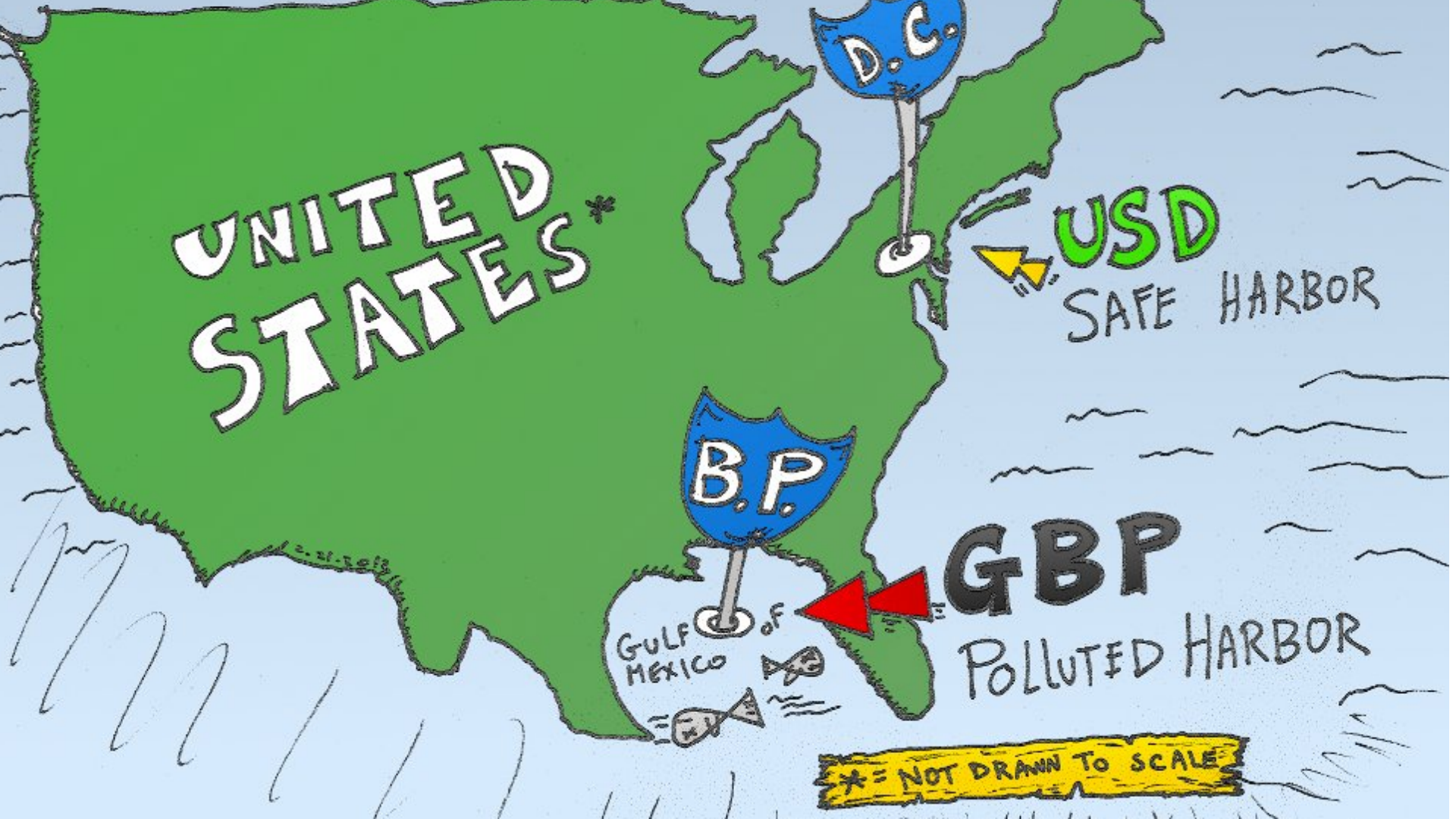
Java One 2015 – Deep Dive Top Performance Mistakes

And other Tips & Tricks to make you a “Performance Expert”

More on http://blog.dynatrace.com

Andreas Grabner - @grabnerandi


UNITED STATES*
D.C.
USD
SAFE HARBOR
B.P.
GBP
POLLUTED HARBOR
GULF OF MEXICO
*= NOT DRAWN TO SCALE

Available Tools

Mission Control

YourKit

Solaris Studio

VisualVM

AppDynamics

NetBeans Profiler

New Relic

JProfiler

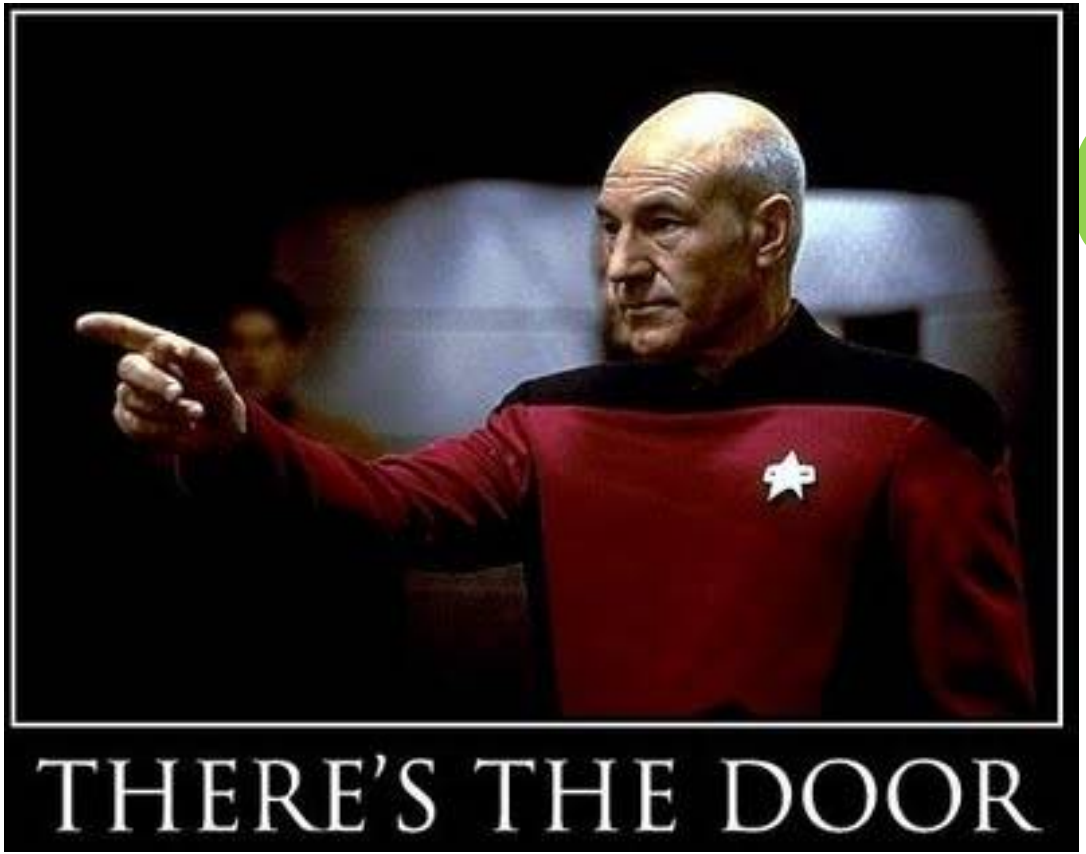
Honest Profiler

JProbe

Dynatrace

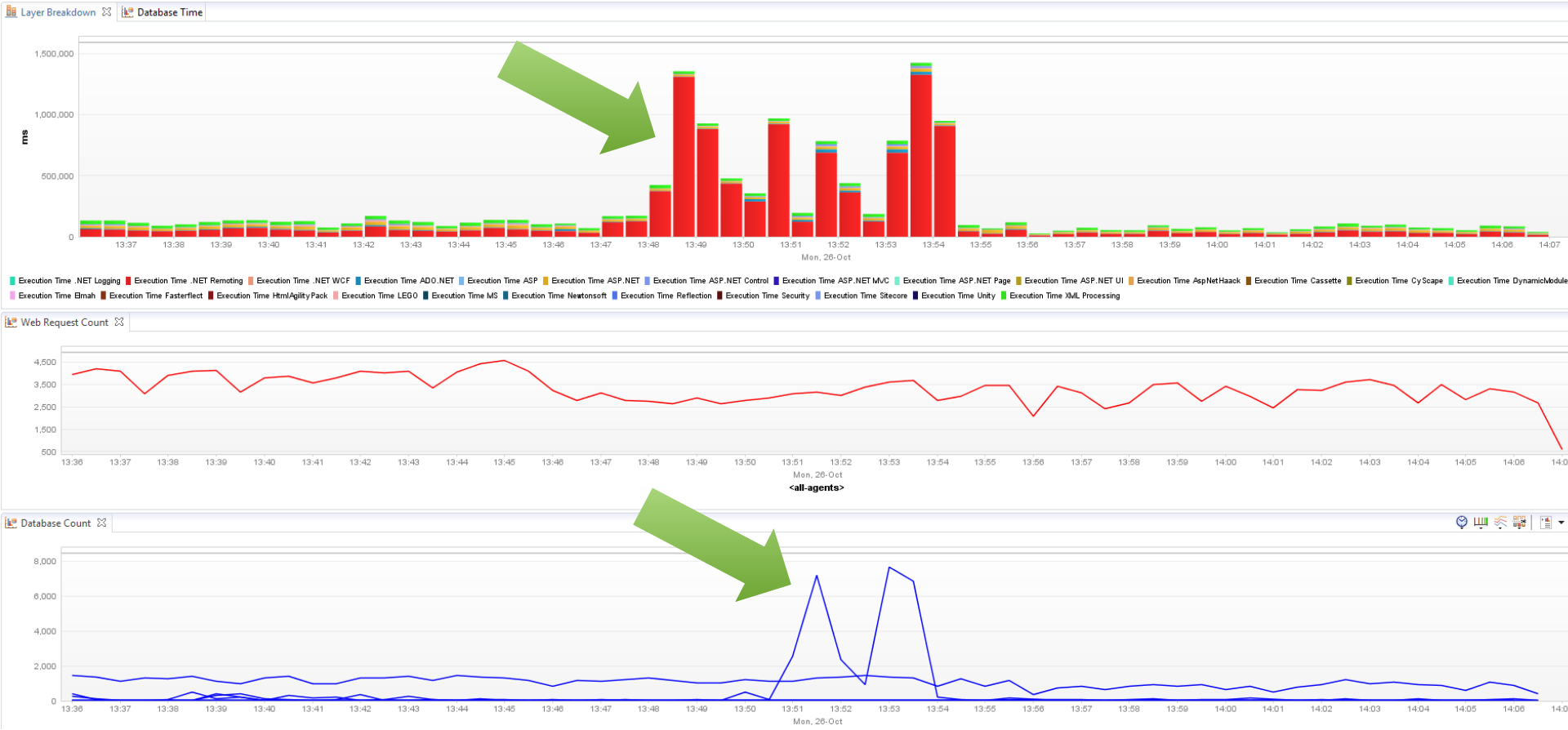
XRebel

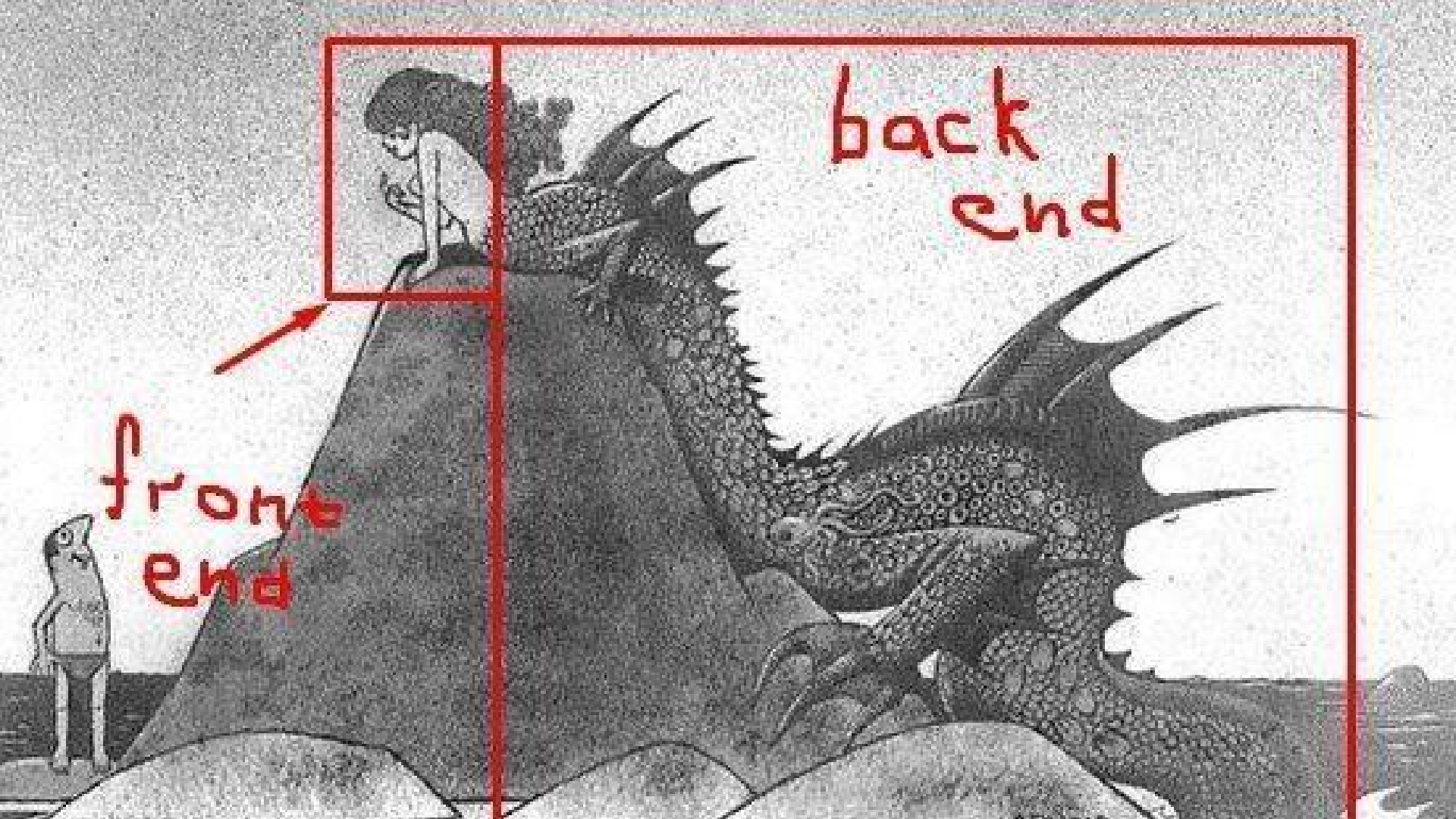
AND MANY MORE



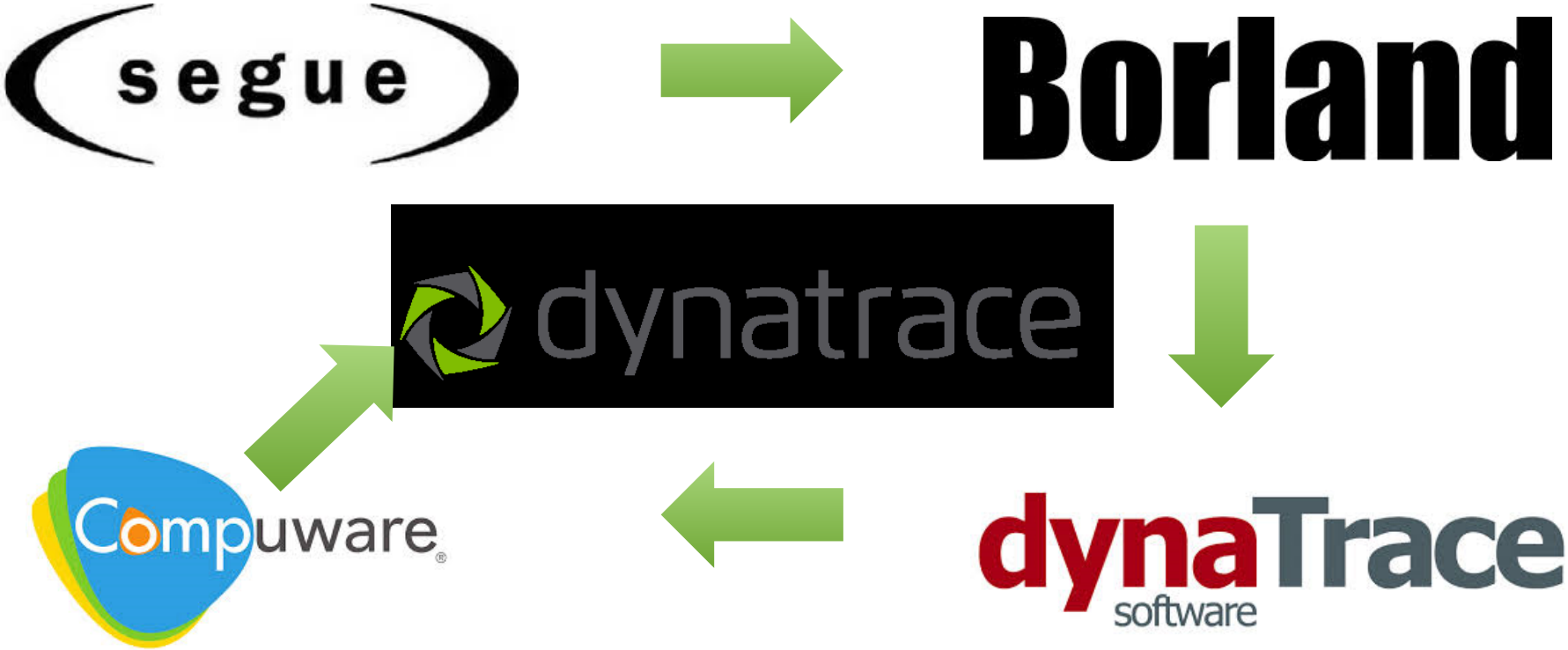
0.01ms

0.02ms


Layer Breakdown
Database Time
1,500,000
1,000,000
500,000
0
ms
13:37 13:38 13:39 13:40 13:41 13:42 13:43 13:44 13:45 13:46 13:47 13:48 13:49 13:50 13:51 13:52 13:53 13:54 13:55 13:56 13:57 13:58 13:59 14:00 14:01 14:02 14:03 14:04 14:05 14:06 14:07
Mon, 26-Oct
Execution Time .NET Logging
Execution Time .NET Remoting
Execution Time .NET WCF
Execution Time ADO.NET
Execution Time ASP
Execution Time ASP.NET
Execution Time ASP.NET Control
Execution Time ASP.NET MVC
Execution Time ASP.NET Page
Execution Time ASP.NET UI
Execution Time AspNetHaack
Execution Time Cassette
Execution Time CyScape
Execution Time DynamicModule
Execution Time Elmah
Execution Time Fasterflect
Execution Time HtmlAgilityPack
Execution Time LEGO
Execution Time MS
Execution Time Newtonsoft
Execution Time Reflection
Execution Time Security
Execution Time Sitecore
Execution Time Unity
Execution Time XML Processing
Web Request Count
4,500
3,500
2,500
1,500
500
13:36 13:37 13:38 13:39 13:40 13:41 13:42 13:43 13:44 13:45 13:46 13:47 13:48 13:49 13:50 13:51 13:52 13:53 13:54 13:55 13:56 13:57 13:58 13:59 14:00 14:01 14:02 14:03 14:04 14:05 14:06 14:0
Mon, 26-Oct
<all-agents>
Database Count
8,000
6,000
4,000
2,000
0
13:36 13:37 13:38 13:39 13:40 13:41 13:42 13:43 13:44 13:45 13:46 13:47 13:48 13:49 13:50 13:51 13:52 13:53 13:54 13:55 13:56 13:57 13:58 13:59 14:00 14:01 14:02 14:03 14:04 14:05 14:06 14:0
Mon, 26-Oct


back end
front end

15 Years: That’s why I ended up talking about performance



Where do your Stories come from?

