Challenges, Benefits and Best Practices of Performance Focused DevOps

Wolfgang Gottesheim

@gottesheim
Status Quo
NEXT EXIT
When do we find performance problems?

Developers → Unit/Integration Tests → Acceptance Tests → Capacity Tests → Release

Software Developers

- what my friends think I do
- what my mum thinks I do
- what users think I should do
- what testers think I do
- what I think I do
- what I really do

#Dynatrace
Developer V/s Tester

6

9
I'M JUST HERE

SO I WON'T GET FINED
I DON'T ALWAYS TEST MY CODE
BUT WHEN I DO I DO IT IN PRODUCTION
Production
What Operations tells Developers...
What Developers would like to know

Where is the user from? What Browser does he use?

This is the exact click path of that frustrated user

No wonder the user is frustrated when waiting 77s to load the landing page

But its not Server-Side related – it happens outside of our Data Center

We also seem to have JavaScript Errors on these pages for IE8
What Developers would like to know

Top contributor is save of Mage_Core_Model_Abstract

70% of that time comes from Sales_Model_Quote

Where are those calls coming from?
"I HAVE A DREAM"...
ONE TEAM  ONE GOAL
Because NOBODY wants this ...
The System is down at the moment.

We're working to resolve the issue as soon as possible. Please try again later.

In a hurry? You might be able to apply faster at our Marketplace call center. Call 1-800-318-2596@ to talk with one of our trained representatives about applying over the phone.

Sorry, something went wrong.

We're working on it and we'll get it fixed as soon as we can.
YES we know this

80% Dev Time in Bug Fixing

$60B Defect Costs

BUT

~80% of problems

caused by ~20% patterns

Broaden the View

• Define metrics that are understood across teams
• Share measurement methods and tools
• Make performance part of agile stories
But…
It’s a culture thing
Culture  Automation  Measure  Share
Starting from...

Developers → CI Server → Testing Environment → Release → Production Environment

#Dynatrace
...or maybe...

Developers — CI Server — Testing Environment — Release — Production Environment

#Dynatrace
We want to get to...

Developers → Commit Stage → Automated Acceptance Testing → Automated Capacity Testing → Release
METRICS to look at, WHY we want them, HOW they help
Queues and Pools

SOME MEN

JUST WANT TO WATCH
THE WORLD BURN
Online Banking: Slow Balance Check

1.69min ($=101s!$) To check balance!

87% spent in IIS

600! SQL Executions
#1: Time Spent in IIS?

<table>
<thead>
<tr>
<th>Method</th>
<th>Elapsed Time [ms]</th>
<th>Thread Name</th>
<th>Argument</th>
<th>Exec Total [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web request</td>
<td>0.00</td>
<td>&lt;0&gt;</td>
<td></td>
<td>101438.99</td>
</tr>
<tr>
<td>Synchronous Invocation</td>
<td>0.43</td>
<td>&lt;0&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronous Path (HTTP)</td>
<td>1.56</td>
<td>&lt;0&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProcessRequestNotificationHelper(IntPtr managedHttpContext, IntPtr nativeR)</td>
<td>1.69</td>
<td>&lt;no name&gt; &lt;32&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExecuteReader(CommandBehavior behavior)</td>
<td>26.16</td>
<td>&lt;no name&gt; &lt;32&gt;</td>
<td>dbo.TempGetStateItem</td>
<td>26.44</td>
</tr>
<tr>
<td>Asynchronous Invocation</td>
<td>28.13</td>
<td>&lt;no name&gt; &lt;32&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asynchronous Path (ASP.NET)</td>
<td>87189.02</td>
<td>&lt;no name&gt; &lt;30&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProcessRequestNotificationHelper(IntPtr managedHttpContext, IntPtr nativeR)</td>
<td>87189.09</td>
<td>&lt;no name&gt; &lt;30&gt;</td>
<td></td>
<td>14281.98</td>
</tr>
</tbody>
</table>

Thread 32 in IIS took **87s** to pass control to Thread 30 in ASP.NET.
#2: SQL Executions!

A new connection for every statement...
#2: SQL Executions! continued ...

