
Continuous Performance Tests

Developer Conference 2012

Tobias Schlitt (@tobySen)
Manuel Pichler (@manuelp)

September 7, 2012

About us

- ▶ **Manuel & Toby**



About us

- ▶ **Manuel & Toby**
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP



About us

- ▶ **Manuel & Toby**
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects



About us

- ▶ **Manuel & Toby**
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects

Co-founders of



About us

- ▶ **Manuel & Toby**
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects

Co-founders of



**Helping people to create
high quality web
applications.**

About us

- ▶ **Manuel & Toby**
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects

Co-founders of



**Helping people to create
high quality web
applications.**

<http://qafoo.com>

Outline

Motivation

Conclusion

Motivation

- ▶ Why should we do performance tests?

Motivation

- ▶ Why should we do performance tests?
 - ▶ Locate unknown bottlenecks
 - ▶ Measure behaviour of the full stack

Motivation

- ▶ Why should we do performance tests?
 - ▶ Locate unknown bottlenecks
 - ▶ Measure behaviour of the full stack
- ▶ Why should we do that continuously?

Motivation

- ▶ Why should we do performance tests?
 - ▶ Locate unknown bottlenecks
 - ▶ Measure behaviour of the full stack
- ▶ Why should we do that continuously?
 - ▶ Find performance regressions
 - ▶ Ensure optimizations are persistent

Often used tools

- ▶ Often misused tools

Often used tools

- ▶ Often misused tools
 - ▶ siege
 - ▶ ApacheBench (ab)

Often used tools

- ▶ Often misused tools
 - ▶ siege
 - ▶ ApacheBench (ab)
- ▶ Testing for micro-optimizations

Often used tools

- ▶ Often misused tools
 - ▶ siege
 - ▶ ApacheBench (ab)
- ▶ Testing for micro-optimizations
 - ▶ Evaluating Hello-World-examples of Frameworks

The problem is more complex

- ▶ Your task: Create a new webshop


The problem is more complex

- ▶ Your task: Create a new webshop
 - ▶ Assume it's march

The problem is more complex


- ▶ Your task: Create a new webshop
 - ▶ Assume it's march
 - ▶ The deadline is October this year, right before Christmas

Awesome Shop

 0 articles
0.00 €

Smartphone

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



1337,-- €
5 items in stock

Comments




● ● ● ● ● ● ● ● ● ●

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor.


A webshop

Awesome Shop

 2 articles
42.32 €


Smartphone

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



1337,- €
5 items in stock

Comments




● ● ● ● ● ● ● ●

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor.


A webshop

Awesome Shop

 2 articles
42.32 €


Smartphone

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



1337,-- €
5 items in stock

Comments




● ● ● ● ● ● ● ● ● ●

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor.


A webshop

Awesome Shop

 2 articles
42.32 €


Smartphone

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



1337,- €
5 items in stock

Comments




5 stars

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor.


A webshop

Awesome Shop

 2 articles
42.32 €


Smartphone

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



1337,-- €
5 items in stock

Comments



● ● ● ● ● ● ● ●

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor.

A real tool for performance tests

- ▶ JMeter

A real tool for performance tests

- ▶ JMeter
 - ▶ Complex user paths

A real tool for performance tests

- ▶ **JMeter**
 - ▶ Complex user paths
 - ▶ Concurrent requests

A real tool for performance tests

- ▶ **JMeter**
 - ▶ Complex user paths
 - ▶ Concurrent requests
 - ▶ Record on proxy

A real tool for performance tests

- ▶ **JMeter**
 - ▶ Complex user paths
 - ▶ Concurrent requests
 - ▶ Record on proxy
 - ▶ **Clustering**

JMeter 101

- ▶ Thread Group
- ▶ Controller
- ▶ Config Element
- ▶ Timer
- ▶ Sampler

Getting started

- ▶ Create a test plan
 - ▶ What do users actually do on your site?