#1: Real Life & Real User Stories

MrJM
@MisterJayEm

Follow

All #UnitedAirlines flights in U.S. are grounded, but they've got their best people working on it. #United

3:37 PM - 8 Jul 2015

46 33

RJ Belles @rjbelles · Sep 15

Great crowd at the @Dynatrace Seattle user group meeting #dynatrace #apm

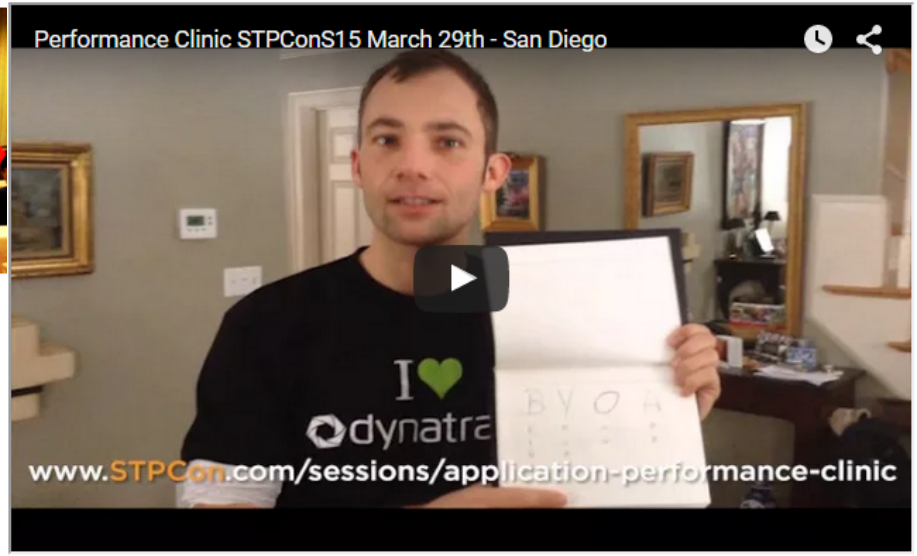
1

#2: http://bit.ly/onlineperfclinic

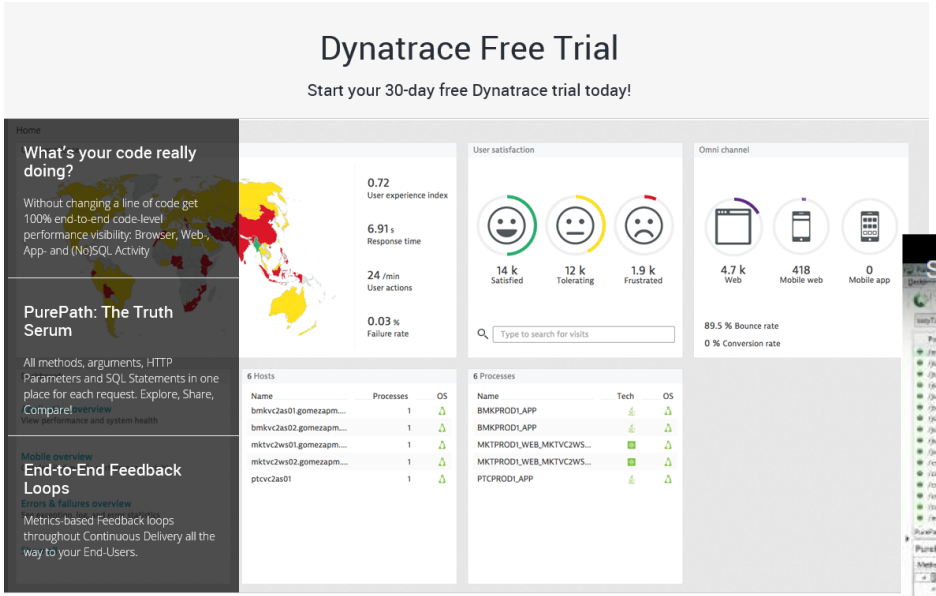
STP SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2015

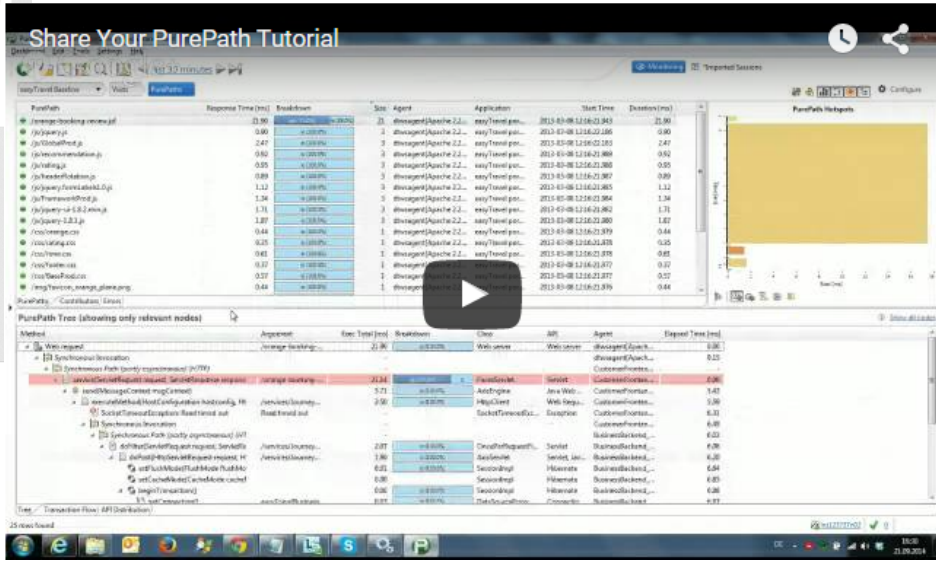
PRICING PROGRAM SPEAKERS SPONSORS HOTEL CONTACT US

Application Performance Clinic



#3: http://bit.ly/sharepurepath



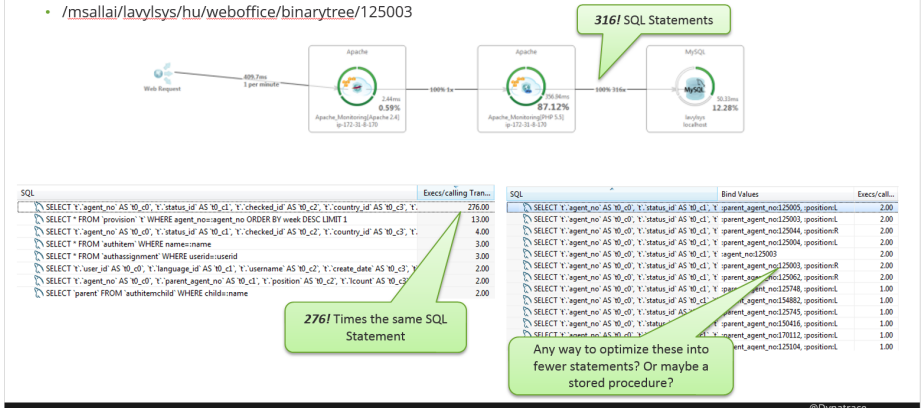


4 Problems Analyzed

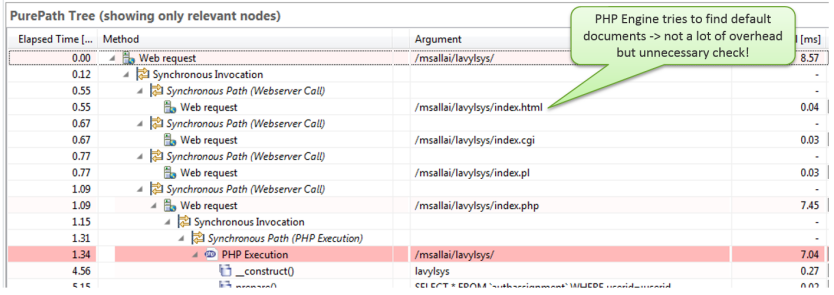
- Unnecessary PHP Document Lookups
  - Minor problem but still optimization potential
- Too many SQL Statements
  - Same statement call very often to iterate through tree structure
  - Maybe put into Stored Procedure?
- Slow SQL Statements
  - Optimize SQL or DB Table Indices?
- CDbCriteria called from BinarytreeController
  - Check implementation of BinarytreeController

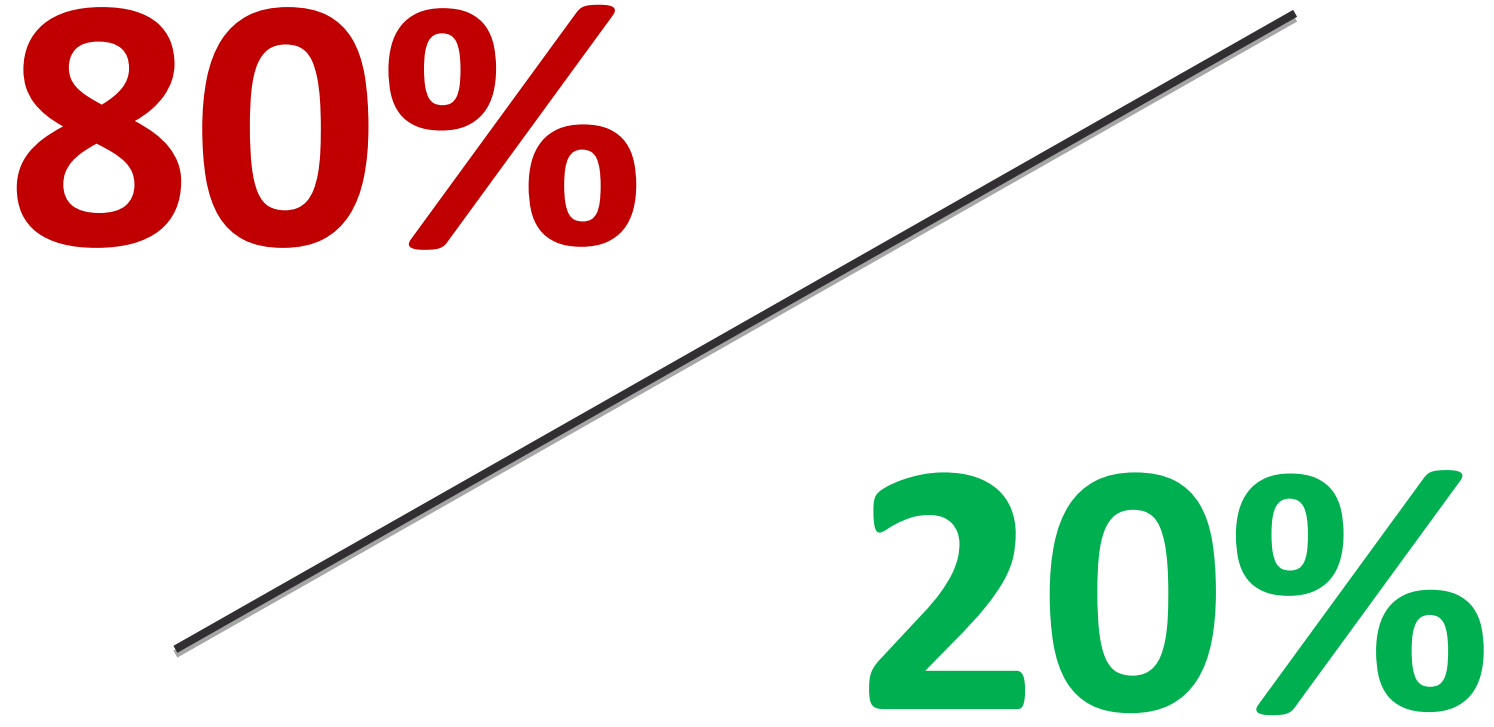
Too many SQL Statements

- /msallai/lavylsys/hu/weboffice/binarytree/125003



Unnecessary PHP Default Document Lookup




80%
20%


metrics
metric
action
achievable
relevant
scope
cost
indicator
strategy
balanced
improvement
marketing
methods
materials
jargon
goals
application
SMART
KPI
lagging
effectiveness
measurable
quantitative
progress
quality
safety
tool
state
brand
important
what
finance-based
employees
shareholders
directional
results
focus
comparison
high-value
time-phased


HavenWorks.com
News
American News
Democratic News:
Republican News:
Atwater Politics
Two Santa Clauses or How The Republican Party Has Conned America for Thirty Years.
Weblog
DEM 2010 TV
OBAMA TV
WHITE HOUSE TV
News' Media Politics
Right-Wing TV Bias
MEDIA BIAS: 2 to 1 in favor of Republicans
TV Online Television
Despite McCain's Comments, Senate GOP Not Offering Detailed Budget
WORLD
INTELLIGENCE
IRAQ NEWS
War Crimes
Poll: Most want inquiry into anti-terror tactics

HTTP Archive - Trends

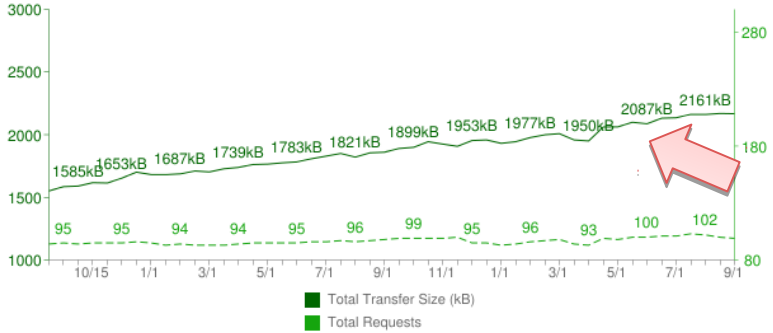
Andreas

httparchive.org/trends.php

http archive

trends stats websites about

Total Transfer Size & Total Requests



JS Transfer Size & JS Requests

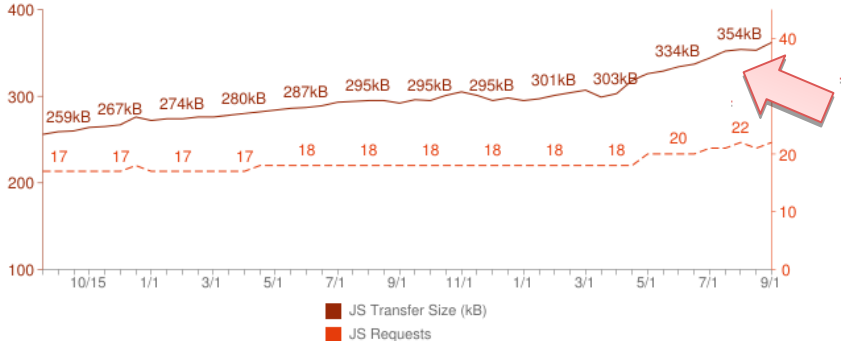
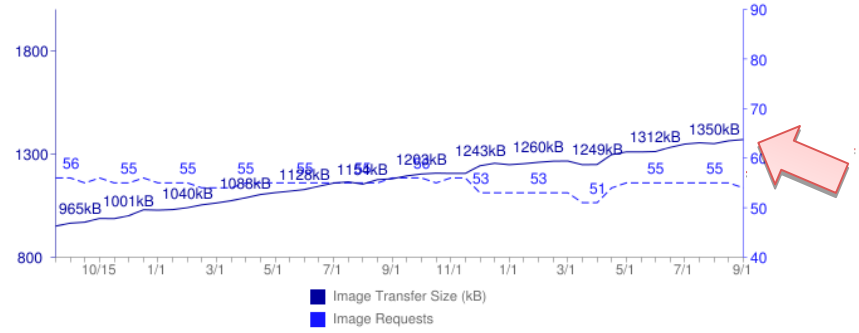
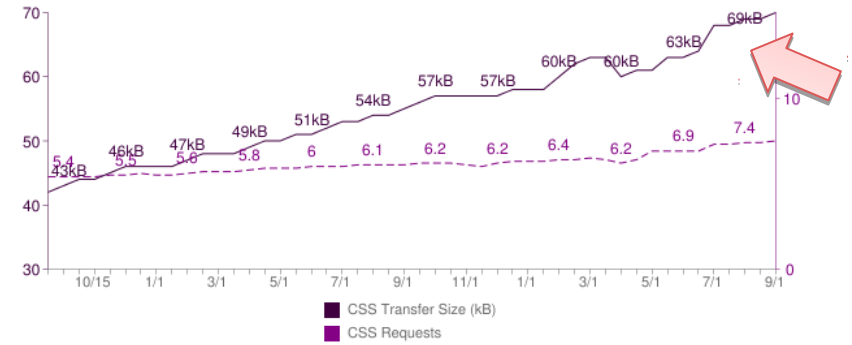


Image Transfer Size & Image Requests



CSS Transfer Size & CSS Requests



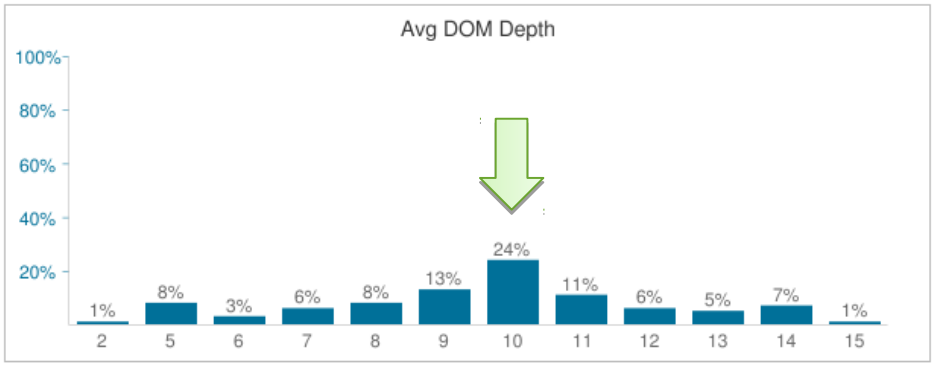
HTTP Archive - Trends

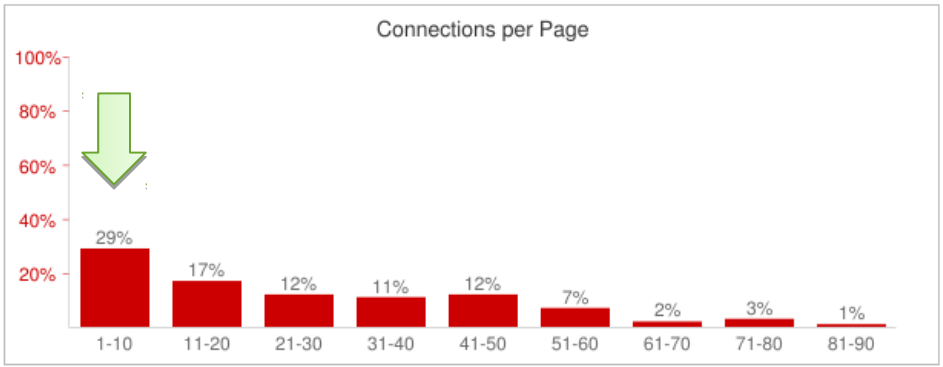
Andreas

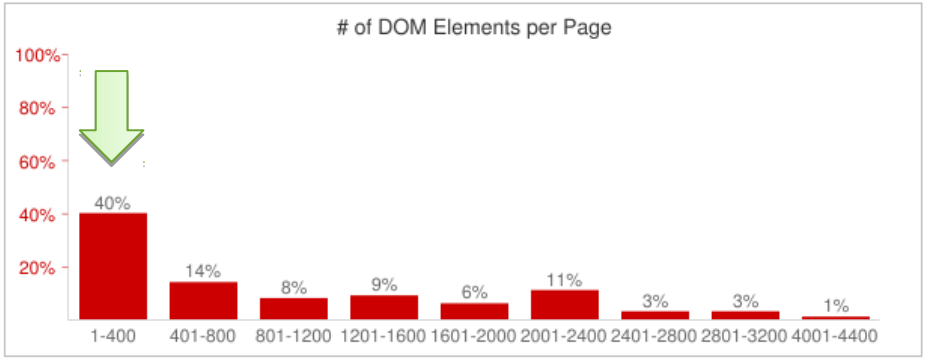
httparchive.org/trends.php

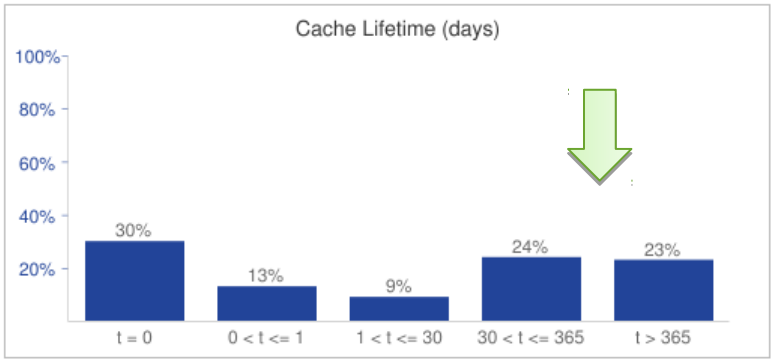
http archive

trends stats websites about

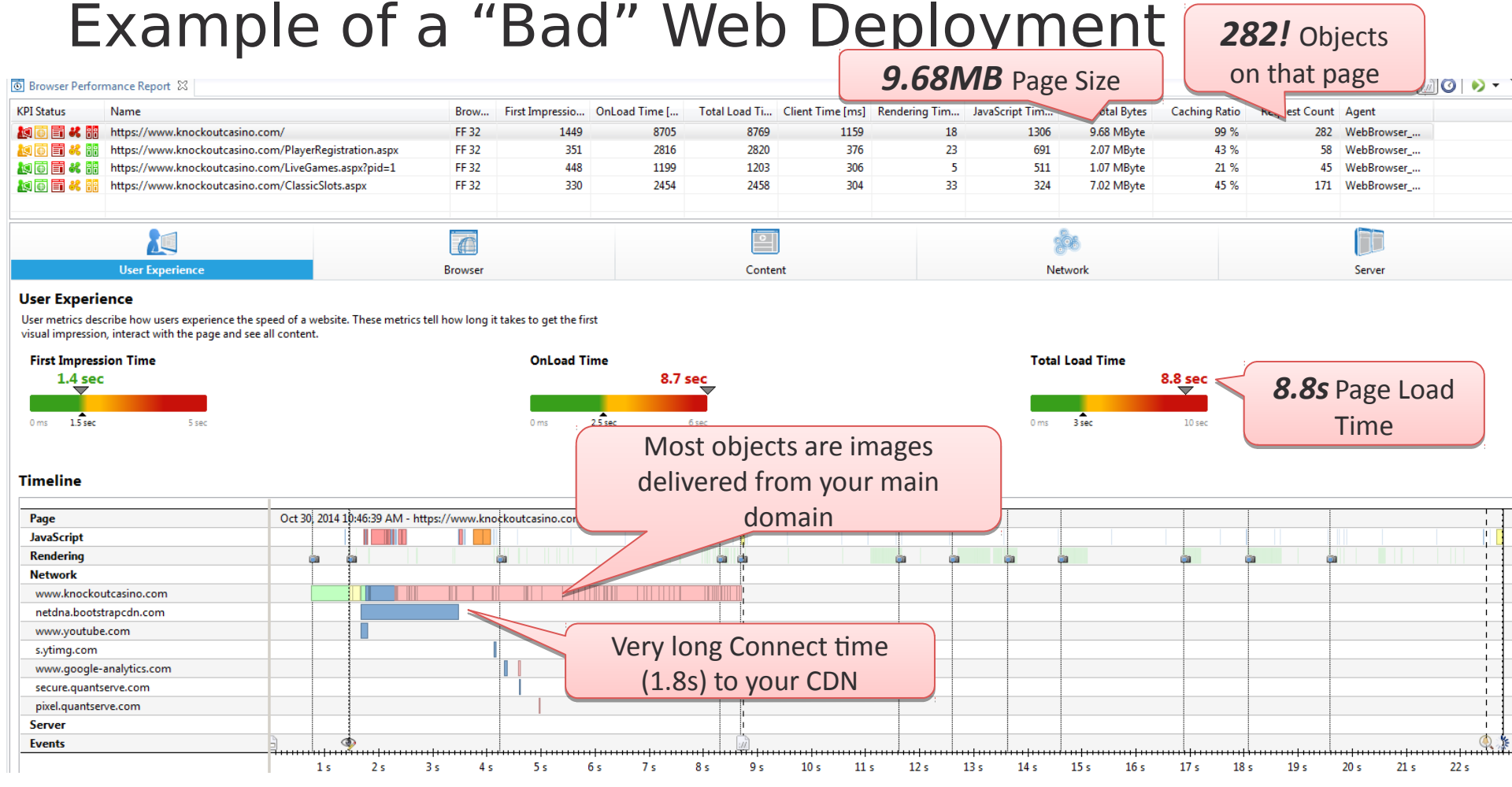








Example of a “Bad” Web Deployment



Mobile landing page of Super Bowl ad

Total size of ~
20MB

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

These values help you compare with other versions of the same page to identify problems or regressions .
Learn more on Key Performance Indicators and how they get calculated.