<table>
<thead>
<tr>
<th>SQL</th>
<th>Execs/calling Trans.</th>
<th>Executions</th>
<th>Preparations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordTypeDetail</td>
<td>67.00</td>
<td>67</td>
<td>0</td>
</tr>
<tr>
<td>TypeByGroupValue</td>
<td>55.00</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td>AccountIDBySPID</td>
<td>39.00</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>GetWithdraw</td>
<td>36.00</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>PlanIDByXXX</td>
<td>33.00</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>GetDistribution</td>
<td>28.00</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>PlanXXXRules</td>
<td>24.00</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>Plan</td>
<td>24.00</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>GetAccountBalances</td>
<td>24.00</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>GetPlanDetail</td>
<td>24.00</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>FindWITHXXXPlanXXXPlanWitXXX</td>
<td>19.00</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>Get_SystemSettings</td>
<td>18.00</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Get_TypeValues</td>
<td>17.00</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>Find_EligibilityRules</td>
<td>16.00</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>GetInvestmentDetail</td>
<td>15.00</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>exec</td>
<td>13.00</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>exec</td>
<td>13.00</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>Put_EligibilityLog</td>
<td>12.00</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>AccountBalance</td>
<td>11.00</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

#1: Same SQL is executed 67! times

#2: NO PREPARATION because everything executed on new Connection
Lessons Learned!

ASP.NET Worker Thread Pool Sizing!

DB Connection Pools More Efficient SQL
Helpful Metrics

- Idle vs. Busy Threads
- # SQLs / Request
- # Opened connections
Push without Plan
Mobile Landing Page of Super Bowl Ad

Total size of ~ 20MB

Find all Key Performance Indicators (KPI) for the selected page

<table>
<thead>
<tr>
<th>First Request</th>
<th>6ms</th>
<th>On Server</th>
<th>5859ms</th>
<th>DNS</th>
<th>0ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Impression Time</td>
<td>277ms</td>
<td>On Client</td>
<td>1490ms</td>
<td>Connect</td>
<td>0ms</td>
</tr>
<tr>
<td>Onload Time</td>
<td>2018ms</td>
<td>Ø Interactive</td>
<td>152ms</td>
<td>Transfer</td>
<td>3341ms</td>
</tr>
<tr>
<td>Total Load Time</td>
<td>15065ms</td>
<td>Ø Wait</td>
<td>1973ms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total size of ~ 20MB


Following table lists details about all 14 different mime-types and their impact on caching

| MIME Type   | Count | Total Size [bytes] | Cached | Cached Size [bytes] | Short Cached | Short Cached Size ... | Not Cached | Not Cached Size [...]
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL</td>
<td>434</td>
<td>21314149</td>
<td>398</td>
<td>21250483</td>
<td>-</td>
<td>2389016</td>
<td>36</td>
<td>63666</td>
</tr>
<tr>
<td>image/jpeg</td>
<td>188</td>
<td>18387983</td>
<td>230</td>
<td>18356932</td>
<td>1</td>
<td>47789</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>image/png</td>
<td>56</td>
<td>568476</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>image/gif</td>
<td>28</td>
<td>286717</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

