

# JavaEE平台技术

## Spring应用的测试

邱明 博士

厦门大学信息学院

mingqiu@xmu.edu.cn

# 1. 测试的方法

- 白盒测试

- 根据软件的逻辑结构，测试软件产品的内部结构和处理过程，证明软件的代码是正确的。
- 静态分析—不执行所测试的程序，扫描所测试程序的代码，对程序的数据流和控制流进行分析，给出测试报告
  - 自动代码审查--阿里巴巴规范插件
  - 人工代码审查
- 动态分析
  - 通过运行代码检查运行结果和预期结果的差异，并分析测试的代码覆盖率。

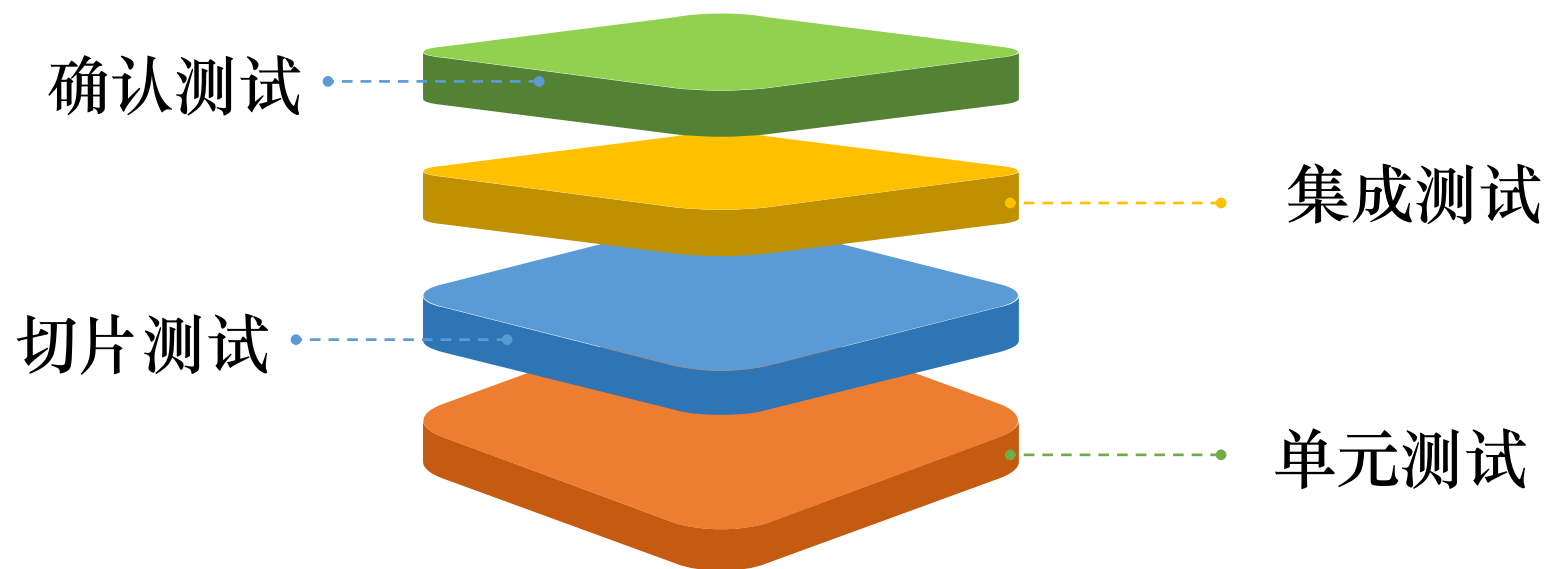
# 1. 测试的方法

- 黑盒测试

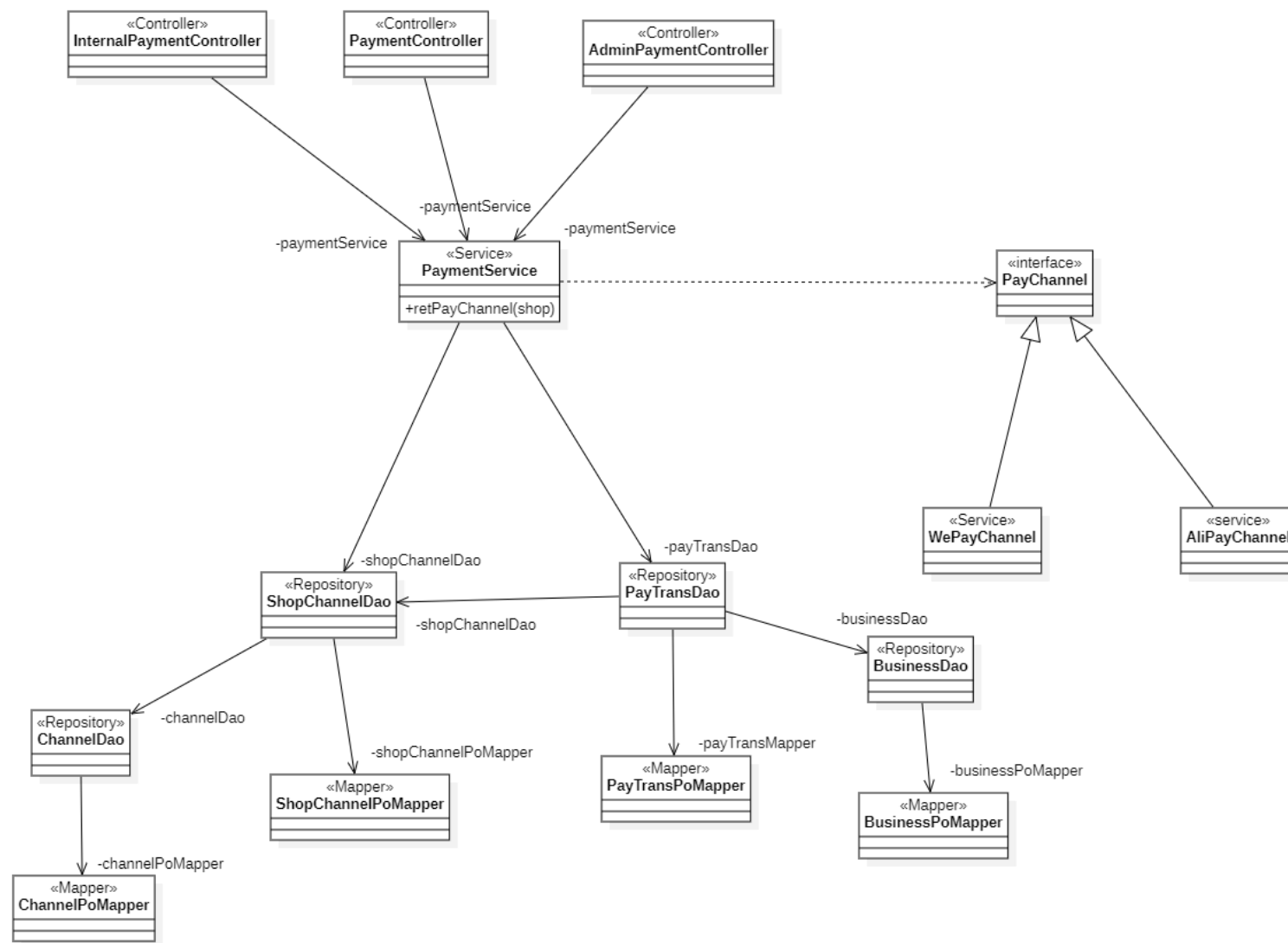
- 根据需求规格对软件进行的测试，以用户的角度，通过各种输入和观察软件的各种输出结果来发现软件存在的缺陷。
  - 等价类划分
  - 边界值分析
  - 错误推断法
  - 因果图

# 1. 测试的方法

- 不同层次的测试



## 2.Spring应用测试对象



## 2.Spring应用测试对象

- 单元测试
  - Junit 5, Mockito, AssertJ
- 切片测试
  - MockBean, MockMvc
- 集成测试

# 3.测试的方法

- 日志---logback

```
<logger name="cn.edu.xmu.oomall.payment" level = "INFO"/>
<logger name="cn.edu.xmu.oomall.payment.mapper" level = "INFO"/>
<logger name="cn.edu.xmu.javaee.core.util.Common" level = "DEBUG"/>
<root level="INFO">
  <appender-ref ref="STDOUT" />
  <appender-ref ref="FILE" />
</root>
```

