

# **Continuous Performance Evaluation using Open Source Tools**

Markus Dlugi, RETIT GmbH



#### **Motivation**

- Continuous Integration (CI) has reached widespread adoption
- Functional requirements are tested with automated tests
- Non-functional requirements such as performance are often neglected

#### How can we track performance if the application changes every day? Every hour? Every minute?



#### **Motivation**

- Modern APM tools often have features to monitor the performance over multiple application versions
- However, they generally come with a high price tag

#### Maybe we can also do it with Open Source Software?



#### **General workflow**



01 February 2017 • www.retit.de • 4

#### **General workflow**



#### **Alternatives: Version Control System**





### **Alternatives: Continuous Integration Systems**



### **Alternatives: Load Testing Tools**



#### **Alternatives: APM Tools**





# Demo



https://github.com/RETIT/continuous-performance-evaluation

#### **Tip #1: Use Pipelines**

- Pipelines in Jenkins allow to specify all steps using Groovy scripts
  - → Avoid manual configuration
  - → Treat pipeline as code

Pipeline Performance\_Test\_Job



#### Stage View

	Checkout	Build	Deploy	Load test	Undeploy
Average stage times:	2s	59s	1min 37s	3min 28s	5s
#1 No Jan 24 Changes	2s	59s	1min 37s	3min 28s	5s



#### **Tip #2: Use Load Tests correctly**

- Load tests only yield meaningful results if used correctly
- When designing your tests, make sure that:
  - Your workload resembles how users would access the application
  - You achieve steady state performance (test runs long enough?)
  - Your test environment is comparable to your production environment
  - Your test environment has no external influences (e.g., from the load test environment)





#### **Tip #3: Find good Thresholds**

- Thresholds are a good way of notifying developers about performance problems
- However, good thresholds are hard to find
  - If the threshold is too low, developers will start ignoring the warnings
  - If the threshold is too high, potential problems might go unnoticed
- Each application and load test is different
  - Finding good thresholds takes time
  - Consider using absolute thresholds
  - You might also need to remove ramp-up using custom scripts



#### **Tip #4: Run the Pipeline often**

- Ideally, you would run your performance tests after every commit
- If you can't do that, minimum is once a day
- Try to minimize the pipeline run time so you can run it more often





### **Tip #5: Optimize your Pipeline**

204, No Content, CargoUser 1-7,, true, 110, 20, 20, 0 Minimize data which needs to be processed, sen ntent,CargoUser 1-4,,true,110,20,20,0 CargoUser 1-8,,true,110,20,20,0 text, true, 4738, 20, 20, 202 CargoUser 1-16,text,true,18606,20,20,26 or stored CargoUser 1-15,,true,235,20,20,263 Content, CargoUser 1-10,, true, 110, 20, 20, 0 "Number of samples in transaction : 18, number of faili

uest, 204, No Content, CargoUser 1-12, true, 110, 20, 20, 0 ,OK, CargoUser 1-17, text, true, 4738, 20, 20, 109

200, "Number of samples in transaction : 18, number of failing

"Number of samples in transaction : 18, number of failing

"Number of samples in transaction : 18, number of fail

200, OK, CargoUser 1-2, text, true, 8055, 20, 20, 14 OK, CargoUser 1-1, text, true, 8056, 20, 20, 10

age,200,0K,CargoUser 1-20,text,true,8056,20,20,11 ontent, CargoUser 1-6,, true, 110, 20, 20, 0 ntent,CargoUser 1-5,,true,110,20,20,0 Sontent, CargoUser 1-11,, true, 110, 20, 20, 0

SargoUser 1-14,,true,235,20,20,544

Jser 1-19, text, true, 4738, 20, 20, 395

3,200,0K,CargoUser 1-3,text,true,8056,20,20,8

Content, CargoUser 1-13, true, 110, 20, 20, 0 00, "Number of samples in transaction : 18, number of fail

JA, Jaryouset 1-20, text, true, 1:30, 20, 20, 24, 24 (n) and 200 of samples in transaction : 18, number of fail

00,0K,CargoUser 1-4,text,true,8057,20,20,8 CargoUser 1-18, text, true, 11144, 20, 20, 19

Content, CargoUser 1-10,, true, 110, 20, 20, 0

ationpage,200,0K,Cargotser 1-5,text,true,8055,20,20,11

204,NO Content,CargoUser 1-11,,true,110,20,20,00 J. OK, CargoUser 1-19, text, true, 16330, 20, 20, 20, 29 Cargotser 1-3, text, true, 4738, 20, 20, 274 Non-the state of the state of t

CargoRequest,204,No Content,Cargouser 1-5,.true,110,20,20, GagRequest,204,No Content,CargoUser 1-13,.true,110,20,20,0 ArnoRoute.302.Found.CargoUser 1-17..true.235.20,20,20,0 GORequest,204,No Content,Cargouser 1-13,,Urue,110,20,2 ArgoRoute,302,Found,CargoUser 1-17,,true,235,20,20,295

CargoUser 1-2, text, true, 4738, 20, 20, 134

K, CargoUser 1-1, text, true, 4738, 20, 20, 376 A, No Content, CargoUser 1-6, true, 110, 20, 20, 0 OK, CargoUser 1-20, text, true, 4738, 20, 20, 244

- Use JMeter's CSV output instead of XML
- Don't store every Jenkins build, introduce log rotation
- data generated during the last test execution data down and the last data down and the last test execution data down and the last down and the last data down and the last down an 2,Found,CargoUser 1-16,,true,235,20,20,188 ,204,No Content,CargoUser 1-15,,true,110,20,20,0 204, No Content, CargoUser 1-14, true, 110, 20, 20, 0
- Use caching effectively
  - Maven's local repository
  - Docker's image cache



## Markus Dlugi dlugi@retit.de



Resource Efficient Technologies & IT Systems



01 February 2017 • www.retit.de • 16