Continuous Performance Testing

Confoo.ca

Kore Nordmann (@koredn)

27. Feb 2013

About me

- Degree in computer sience

About me

- Degree in computer sience
- Professional PHP since 2000

About me

- Degree in computer sience
- Professional PHP since 2000
- Open source enthusiast

About me

- Degree in computer sience
- Professional PHP since 2000
- Open source enthusiast
- Passion for
  - Software Design
  - Automated Testing

Co-founder of

Co-founder of

Qafoo

passion for software quality

Helping people to create high quality web applications.

`http://qafoo.com`

Co-founder of

Helping people to create high quality web applications.

`http://qafoo.com`

- Expert consulting
- Individual training

Co-founder of

passion for software quality

Helping people to create high quality web applications.

`http://qafoo.com`

- Expert consulting
- Individual training

Get a training on object oriented design for your team!

Outline

Motivation

Conclusion

Motivation

- Why should we do performance tests?

Motivation

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

Motivation

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

Motivation

- Why should we do performance tests?
  - Locate unknown bottlenecks
  - Mesaure 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

Often used tools

- Often misused tools
  - siege
  - ApacheBench (ab)
- Testing for micro-optimizations
  - Evaluating Hello-World-examples of Frameworks
- Useful tools, I won't talk about
  - xDebug Profiling
  - xhProf
  - Database profiling tools
  - System metrics (Graphite, . . .)

The problem is more complex

- Your task: Create a new webshop

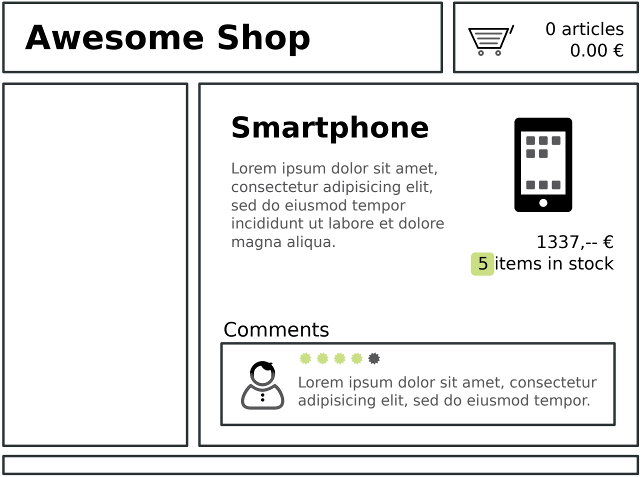
The problem is more complex

- 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

A webshop



Awesome Shop

0 articles
0.00 €

Smartphone

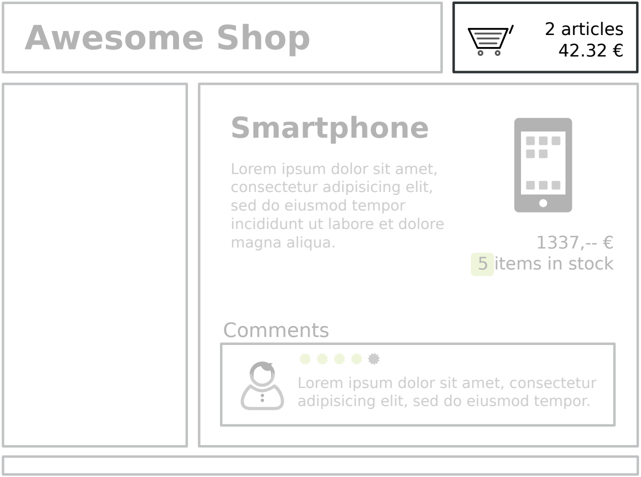
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

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

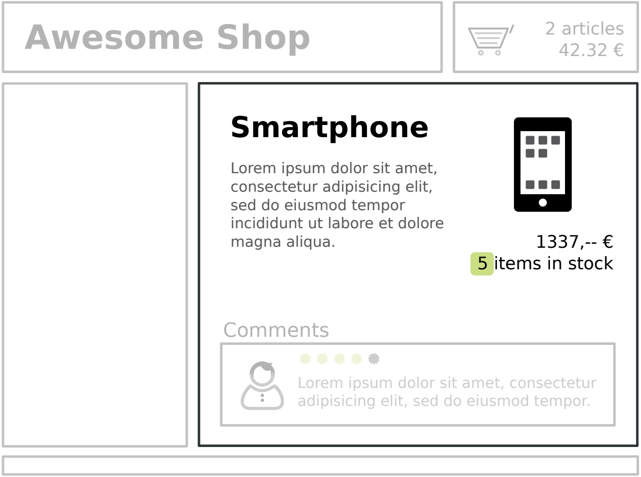
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