434 Resources in total on that page:
230 JPEGs, 75 PNGs, 50 GIFs, ...
### m.store.com redirects to www.store.com

**ALL** CSS and JS files are redirected to the www domain

This is a lot of time “wasted” especially on high latency mobile connections

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://m.store.com/grip_783785955/optimizedBundles/globalOptimized.css">http://m.store.com/grip_783785955/optimizedBundles/globalOptimized.css</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/css</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../optimizedBundles/globalOptimized.css" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.28</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/content/0005/771/822/universal_button_sprites.css">http://m.store.com/content/0005/771/822/universal_button_sprites.css</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/css</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../universal_button_sprites.css" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.13</td>
<td>0.13</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/content/0002/444/294/brand_buttons_and_assets_sprites.css">http://m.store.com/content/0002/444/294/brand_buttons_and_assets_sprites.css</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/css</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../brand_buttons_and_assets_sprites.css" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.12</td>
<td>0.12</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/content/0007/279/407/universal_bar_sprites.css">http://m.store.com/content/0007/279/407/universal_bar_sprites.css</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/css</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../universal_bar_sprites.css" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.28</td>
<td>0.28</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/css/common/layouts/en/universal.css">http://m.store.com/grid/css/common/layouts/en/universal.css</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/css</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../universal.css" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/css/browse/layouts/en/browse.css">http://m.store.com/css/browse/layouts/en/browse.css</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/css</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../browse.css" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.30</td>
<td>0.30</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/css/buy/shoppingBag/en/shoppingBag.css">http://m.store.com/grid/css/buy/shoppingBag/en/shoppingBag.css</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/css</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../shoppingBag.css" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.28</td>
<td>0.28</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grip_87832942/optimizedBundles/tealeafOptimized.js">http://m.store.com/grip_87832942/optimizedBundles/tealeafOptimized.js</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../tealeafOptimized.js" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/ju/components/layout/en/Pad2.js">http://m.store.com/grid/ju/components/layout/en/Pad2.js</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../Pad2.js" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.31</td>
<td>0.31</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/ju/common/layouts/en/pixelManagement/pixelManagementSe">http://m.store.com/grid/ju/common/layouts/en/pixelManagement/pixelManagementSe</a>...</td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../pixelManagement/pixelManagementSe..." alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/ju/buy/applicationFunctions.js">http://m.store.com/grid/ju/buy/applicationFunctions.js</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../applicationFunctions.js" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grip_N690406239/optimizedBundles/globalOptimized.js">http://m.store.com/grip_N690406239/optimizedBundles/globalOptimized.js</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../globalOptimized.js" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/ju/components/layout/en/DeviceSe">http://m.store.com/grid/ju/components/layout/en/DeviceSe</a>...</td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../DeviceSe..." alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/Asset_Archive/GPWeb/content/0007/390/104/assets/wcLib_052513.js">http://m.store.com/Asset_Archive/GPWeb/content/0007/390/104/assets/wcLib_052513.js</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../wcLib_052513.js" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/Asset_Archive/GPWeb/content/0006/955/577/cn955557.js">http://m.store.com/Asset_Archive/GPWeb/content/0006/955/577/cn955557.js</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../cn955557.js" alt="Time Chart" /></td>
<td>1.36</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
</tr>
<tr>
<td><a href="http://m.store.com/assets/common/clear.gif">http://m.store.com/assets/common/clear.gif</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>image/gif</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../clear.gif" alt="Time Chart" /></td>
<td>1.37</td>
<td>0.41</td>
<td>0.41</td>
<td>0.28</td>
</tr>
<tr>
<td><a href="http://m.store.com/content/0001/195/897/badgeLarge.gif">http://m.store.com/content/0001/195/897/badgeLarge.gif</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>image/gif</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../badgeLarge.gif" alt="Time Chart" /></td>
<td>1.37</td>
<td>0.42</td>
<td>0.42</td>
<td>0.28</td>
</tr>
<tr>
<td><a href="http://m.store.com/webcontent/0005/895/615/cn5895616.jpg">http://m.store.com/webcontent/0005/895/615/cn5895616.jpg</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>image/jpeg</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../5895616.jpg" alt="Time Chart" /></td>
<td>1.37</td>
<td>0.41</td>
<td>0.41</td>
<td>0.12</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/ju/components/layout/en/reporting/sCode.js">http://m.store.com/grid/ju/components/layout/en/reporting/sCode.js</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>application/javascript</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../sCode.js" alt="Time Chart" /></td>
<td>1.37</td>
<td>0.41</td>
<td>0.41</td>
<td>0.12</td>
</tr>
<tr>
<td><a href="http://m.store.com/grid/assets/common/en/lineitemDisplayBackground.gif">http://m.store.com/grid/assets/common/en/lineitemDisplayBackground.gif</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>image/gif</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../lineitemDisplayBackground.gif" alt="Time Chart" /></td>
<td>3.65</td>
<td>0.13</td>
<td>0.13</td>
<td>0.28</td>
</tr>
<tr>
<td><a href="http://m.store.com/assets/common/strike.gif">http://m.store.com/assets/common/strike.gif</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>image/gif</td>
<td>0</td>
<td><img src="https://s3.amazonaws.com/.../strike.gif" alt="Time Chart" /></td>
<td>3.80</td>
<td>0.13</td>
<td>0.13</td>
<td>0.28</td>
</tr>
<tr>
<td><a href="http://m.store.com/fisdoubleclick/actvitysource=click&amp;activitytype=domainclick&amp;domain=m.store.com">http://m.store.com/fisdoubleclick/actvitysource=click&amp;activitytype=domainclick&amp;domain=m.store.com</a></td>
<td>no</td>
<td>Net</td>
<td>302 Found</td>
<td>text/html</td>
<td>456</td>
<td><img src="https://s3.amazonaws.com/.../fisdoubleclick/actvitysource=click&amp;activitytype=domainclick&amp;domain=m.store.com" alt="Time Chart" /></td>
<td>4.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Fifa.com during Worldcup