First Request	6ms	*On Server*	5859ms	*DNS*	0ms	*Network*	**16307ms**	*Total Size*	**20814kb**
First Impression Tin	277ms	*On Client*	1490ms	*Connect*	0ms	*JavaScript*	**3268ms**	*# of Requests*	437
OnLoad Time	2018ms	*Ø Interactive*	152ms	*Transfer*	3341ms	*Rendering*	**3402ms**	*# of XHR*	2
Total Load Time	15065ms			*Ø Wait*	1973ms				

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

Analyze which types of resources are cached vs. not cached and how much size you can safe by caching more of these resources

MIME	Count	Total Size [bytes]	Cached	Cached Size [bytes]	Short Cach...	Short Cached Size ...	Not Cached	Not Cached Size [...
TOTAL	434	21314149	398	21250483	-	2389916	36	63666
image/jpeg	230	18387943	230	18387943	1	47789	0	0
image/png	75	424580	75	424580	0	908	0	0
image/gif	50	5762	50	5762	-	-	0	0

434 Resources in total on that page:
230 JPEGs, 75 PNGs, 50 GIFs, …

Fifa.com during Worldcup

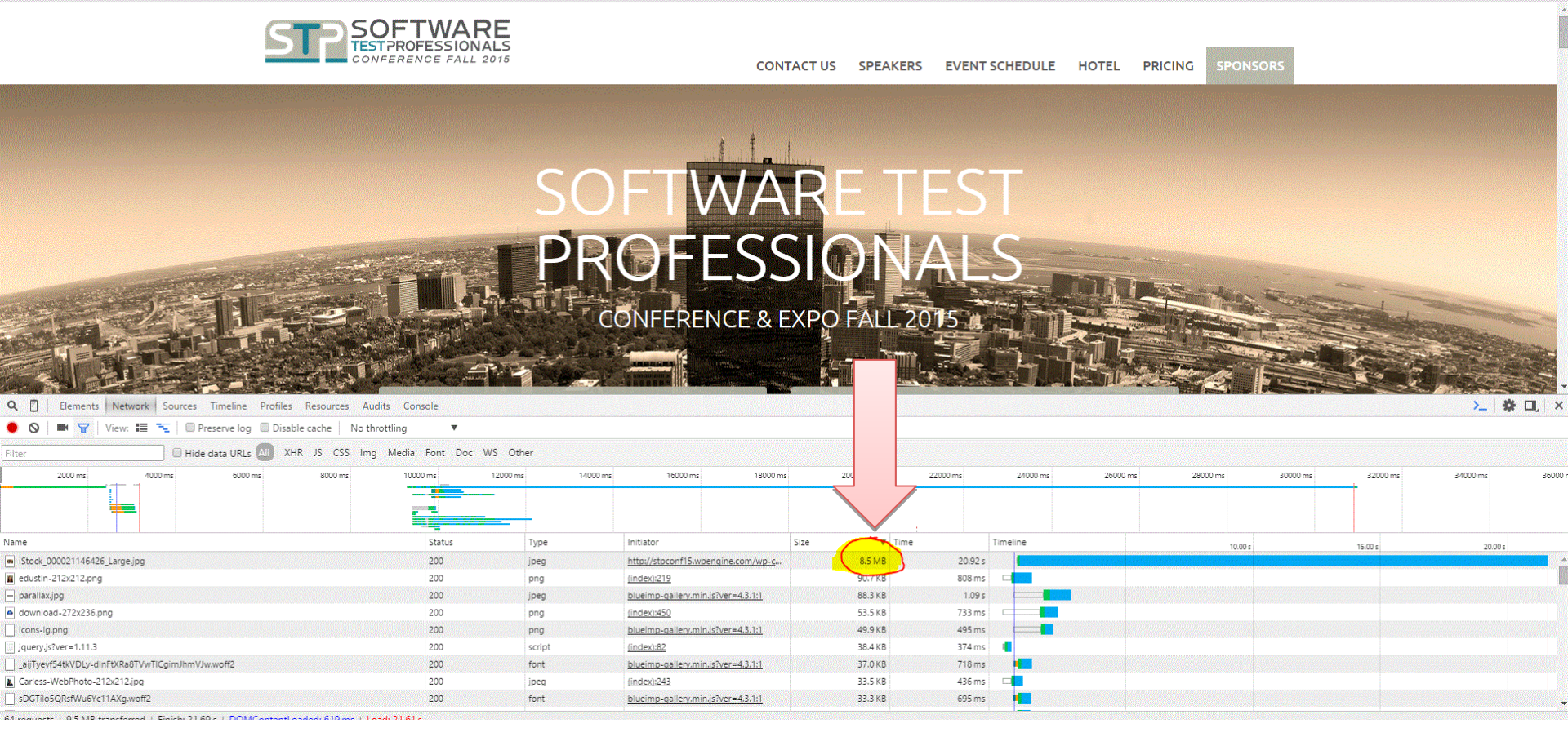
Largest Items on page:
favicon with 370kb

	URL	Size [byt...	Total [s]	MIME
	http://m.fifa.com/imgml/favicon/favicon.ico	370070	1.22	image/x-icon
CSS	http://css.m.fifa.com/components/style/framework/lang=en/base.css?v=20140506111842	288056	1.09	text/css
JS	http://js.m.fifa.com/components/script/require-libs/frameworks/bundle.js?v=635337579533169403	172349	0.64	application/x-javasc...
HTM	http://m.fifa.com/worldcup/	113996	1.00	text/html
CSS	http://css.m.fifa.com/components/style/framework/lang=en/worldcup/base.css?v=20140430173346	109845	0.92	text/css
JS	http://js.m.fifa.com/components/script/frameworks/lang=en/require-2.1.9.js?v=20140422110615	98981	1.07	application/x-javasc...
JS	http://partner.googleadservices.com/gpt/pubads_impl_38.js	78421	1.13	text/javascript

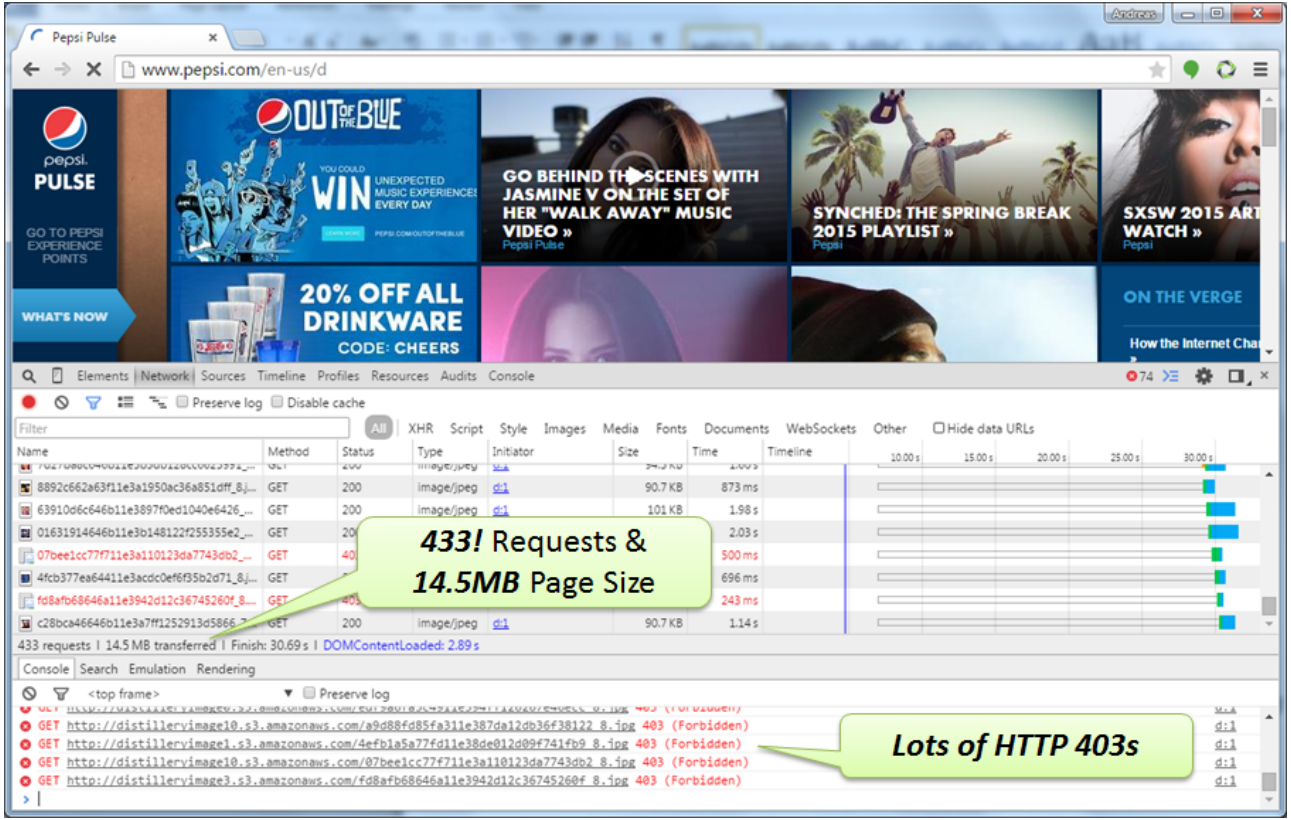
But also some heavyweight
CSS and JS files with > 150kb

Source: http://apmblog.compuware.com/2014/05/21/is-the-fifa-world-cup-website-ready-for-the-tournament/

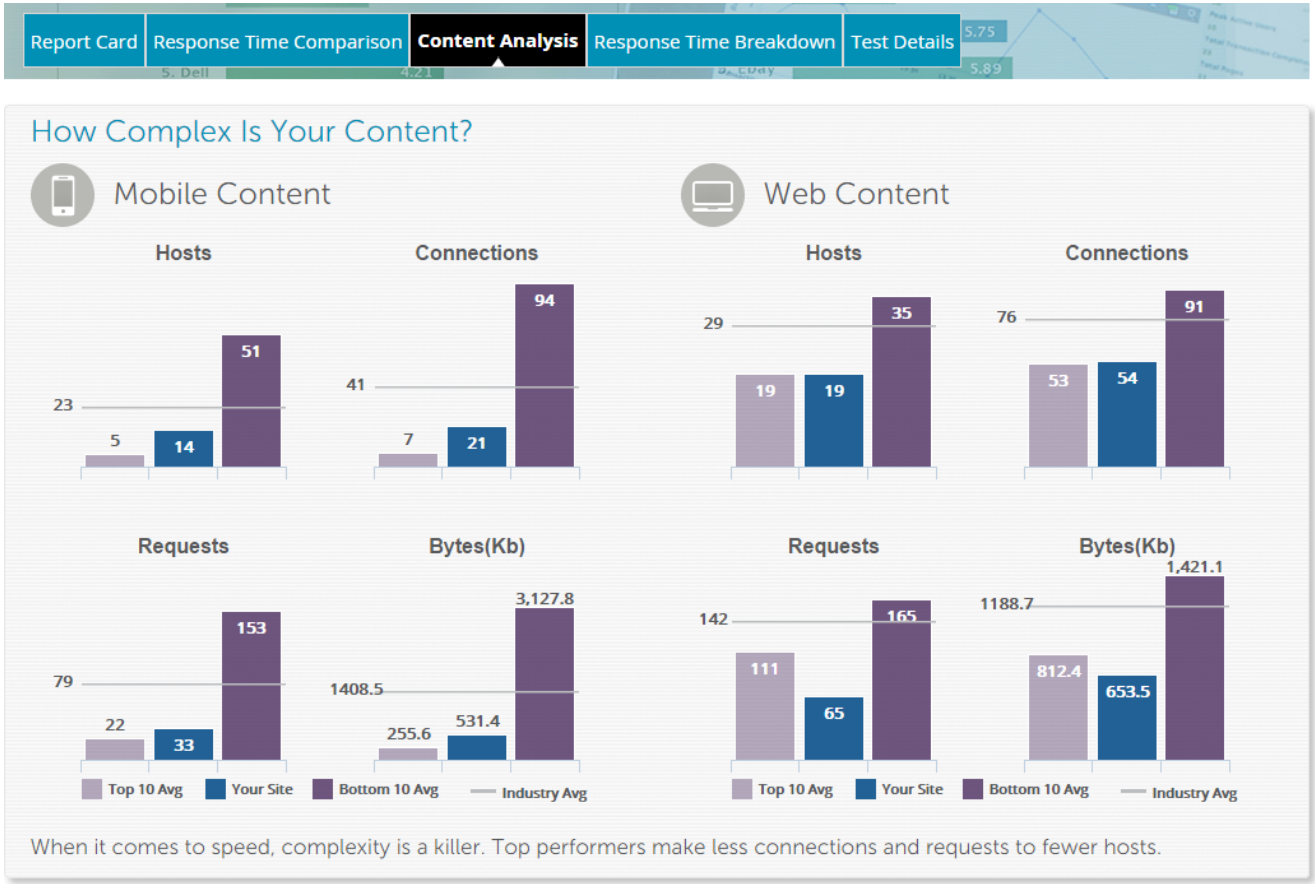
8MB of background image for STPCon (Word Press)



Make F12 or Browser Agent your friend!



Compare yourself Online!



Key Metrics

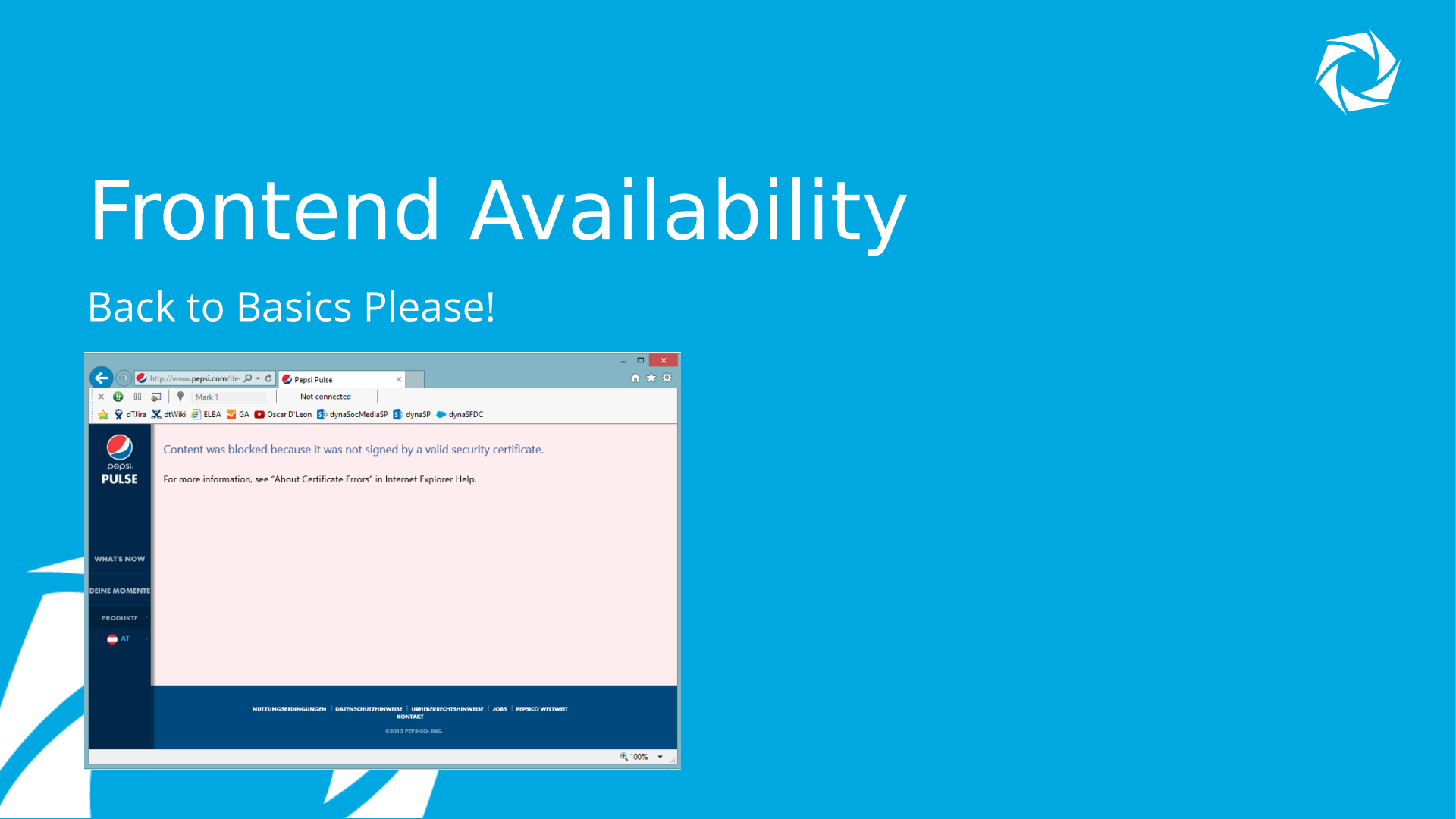
of Resources

Size of Resources

Total Size of Content

Tooling

- Browser Built-In Developer Tools
- Extensions such as YSlow, PageSpeed
- Online Tools
  - WebPageTest
  - Google PageSpeed Insights
  - Dynatrace Performance Center
  - …
- Automate!! With Selenium, WebDriver, Cucumber, …

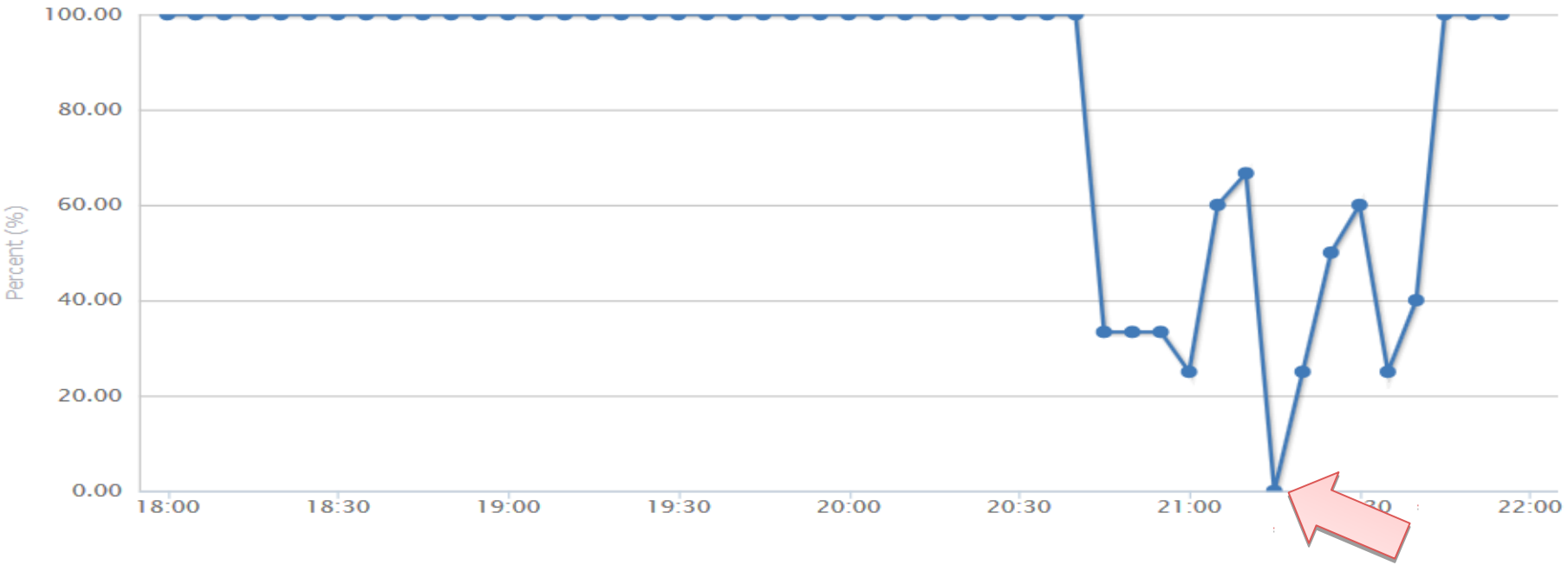


Frontend Availability

Back to Basics Please!

