Root-cause Analysis of Performance Anomalies in Web-based Applications

João Paulo Magalhães (CIICESI, ESTGF.IPP)
Luis Moura Silva (CISUC, University of Coimbra)

20 April 2011
CISUC SSE 2011
Outline

The problem

Research goals

Framework

Root-cause Analysis of Performance Anomalies

Experimental evaluation

Conclusion and future work
The problem

Performance Anomalies

- Stress
- Financial Issues
Examples of impact

Greg Linden (Amazon)

"+500ms latency @Google = -20% of its traffic"
"+100ms latency @Amazon = -1% of sales"
Examples of impact

Greg Linden (Amazon)

"+500ms latency @Google = -20% of its traffic"
"+100ms latency @Amazon = -1% of sales"

Sean Power (Metrics 101)

"Between 37% to 49% of users who experience performance issues when completing a transaction will either abandon the site or switch to a competitor. Of these, 77% will share their experiences with others."

Examples of impact

Aberdeen Group

"Issues with application performance are affecting overall business revenues by up to 9%"
Examples of impact

Aberdeen Group

"Issues with application performance are affecting overall business revenues by up to 9%"

Aberdeen Group

"A 1-second delay in page load time equals 11% fewer page views, a 16% decrease in customer satisfaction, and 7% loss in conversions. (In dollar terms, this means that if your site typically earns $100,000 a day, this year you could lose $2.5 million in sales.)"

My research goals

- Timely detection of performance anomalies
- Root-cause analysis for the observable performance anomalies
- Prediction of performance anomalies
- Proactive recovery (self-healing web servers)
- Easy to apply on existing/legacy web applications
Framework

- Monitoring
- Workload Variation
- System Dimensioning
- Anomaly Detector
- Recovery
- Change Analysis
  - APP
  - SYS
  - CONF
- Root-cause Failure Analysis
Root-cause analysis

1. Identify the components associated with a performance anomaly

2. Provide timely analysis of root-cause

3. Distinguish if a performance anomaly is related to:
   i. application change
   ii. system change
   iii. remote service change
Monitoring - Aspect-Oriented Programming

- online
- easy to apply and do not require source code changes
- collect multiple system and application parameters
Monitoring - Overhead

at $t_1$ and $t_n$

-1% throughput : +2ms latency

from $t_1$ to $t_n$

-6% throughput : +313ms latency

at $t_1$ and $t_n$ and (from $t_2$ to $t_{n-1}$ if $j \% 15 = \text{true}$)

-1% throughput : +11ms latency
Data analysis: overall approach

1. Distinguish workload variations from performance anomalies

2. Verify existence of system or application server changes

3. Pinpointing
1) Workload Change or Performance Anomaly?

\[ r = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}} \]

**Interpretation**

<table>
<thead>
<tr>
<th>Correlation Coefficient</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>([-1.0, -0.5])</td>
<td>large</td>
</tr>
<tr>
<td>([0.5, 1.0])</td>
<td>large</td>
</tr>
<tr>
<td>([-0.49, -0.3])</td>
<td>medium</td>
</tr>
<tr>
<td>([0.3, 0.49])</td>
<td>medium</td>
</tr>
<tr>
<td>([-0.29, -0.1])</td>
<td>small</td>
</tr>
<tr>
<td>([0.1, 0.29])</td>
<td>small</td>
</tr>
<tr>
<td>([-0.09, 0.09])</td>
<td>no correlation</td>
</tr>
</tbody>
</table>
2) System or Application Server Change?

\[ r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2} \sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \]
Pinpointing: 3) Internal or Remote Service Change?

We track how much the total response time variance explains each one of the components response time

• Anova - coefficient of determination:

\[
R^2 = \frac{\sum_{i=1}^{n} (f_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\]

When a performance anomaly is detected:

• Observe which component(s) become more explanatory for the regression
Experimental evaluation

- Benchmark application - TPC-W
- O.S. - Linux, kernel 2.6, 64-bits
- LAN - 1Gbps
- AOP - AspectJ
- Math tool - R
- Data aggregation - epoch=5sec
Leading questions

• Is the root-cause analysis able to identify the components associated with a performance anomaly?

• Is the pinpointing useful for further recovery?

• Does the root-cause analysis provide timely results?
Scenarios

A - Remote Service Change: Injected queries directly in the database. Queries involve tables which are missing indexes

B - Application Change: Instrumented one of the user transactions to gradually starts performing slowly
Remote service change

- Performance anomaly
- None of the system or application parameters reveals change
Remote service change - Root-cause analysis

Pinpointing: Calls interacting with the DB server

Transaction 4

Transaction 7
Application change

- Performance anomaly
- None of the system or application parameters reveals change
Application change - Root-cause analysis

Pinpointing:
Correctly identified the instrumented transaction and corresponding call
Experimental tests -conclusions

For all the scenarios:

- The root-cause analysis have correctly pinpointed the components associated with the performance anomaly

Timely detection:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Periods</th>
<th>+100ms</th>
<th>+2sec</th>
<th>#Req</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>before</td>
<td>10</td>
<td>0</td>
<td>124703</td>
</tr>
<tr>
<td></td>
<td>injected:pinpoint</td>
<td>15</td>
<td>7</td>
<td>2361</td>
</tr>
<tr>
<td></td>
<td>after</td>
<td>427</td>
<td>173</td>
<td>66953</td>
</tr>
<tr>
<td>B</td>
<td>before</td>
<td>3</td>
<td>0</td>
<td>105707</td>
</tr>
<tr>
<td></td>
<td>injected:pinpoint</td>
<td>2</td>
<td>0</td>
<td>2258</td>
</tr>
<tr>
<td></td>
<td>after</td>
<td>2142</td>
<td>1988</td>
<td>75884</td>
</tr>
</tbody>
</table>
Conclusions

Timely detect performance anomalies AND root-cause analysis

• supports IT staff job
• contributes to reduce the mean-time-to-detect
• contributes to reduce the mean-time-to-repair
Future work

- adaptive and selective monitoring
- forecasting of performance anomalies
- capacity planning / dynamic dimensioning