Getting started

- ▶ Create a test plan
 - ▶ What do users actually do on your site?
- ▶ Example:
 - ▶ Random browser
 - ▶ User registration
 - ▶ Sign on
 - ▶ Shopping with checkout
 - ▶ Commenting products

Using JMeter

Demo

Automation

- ▶ Ant JMeter integration

Automation

- ▶ Ant JMeter integration
- ▶ **Automation of your environment**

Automation

- ▶ Ant JMeter integration
- ▶ Automation of your environment
 - ▶ Setting up different software versions

Automation

- ▶ Ant JMeter integration
- ▶ Automation of your environment
 - ▶ Setting up different software versions
 - ▶ Testing with different extensions

Automation

- ▶ Ant JMeter integration
- ▶ **Automation of your environment**
 - ▶ Setting up different software versions
 - ▶ Testing with different extensions
 - ▶ Running different database setups

Automation

- ▶ Ant JMeter integration
- ▶ **Automation of your environment**
 - ▶ Setting up different software versions
 - ▶ Testing with different extensions
 - ▶ Running different database setups
 - ▶ **Handling multi-node environments**

Automation

- ▶ Ant JMeter integration
- ▶ **Automation of your environment**
 - ▶ Setting up different software versions
 - ▶ Testing with different extensions
 - ▶ Running different database setups
 - ▶ Handling multi-node environments
 - ▶ **Plain Ant or Puppet, Chef, Vagrant**

Apache Ant example

```
26 <target name="build-apc-bytecode-cache-user-cache-file"  
27     depends="setup-apc-bytecode-file-cache, _build"  
28     description="Run with APC opcode cache and file based cache." />  
29  
30 <target name="build-apc-bytecode-cache-user-cache-apc"  
31     depends="setup-apc-bytecode-apc-cache, _build"  
32     description="Run with APC opcode cache and APC based cache." />  
  
211 <target name="-remote-exec-parallel">  
212     <subant target="${target}" inheritall="true">  
213         <fileset dir="${project.dir}" includes="server*.xml" />  
214     </subant>  
215 </target>  
  
216  
217 <target name="-remote-exec">  
218     <sshexec command="${command}"  
219         username="${ssh.username}"  
220         password="${ssh.password}"  
221         host="${hostname}"  
222         trust="true" />  
223 </target>
```

Apache Ant example

```
26 <target name="build-apc-bytecode-cache-user-cache-file"  
27     depends="setup-apc-bytecode-file-cache, _build"  
28     description="Run with APC opcode cache and file based cache." />  
29  
30 <target name="build-apc-bytecode-cache-user-cache-apc"  
31     depends="setup-apc-bytecode-apc-cache, _build"  
32     description="Run with APC opcode cache and APC based cache." />  
  
211 <target name="-remote-exec-parallel">  
212     <subant target="{target}" inheritall="true">  
213         <fileset dir="{project.dir}" includes="server*.xml" />  
214     </subant>  
215 </target>  
  
216 <target name="-remote-exec">  
217     <sshexec command="{command}"  
218         username="{ssh.username}"  
219         password="{ssh.password}"  
220         host="{hostname}"  
221         trust="true" />  
222 </target>
```

Apache Ant example

```
26 <target name="build-apc-bytecode-cache-user-cache-file"  
27     depends="setup-apc-bytecode-file-cache, _build"  
28     description="Run with APC opcode cache and file based cache." />  
29  
30 <target name="build-apc-bytecode-cache-user-cache-apc"  
31     depends="setup-apc-bytecode-apc-cache, _build"  
32     description="Run with APC opcode cache and APC based cache." />  
  
211 <target name="-remote-exec-parallel">  
212     <subant target="{target}" inheritall="true">  
213         <fileset dir="{project.dir}" includes="server*.xml" />  
214     </subant>  
215 </target>  
216  
217 <target name="-remote-exec">  
218     <sshexec command="{command}"  
219         username="{ssh.username}"  
220         password="{ssh.password}"  
221         host="{hostname}"  
222         trust="true" />  
223 </target>
```

Apache Ant example

```
26 <target name="build-apc-bytecode-cache-user-cache-file"  
27     depends="setup-apc-bytecode-file-cache, _build"  
28     description=">_Run_with_APC_opcode_cache_and_file_based_cache." />  
29  
30 <target name="build-apc-bytecode-cache-user-cache-apc"  
31     depends="setup-apc-bytecode-apc-cache, _build"  
32     description=">_Run_with_APC_opcode_cache_and_APC_based_cache." />
```

```
211 <target name="-remote-exec-parallel">  
212     <subant target="${target}" inheritall="true">  
213         <fileset dir="${project.dir}" includes="server*.xml" />  
214     </subant>  
215 </target>  
216  
217 <target name="-remote-exec">  
218     <sshexec command="${command}"  
219         username="${ssh.username}"  
220         password="${ssh.password}"  
221         host="${hostname}"  
222         trust="true" />  
223 </target>
```