Metric: Availability | Group By: Test | View By: Time | Calculation: Percent | Data Level: Step | Type | [+]

Show Failure Status

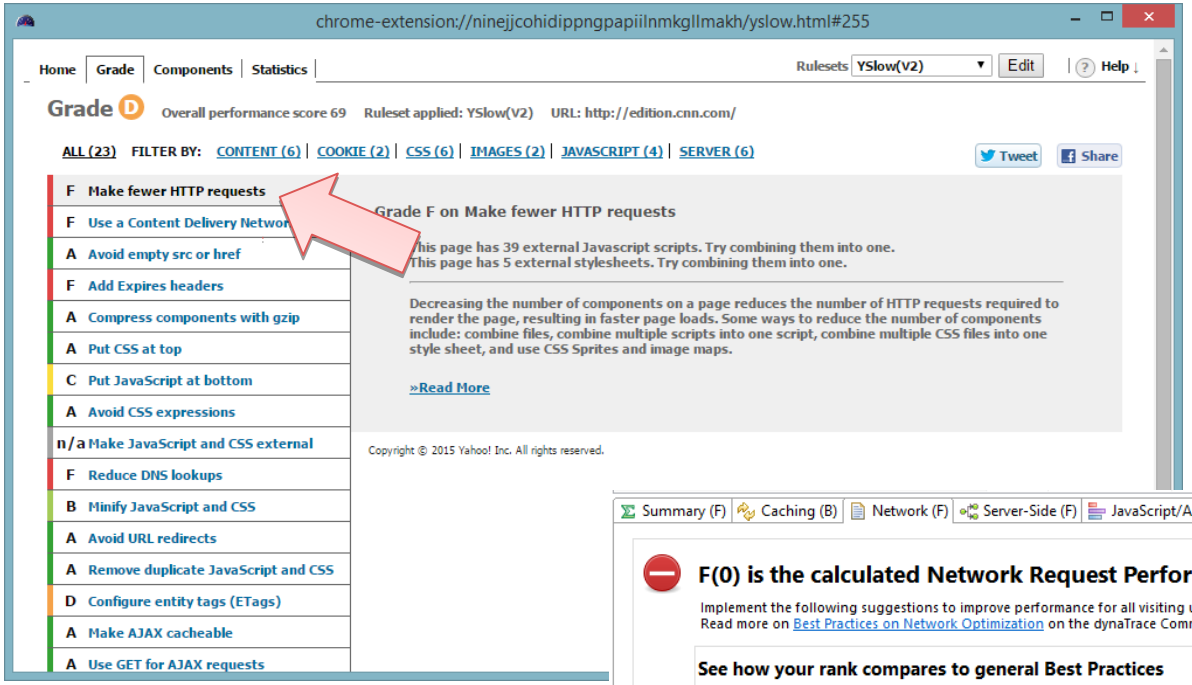


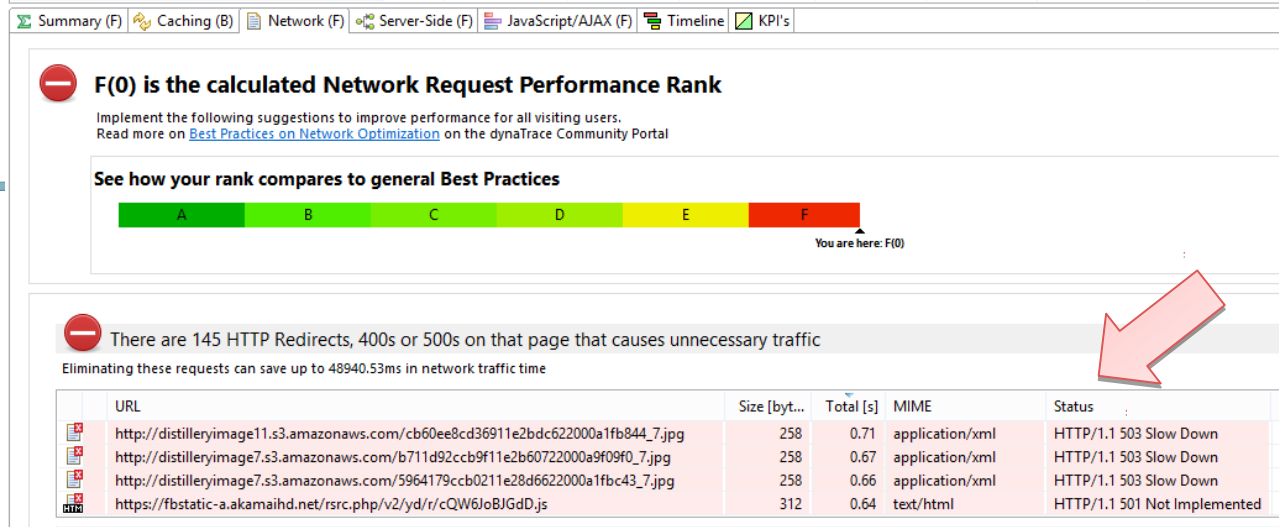
Click and drag in the plot area to zoom in.

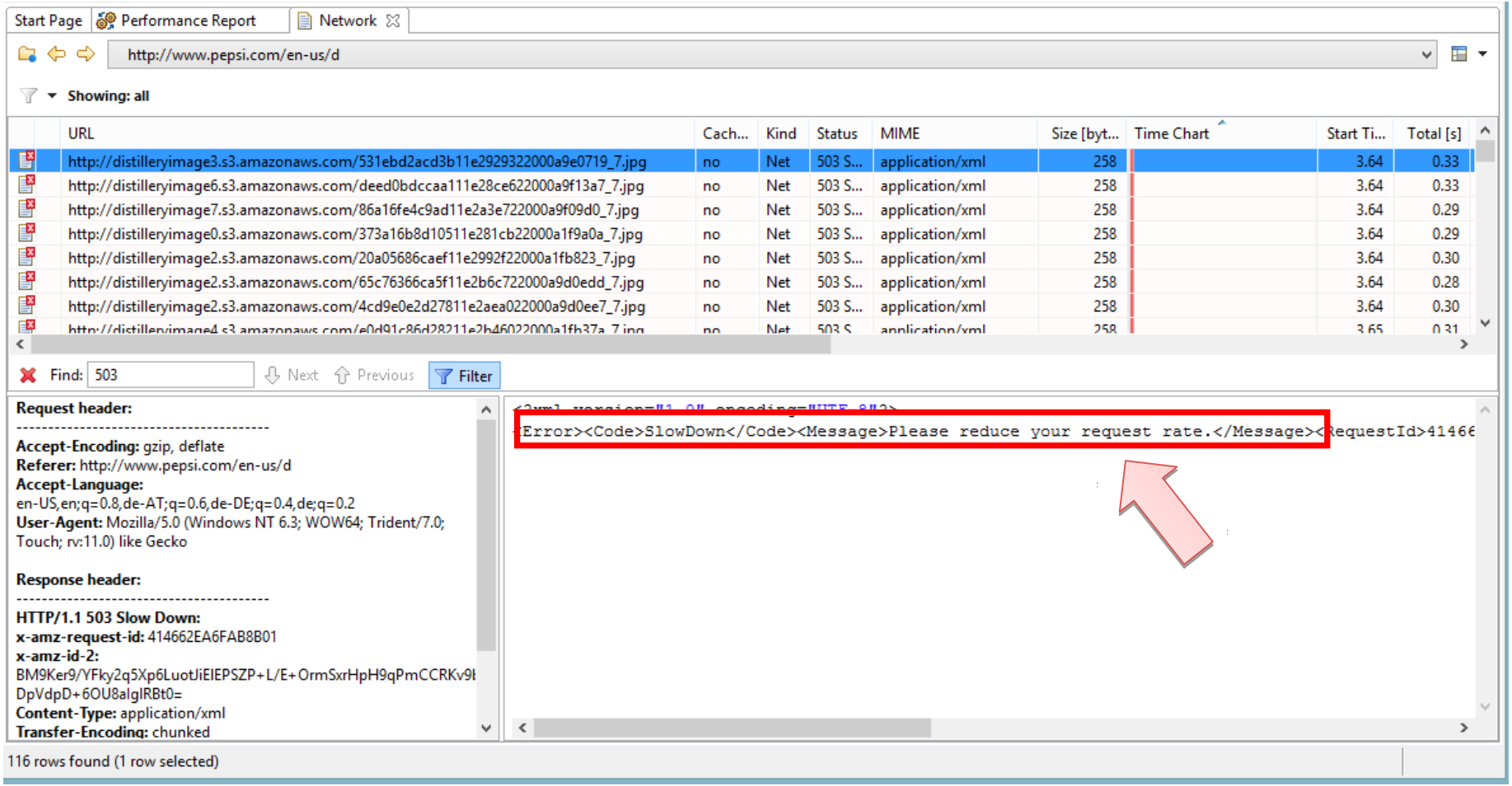
› Hide data

Synthetic Mixed

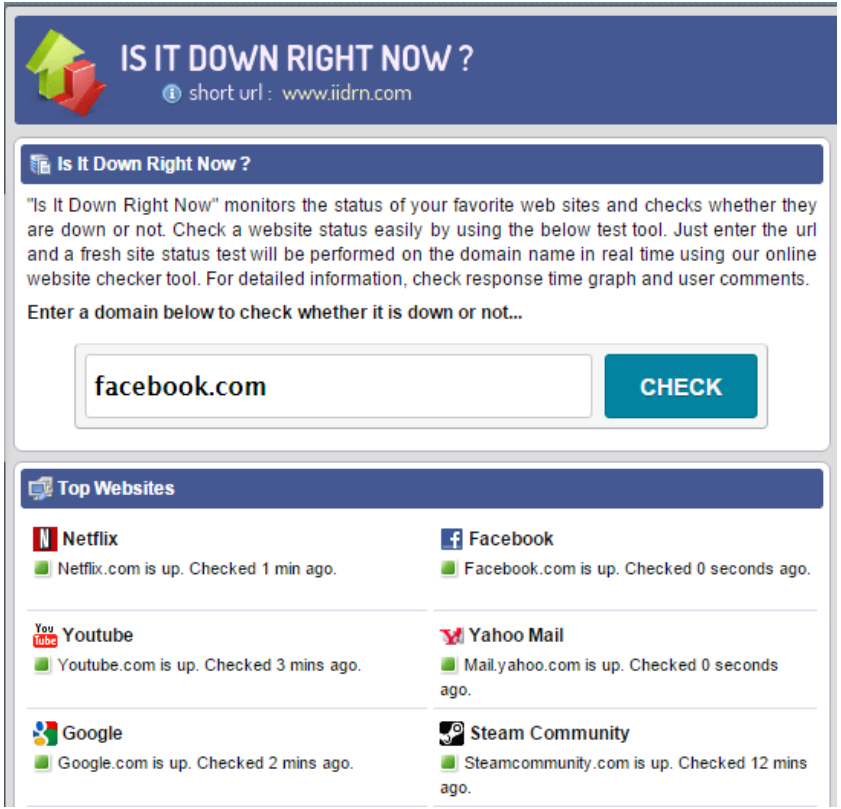
	Name	Percent Availability (%)	Product
■	Kia SB Ad - Backbone	84.47	

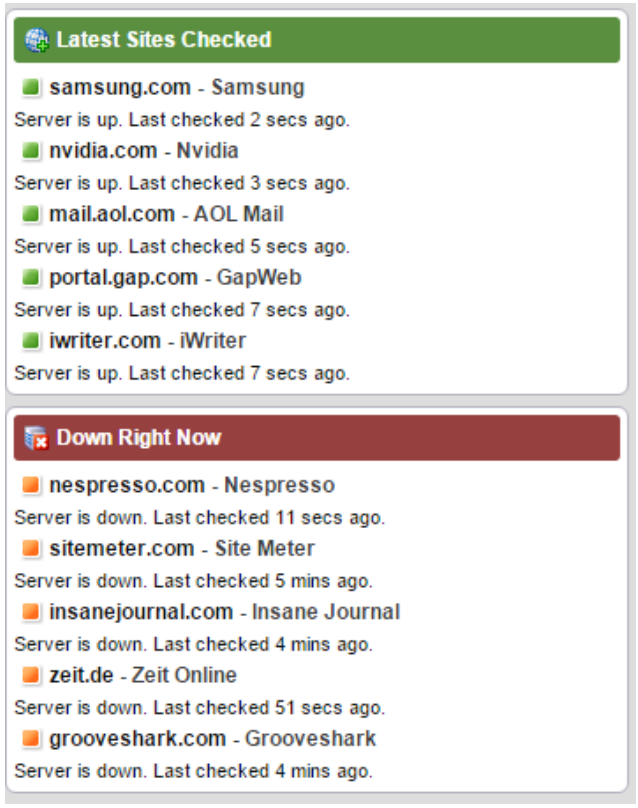

chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#255
Home
Grade
Components
Statistics
Rulesets
YSlow(V2)
Edit
Help
Grade D
Overall performance score 69
Ruleset applied: YSlow(V2)
URL: http://edition.cnn.com/
ALL (23)
FILTER BY:
CONTENT (6)
COOKIE (2)
CSS (6)
IMAGES (2)
JAVASCRIPT (4)
SERVER (6)
Tweet
Share
F Make fewer HTTP requests
F Use a Content Delivery Networ
A Avoid empty src or href
F Add Expires headers
A Compress components with gzip
A Put CSS at top
C Put JavaScript at bottom
A Avoid CSS expressions
n/a Make JavaScript and CSS external
F Reduce DNS lookups
B Minify JavaScript and CSS
A Avoid URL redirects
A Remove duplicate JavaScript and CSS
D Configure entity tags (ETags)
A Make AJAX cacheable
A Use GET for AJAX requests
Grade F on Make fewer HTTP requests
This page has 39 external Javascript scripts. Try combining them into one.
This page has 5 external stylesheets. Try combining them into one.
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.
»Read More
Copyright © 2015 Yahoo! Inc. All rights reserved.


Summary (F)
Caching (B)
Network (F)
Server-Side (F)
JavaScript/AJAX (F)
Timeline
KPI's
F(0) is the calculated Network Request Performance Rank
Implement the following suggestions to improve performance for all visiting users.
Read more on Best Practices on Network Optimization on the dynaTrace Community Portal
See how your rank compares to general Best Practices
A
B
C
D
E
F
You are here: F(0)
There are 145 HTTP Redirects, 400s or 500s on that page that causes unnecessary traffic
Eliminating these requests can save up to 48940.53ms in network traffic time
URL
Size [byt...
Total [s]
MIME
Status
http://distilleryimage11.s3.amazonaws.com/cb60ee8cd36911e2bdc622000a1fb844_7.jpg 258 0.71 application/xml HTTP/1.1 503 Slow Down
http://distilleryimage7.s3.amazonaws.com/b711d92ccb9f11e2b60722000a9f09f0_7.jpg 258 0.67 application/xml HTTP/1.1 503 Slow Down
http://distilleryimage7.s3.amazonaws.com/5964179ccb0211e28d6622000a1fbc43_7.jpg 258 0.66 application/xml HTTP/1.1 503 Slow Down
https://fbstatic-a.akamaihd.net/rsrc.php/v2/yd/r/cQW6JoBJGdD.js 312 0.64 text/html HTTP/1.1 501 Not Implemented

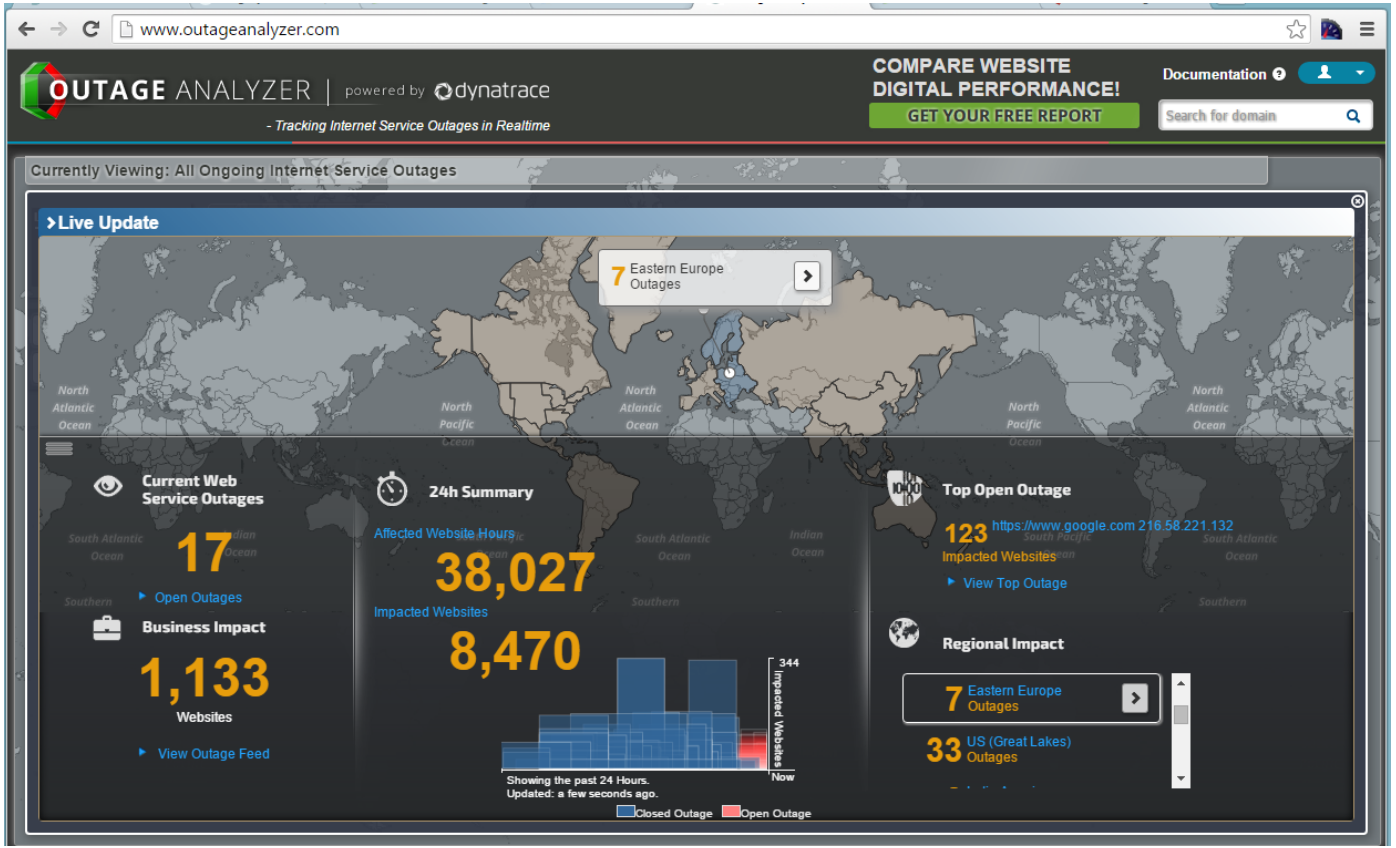

Start Page
Performance Report
Network
http://www.pepsi.com/en-us/d
Showing: all
URL
Cach...
Kind
Status
MIME
Size [byt...
Time Chart
Start Ti...
Total [s]
http://distilleryimage3.s3.amazonaws.com/531ebd2acd3b11e2929322000a9e0719_7.jpg no Net 503 S... application/xml 258 3.64 0.33
http://distilleryimage6.s3.amazonaws.com/deed0bdccaa111e28ce622000a9f13a7_7.jpg no Net 503 S... application/xml 258 3.64 0.33
http://distilleryimage7.s3.amazonaws.com/86a16fe4c9ad11e2a3e722000a9f09d0_7.jpg no Net 503 S... application/xml 258 3.64 0.29
http://distilleryimage0.s3.amazonaws.com/373a16b8d10511e281cb22000a1f9a0a_7.jpg no Net 503 S... application/xml 258 3.64 0.29
http://distilleryimage2.s3.amazonaws.com/20a05686caef11e2992f22000a1fb823_7.jpg no Net 503 S... application/xml 258 3.64 0.30
http://distilleryimage2.s3.amazonaws.com/65c76366ca5f11e2b6c722000a9d0edd_7.jpg no Net 503 S... application/xml 258 3.64 0.28
http://distilleryimage2.s3.amazonaws.com/4cd9e0e2d27811e2aea022000a9d0ee7_7.jpg no Net 503 S... application/xml 258 3.64 0.30
Find: 503
Next
Previous
Filter
Request header:
Accept-Encoding: gzip, deflate
Referer: http://www.pepsi.com/en-us/d
Accept-Language:
en-US,en;q=0.8,de-AT;q=0.6,de-DE;q=0.4,de;q=0.2
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Response header:
HTTP/1.1 503 Slow Down:
x-amz-request-id: 414662EA6FAB8B01
x-amz-id-2:
Content-Type: application/xml
Transfer-Encoding: chunked
<Error><Code>SlowDown</Code><Message>Please reduce your request rate.</Message>
116 rows found (1 row selected)

Online Services for you: Is it down right now?