# 3.测试的方法

- 控制测试的范围

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>
    <includes>
      <include>**/*Test.java</include>
    </includes>
    <skip>>false</skip>
  </configuration>
</plugin>
```



# 4.测试的结果

- jacoco

## payment

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">cn.edu.xmu.oomall.payment.mapper.generator.po</a>	<div><div></div></div>	4%	<div><div></div></div>	14%	1,730	1,870	3,355	3,549	1,651	1,786	30	35
<a href="#">cn.edu.xmu.oomall.payment.mapper.generator</a>	<div><div></div></div>	3%	<div><div></div></div>	5%	473	484	1,306	1,369	44	47	6	7
<a href="#">cn.edu.xmu.oomall.payment.dao.bo</a>	<div><div></div></div>	18%	<div><div></div></div>	2%	262	337	40	125	60	132	3	9
<a href="#">cn.edu.xmu.javaee.core.util</a>	<div><div></div></div>	41%	<div><div></div></div>	23%	99	126	260	405	46	68	5	9
<a href="#">cn.edu.xmu.oomall.payment.service.openfeign.param</a>	<div><div></div></div>	11%	<div><div></div></div>	0%	87	104	11	20	27	44	0	4
<a href="#">cn.edu.xmu.javaee.core.model</a>	<div><div></div></div>	47%	<div><div></div></div>	1%	76	116	29	96	24	63	1	6
<a href="#">cn.edu.xmu.oomall.payment.controller.vo</a>	<div><div></div></div>	19%	<div><div></div></div>	0%	61	88	0	16	7	34	0	2
<a href="#">cn.edu.xmu.javaee.core.aop</a>	<div><div></div></div>	59%	<div><div></div></div>	35%	36	51	55	148	4	15	1	4