Largest Items on page: 
*favicon with 370kb*

But also some heavyweight 
**CSS and JS files with > 150kb**

http://apmblog.dynatrace.com/2014/05/21/is-the-fifa-world-cup-website-ready-for-the-tournament/
Helpful Metrics

• Load Time
• # Images
• # Redirects
• # HTTP 3xx, 4xx, 5xx
• Size of resources
Reusing Components
Requirement: We need a report

Monitor trials

Only my ✓ Active only ✓

Normal Status Flow

Invite > Invitation Accepted > License Accepted > Trial Run > Terminated

Filter: 

3 trial(s) found, displaying 3 from 1 to 3. 

ID  Name                      Status              Running                  Creator                               Company  License Key   Events  Actions
---  ------------------------- -------------------- ------------------------- ------------------------------------ -------- -------------- ------- ----------------
606  Przemyslaw                Invited             ✔️                         andreas.grabner@compuware.com        ✔️        ✔️            1       Cancel Trial
605  Simon                     Invitation Accepted ✔️                         andreas.grabner@compuware.com        ✔️        ✔️            2       Cancel Trial
758  David                      Invitation Accepted ✔️                         andreas.grabner@compuware.com        ✔️        ✔️            2       Cancel Trial
Using Hibernate results in 4k+ SQL Statements to display 3 items!

Hibernate Executes 4k+ Statements

Individual Execution VERY FAST

But Total SUM takes 6s

http://apmblog.dynatrace.com/2014/04/23/database-access-quality-metrics-for-your-continuous-delivery-pipeline/
Helpful Metrics

• # SQL Statement Executions
• # of same SQLs
• Result Set Size
There are no shortcuts to perfection.

Tools and Frameworks
Online Bank: Transaction History CSV Download!

Problem: Takes 207s! To download. 87% of Time spent in Garbage Collection

Building CSV output in memory...
Online Store: Rendering Search Result

**Problem:** 4.4s to render result page

<table>
<thead>
<tr>
<th>Method</th>
<th>Argument</th>
<th>Exec. Total [ms]</th>
<th>Result</th>
<th>CSS</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronous Path (HTTP)</td>
<td></td>
<td>4,982.23</td>
<td>-</td>
<td>-</td>
<td>Web server</td>
</tr>
<tr>
<td>ProcessRequestNotification</td>
<td></td>
<td>120.32</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1 = GetProductCategories</td>
<td></td>
<td>1,210.20</td>
<td>-</td>
<td>-</td>
<td>ADONET</td>
</tr>
<tr>
<td>1 = Connection Acquire</td>
<td></td>
<td>8.62</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1 = CallEnd Unsynchronized (SearchResult)</td>
<td></td>
<td>4034.26</td>
<td>-</td>
<td>-</td>
<td>ASYNCH</td>
</tr>
<tr>
<td>ResultSearchViewModel model, String manufacture application, Nullable&lt;int32&gt;, int p</td>
<td></td>
<td>1908.73</td>
<td>-</td>
<td>-</td>
<td>SearchController</td>
</tr>
</tbody>
</table>

**Root Cause:** Custom RegEx Library with performance issues on large strings
Helpful Metrics

• Memory Usage
• Time Spent in APIs
Architectural Decisions
Project: Online Room Reservation System

• Symptoms
  • HTML takes between 60 and 120s to render
  • High GC Time

• Assumptions
  • Bad GC Tuning
  • Probably bad Database Performance as rendering was simple
Developers built own monitoring

```java
void roomreservationReport(int roomid) {
    long startTime = System.currentTimeMillis();
    Object data = loadDataForRoom(roomid);
    long dataLoadTime = System.currentTimeMillis() - startTime;
    generateReport(data, roomid);
}
```

**Result:**
Avg. Data Load Time: **45s**!

**DB Tool says:**
Avg. SQL Query: **<1ms**
#1: Loading too much data

24889! Calls to the Database API!