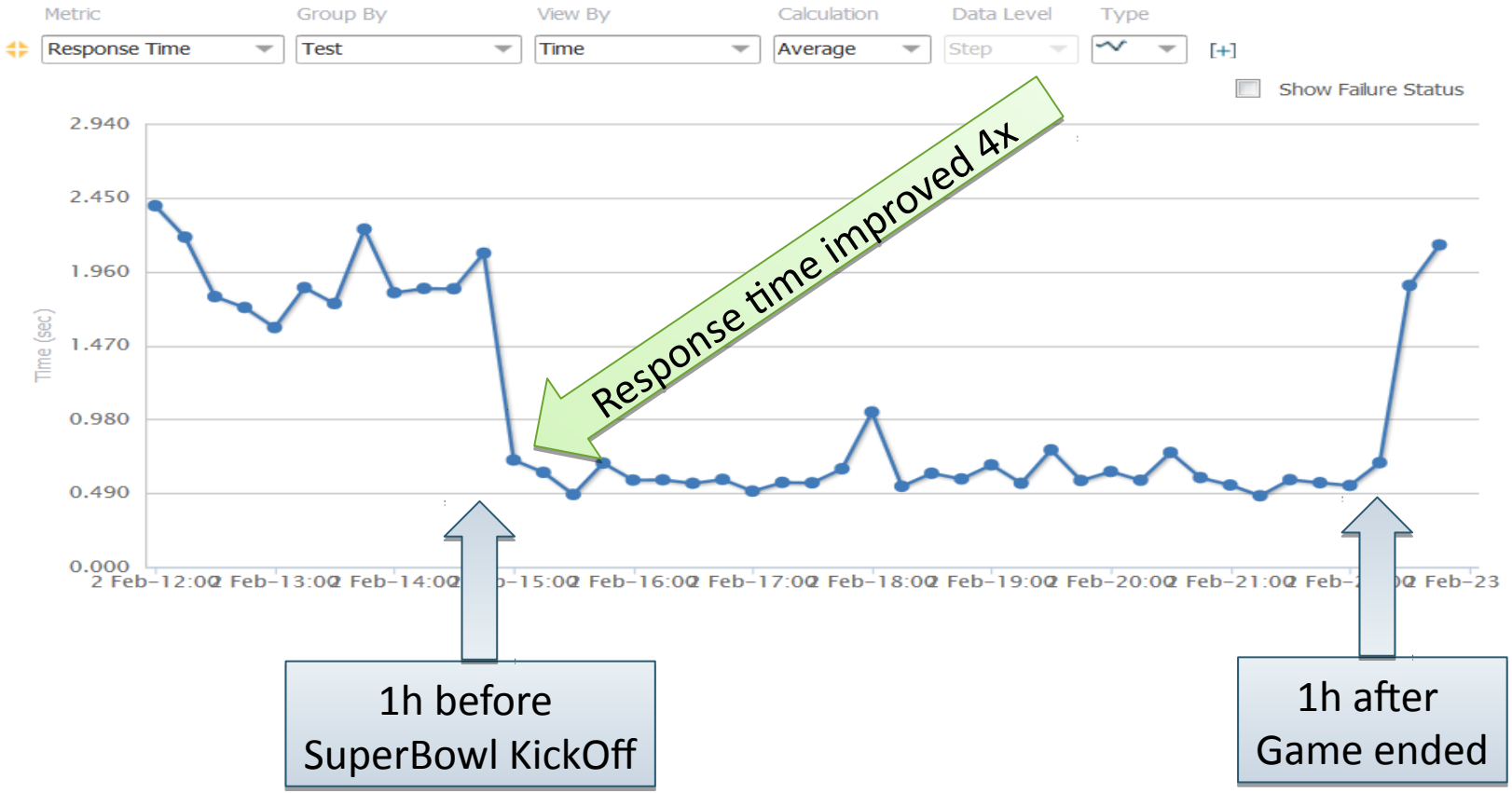




Online Services for you: Outage Analyzer



Tip for handling Spike Load: GO LEAN!!



Key Metrics

HTTP 3xx, 4xx, 5xx

of Domains

Online Services

- Dynatrace Synthetic
- Ruxit Synthetic
- NewRelic Synthetic
- AppDynamics
- PingDom
- … Just Google for „Synthetic Monitoring“

Backend Performance

The Usual Suspects

Project: Online Room Reservation System

- Symptoms
  - HTML takes between 60 and 120s to render
  - High GC Time
- Developer Assumptions
  - Bad GC Tuning
  - Probably bad Database Performance as rendering was simple
- ***Result***: 2 Years of Finger pointing between Dev and DBA

Developers built own monitoring

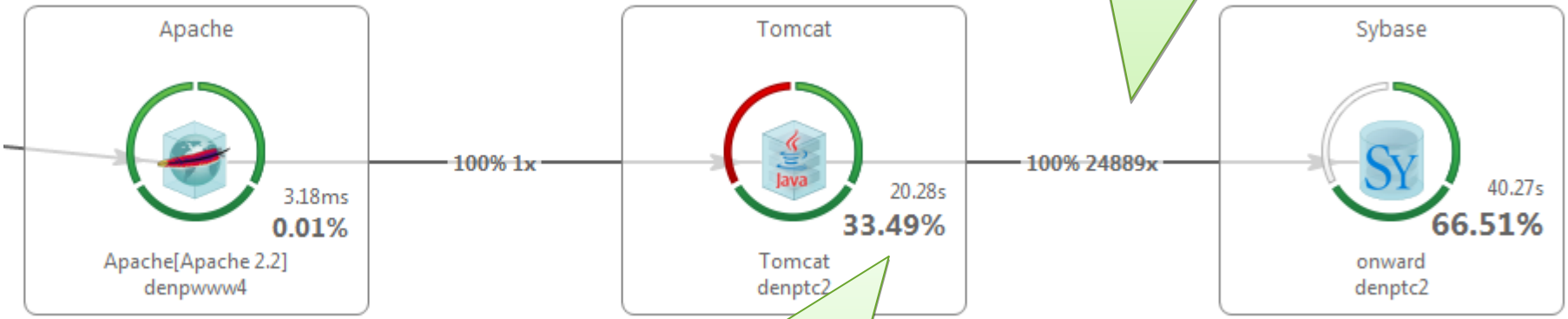
```
void roomreservationReport(int officeId)
{
  long startTime = System.currentTimeMillis();
  Object data = loadDataForOffice(officeId);
  long dataLoadTime = System.currentTimeMillis() - startTime;

  generateReport(data, officeId);
}
```

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

SQL	Execs/calling Tran...	Executions	Pr...	
set clientname '[redacted]'	12444.00	12444	0	1.28
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=167102829	2.00	2	0	1.53
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257623	2.00	2	0	0.51
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257626	2.00	2	0	0.73
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257624	2.00	2	0	0.68
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257629	2.00	2	0	0.57
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257613	2.00	2	0	0.78
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257614	2.00	2	0	1.48
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257615	2.00	2	0	0.65
selecT attribute_value.attributevalue_pk, attribute_value.object_sg,attribute_value.blob_value,attribute_value.objec	2.00	2	0	1.08
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=164257612	2.00	2	0	0.69
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=167103926	2.00	2	0	0.91
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=167102878	2.00	2	0	2.20
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=167102895	2.00	2	0	0.72
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where attributevalue_pk=167102889	2.00			0.81
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value [illegible]here attributevalue_pk=85534525	2.00			0.90
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value wh[illegible]tributevalue_pk=85534529	2.00			29.48
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value whe[illegible]evalue_pk=138498042	2.00			0.51
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_value where[illegible]pk=85674392	2.00			0.64
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_valu[illegible]	2.00	2	0	0.49
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_valu[illegible]	2.00	2	0	0.60
selecT 1 from address where address_pk=67171152	2.00	2	0	0.67
selecT 1 from device where device_pk=67171152	2.00	2	0	2.97
selecT attribute_fk,value,valid_value_fk,blob_value from attribute_valu[illegible]	2.00	2	0	1.23

12444! individual connections

Individual SQL really ***<1ms***

Classical N+1 Query Problem

#3: Putting all data in temp Hashtable

Method	Exec Sum	Breakdown	Class	APIs
service(ServletRequest, ServletResponse)	396.57s	cpu (44.0%) io (56.0%)	javax.faces.webapp.FacesServlet	Servlet
lookup(Name)	66.35s	io (100.0%)	$Proxy55	RMI
_jspService(HttpServletRequest, HttpServlet...	42.20s	io (100.0%)	org.apache.jsp.meeting_005fcenter.login.Process_005fLogin_jsp	Servlet
get(Object)	36.95s	cpu (100.0%)	java.util.Hashtable	Ajax4jsf, [redacted]
doFilter(ServletRequest, ServletResponse, Fil...	8.63s	cpu (45.0%) io (55.0%)	org.ajax4jsf.webapp.BaseFilter	Servlet
getSitesFromManager(SubscriberBean, Con...	6.57s	[redacted].StepSites	P[redacted]	

Lots of time spent in Hashtable.get

Caller Breakdown of 'get(Object)'

Find out from what components the method is called and

Method	tion	APIs	Package
Hashtable.get(Object)		Pe[redacted]l, Ajax4...	java.util
Entity.get(String, int, Class)	82.0%	Pe[redacted]	com.[redacted]entity
Entity.get(String, int)	81.0%	Pe[redacted]	com.[redacted]entity
Entity.getEntityInteger(String)	78.0%	Pe[redacted]	com.[redacted]entity
AbstractSiteEntity.getRoomCapacity()	77.0%	Pe[redacted]	com.[redacted]entity
StepSites.getShowCapacity()		Pe[redacted]	com.[redacted].reservation
Entity.getEntityValidValue(String)		Pe[redacted]	com.[redacted]entity
AbstractDeviceControlEntity.getRDMFk()			com.[redacted]entity
AttributeValueEntity.getAttributeFk()			com.[redacted]entity
Entity.getEntityBoolean(String)			com.[redacted]entity
Entity.getEntityString(String)			com.[redacted]entity
Entity.getEntity(String, Class)			com.[redacted]entity
Entity.getEntityPk(String)			com.[redacted]entity

Called from their Entity Objects

Lessons Learned – ***Don't Assume*** …

- … you know what code is doing you inherited!!
- … you are not making mistakes like this ☺

- Explore the Right Tools
  - Built-In Database Analysis Tools
  - "Logging" options of Frameworks such as Hibernate, …
  - JMX, Perf Counters, … of your Application Servers
  - Performance Tracing Tools: Dynatrace, Ruxit, NewRelic, AppDynamics, Your Profiler of Choice …

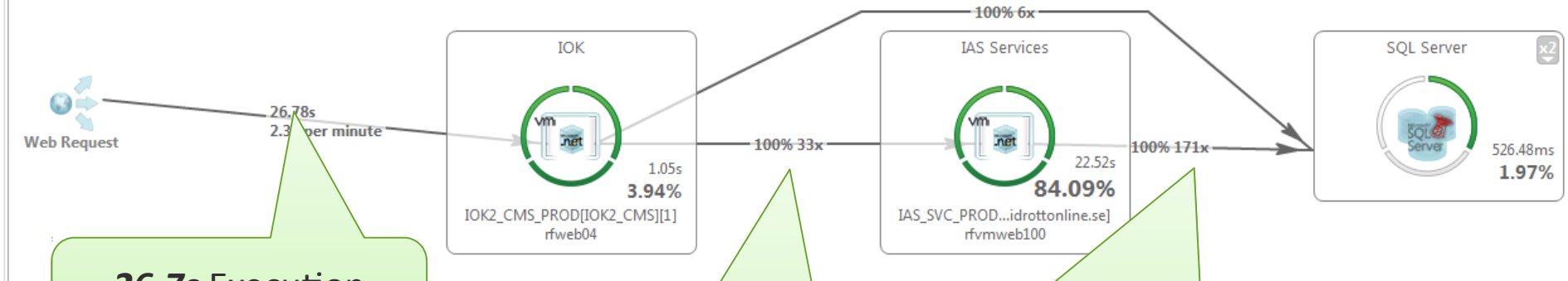
Key Metrics

of SQL Calls
# of same SQL Execs (1+N)
# of Connections
Rows/Data Transferred

Backend Performance

Architectural Mistakes with „Migrating“ to (Micro)Services

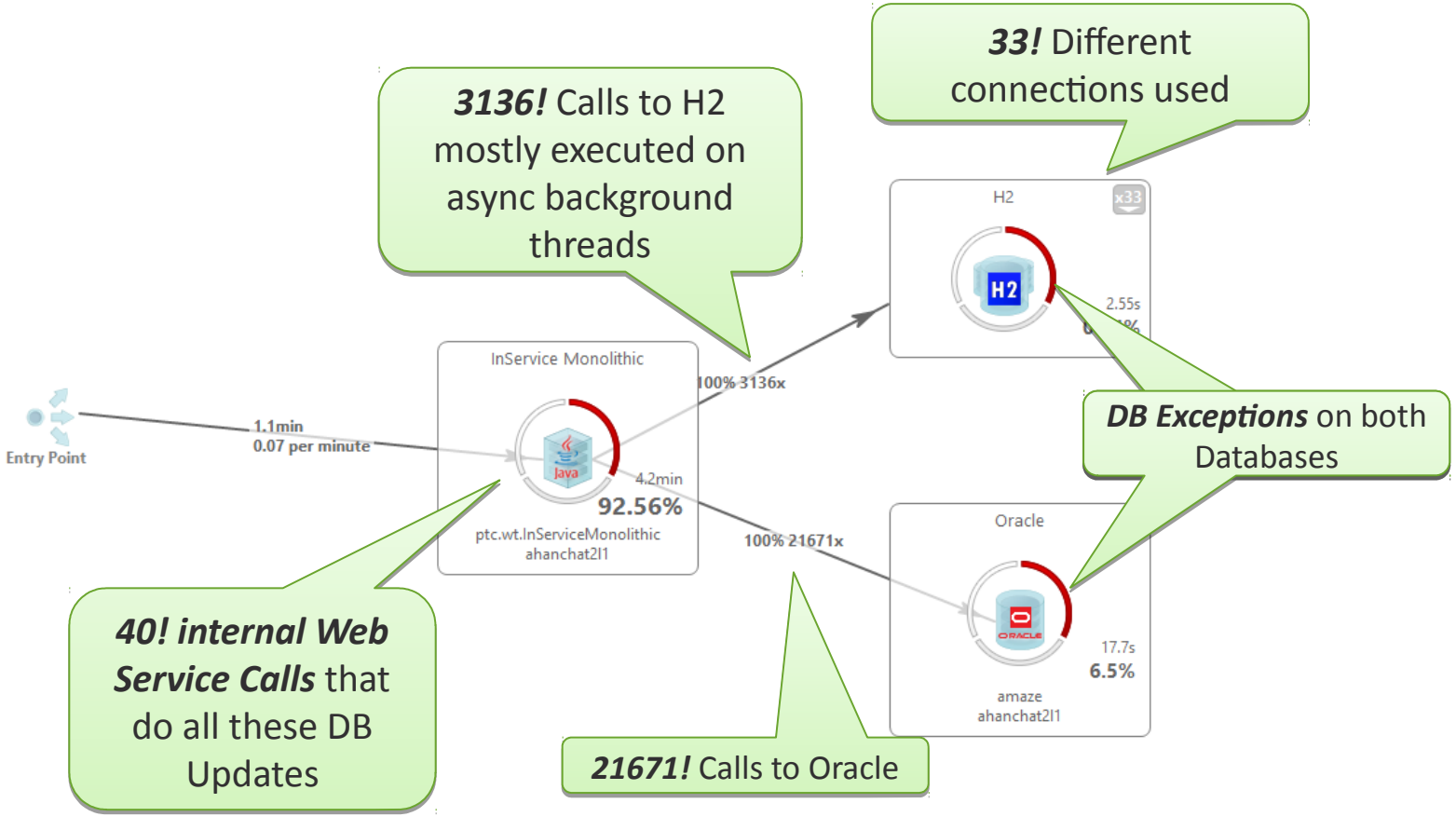
Architecture Violation: Direct access to DB instead from frontend logic



26.7s Execution Time

33! Calls to the same Web Service

171! SQL Queries through LINQ by this Web Service – request ***similar data*** for each call


33! Different connections used
3136! Calls to H2 mostly executed on async background threads
H2
x33
2.55s
InService Monolithic
100% 3136x
Entry Point
1.1min
0.07 per minute
4.2min
92.56%
ptc.wt.InServiceMonolithic
ahanchat2l1
DB Exceptions on both Databases
Oracle
100% 21671x
17.7s
6.5%
amaze
ahanchat2l1
40! internal Web Service Calls that do all these DB Updates
21671! Calls to Oracle

Key Metrics

of Service Calls

Payload of Service Calls

# of Involved Threads

1+N Service Call Pattern!

Tooling

- Dynatrace
- Ruxit
- NewRelic
- AppDynamics
- Any Profiler that can trace across tiers
- Google for Tracing or APM (Application Performance Management)

Logging

WE CAN LOG THIS!!

Log Hotspots in Frameworks!

callAppenders clear CPU and I/O Hotspot

Method Breakdown by Execution Time

Select a method to find out where it is called from

Method	Exec Sum	Breakdown	Class	APIs
socketRead0(FileDescriptor, byte[], int, int, i...	628.03s	io (98.0%)	java.net.SocketInputStream	Com Sbi, Ibm, JDBC
callAppenders(LoggingEvent)	382.52s	cpu (43.0%) io (56.0%)	org.apache.log4j.Category	Log
doPost(HttpServletRequest, HttpServletRes...	85.22s	cpu io (67.0%)	org.springframework.web.servlet.FrameworkServlet	Servlet
executeQuery()	57.07s	io (97.0%)	com.ibm.ws.rsadapter.jdbc.WSJdbcPreparedStatement	JDBC
wait(long, int)	53.55s	wait (98.0%)	java.lang.Object	Ibm

Caller Breakdown of 'callAppenders(LoggingEvent)'

Find out from what components the method is called and which call path has the biggest performance impact

Method	Contribution	APIs	Package
Category.callAppenders(LoggingEvent)		**Log**	**org.apache.log4j**
FrameworkServlet.doPost(HttpServletRequest, HttpServletResponse)	59.0%	Servlet	org.springframework.web.servlet
MerchantPreLoginHandler.merchantPreLoginHandler(HttpServletRequest, HttpServletResponse)	18.0%	Com Sbi	com.sbi.merchant.handler
FrameworkServlet.doGet(HttpServletRequest, HttpServletResponse)		Servlet	org.springframework.we...
DoubleVerificationHandler.handleRequest(HttpServletRequest, HttpServletResponse)		Com Sbi	com.sbi.merchantut...
Category.forcedLog(String, Priority, Object, Throwable)		Log	org.apache.log4...
LogonBP.validateLogin(String, String, String, String, String, String)		Com Sbi	com.sbi.co...
SMSGatewayDynaPortClient.sendMessageToSgate(String, String)		Com Sbi	com.sb...
_findbankcode._jspService(HttpServletRequest, HttpServletResponse)			
_merchantverification._jspService(HttpServletRequest, HttpServletResponse)			
_merchanterror._jspService(HttpServletRequest, HttpServletResponse)			
MerchantParamsDAOImpl.getMerchantParams(String, String)			
_logon._jspService(HttpServletRequest, HttpServletResponse)		Servlet	com.ibm._jsp
_error._jspService(HttpServletRequest, HttpServletResponse)		Servlet	com.ibm._jsp
_HostedPaymentRedirect._jspService(HttpServletRequest, HttpServletResponse)		Servlet	com.ibm._jsp

Excessive logging through Spring Framework

Debug Log and outdated log4j library

#1: Top Problem: log4j.callAppenders -> 71% Sync Time

Method Breakdown by Execution Time

Select a method to find out where it is called from

Method	Exec Sum	Breakdown	Class	APIs
callAppenders(LoggingEvent)	116.26s	cpu sync (71.0%)	org.apache.log4j.Category	Log
socketWrite0(FileDescriptor, byte[], int, int)	98.38s	cpu io (72.0%)	java.net.SocketOutputStream	Servlet
nativeLayout(Font2D, FontStrike, float[], int,...	49.19s	cpu io (72.0%)	sun.font.SunLayoutEngine	AWT
nativeBidiChars(Bidi, char[], int, byte[], int, i...	8.94s	cpu io (72.0%)	java.text.Bidi	AWT
<init>()	4.47s	cpu io (72.0%)	java.lang.Object	AWT
indexedBinarySearch(List, Object)	4.47s	cpu io (72.0%)	java.util.Collections	Sf
hashCode()	4.47s	cpu io (72.0%)	java.text.AttributeEntry	AWT
append(String)	4.47s	cpu io (72.0%)	java.lang.AbstractStringBuilder	Sf
append(String)	4.47s	cpu io (72.0%)	java.lang.StringBuilder	Sf
doFilter(ServletRequest, ServletResponse, Fil...	0ms	cpu (95.0%) io	org.jboss.web.tomcat.filters.ReplyHeaderFilter	Servlet

Caller Breakdown of 'callAppenders(LoggingEvent)'

Find out from what components the method is called and which call path has the biggest performance impact

Method	Contribution	APIs	Package
Category.callAppenders(LoggingEvent)		**Log**	**org.apache.log4j**
Category.forcedLog(String, Priority, Object, Throwable)		Log	org.apache.log4j
Category.log(String, Priority, Object, Throwable)			
GeneratedMethodAccessor101.invoke(Object,			
DelegatingMethodAccessorImpl.invoke(Obj			
Method.invoke(Object, Object[])			
Log4jProxy.log(Object, Object, Throwa			
Log4jProxy.debug(Object, Throwable)			org.apache.commo
Log4JLogger.debug(Object)		Sf	org.apache.commo
JRVerticalFiller.fillDetail()	92.0%	Sf	net.sf.jasperreports.
JRBaseFiller.addPage(JRPrintPage)		Sf	net.sf.jasperreports.
Category.info(Object)		Log	org.apache.log4j

#3: Doing “DEBUG” log output: Is this necessary?