<a href="#">cn.edu.xmu.oomall.payment.dao</a>	<div><div></div></div>	82%	<div><div></div></div>	75%	3	15	9	65	1	9	0	4
<a href="#">cn.edu.xmu.javaee.core.exception</a>		38%		n/a	2	3	4	7	2	3	0	1
<a href="#">cn.edu.xmu.oomall.payment.service</a>		91%		50%	2	5	2	29	0	3	0	1
<a href="#">cn.edu.xmu.oomall.payment</a>		37%		n/a	1	2	2	3	1	2	0	1
<a href="#">cn.edu.xmu.oomall.payment.service.openfeign</a>		60%		n/a	1	2	1	2	1	2	0	1
<a href="#">cn.edu.xmu.oomall.payment.controller</a>	<div><div></div></div>	100%		100%	0	5	0	23	0	4	0	2
<a href="#">cn.edu.xmu.javaee.core.config</a>	<div><div></div></div>	100%		n/a	0	4	0	24	0	4	0	2
<a href="#">cn.edu.xmu.oomall.payment.service.channel</a>		100%		n/a	0	2	0	23	0	2	0	1
<a href="#">cn.edu.xmu.oomall.payment.config</a>		100%		n/a	0	2	0	6	0	2	0	1
Total	23,212 of 26,849	13%	1,837 of 1,987	7%	2,833	3,216	5,074	5,910	1,868	2,220	46	90

# 5. 性能测试

- 性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。
  - 负载测试
    - 通过在被测系统上不断加压，直到性能指标达到极限状态。确定各种负载下系统的性能，目标是测试负载逐渐增加时各项指标的变化。
  - 并发测试
    - 通过模拟用户并发，测试多用户同时访问同一个应用时是否存在死锁或者其他性能问题。

# 5. 性能测试

- Jmeter
  - 线程组 (Thread group)
    - 设置多个用户 (Number of Thread) 和启动时间 (Ramp-Up Period) 来加载
  - 取样器 (Sampler)
    - 性能测试中向服务器发送Request, 记录Response
  - 逻辑控制器 (Logic Controller)
