



# PerfCake

Good, Better, PerfCake!



Pavel Macík ([pmacik@perfcake.org](mailto:pmacik@perfcake.org))  
Martin Večeřa ([mvecera@perfcake.org](mailto:mvecera@perfcake.org))  
<http://goo.gl/6eazZe>

[www.perfcake.org](http://www.perfcake.org)

# Jak jsme se sem dostali?

14:46 z Brna, skupina A - Renault Scénic III  
Bratislava, Nitra, Zvolen, Lučenec (zastávka),  
Rimavská Sobota, Rožňava - 550km



# Jak jsme se sem dostali?

- spolupráce s univerzitami
  - BP, DP, kurzy, hledání talentů
- co je cílem studia?
- co vám můžeme ukázat?
  - jak se dělá open-source projekt
  - měření výkonu aplikací



# Testování výkonu...

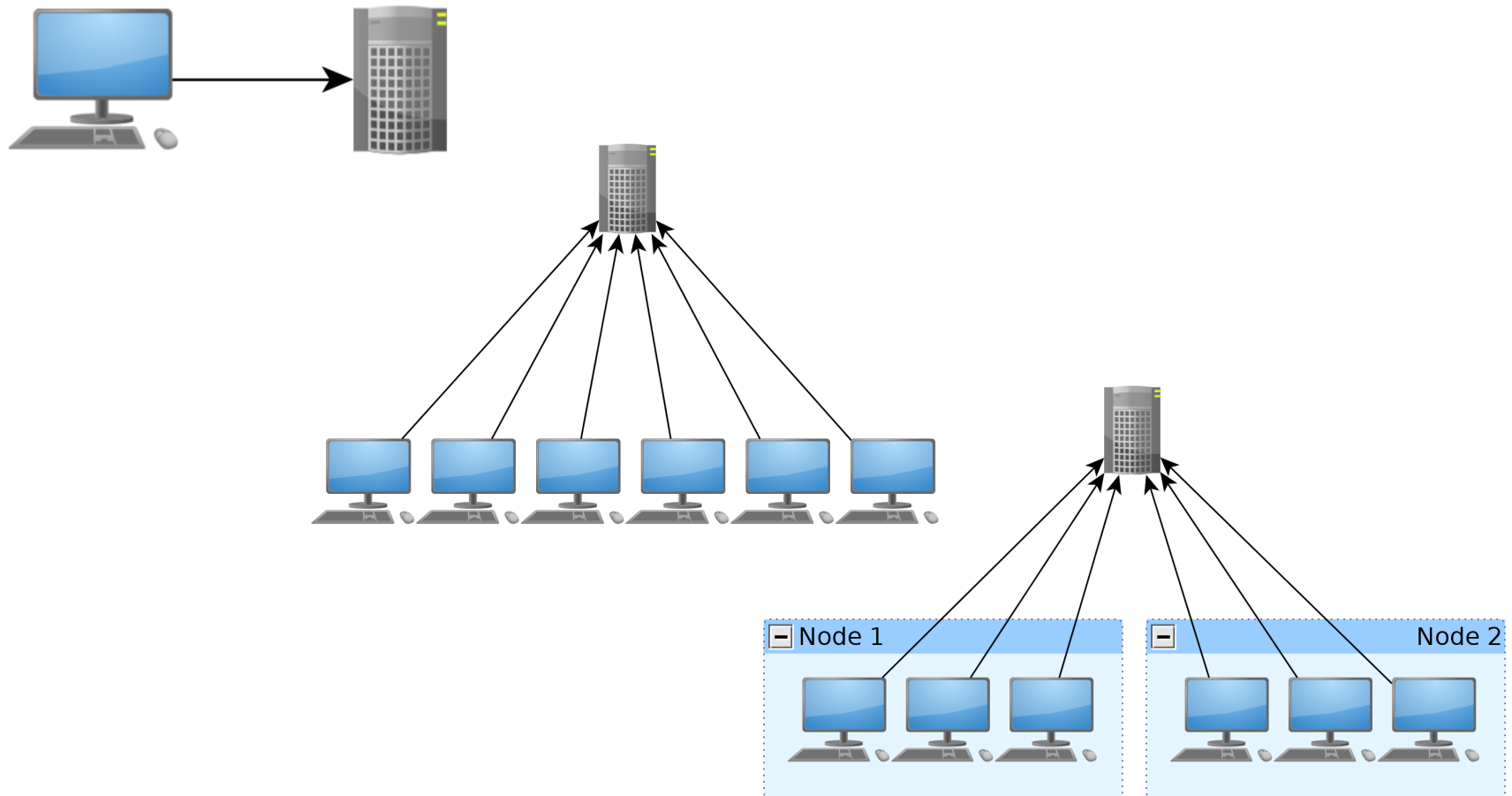




...než systém havaruje



# Aspekty testování výkonu



# Aspekty testování výkonu

- počet klientů/vláken
- velikost požadavků/zpráv
- způsob generování zátěže
  - backlog, throughput, ramp up
- propustnost - maximální nebo cílová
- počet iterací, doba
- čas odpovědi, spotřeba paměti
- jak často měřit
- vliv na měřený systém