#2: Most of logging done from fillDetail method

Key Metrics

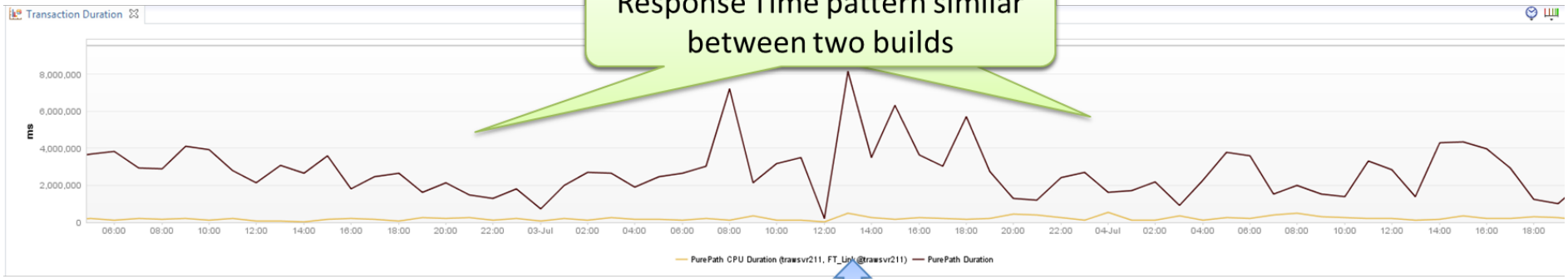
of Log Entries

Size of Logs per Use Case

Response Time is not the only Performance Indicator

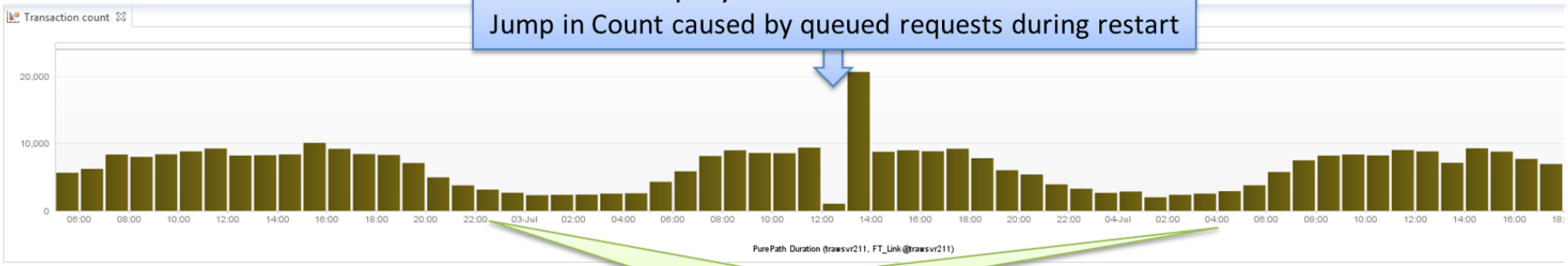
Look at Resources as well

Is this a successful new Build?



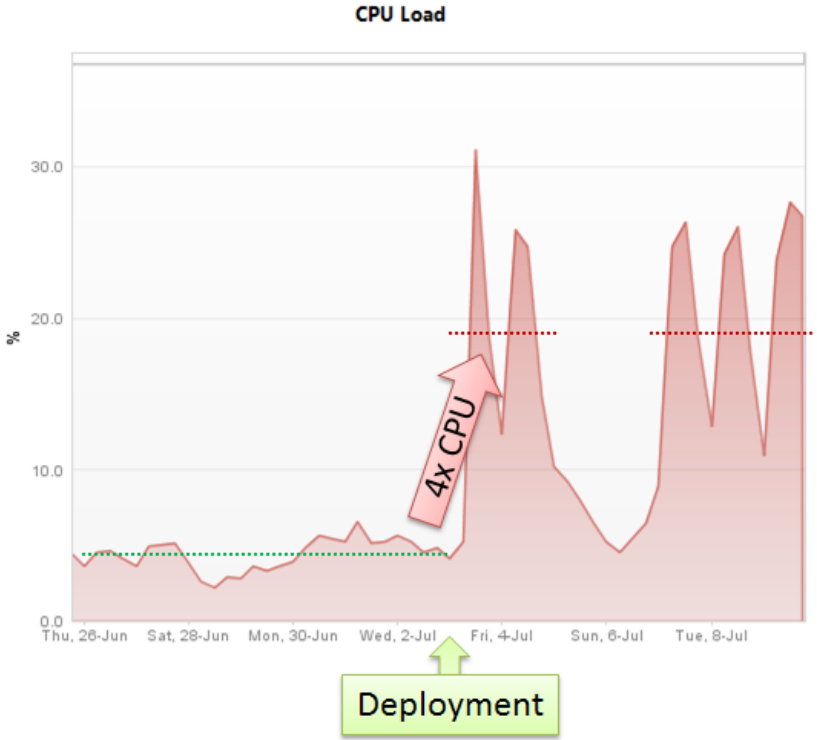
Response Time pattern similar between two builds

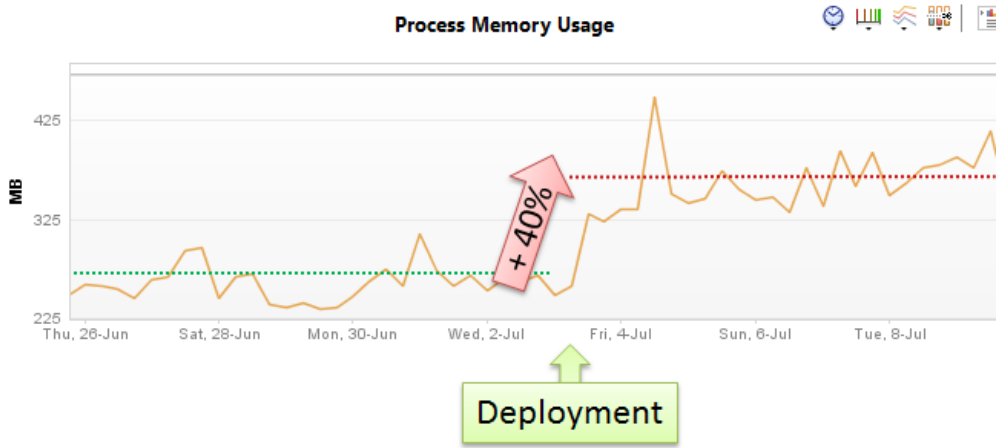
Deploy New Version at 12:00
Jump in Count caused by queued requests during restart



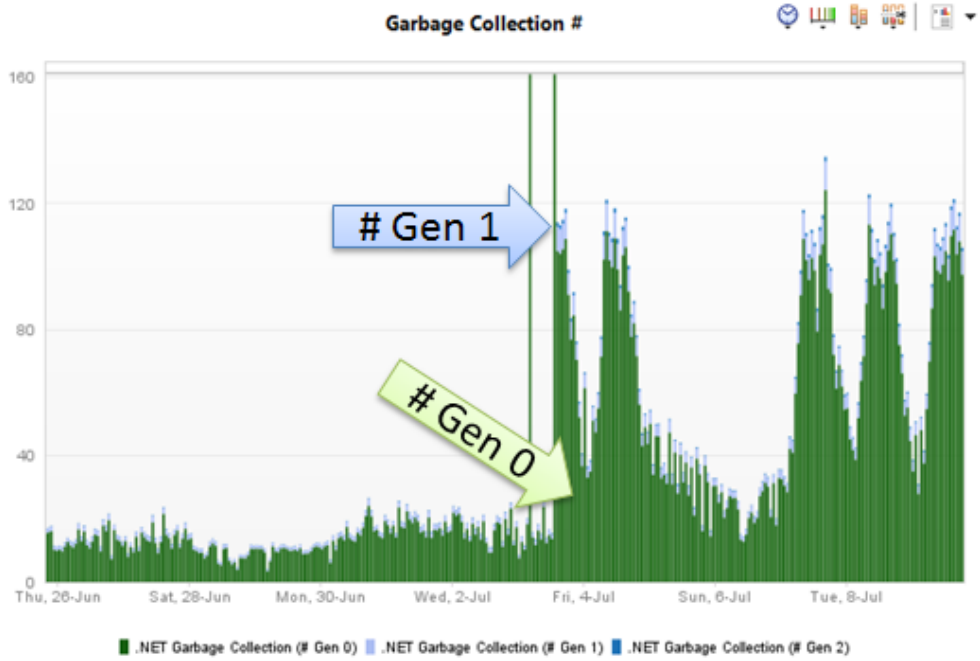
Application is able to handle the same load pattern

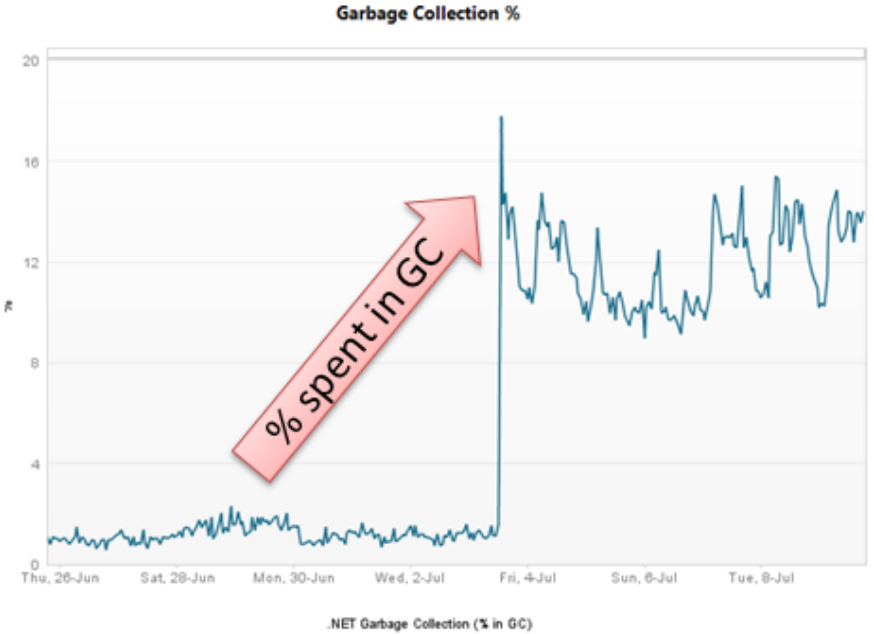
Look at Resource Usage: CPU, Memory, ...



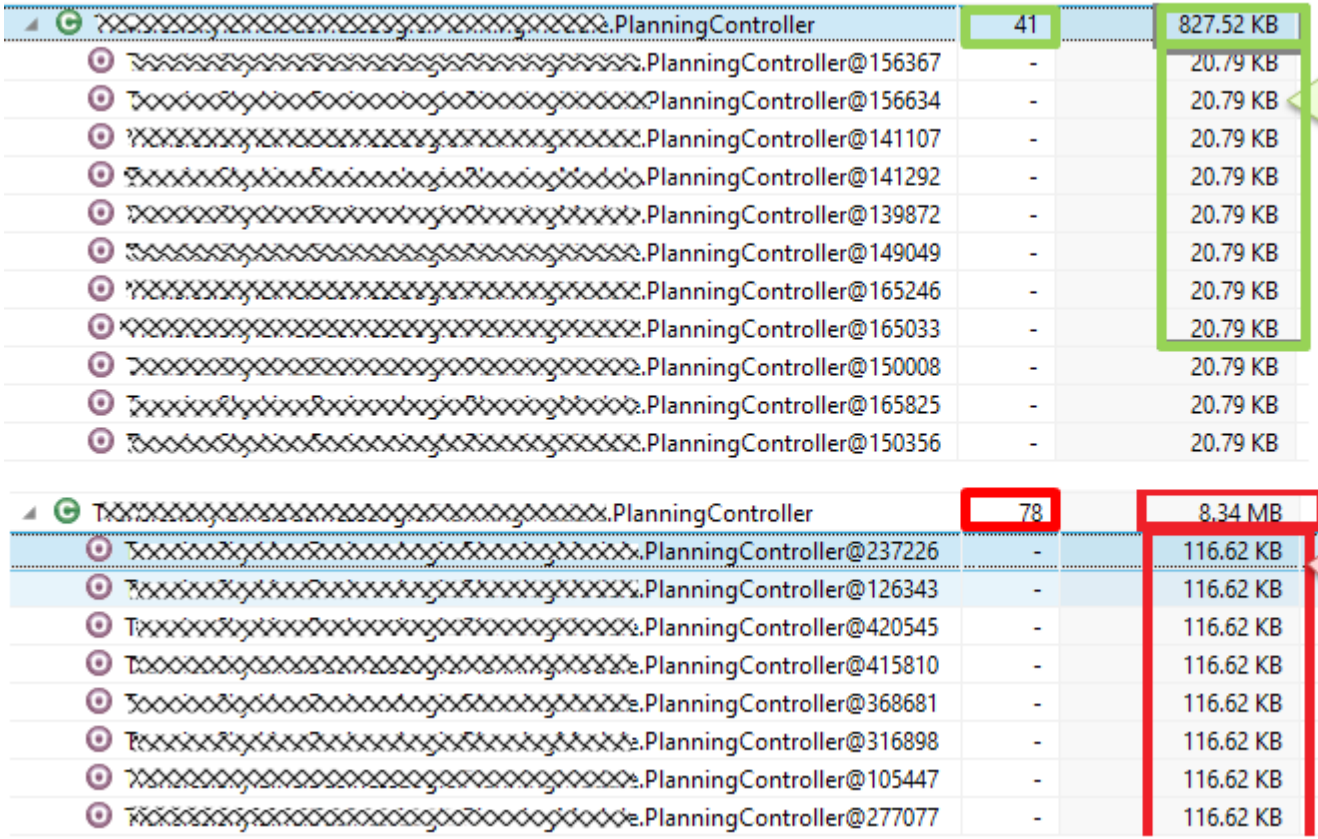


Memory? Look at Heap Generations





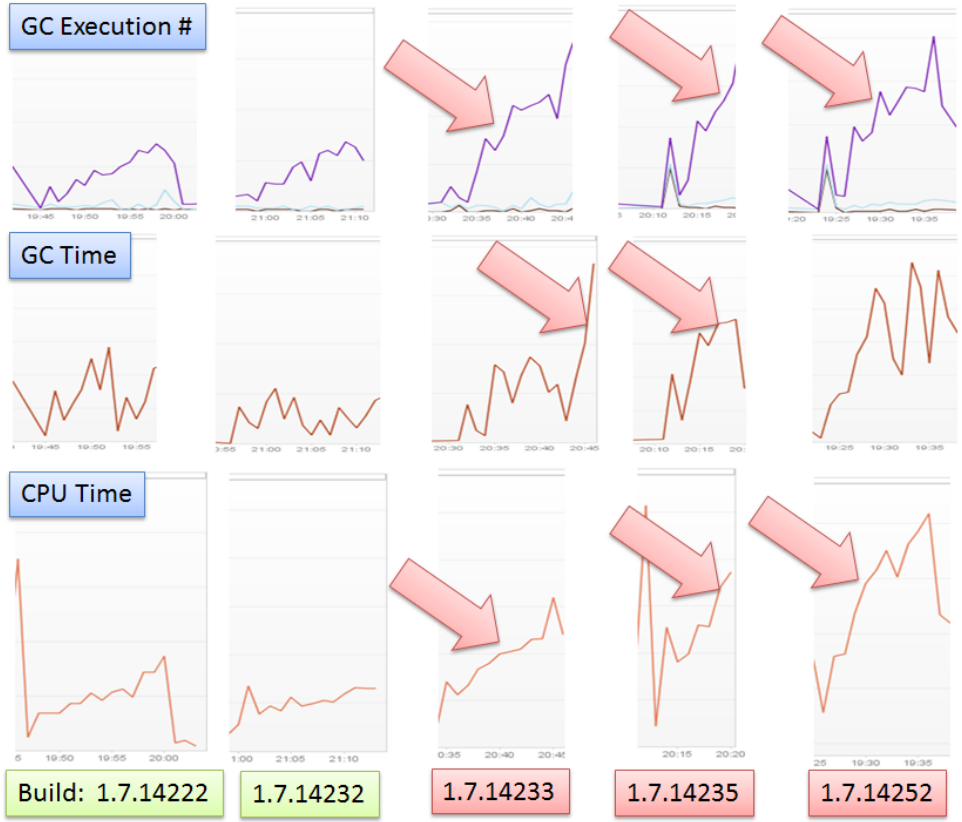
Root Cause: Dependency Injection

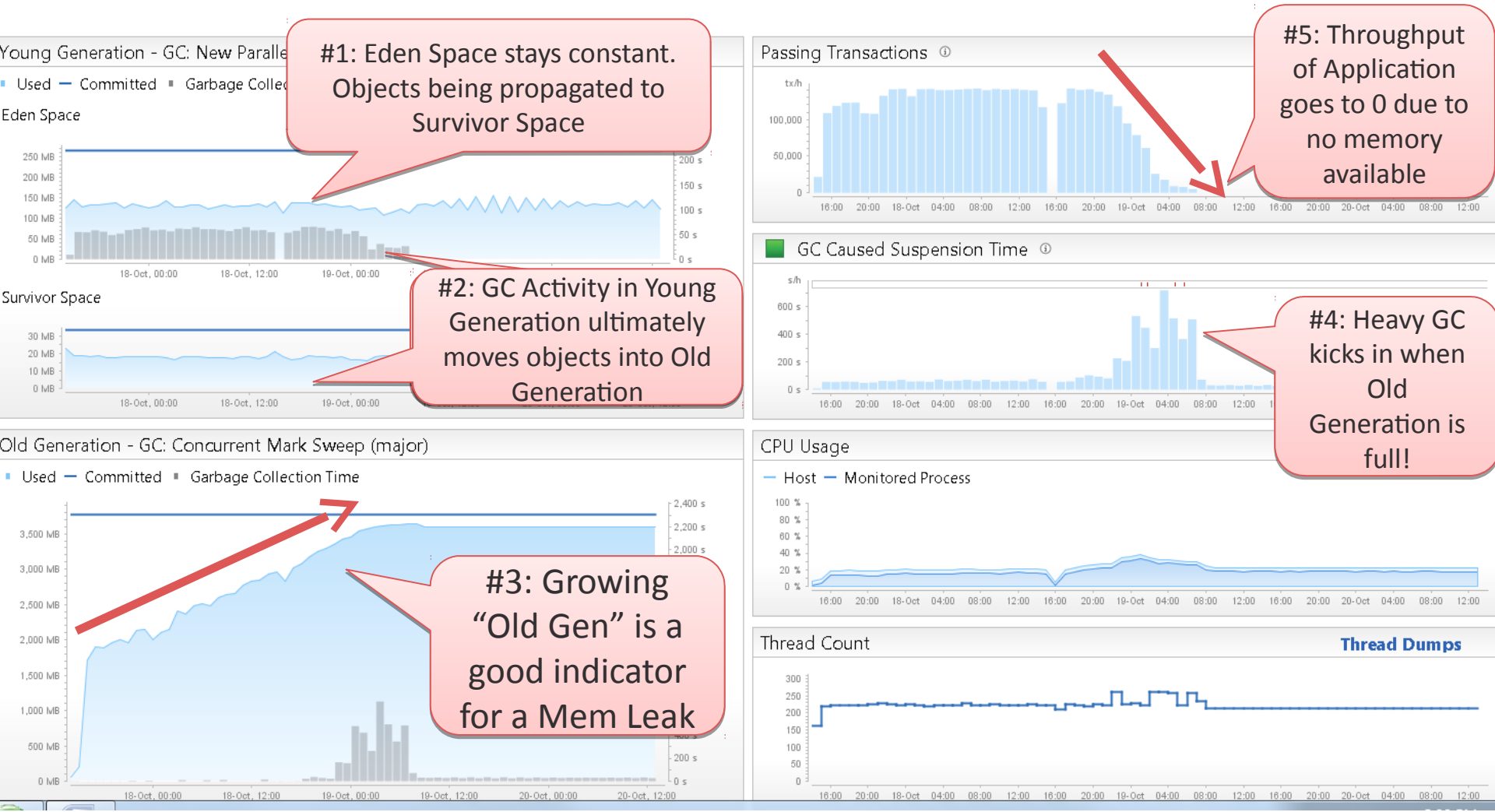


Before Deploy: 41 Controller Objects with Avgerage size of 20.79kB

After Deploy: ~6x larger Object size and ~2x as many objects on the heap

Prevent: Monitor Memory Metrics for every Build




Young Generation - GC: New Paralle
Used
Committed
Garbage Collec
Eden Space
250 MB
200 MB
150 MB
100 MB
50 MB
0 MB
200 s
150 s
100 s
50 s
0 s
18-Oct, 00:00
18-Oct, 12:00
19-Oct, 00:00
Survivor Space
30 MB
20 MB
10 MB
0 MB
18-Oct, 00:00
18-Oct, 12:00
19-Oct, 00:00
#1: Eden Space stays constant. Objects being propagated to Survivor Space
#2: GC Activity in Young Generation ultimately moves objects into Old Generation
Old Generation - GC: Concurrent Mark Sweep (major)
Used
Committed
Garbage Collection Time
3,500 MB
3,000 MB
2,500 MB
2,000 MB
1,500 MB
1,000 MB
500 MB
0 MB
2,400 s
2,200 s
2,000 s
200 s
0 s
18-Oct, 00:00
18-Oct, 12:00
19-Oct, 00:00
19-Oct, 12:00
20-Oct, 00:00
20-Oct, 12:00
#3: Growing "Old Gen" is a good indicator for a Mem Leak
Passing Transactions
tx/h
100,000
50,000
0
16:00
20:00
18-Oct
04:00
08:00
12:00
16:00
20:00
19-Oct
04:00
08:00
12:00
16:00
20:00
20-Oct
04:00
08:00
12:00
#5: Throughput of Application goes to 0 due to no memory available
GC Caused Suspension Time
s/h
600 s
400 s
200 s
0 s
#4: Heavy GC kicks in when Old Generation is full!
CPU Usage
Host
Monitored Process
100 %
80 %
60 %
40 %
20 %
0 %
Thread Count
Thread Dumps
300
250
200
150
100
50
0