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

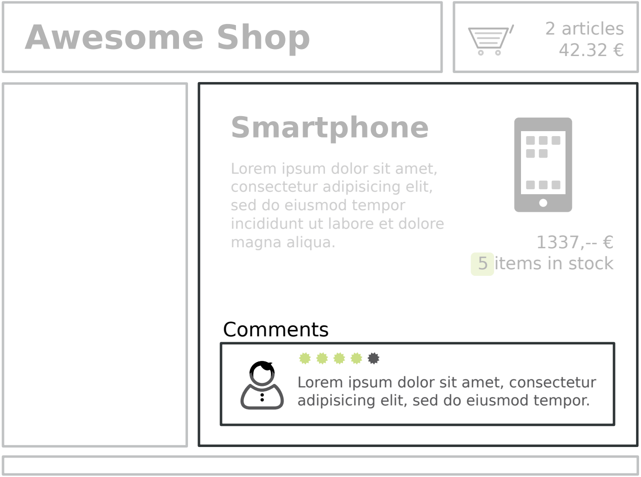
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

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

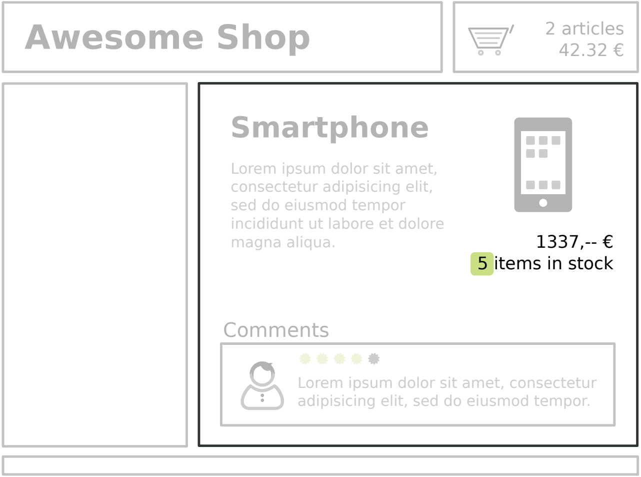
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

1337,-- €
5 items in stock

Comments

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
  - Concurrend requests

A real tool for performance tests

- JMeter
  - Complex user paths
  - Concurrend requests
  - Record on proxy

A real tool for performance tests

- JMeter
  - Complex user paths
  - Concurrend requests
  - Record on proxy
  - Clustering

JMeter 101

- Thread Group
  - An execution plan (like a user registration)

JMeter 101

- Thread Group
  - An execution plan (like a user registration)
- Controller
  - Controls how samplers are executed (loop, random, . . .)

JMeter 101

- Thread Group
  - An execution plan (like a user registration)
- Controller
  - Controls how samplers are executed (loop, random, . . .)
- Config Element
  - Configration, optionally from external sources

JMeter 101

- Thread Group
  - An execution plan (like a user registration)
- Controller
  - Controls how samplers are executed (loop, random, . . .)
- Config Element
  - Configration, optionally from external sources
- Timer
  - Defining timing constraints for sampler execution

JMeter 101

- Thread Group
  - An execution plan (like a user registration)
- Controller
  - Controls how samplers are executed (loop, random, . . .)
- Config Element
  - Configration, optionally from external sources
- Timer
  - Defining timing constraints for sampler execution
- Sampler
  - Perform the actual work (like HTTP requests)

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

```
<target name="build-apc-bytecode-cache-user-cache-file"
        depends="setup-apc-bytecode-file-cache,␣-build"
        description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣file␣based␣cache." />

<target name="build-apc-bytecode-cache-user-cache-apc"
        depends="setup-apc-bytecode-apc-cache,␣-build"
        description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣APC␣based␣cache." />
```

```
<target name="-remote-exec-parallel">
    <subant target="${target}" inheritall="true">
        <fileset dir="${project.dir}" includes="server*.xml" />
    </subant>
</target>

<target name="-remote-exec">
    <sshexec command="${command}"
             username="${ssh.username}"
             password="${ssh.password}"
             host="${hostname}"
             trust="true" />
</target>
```