    - 控制 sampler 节点发送请求的逻辑顺序的控制器
  - 配置元件 (config element)
    - 用于提供对 静态数据配置的支持, 如读取csv文件
  - 断言 (Assertions)
    - Response Assertion与JSONPathAssertion
  - 监听器 (Listener)
    - 对测试结果数据进行处理和可视化展示

# 5. 性能测试

- 测试指标

- 响应时间 (Response Time) :

- 从客户端发出请求开始，到接收到服务器的响应为止。
    - 响应时间=网络传输的时间+服务器的处理时间

# 5. 性能测试

Statistics

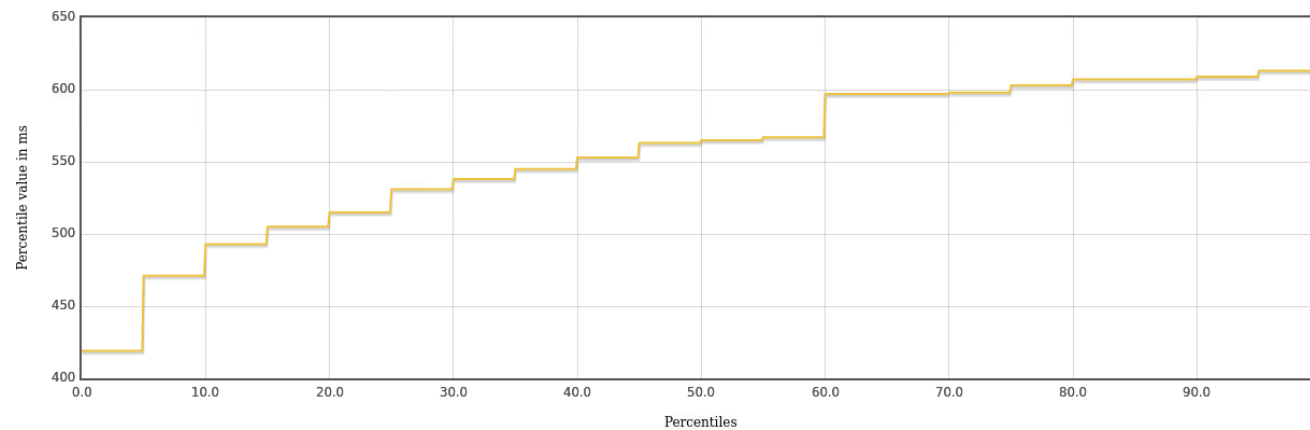
Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label ^	#Samples ⇅	FAIL ⇅	Error % ⇅	Average ⇅	Min ⇅	Max ⇅	Median ⇅	90th pct ⇅	95th pct ⇅	99th pct ⇅	Transactions/s ⇅	Received ⇅	Sent ⇅
Total	20	0	0.00%	554.80	419	613	564.00	608.80	612.80	613.00	32.41	30.89	0.00
HTTP Request	20	0	0.00%	554.80	419	613	564.00	608.80	612.80	613.00	32.41	30.89	0.00

Statistics

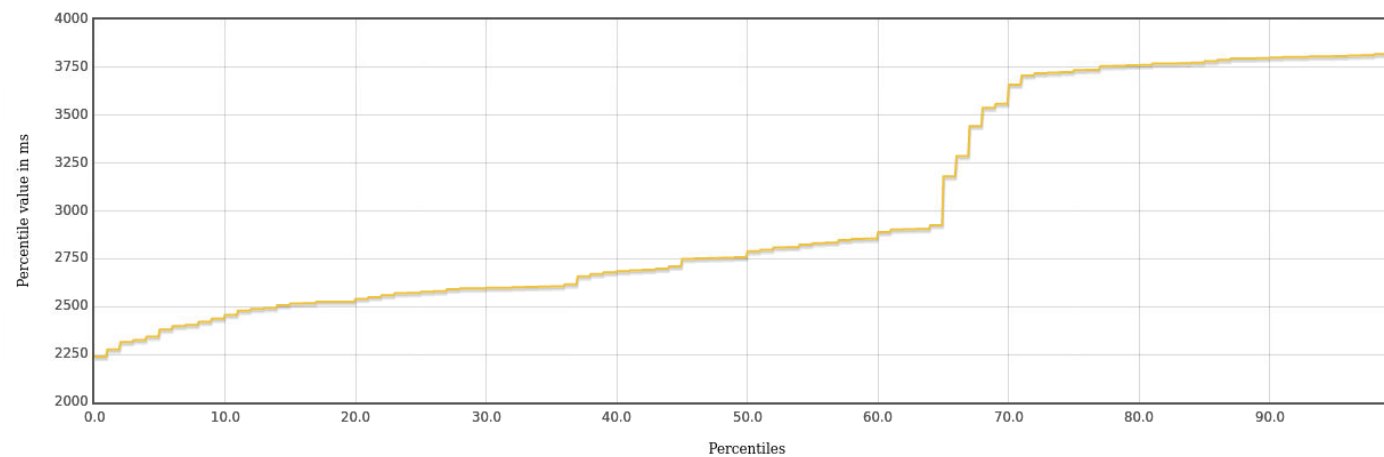
Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label ^	#Samples ⇅	FAIL ⇅	Error % ⇅	Average ⇅	Min ⇅	Max ⇅	Median ⇅	90th pct ⇅	95th pct ⇅	99th pct ⇅	Transactions/s ⇅	Received ⇅	Sent ⇅
Total	100	1	1.00%	3007.74	2239	3824	2773.50	3800.80	3808.90	3823.95	26.04	24.70	0.00
HTTP Request	100	1	1.00%	3007.74	2239	3824	2773.50	3800.80	3808.90	3823.95	26.04	24.70	0.00

# 5. 性能测试

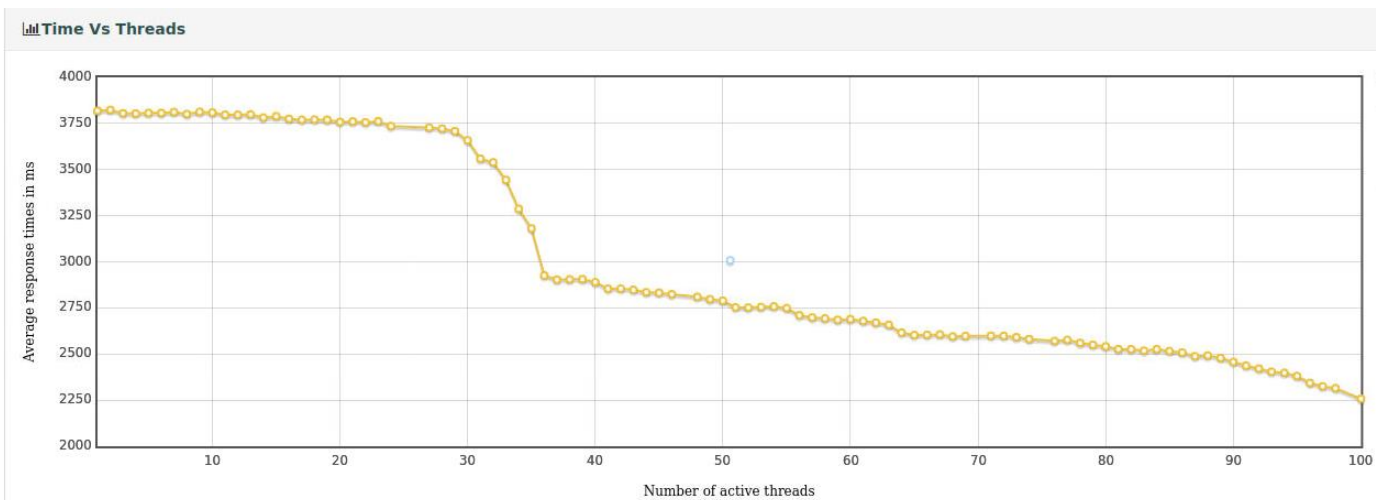
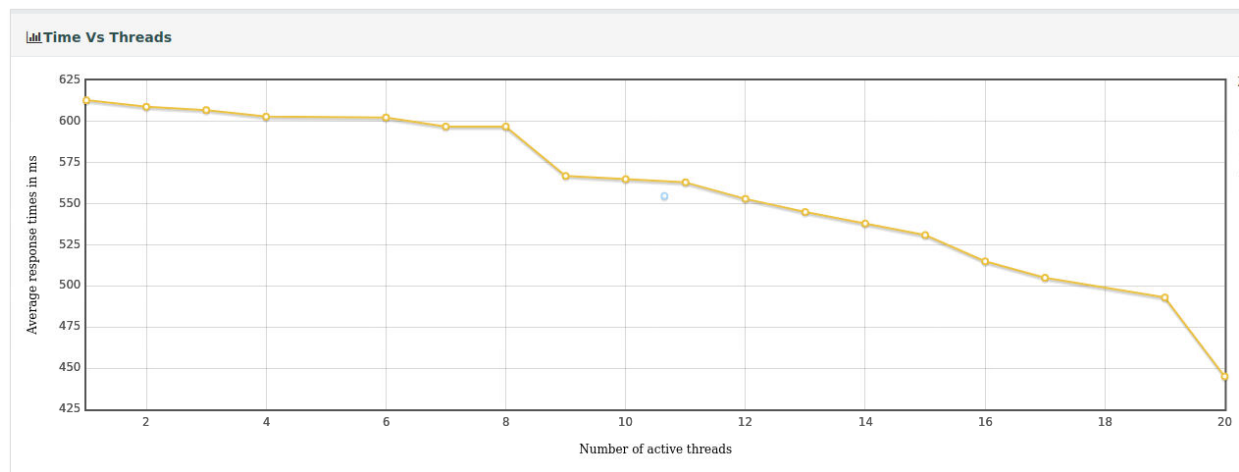
Response Time Percentiles



Response Time Percentiles



# 5. 性能测试



# 5. 性能测试