Apache Ant example

```
26 <target name="build-apc-bytecode-cache-user-cache-file"
27     depends="setup-apc-bytecode-file-cache, build"
28     description="Run with APC opcode cache and file based cache." />
29
30 <target name="build-apc-bytecode-cache-user-cache-apc"
31     depends="setup-apc-bytecode-apc-cache, build"
32     description="Run with APC opcode cache and APC based cache." />
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111 <target name="-remote-exec-parallel">
112     <subant target="{target}" inheritall="true">
113         <fileset dir="{project.dir}" includes="server*.xml" />
114     </subant>
115 </target>
116
117 <target name="-remote-exec">
118     <sshexec command="{command}"
119         username="{ssh.username}"
120         password="{ssh.password}"
121         host="{hostname}"
122         trust="true" />
123 </target>
```

Apache Ant example

```
454 <target name="-restart-host">
455     <antcall target="-remote-exec">
456         <param name="command" value="shutdown -r now" />
457     </antcall>
458
459     <echo taskname="waitfor" message="Wait for ${hostname} to stop ..." />
460     <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
461         <not>
462             <http url="http://${hostname}"/>
463         </not>
464     </waitfor>
465
466     <echo taskname="waitfor" message="Wait for ${hostname} is up again ..." />
467     <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
468         <http url="http://${hostname}"/>
469     </waitfor>
470 </target>
```

Apache Ant example

```
454 <target name="-restart-host">
455   <antcall target="-remote-exec">
456     <param name="command" value="shutdown_r_now" />
457   </antcall>
458
459   <echo taskname="waitfor" message="Wait_for_${hostname}_to_stop ..." />
460   <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
461     <not>
462       <http url="http://${hostname}"/>
463     </not>
464   </waitfor>
465
466   <echo taskname="waitfor" message="Wait_for_${hostname}_is_up_again ..." />
467   <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
468     <http url="http://${hostname}"/>
469   </waitfor>
470 </target>
```

Apache Ant example

```
454 <target name="-restart-host">
455     <antcall target="-remote-exec">
456         <param name="command" value="shutdown_r_now" />
457     </antcall>
458
459     <echo taskname="waitfor" message="Wait_for_${hostname}_to_stop..." />
460     <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
461         <not>
462             <http url="http://${hostname}"/>
463         </not>
464     </waitfor>
465
466     <echo taskname="waitfor" message="Wait_for_${hostname}_is_up_again..." />
467     <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
468         <http url="http://${hostname}"/>
469     </waitfor>
470 </target>
```


Apache Ant example

```
454 <target name="-restart-host">
455   <antcall target="-remote-exec">
456     <param name="command" value="shutdown_r_now" />
457   </antcall>
458
459   <echo taskname="waitfor" message="Wait_for_${hostname}_to_stop ..." />
460   <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
461     <not>
462       <http url="http://${hostname}"/>
463     </not>
464   </waitfor>
465
466   <echo taskname="waitfor" message="Wait_for_${hostname}_is_up_again ..." />
467   <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
468     <http url="http://${hostname}"/>
469   </waitfor>
470 </target>
```

Apache Ant example

```
421 <target name="jmeter" depends="-settings-init,-start-jmeter" />
422
423 <target name="-start-jmeter">
424   <antcall target="-start-jmeter-before-hook" />
425
426   <jmeter jmeterhome="${local.jmeter.home.dir}"
427     resultlog="${local.jmeter.log.file}"
428     testplan="${local.jmeter.test.dir}/${jmeter.file}">
429
430     <property name="jmeter.data.dir" value="${local.project.data.dir}" />
431     <property name="jmeter.rampup.time" value="${jmeter.rampup.time}" />
432     <property name="jmeter.execution.time" value="${jmeter.execution.time}" />
433   </jmeter>
434
435   <antcall target="-start-jmeter-after-hook" />
436 </target>
```