Apache Ant example

```
    <target name="build-apc-bytecode-cache-user-cache-file"
            depends="setup-apc-bytecode-file-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣file␣based␣cache." />

    <target name="build-apc-bytecode-cache-user-cache-apc"
            depends="setup-apc-bytecode-apc-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣APC␣based␣cache." />
```

```
    <target name="-remote-exec-parallel">
        <subant target="${target}" inheritall="true">
            <fileset dir="${project.dir}" includes="server*.xml" />
        </subant>
    </target>

    <target name="-remote-exec">
        <sshexec command="${command}"
                 username="${ssh.username}"
                 password="${ssh.password}"
                 host="${hostname}"
                 trust="true" />
    </target>
```

Apache Ant example

```
    <target name="build-apc-bytecode-cache-user-cache-file"
            depends="setup-apc-bytecode-file-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣file␣based␣cache." />

    <target name="build-apc-bytecode-cache-user-cache-apc"
            depends="setup-apc-bytecode-apc-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣APC␣based␣cache." />
```

```
    <target name="-remote-exec-parallel">
        <subant target="${target}" inheritall="true">
            <fileset dir="${project.dir}" includes="server*.xml" />
        </subant>
    </target>

    <target name="-remote-exec">
        <sshexec command="${command}"
                 username="${ssh.username}"
                 password="${ssh.password}"
                 host="${hostname}"
                 trust="true" />
    </target>
```

Apache Ant example

```xml
    <target name="build-apc-bytecode-cache-user-cache-file"
            depends="setup-apc-bytecode-file-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣file␣based␣cache." />

    <target name="build-apc-bytecode-cache-user-cache-apc"
            depends="setup-apc-bytecode-apc-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣APC␣based␣cache." />
```

```xml
    <target name="-remote-exec-parallel">
        <subant target="${target}" inheritall="true">
            <fileset dir="${project.dir}" includes="server*.xml" />
        </subant>
    </target>

    <target name="-remote-exec">
        <sshexec command="${command}"
                 username="${ssh.username}"
                 password="${ssh.password}"
                 host="${hostname}"
                 trust="true" />
    </target>
```

Apache Ant example

```
    <target name="build-apc-bytecode-cache-user-cache-file"
            depends="setup-apc-bytecode-file-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣file␣based␣cache." />

    <target name="build-apc-bytecode-cache-user-cache-apc"
            depends="setup-apc-bytecode-apc-cache,␣-build"
            description="->␣␣Run␣with␣APC␣opcode␣cache␣and␣APC␣based␣cache." />
```

```
    <target name="-remote-exec-parallel">
        <subant target="${target}" inheritall="true">
            <fileset dir="${project.dir}" includes="server*.xml" />
        </subant>
    </target>

    <target name="-remote-exec">
        <sshexec command="${command}"
                 username="${ssh.username}"
                 password="${ssh.password}"
                 host="${hostname}"
                 trust="true" />
    </target>
```

Apache Ant example

```
    <target name="-restart-host">
        <antcall target="-remote-exec">
            <param name="command" value="shutdown␣-r␣now" />
        </antcall>

        <echo taskname="waitfor" message="Wait␣for␣${hostname}␣to␣stop..." />
        <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
            <not>
                <http url="http://${hostname}"/>
            </not>
        </waitfor>

        <echo taskname="waitfor" message="Wait␣for␣${hostname}␣is␣up␣again..." />
        <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
            <http url="http://${hostname}"/>
        </waitfor>
    </target>
```

Apache Ant example

```
    <target name="-restart-host">
        <antcall target="-remote-exec">
            <param name="command" value="shutdown␣-r␣now" />
        </antcall>

        <echo taskname="waitfor" message="Wait␣for␣${hostname}␣to␣stop..." />
        <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
            <not>
                <http url="http://${hostname}"/>
            </not>
        </waitfor>

        <echo taskname="waitfor" message="Wait␣for␣${hostname}␣is␣up␣again..." />
        <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
            <http url="http://${hostname}"/>
        </waitfor>
    </target>
```

Apache Ant example

```
    <target name="-restart-host">
        <antcall target="-remote-exec">
            <param name="command" value="shutdown␣-r␣now" />
        </antcall>

        <echo taskname="waitfor" message="Wait␣for␣${hostname}␣to␣stop..." />
        <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
            <not>
                <http url="http://${hostname}"/>
            </not>
        </waitfor>

        <echo taskname="waitfor" message="Wait␣for␣${hostname}␣is␣up␣again..." />
        <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
            <http url="http://${hostname}"/>
        </waitfor>
    </target>
```

