure

Practical Differences
- **Random Sampling** has the lowest effectiveness.

- On Avg. and in all experiments, **PCA** performs better than **Clustering** approach.

- **WRAPPER** dominates the best supervised approach, i.e., **PCA**.
Overall, there is an excellent balance of high precision and recall of both the **WRAPPER** and **PCA** approaches (on average **0.95, 0.94** and **0.82, 0.84** respectively) for deviation detection.
CASE STUDY FINDINGS
(Practical Differences)

Real Time Analysis

Stability

Manual Overhead
REAL TIME ANALYSIS

- **WRAPPER** --- deviations on a per-observation basis.

- **PCA** --- requires a certain amount of observations (wait time).
We refer to ‘Stability’ as the ability of an approach to remain effective while its signature size is reduced.
WRAPPER methodology is more stable than PCA approach.
WRAPPER approach requires all observations of the baseline performance counter data to be labeled as *Pass/Fail*
MANUAL OVERHEAD

Marking each observation is time consuming
HIGHLEVEL OVERVIEW OF OUR METHODOLOGIES

UNSUPERVISED SIGNATURE GENERATION

DEVIAION DETECTION

SUBJECT OF STUDY

BlackBerry
System: Industrial System
Domain: Telecom

Open Source
Domain: Ecommerce

Type of data:
1. Load Test Repository
2. Data From Our Experiments on the Company’s Testing Platform

Type of data:
1. Data From Our Experiments with and Open Source Benchmark Application

CASE STUDY FINDINGS (Effectiveness)

Overall, there is an excellent balance of high precision and recall of both the WRAPPER and PCA approaches (on average 0.95, 0.94 and 0.82, 0.84 respectively) for deviation detection.