High CPU and High Memory Usage to keep all data in Memory
#2: On individual connections

Classical N+1 Query Problem

Individual SQL really <1ms

12444! individual connections
#3: Putting all data in temp Hashtable

Lots of time spent in Hashtable.get

Called from their Entity Objects
Lesson Learned

• Custom Measuring
  • Was impacted by Garbage Collection
  • Just measured overall time but not # SQL Executions

```java
void roomreservationReport(int roomid) {
    long startTime = System.currentTimeMillis();
    Object data = loadDataForRoom(roomid);
    long dataLoadTime = System.currentTimeMillis() - startTime;

    generateReport(data, roomid);
}
```

• Learn SQL and don’t use Hashtables as Workaround
Helpful Metrics

- # SQL Executions
- # of SAME SQLs
- Connection Acquisition Time
Performance as a Quality Gate

- Automated collection of performance metrics in test runs
- Automated analysis of performance metrics to identify outliers
- Comparison of performance metrics across builds
- Integration with build automation tools and practices
- Actionable data through deep transactional insight
- Automated notifications on performance issues in tests
- Measurements accessible and shareable across teams
PERFORMANCE as part of our Continuous Delivery Process

Developers → Commit Stage → Automated Acceptance Testing → Automated Capacity Testing → Release
Remember: Use Tools to measure...

- # Images
- # Redirects
- Size of Resources
- # SQL Executions
- # of SAME SQLs
- # of Connections
- Time Spent in API

- # Calls into API
- # Functional Errors
- 3rd Party calls
- # of Domains
- Total Page Size
- # Items per Page
- # AJAX per Page
If we do all that ...
Software Developers

- What my friends think I do
- What my mom thinks I do
- What users should do
- Bugs!!!
- What testers think I do
- What I think I do
- What I really do

Developers → Unit/Integration Tests → Acceptance Tests → Capacity Tests → Release
A FINAL THOUGHT!
Putting it into Test Automation

<table>
<thead>
<tr>
<th>Test Framework Results</th>
<th>Architectural Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Build #</strong></td>
<td><strong># SQL</strong></td>
</tr>
<tr>
<td>Build 17</td>
<td>12</td>
</tr>
<tr>
<td>testPurchase</td>
<td></td>
</tr>
<tr>
<td>testSearch</td>
<td>3</td>
</tr>
<tr>
<td>Build 18</td>
<td>12</td>
</tr>
<tr>
<td>testPurchase</td>
<td></td>
</tr>
<tr>
<td>testSearch</td>
<td>3</td>
</tr>
<tr>
<td>Build 19</td>
<td>75</td>
</tr>
<tr>
<td>testPurchase</td>
<td></td>
</tr>
<tr>
<td>testSearch</td>
<td>3</td>
</tr>
<tr>
<td>Build 20</td>
<td>9</td>
</tr>
<tr>
<td>testPurchase</td>
<td></td>
</tr>
<tr>
<td>testSearch</td>
<td></td>
</tr>
</tbody>
</table>

We identified a regression

Problem solved

Now we have the functional and architectural confidence

Let’s look behind the scenes

Exceptions probably reason for failed tests

Problem fixed but now we have an architectural regression
And in your Pipeline

**Unit & Integration Tests**
- Commit Stage
  - Compile
  - Execute Unit Test
  - Code Analysis
  - Build installers

**Automated Acceptance Testing**

**Performance Tests**
- Automated Capacity Testing

**Manual testing**
- Key showcases
- Exploratory testing

**Production Monitoring**
- Release

**Functional Tests**
Wolfgang Gottesheim
Follow me @gottesheim
Email me wolfgang.Gottesheim@dynatrace.com
http://blog.dynatrace.com