Apache Ant example

```
421 <target name="jmeter" depends="-settings-init,-start-jmeter" />
422
423 <target name="-start-jmeter">
424   <antcall target="-start-jmeter-before-hook" />
425
426   <jmeter jmeterhome="${local.jmeter.home.dir}"
427     resultlog="${local.jmeter.log.file}"
428     testplan="${local.jmeter.test.dir}/${jmeter.file}">
429
430     <property name="jmeter.data.dir" value="${local.project.data.dir}" />
431     <property name="jmeter.rampup.time" value="{jmeter.rampup.time}" />
432     <property name="jmeter.execution.time" value="{jmeter.execution.time}" />
433   </jmeter>
434
435   <antcall target="-start-jmeter-after-hook" />
436 </target>
```

Apache Ant example

```
351 <target name="-backup-data-from-host">
352   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      error.log}'"
353     localtofile="${local.builddir}/${hostname}-error.log"
354     trust="true" />
355
356   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      access.log}'"
357     localtofile="${local.builddir}/${hostname}-access.log"
358     trust="true" />
359
360   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.
      log}'"
361     localtofile="${local.builddir}/${hostname}-php_errors.log"
362     trust="true" />
363
364   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/
      scripts/load.log'"
365     localtofile="${local.builddir}/${hostname}-load.log"
366     trust="true" />
367
368 </target>
```

Apache Ant example

```
351 <target name="-backup-data-from-host">
352   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
353     error.log}'"
354     localtofile="${local.builddir}/${hostname}-error.log"
355     trust="true" />
356
357   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
358     access.log}'"
359     localtofile="${local.builddir}/${hostname}-access.log"
360     trust="true" />
361
362   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.
363     log}'"
364     localtofile="${local.builddir}/${hostname}-php_errors.log"
365     trust="true" />
366
367   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/
368     scripts/load.log'"
369     localtofile="${local.builddir}/${hostname}-load.log"
370     trust="true" />
371 </target>
```

Apache Ant example

```
351 <target name="-backup-data-from-host">
352   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      error.log}'"
353     localtofile="${local.builddir}/${hostname}-error.log"
354     trust="true" />
355
356   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      access.log}'"
357     localtofile="${local.builddir}/${hostname}-access.log"
358     trust="true" />
359
360   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.
      log}'"
361     localtofile="${local.builddir}/${hostname}-php_errors.log"
362     trust="true" />
363
364   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/
      scripts/load.log'"
365     localtofile="${local.builddir}/${hostname}-load.log"
366     trust="true" />
367
368 </target>
```

Apache Ant example

```
351 <target name="-backup-data-from-host">
352   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      error.log}'"
353     localtofile="${local.builddir}/${hostname}-error.log"
354     trust="true" />
355
356   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      access.log}'"
357     localtofile="${local.builddir}/${hostname}-access.log"
358     trust="true" />
359
360   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.
      log}'"
361     localtofile="${local.builddir}/${hostname}-php_errors.log"
362     trust="true" />
363
364   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/
      scripts/load.log'"
365     localtofile="${local.builddir}/${hostname}-load.log"
366     trust="true" />
367
368 </target>
```

Apache Ant example

```
351 <target name="-backup-data-from-host">
352   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      error.log}'"
353     localtofile="${local.builddir}/${hostname}-error.log"
354     trust="true" />
355
356   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.
      access.log}'"
357     localtofile="${local.builddir}/${hostname}-access.log"
358     trust="true" />
359
360   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.
      log}'"
361     localtofile="${local.builddir}/${hostname}-php_errors.log"
362     trust="true" />
363
364   <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/
      scripts/load.log'"
365     localtofile="${local.builddir}/${hostname}-load.log"
366     trust="true" />
367
368 </target>
```