Apache Ant example

```
<target name="-restart-host">
    <antcall target="-remote-exec">
        <param name="command" value="shutdown␣-r␣now" />
    </antcall>

    <echo taskname="waitfor" message="Wait␣for␣${hostname}␣to␣stop..." />
    <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
        <not>
            <http url="http://${hostname}"/>
        </not>
    </waitfor>

    <echo taskname="waitfor" message="Wait␣for␣${hostname}␣is␣up␣again..." />
    <waitfor maxwait="5" maxwaitunit="minute" checkevery="100">
        <http url="http://${hostname}"/>
    </waitfor>
</target>
```

Apache Ant example

```
    <target name="jmeter" depends="-settings-init,-start-jmeter" />

    <target name="-start-jmeter">
        <antcall target="-start-jmeter-before-hook" />

        <jmeter jmeterhome="${local.jmeter.home.dir}"
                resultlog="${local.jmeter.log.file}"
                testplan="${local.jmeter.test.dir}/${jmeter.file}">

            <property name="jmeter.data.dir" value="${local.project.data.dir}" />
            <property name="jmeter.rampup.time" value="${jmeter.rampup.time}" />
            <property name="jmeter.execution.time" value="${jmeter.execution.time}" />
        </jmeter>

        <antcall target="-start-jmeter-after-hook" />
    </target>
```

Apache Ant example

```
    <target name="jmeter" depends="-settings-init,-start-jmeter" />

    <target name="-start-jmeter">
        <antcall target="-start-jmeter-before-hook" />

        <jmeter jmeterhome="${local.jmeter.home.dir}"
                resultlog="${local.jmeter.log.file}"
                testplan="${local.jmeter.test.dir}/${jmeter.file}">

            <property name="jmeter.data.dir" value="${local.project.data.dir}" />
            <property name="jmeter.rampup.time" value="${jmeter.rampup.time}" />
            <property name="jmeter.execution.time" value="${jmeter.execution.time}" />
        </jmeter>

        <antcall target="-start-jmeter-after-hook" />
    </target>
```

Apache Ant example

```
    <target name="jmeter" depends="-settings-init,-start-jmeter" />

    <target name="-start-jmeter">
        <antcall target="-start-jmeter-before-hook" />

        <jmeter jmeterhome="${local.jmeter.home.dir}"
                resultlog="${local.jmeter.log.file}"
                testplan="${local.jmeter.test.dir}/${jmeter.file}">

            <property name="jmeter.data.dir" value="${local.project.data.dir}" />
            <property name="jmeter.rampup.time" value="${jmeter.rampup.time}" />
            <property name="jmeter.execution.time" value="${jmeter.execution.time}" />
        </jmeter>

        <antcall target="-start-jmeter-after-hook" />
    </target>
```

Apache Ant example

```
        <target name="-backup-data-from-host">
            <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.error.log}'"
                localtofile="${local.builddir}/${hostname}-error.log"
                trust="true" />

            <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.access.log}'"
                localtofile="${local.builddir}/${hostname}-access.log"
                trust="true" />

            <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.log}'"
                localtofile="${local.builddir}/${hostname}-php_errors.log"
                trust="true" />

            <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/scripts/load.log'"
                localtofile="${local.builddir}/${hostname}-load.log"
                trust="true" />

        </target>
```

Apache Ant example

```
    <target name="-backup-data-from-host">
        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.error.log}'"
             localtofile="${local.builddir}/${hostname}-error.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.access.log}'"
             localtofile="${local.builddir}/${hostname}-access.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.log}'"
             localtofile="${local.builddir}/${hostname}-php_errors.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/scripts/load.log'"
             localtofile="${local.builddir}/${hostname}-load.log"
             trust="true" />

    </target>
```

Apache Ant example

```
    <target name="-backup-data-from-host">
        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.error.log}'"
             localtofile="${local.builddir}/${hostname}-error.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.access.log}'"
             localtofile="${local.builddir}/${hostname}-access.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.log}'"
             localtofile="${local.builddir}/${hostname}-php_errors.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/scripts/load.log'"
             localtofile="${local.builddir}/${hostname}-load.log"
             trust="true" />

    </target>
```