Key Metrics

of Objects per Generation

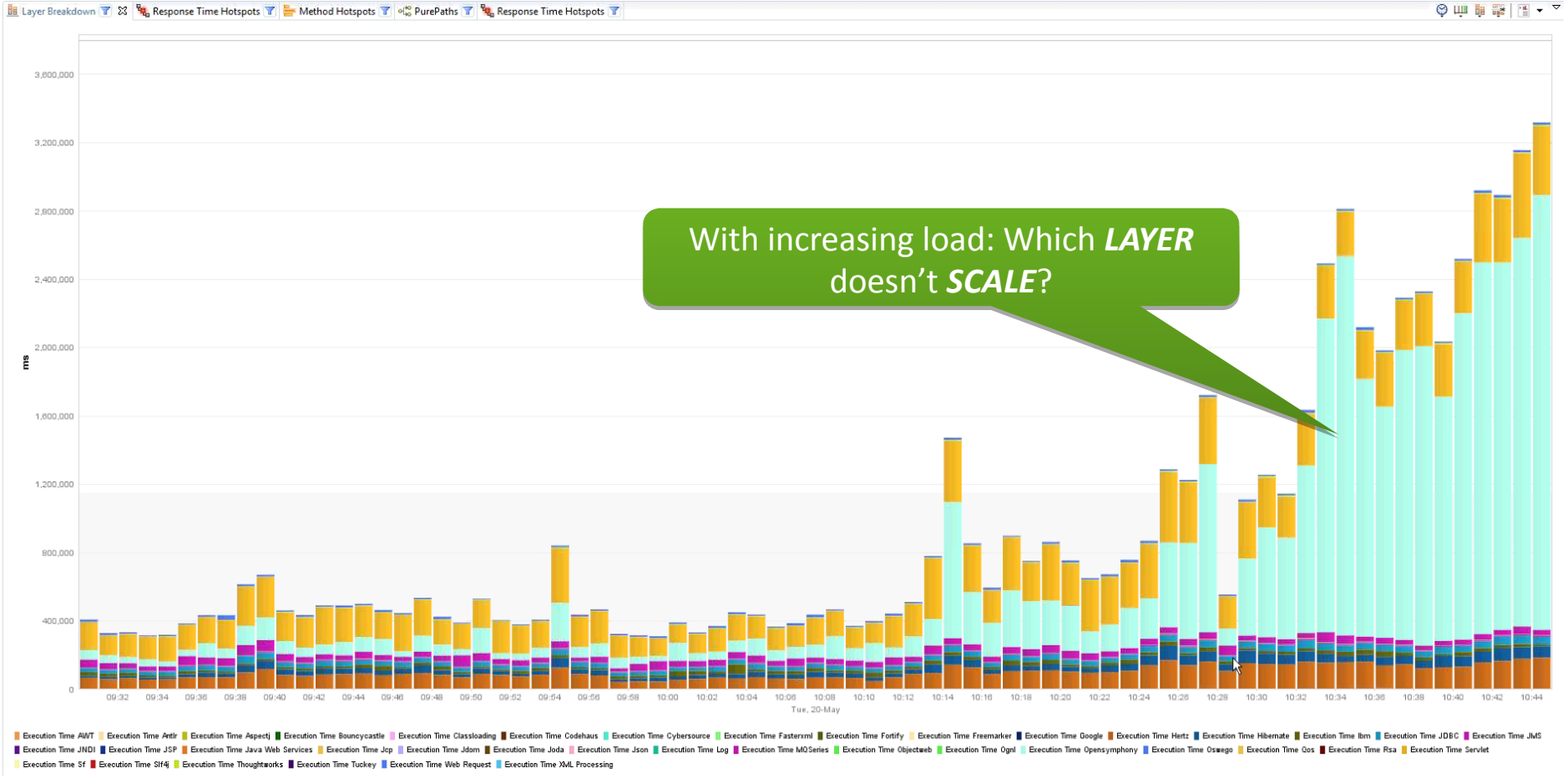
of GC Runs

Total Impact of GC

Tips & Tricks

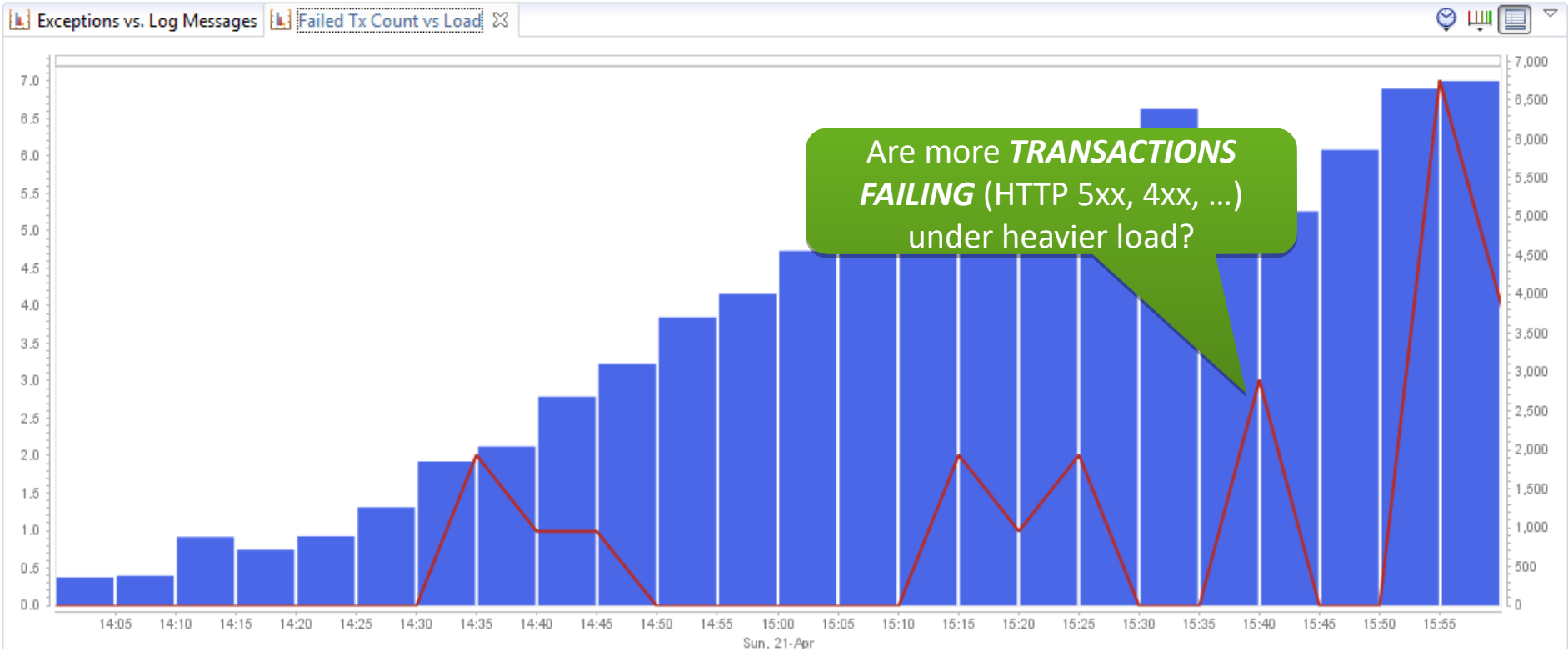
And more Metrics of course ☺

Tip: Layer Breakdown over Time

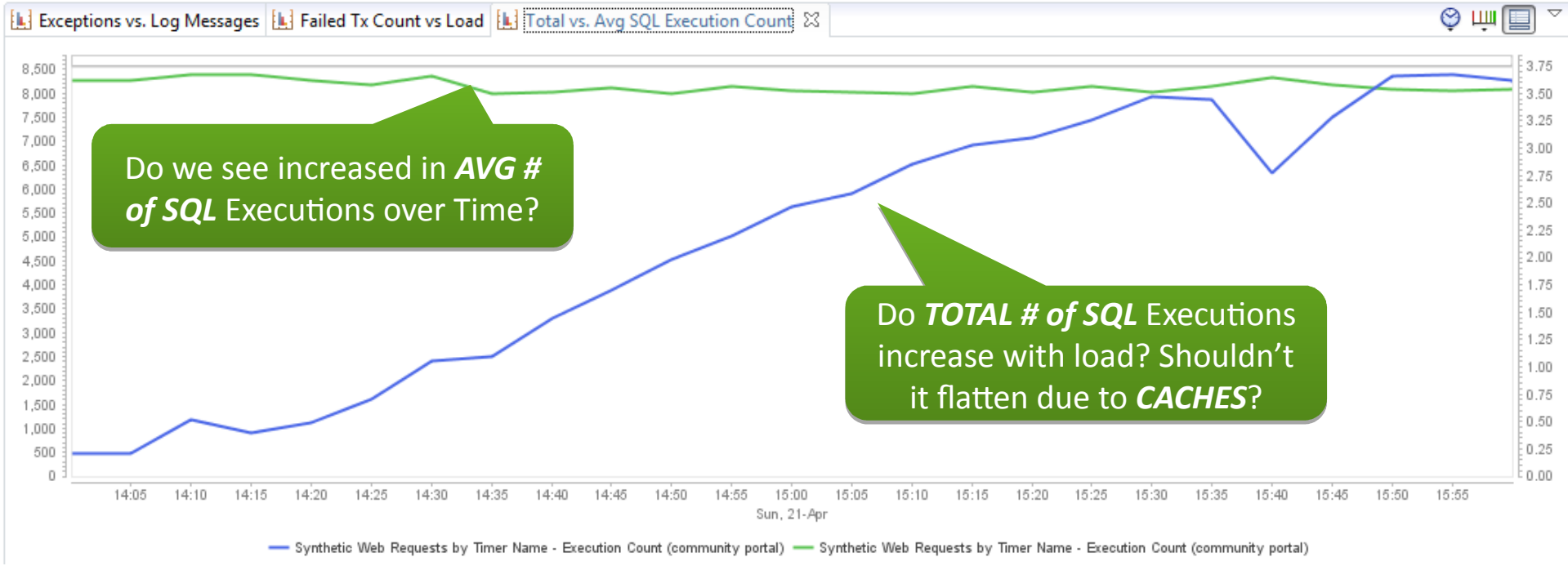


Tip: Exceptions and Log Messages

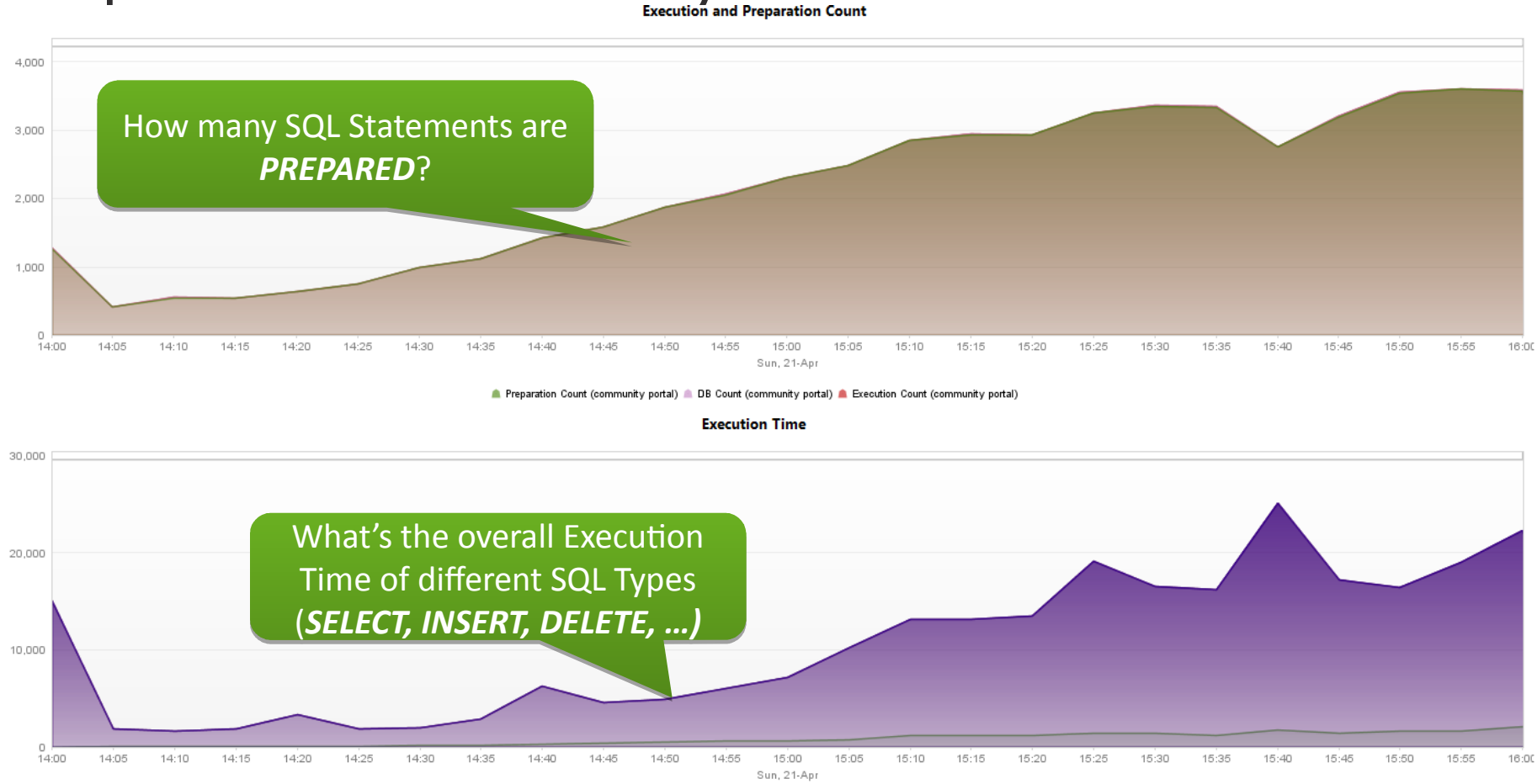
Tip: Failed Transactions



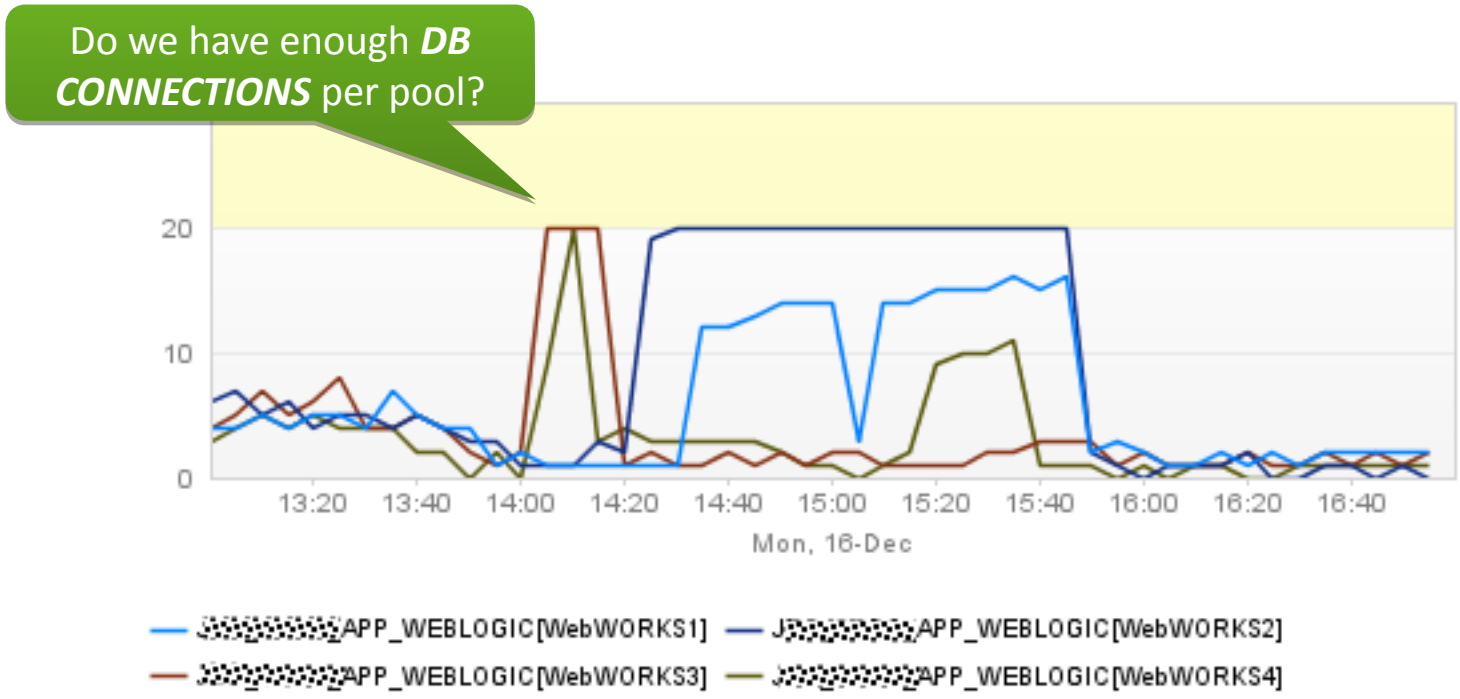
Tip: Database Activity



Tip: Database History Dashboard



Tip: DB Connection Pool Utilization

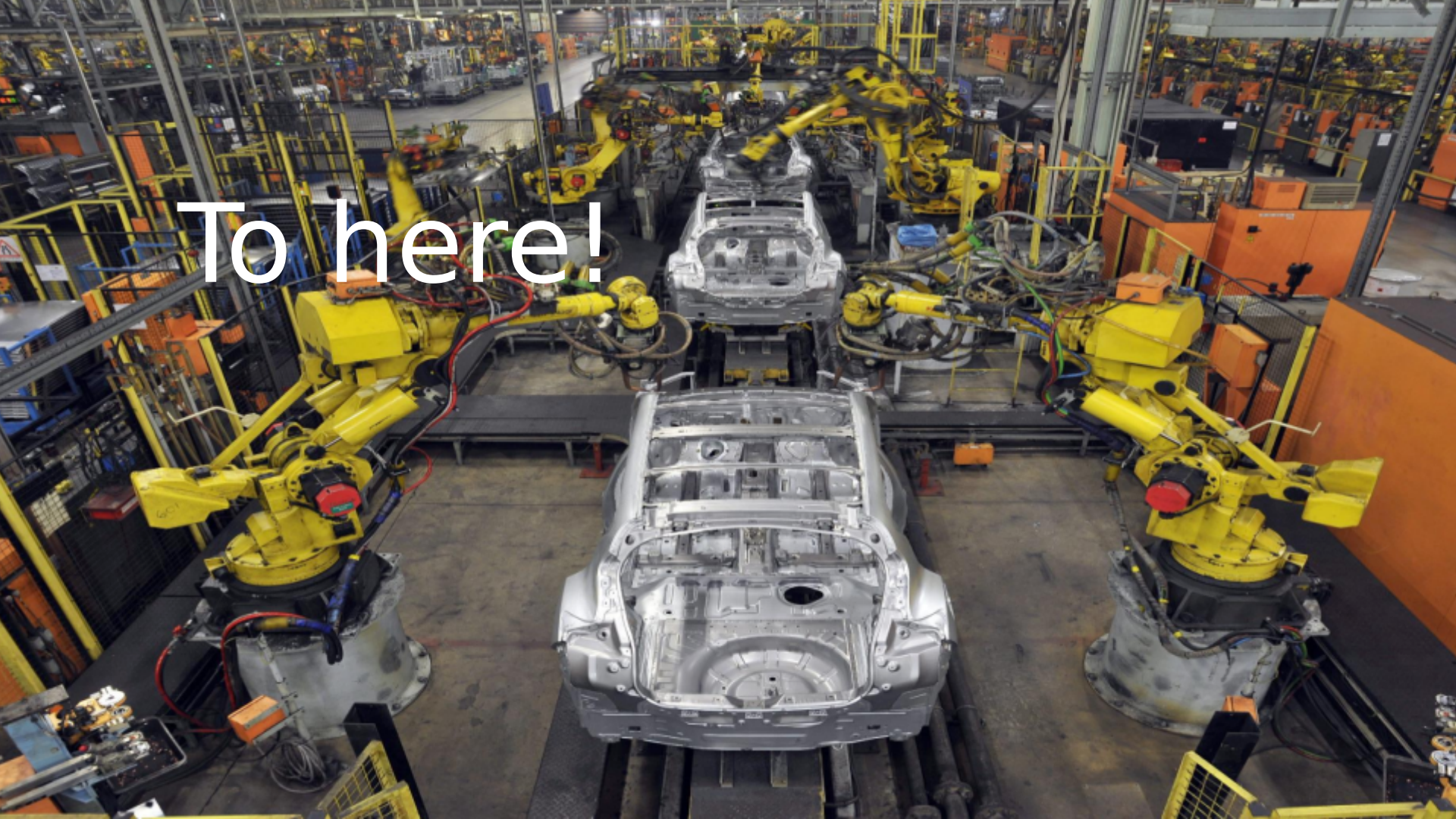


For more Key Metrics

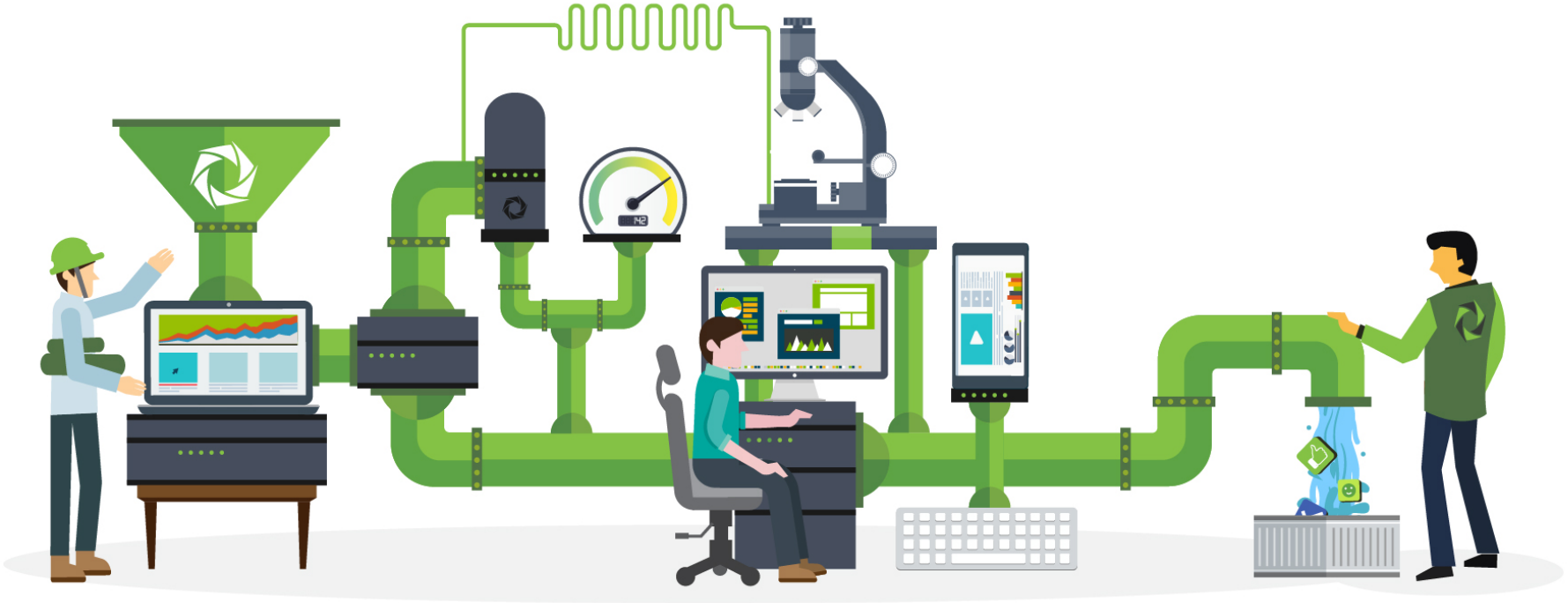
http://blog.dynatrace.com

http://blog.ruxit.com

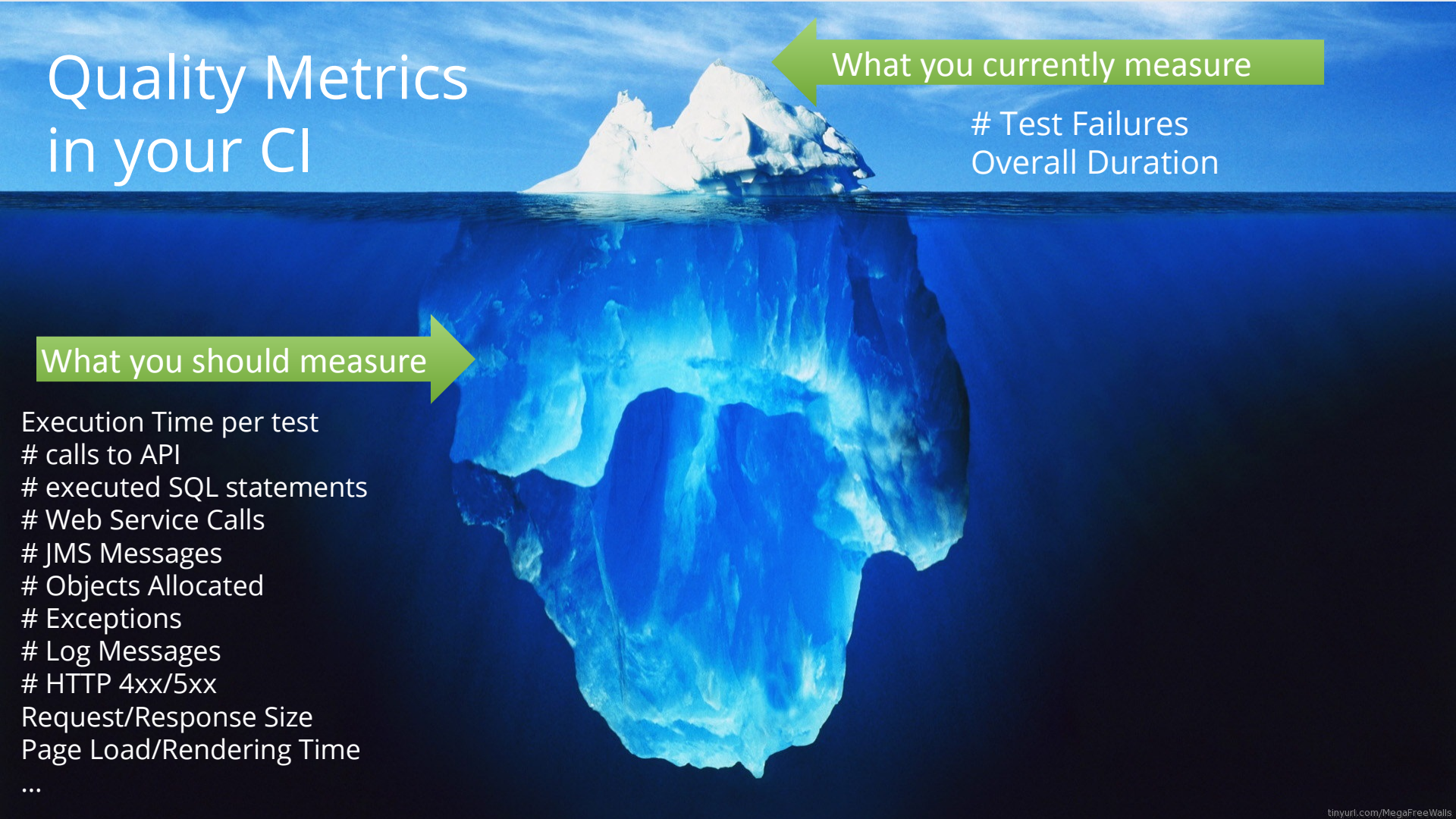

We want to get from here
…


To here!

Use these ***application metrics*** as additional ***Quality Gates***



Quality Metrics in your CI



What you currently measure

Test Failures
Overall Duration

What you should measure

Execution Time per test
calls to API
executed SQL statements
Web Service Calls
JMS Messages
Objects Allocated
Exceptions
Log Messages
HTTP 4xx/5xx
Request/Response Size
Page Load/Rendering Time
...

tinyurl.com/MegaFreeWalls

Connecting your Tests with Quality

Let's look behind the scenes

Test Framework Results			Architectural Data		
Build #	Test Case	Status	# SQL	# Excep	CPU
Build 17	testPurchase	OK	12	0	120ms
	testSearch	OK	3	1	68ms
Build 18	testPurchase	FAILED	12	5	60ms
	testSearch	OK	3	1	68ms
Build 19	testPurchase	OK	75	0	230ms
	testSearch	OK	3	1	68ms
Build 20	testPurchase	O	12	0	120ms
	testSea				ms

We identified a re sion

Problem solved

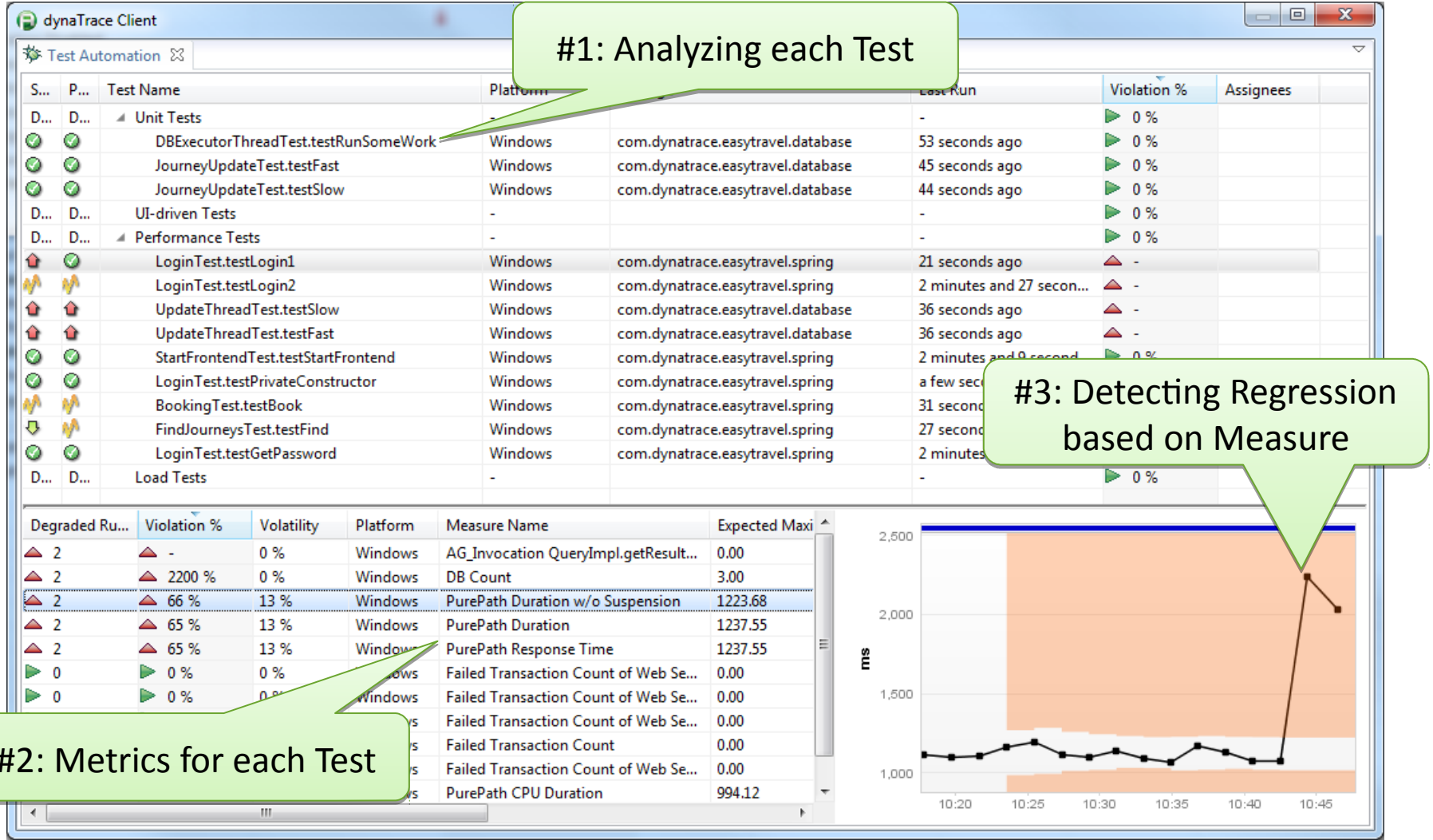
tions probabl for

failed te

Problem fixed but now we have an

ectural regression

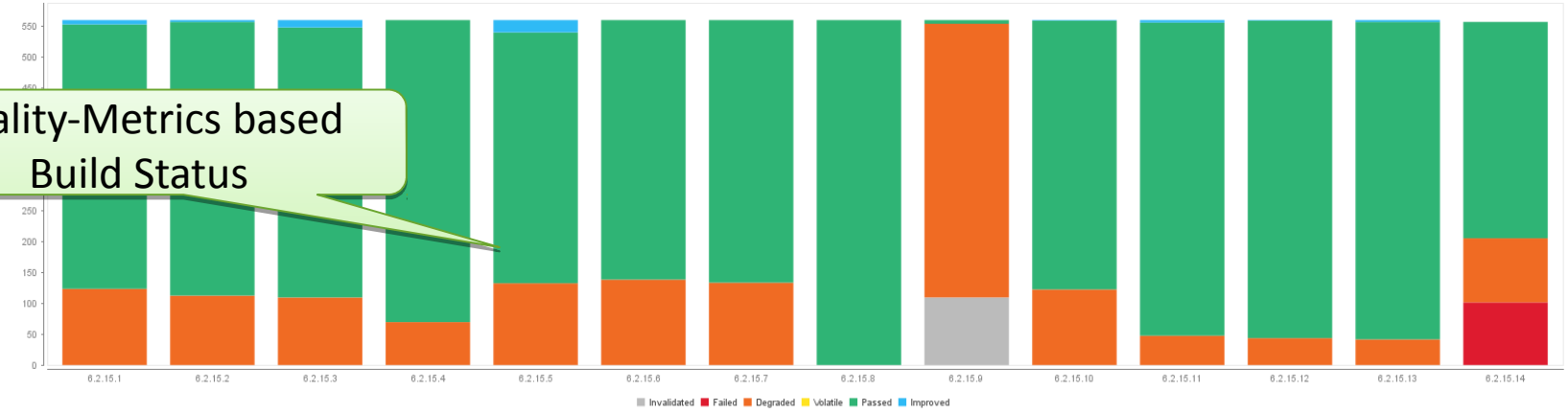
Now we have the functional and architectural confidence


dynaTrace Client
Test Automation
#1: Analyzing each Test
S... P... Test Name Platform Last Run Violation % Assignees
D... D... Unit Tests - 0 %
DBExecutorThreadTest.testRunSomeWork Windows com.dynatrace.easytravel.database 53 seconds ago 0 %
JourneyUpdateTest.testFast Windows com.dynatrace.easytravel.database 45 seconds ago 0 %
JourneyUpdateTest.testSlow Windows com.dynatrace.easytravel.database 44 seconds ago 0 %
D... D... UI-driven Tests - 0 %
D... D... Performance Tests - 0 %
LoginTest.testLogin1 Windows com.dynatrace.easytravel.spring 21 seconds ago -
LoginTest.testLogin2 Windows com.dynatrace.easytravel.spring 2 minutes and 27 secon... -