# Skrytá rizika

- zahřívání aplikace
- spolehlivost výsledků
  - rozptyl
  - propustnost měřicího nástroje
- měřený systém příliš ovlivněný
- stabilita prostředí
  - operační systém
  - verze knihoven
  - komunikační kanály - síť
- validní odpovědi od aplikace
- virtuální prostředí

# Požadavky na nástroj

- malý vliv na měřený systém
- flexibilita
  - způsob generování, množství požadavků/klientů
  - transportní protokol, prezentace výsledků
  - validace odpovědí
- jednoduché použití - uživatelská přítulnost
- vysoká propustnost, spolehlivost
- opakovatelnost měření
- přenositelnost, obecnost použití
- vizualizace výsledků

# Konkurence



## The Grinder



radargun



# Konkurence

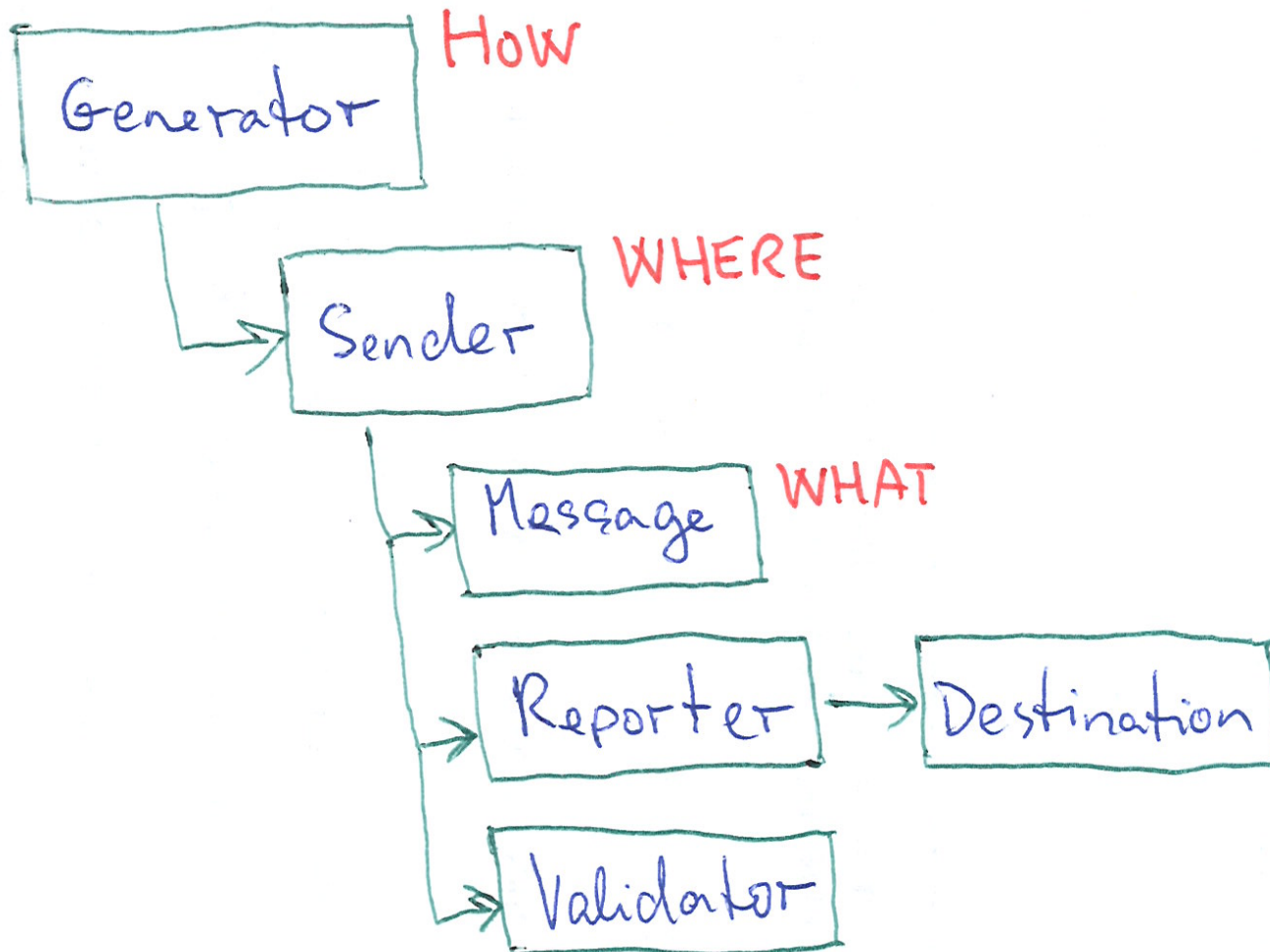


radargun



The Grinder

# PerfCake Architecture





# Konfigurace

```
<?xml version="1.0" encoding="utf-8"?>
<scenario xmlns="urn:perfcake:scenario:3.0">
  <properties>
    <property name="applicationName" name="testApp"/>
  </properties>
  <generator class="DefaultMessageGenerator" threads="${thread.count:10}">
    <run type="time" value="10000"/>
  </generator>
  <sender class="HttpSender">
    <property name="target" value="http://your-app.com/cool-app"/>
  </sender>
  <reporting>
    <reporter class="WarmUpReporter"/>
    <reporter class="ThroughputStatsReporter">
      <destination class="CsvDestination" enabled="true">
        <period type="time" value="3000"/>
        <property name="path" value="${perfcake.scenario}-stats.csv"/>
      </destination>
      <destination class="ConsoleDestination">
        <period type="percentage" value="5"/>
      </destination>
    </reporter>
  </reporting>
  <messages>
    <message uri="plain.txt" multiplicity="10">
      <header name="hello" value="hi!"/>
      <property name="myProp" value="value"/>
      <validatorRef id="text1"/>
    </message>
    <message content="Hello World!"/>
  </messages>
  <validation fastForward="false" enabled="false">
    <validator id="text1" class="RegExpValidator">I'm a fish!</validator>
  </validation>
</scenario>
```

# DSL je srozumitelnější

```
scenario "My Cool Scenario"
  applicationName "testApp"
  run 10.sec in ${thread.count:10}.threads