Apache Ant example

```
    <target name="-backup-data-from-host">
        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.error.log}'"
             localtofile="${local.builddir}/${hostname}-error.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.access.log}'"
             localtofile="${local.builddir}/${hostname}-access.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.log}'"
             localtofile="${local.builddir}/${hostname}-php_errors.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/scripts/load.log'"
             localtofile="${local.builddir}/${hostname}-load.log"
             trust="true" />

    </target>
```

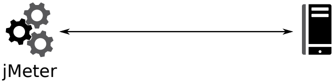
Apache Ant example

```
    <target name="-backup-data-from-host">
        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.error.log}'"
             localtofile="${local.builddir}/${hostname}-error.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.webserver.access.log}'"
             localtofile="${local.builddir}/${hostname}-access.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.php.error.log}'"
             localtofile="${local.builddir}/${hostname}-php_errors.log"
             trust="true" />

        <scp remotefile="${ssh.username}:${ssh.password}@${hostname}:'${remote.basedir}/scripts/load.log'"
             localtofile="${local.builddir}/${hostname}-load.log"
             trust="true" />

    </target>
```

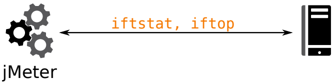
Test Setup

What do we actually test?

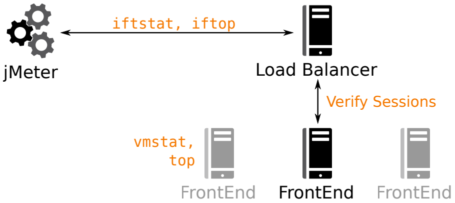
Test Setup



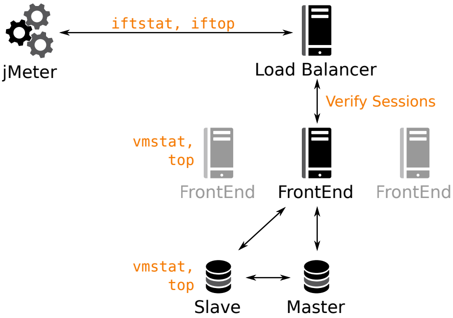
Test Setup



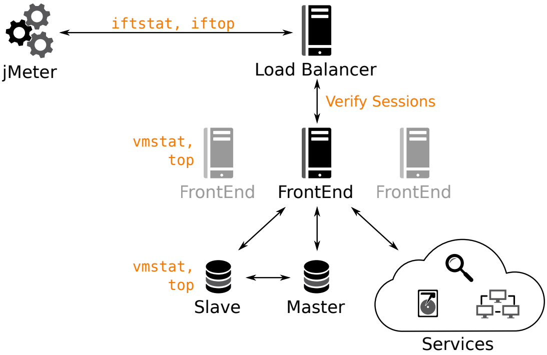
Test Setup


iftstat, iftop
jMeter
Load Balancer
Verify Sessions
vmstat,
top
FrontEnd
FrontEnd
FrontEnd

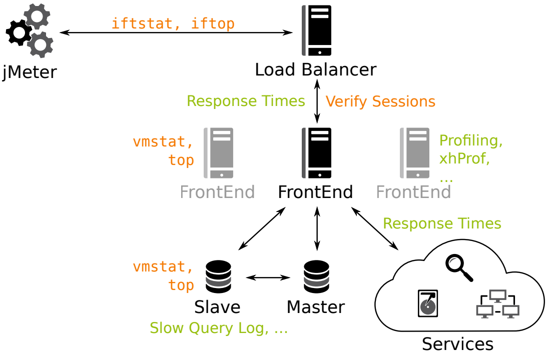
Test Setup


iftstat, iftop
jMeter
Load Balancer
Verify Sessions
vmstat,
top
FrontEnd
FrontEnd
FrontEnd
vmstat,
top
Slave
Master

Test Setup


iftstat, iftop
jMeter
Load Balancer
Verify Sessions
vmstat,
top
FrontEnd
FrontEnd
FrontEnd
vmstat,
top
Slave
Master
Services

Test Setup


iftstat, iftop
jMeter
Load Balancer
Response Times
Verify Sessions
vmstat,
top
FrontEnd
FrontEnd
FrontEnd
Profiling,
xhProf,
...
Response Times
vmstat,
top
Slave
Master
Slow Query Log, ...
Services