Ant 2.0

```
78 <commons.scriptdef name="parallel.extension-point" language="javascript">
79   <attribute name="target" /><![CDATA[
80     var targets = project.getTargets();
81     var dependencies = targets.get(
82       attributes.get("target")
83     ).getDependencies();
84
85     var parallel = project.createTask("parallel");
86     while(dependencies.hasMoreElements()) {
87       (function(dependency) {
88         var target = targets.get(dependency);
89         var task = project.createTask("antcall");
90         task.setTarget(target);
91         parallel.addTask(task);
92       })(dependencies.nextElement());
93
94     self.project.setProperty( 'commons.scriptdef.failed', false );
95     try {
96       parallel.perform();
97     } catch ( e ) {
98       self.project.setProperty( 'commons.scriptdef.failed', true );
99     }
100   } ]]>
101 </commons.scriptdef>
```

Hardware

- ▶ Test in a realistic environment

Hardware

- ▶ Test in a realistic environment
 - ▶ If your software runs in the cloud test against virtual environment
 - ▶ If you use real hardware, also test against real hardware

Hardware

- ▶ Test in a realistic environment
 - ▶ If your software runs in the cloud test against virtual environment
 - ▶ If you use real hardware, also test against real hardware
- ▶ JMeter might have serious hardware requirements

Hardware

- ▶ Test in a realistic environment
 - ▶ If your software runs in the cloud test against virtual environment
 - ▶ If you use real hardware, also test against real hardware
- ▶ JMeter might have serious hardware requirements
 - ▶ Use real hardware
 - ▶ Use the biggest VM available
 - ▶ Ensure that not the JMeter hardware is the bottleneck

Hardware

- ▶ Test in a realistic environment
 - ▶ If your software runs in the cloud test against virtual environment
 - ▶ If you use real hardware, also test against real hardware
- ▶ JMeter might have serious hardware requirements
 - ▶ Use real hardware
 - ▶ Use the biggest VM available
 - ▶ Ensure that not the JMeter hardware is the bottleneck
- ▶ Be sure that the network is not the bottleneck

Hardware

- ▶ Test in a realistic environment
 - ▶ If your software runs in the cloud test against virtual environment
 - ▶ If you use real hardware, also test against real hardware
- ▶ JMeter might have serious hardware requirements
 - ▶ Use real hardware
 - ▶ Use the biggest VM available
 - ▶ Ensure that not the JMeter hardware is the bottleneck
- ▶ Be sure that the network is not the bottleneck
 - ▶ See `ifstat`, `iftop`

Hardware

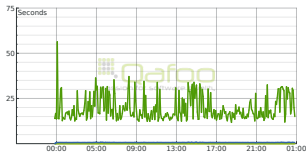
- ▶ Test in a realistic environment
 - ▶ If your software runs in the cloud test against virtual environment
 - ▶ If you use real hardware, also test against real hardware
- ▶ JMeter might have serious hardware requirements
 - ▶ Use real hardware
 - ▶ Use the biggest VM available
 - ▶ Ensure that not the JMeter hardware is the bottleneck
- ▶ Be sure that the network is not the bottleneck
 - ▶ See `ifstat`, `iftop`
- ▶ Measure several system metrics

Hardware

- ▶ Test in a realistic environment
 - ▶ If your software runs in the cloud test against virtual environment
 - ▶ If you use real hardware, also test against real hardware
- ▶ JMeter might have serious hardware requirements
 - ▶ Use real hardware
 - ▶ Use the biggest VM available
 - ▶ Ensure that not the JMeter hardware is the bottleneck
- ▶ Be sure that the network is not the bottleneck
 - ▶ See `ifstat`, `iftop`
- ▶ Measure several system metrics
 - ▶ See `vmstat`, `top`

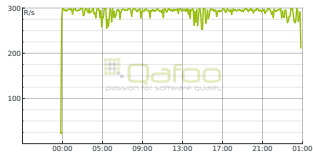
Extended system metrics

Response time



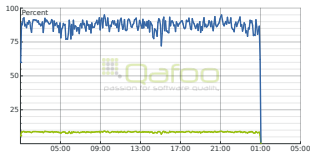
Minimum Average Maximum

Requests per second



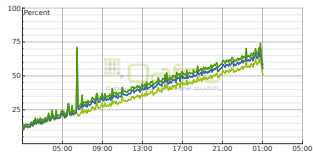
Average

CPU usage



System User

Memory usage



Used Memory Shared Cached

Continuous Performance

- ▶ Jenkins

Continuous Performance

- ▶ Jenkins
 - ▶ Plugin available

Continuous Performance

- ▶ Jenkins
 - ▶ Plugin available
- ▶ Sonar

Continuous Performance

- ▶ Jenkins
 - ▶ Plugin available
- ▶ Sonar
 - ▶ JMeter Plugin available

Continuous testing with Jenkins

Demo

Getting realistic settings

- ▶ Your customer usually only knows his AGOV/IVW values, like:

Getting realistic settings

- ▶ Your customer usually only knows his AGOV/IVW values, like:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month

Getting realistic settings

- ▶ Your customer usually only knows his AGOV/IVW values, like:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Believe us, you will get nothing more :-)

Getting realistic settings

- ▶ Your customer usually only knows his AGOV/IVW values, like:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Believe us, you will get nothing more :-)
 - ▶ Ask for access statistics before Christmas and the ratio compared with regular months

Getting realistic settings

- ▶ Your customer usually only knows his AGOV/IVW values, like:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Believe us, you will get nothing more :-)
 - ▶ Ask for access statistics before Christmas and the ratio compared with regular months
 - ▶ Ask for hours with the highest conversion rates

Getting realistic settings

- ▶ Your customer usually only knows his AGOV/IVW values, like:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Believe us, you will get nothing more :-)
 - ▶ Ask for access statistics before Christmas and the ratio compared with regular months
 - ▶ Ask for hours with the highest conversion rates
 - ▶ Maybe get the aggregated access logs from existing similar shops

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Per day: $1.000.000/26 = 38.500$ (non-business)

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Per day: $1.000.000/26 = 38.500$ (non-business)
- ▶ Per hour: $38.500/12 = 3.200$ (national shop)

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Per day: $1.000.000/26 = 38.500$ (non-business)
- ▶ Per hour: $38.500/12 = 3.200$ (national shop)
- ▶ Peak hour: $3.200 * 8 = 25.500$ (18:00 to 19:00)

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Per day: $1.000.000/26 = 38.500$ (non-business)
- ▶ Per hour: $38.500/12 = 3.200$ (national shop)
- ▶ Peak hour: $3.200 * 8 = 25.500$ (18:00 to 19:00)
- ▶ Per second: $25.500/3600 = 7PI/s$

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Per day: $1.000.000/26 = 38.500$ (non-business)
- ▶ Per hour: $38.500/12 = 3.200$ (national shop)
- ▶ Peak hour: $3.200 * 8 = 25.500$ (18:00 to 19:00)
- ▶ Per second: $25.500/3600 = 7PI/s$
- ▶ Add Christmas / Easter bonus

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Per day: $1.000.000/26 = 38.500$ (non-business)
- ▶ Per hour: $38.500/12 = 3.200$ (national shop)
- ▶ Peak hour: $3.200 * 8 = 25.500$ (18:00 to 19:00)
- ▶ Per second: $25.500/3600 = 7PI/s$
- ▶ Add Christmas / Easter bonus
- ▶ Add launch bonus

Example calculation

- ▶ Customer provided values, for a classic webshop:
 - ▶ 1.000.000 PIs per month
 - ▶ 30.000 sold articles per month
 - ▶ 45.000 registrations per month
- ▶ Per day: $1.000.000/26 = 38.500$ (non-business)
- ▶ Per hour: $38.500/12 = 3.200$ (national shop)
- ▶ Peak hour: $3.200 * 8 = 25.500$ (18:00 to 19:00)
- ▶ Per second: $25.500/3600 = 7PI/s$
- ▶ Add Christmas / Easter bonus
- ▶ Add launch bonus
- ▶ So ... **50 PI/s** should be safe?
 - ▶ Spare resources for scaling are always a business decision
 - ▶ Provide with trade-off: Costs vs. downtime / slowness
 - ▶ Fail gracefully

Outline

Motivation

Conclusion

Conclusion

- ▶ Plan your test scenario

Conclusion

- ▶ Plan your test scenario
- ▶ Use realistic thresholds

Conclusion

- ▶ Plan your test scenario
- ▶ Use realistic thresholds
- ▶ Choose the right tool

Conclusion

- ▶ Plan your test scenario
- ▶ Use realistic thresholds
- ▶ Choose the right tool

with care

Thanks for listening

- ▶ More about us:
 - ▶ <http://qafoo.com>