  generator "DefaultMessageGenerator"
  sender "HttpSender" target "http://your-app.com/cool-app"
  reporter "WarmUpReporter"
  reporter "ThrouputStatsReporter"
    destination "CsvDestination" every 3.sec path "${perfcake.scenario}-stats.csv" enabled
    destination "ConsoleDestination" every 5.percent
  message "plain.txt" 10.times headers ["hello":"hi!"] myProp "value" validateWith "text1"
  message content "Hello World!"
  validation fastForward false disabled
    validator "text1" "RegexValidator" pattern "I am a fish!"
end
```

# A teď je řada na vás!

## #23 Add check for code-style, code coverage and license

university project

<https://github.com/PerfCake/PerfCake/issues/23>

Missing just the code-style plugin.

## #50 Increase code coverage

university project

<https://github.com/PerfCake/PerfCake/issues/50>

Tests, tests, tests. Nobody is perfect, neither are we. We need more tests.

## #140 Develop Validators integration test








university project

<https://github.com/PerfCake/PerfCake/issues/140>

We have no clue whether the validation works as expected. Anyone to prove?

# Další výzvy

## Open Source Contest 2014

-  **Implement script sender** enhancement open source contest  0  
#139 opened 6 days ago by marvec Release 4.0
-  **Add support for classpath definition to the Groovy sender** enhancement open source contest  0  
#138 opened 6 days ago by marvec Release 4.0
-  **Implement InfluxDB destination** enhancement open source contest  0  
#137 opened 7 days ago by marvec Release 4.0
-  **Implement ChannelSender** cake enhancement open source contest  0  
#100 opened on 11 Mar by pmacik

# Co nás čeká

- podpora v IDE
- spouštění v clusteru
- vyleštění dokumentace
- lepší PR
- integrace s TestNG a junit
- vizualizace výsledků



# Links

- [www.perfcake.org](http://www.perfcake.org)
- [github.com/PerfCake/PerfCake/issues](https://github.com/PerfCake/PerfCake/issues)
- [site.perfcake.org](http://site.perfcake.org)
- [javadoc.perfcake.org](http://javadoc.perfcake.org)
- [perfcake.ci.cloudbees.com](https://perfcake.ci.cloudbees.com)
- [twitter.com/perfcaketool](https://twitter.com/perfcaketool)
- [perfcake.blogspot.com](http://perfcake.blogspot.com)



?