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 Statistics

Demo

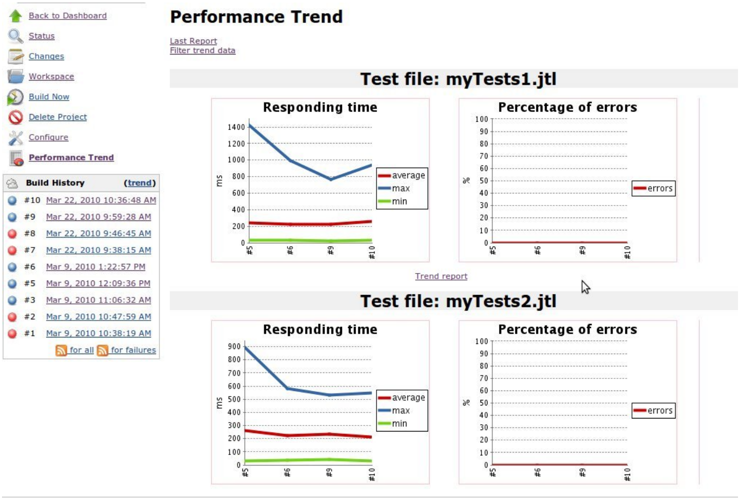
Continuous Performance

- Plugins available for:
  - Jenkins
  - Sonar

Continuous Performance

- Plugins available for:
  - Jenkins
  - Sonar
- Maintaining all those servers can be expensive

Continuous testing with Jenkins



Build History	(trend)
#10	Mar 22, 2010 10:36:48 AM
#9	Mar 22, 2010 9:59:28 AM
#8	Mar 22, 2010 9:46:45 AM
#7	Mar 22, 2010 9:38:15 AM
#6	Mar 9, 2010 1:22:57 PM
#5	Mar 9, 2010 12:09:36 PM
#3	Mar 9, 2010 11:06:32 AM
#2	Mar 9, 2010 10:47:59 AM
#1	Mar 9, 2010 10:38:19 AM

Performance Trend

Last Report
Filter trend data

Test file: myTests1.jtl

Trend report

Test file: myTests2.jtl

Questions

- Common questions:
  - Can we survive christmas?

Questions

- Common questions:
  - Can we survive christmas?
  - What is the maximum we can accomplish with the current setup?

Questions

- Common questions:
  - Can we survive christmas?
  - What is the maximum we can accomplish with the current setup?
  - Does (APC|XCache|MemCache) really help us?

Getting realistic settings

- Ask for real access logs:
  - Extract exact request model from those

Getting realistic settings

- Ask for real access logs:
  - Extract exact request model from those
- Your customer usually only knows very broad values, like:
  - 1.000.000 PIs per month
  - 30.000 sold articles per month
  - 45.000 registrations per month

Getting realistic settings

- Ask for real access logs:
  - Extract exact request model from those
- Your customer usually only knows very broad values, like:
  - 1.000.000 PIs per month
  - 30.000 sold articles per month
  - 45.000 registrations per month
- Very seldom, that you get more:

Getting realistic settings

- Ask for real access logs:
  - Extract exact request model from those
- Your customer usually only knows very broad values, like:
  - 1.000.000 PIs per month
  - 30.000 sold articles per month
  - 45.000 registrations per month
- Very seldom, that you get more:
  - Ask for access statistics before Christmas and the ratio compared with regular months

Getting realistic settings

- Ask for real access logs:
  - Extract exact request model from those
- Your customer usually only knows very broad values, like:
  - 1.000.000 PIs per month
  - 30.000 sold articles per month
  - 45.000 registrations per month
- Very seldom, that you get 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

- Ask for real access logs:
  - Extract exact request model from those
- Your customer usually only knows very broad values, like:
  - 1.000.000 PIs per month
  - 30.000 sold articles per month
  - 45.000 registrations per month
- Very seldom, that you get 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: $3.200 \ast 8 = 25.500$ (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

Verification

- Do your requests actually model customer requirements?
  - Compare generated accesss logs with real access logs
  - Compare user registrations / checkouts per hour with requested values

Outline

Motivation

Conclusion

Conclusion

- Plan your test scenario

Conclusion

- Plan your test scenario
- Use realisitic thresholds

Conclusion

- Plan your test scenario
- Use realisitic thresholds
- Choose the right tool

Conclusion

- Plan your test scenario
- Use realisitic thresholds
- Choose the right tool

with care

Thanks for Listening

Rate this talk: `https://joind.in/7842`

Thanks for Listening

Rate this talk: `https://joind.in/7842`

Stay in touch

- Kore Nordmann
- kore@qafoo.com
- @koredn / @qafoo

Rent a web quality expert:
`http://qafoo.com`