UpdateThreadTest.testSlow Windows com.dynatrace.easytravel.database 36 seconds ago -
UpdateThreadTest.testFast Windows com.dynatrace.easytravel.database 36 seconds ago -
StartFrontendTest.testStartFrontend Windows com.dynatrace.easytravel.spring
LoginTest.testPrivateConstructor Windows com.dynatrace.easytravel.spring
BookingTest.testBook Windows com.dynatrace.easytravel.spring 31 second
FindJourneysTest.testFind Windows com.dynatrace.easytravel.spring 27 second
LoginTest.testGetPassword Windows com.dynatrace.easytravel.spring 2 minutes
D... D... Load Tests - 0 %
#3: Detecting Regression based on Measure
Degraded Ru... Violation % Volatility Platform Measure Name Expected Maxi
2 - 0 % Windows AG_Invocation QueryImpl.getResult... 0.00
2 2200 % 0 % Windows DB Count 3.00
2 66 % 13 % Windows PurePath Duration w/o Suspension 1223.68
2 65 % 13 % Windows PurePath Duration 1237.55
2 65 % 13 % Windows PurePath Response Time 1237.55
0 0 % 0 % Failed Transaction Count of Web Se... 0.00
0 0 % Failed Transaction Count of Web Se... 0.00
Failed Transaction Count of Web Se... 0.00
Failed Transaction Count 0.00
Failed Transaction Count of Web Se... 0.00
PurePath CPU Duration 994.12
#2: Metrics for each Test
ms
2,500
2,000
1,500
1,000
10:20
10:25
10:30
10:35
10:40
10:45

Test Overview

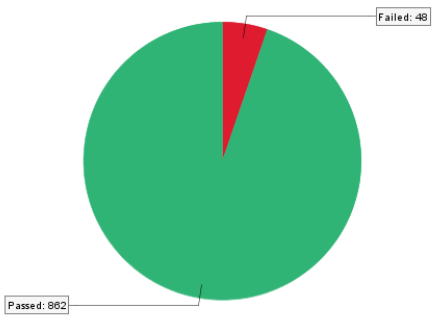
Test Automation Overview

Quality-Metrics based Build Status

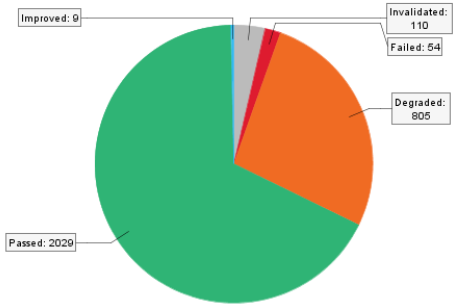


Details for all builds

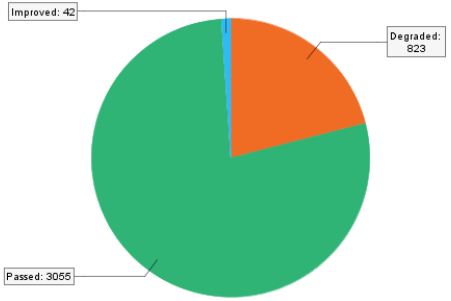
Unit Tests



Performance Tests



Web API Tests



Analyze Load Test Results

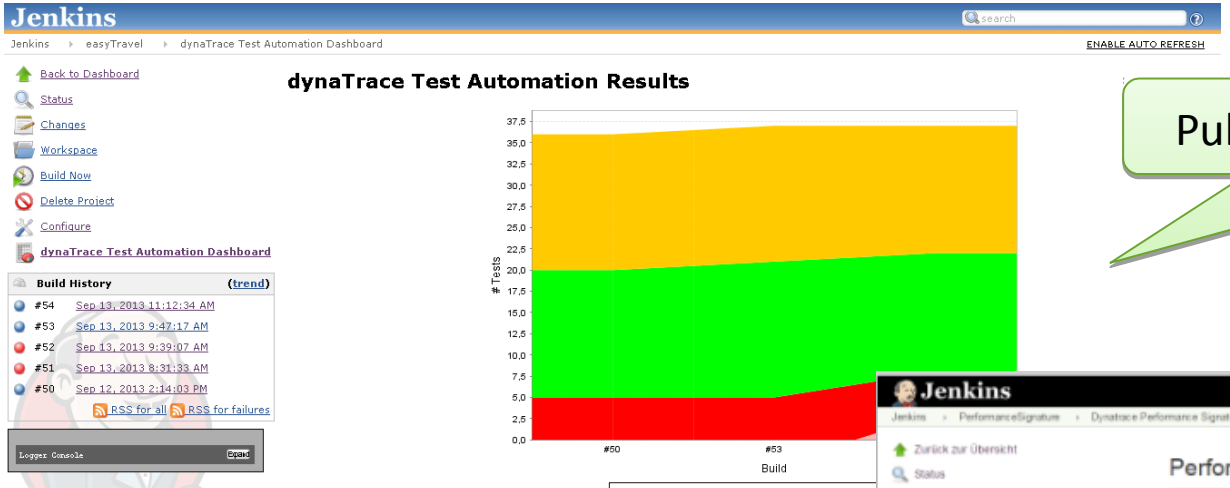
Analyze the synthetic transactions of a load test and quickly identify your performance and scalability bottlenecks.

Learn more about integrating load test tools with dynaTrace.

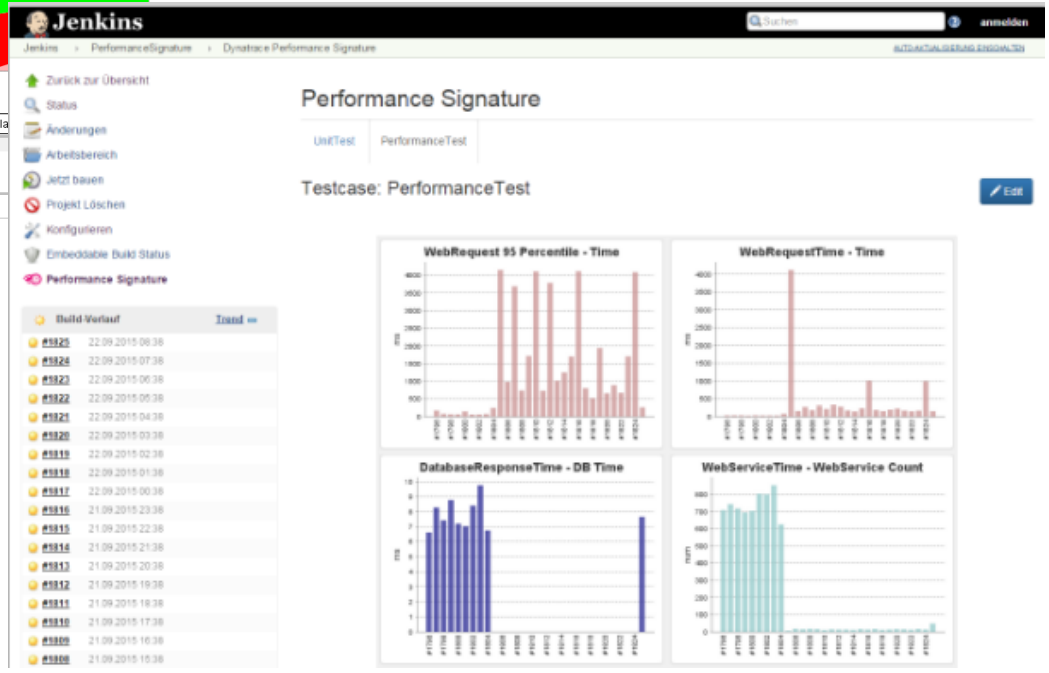
Integrate dynaTrace in your build process

Integrate dynaTrace with Ant, Maven, Jenkins, ...

Learn more about integrating dynaTrace in all stages of your delivery pipeline.



Pull data into Jenkins, Bamboo ...



Making Quality a first-class citizen

„Too hard“

„not cool enough“

United flights resume after glitch grounds planes

„we‘ll get round to this later“

Questions and/or Demo

Slides: slideshare.net/grabnerandi

Get Tools: bit.ly/dttrial

YouTube Tutorials: bit.ly/dttutorials

Contact Me: agrabner@dynatrace.com

Follow Me: @grabnerandi

Read More: blog.dynatrace.com


Andreas Grabner
Dynatrace Developer Advocate
@grabnerandi
http://blog.dynatrace.com