



携程技术中心

携程技术沙龙

深度学习在移动端的实践和自动化测试

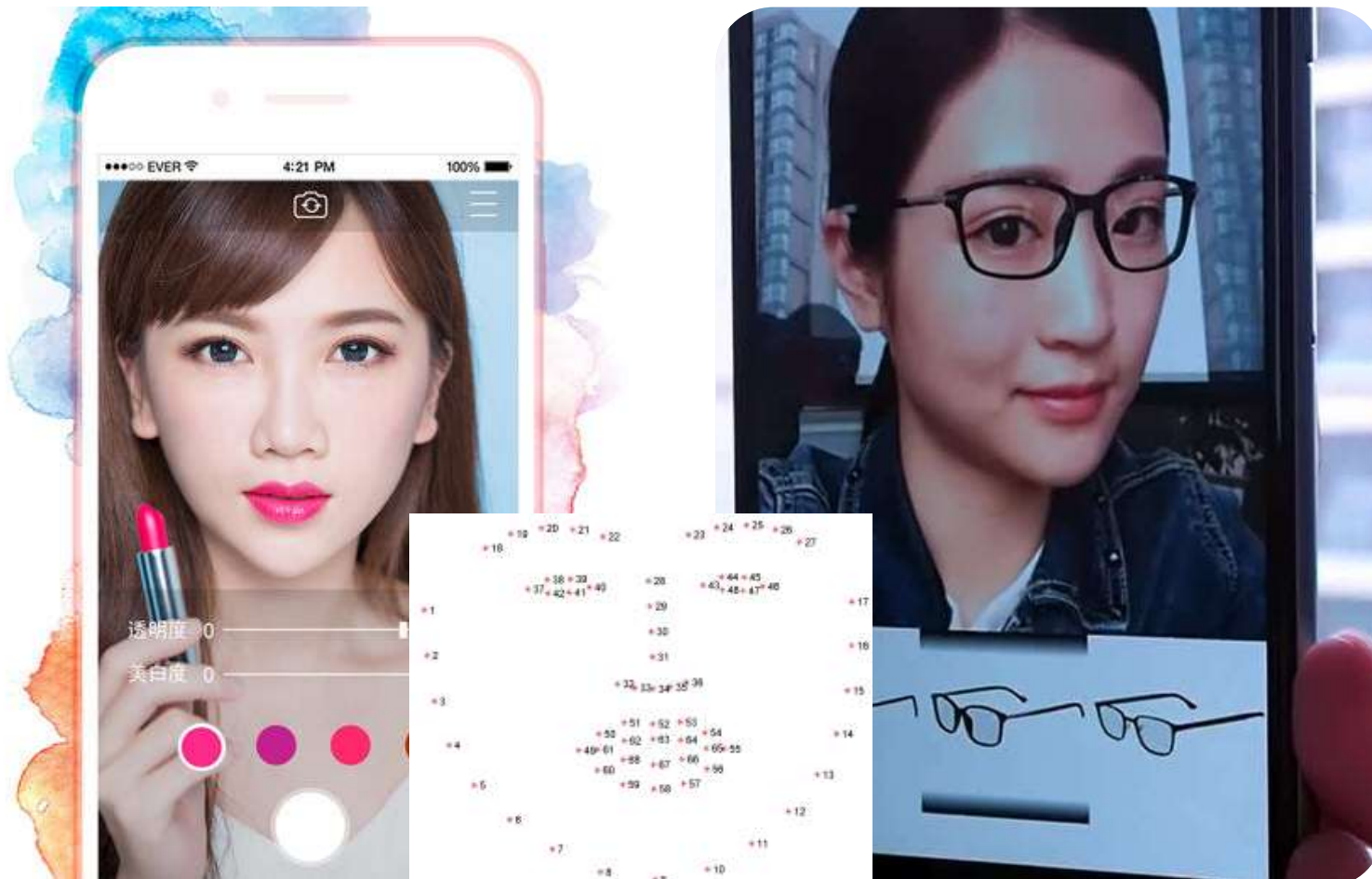
唯品会研究院 张凯通



张凯通

- 唯品会研究院资深开发工程师，负责深度学习在移动端的应用，以及自动化测试等工作。曾参与“唯品会”、“学霸君”、“imo云办公室”等多款千万级用户的手机应用开发和架构设计。

我们在做的事



目录

CONTENTS

- 1 技术选型
- 2 移动端部署
- 3 性能优化
- 4 性能测试
- 5 自动化

人脸检测

传统的人脸检测

效率

精度

工程化

基于深度学习的人脸检测

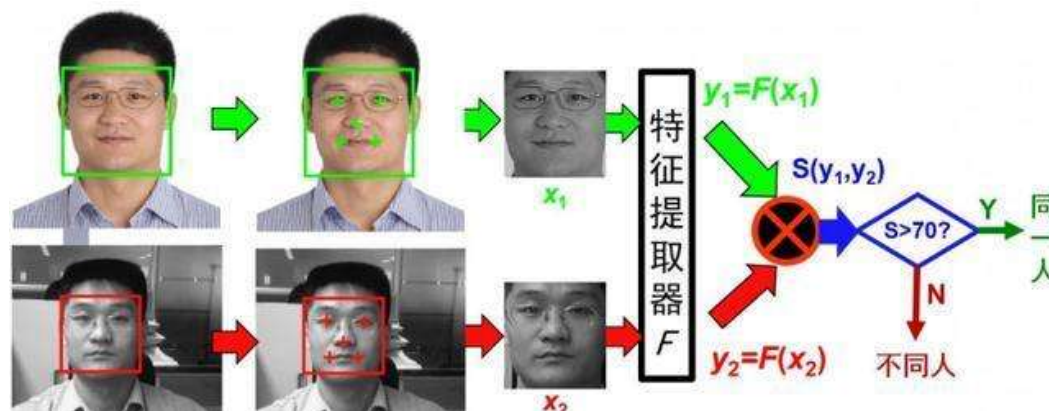
识别效率高

当网络的构造设计完成之后，深度神经网络能够自发去分类、提取需要的特征，这就大大节省了人工获取特点以及设计分类装置的不必要程序。深度神经网络的学习方式使得识别器不会进入到常见的多层次网络，避免局部选择。并且深度神经网络在训练时不需要监督，同时也不需要标签，节省了标本取样的时间。在表达复杂函数时，深度神经网络由于不需要过多的神经元，因此，识别效率得到了极大提升。

基于深度学习的人脸检测

识别效果好

深度模型通过逐层抽象的方式获得人脸的高层特征，具备更强的表达能力，能够充分发挥大数据的优势，准确率比传统方式有着质的飞跃。当调整神经网络权值时，权值就会自动接近最优点，最终获取一个完美的数据。



Tensorflow

Why Tensorflow?

Why Tensorflow?

- 可用性

兼容性

API 稳定

开发调试便捷

更多扩展支持

Why Tensorflow?

- 灵活性

多种类型设备支持

CPU GPU混合运算支持

Why Tensorflow?

- 效率

快速提升的效率

Why Tensorflow?

- 支持

google的开发资源

共享的模型

强大的社区

Why Tensorflow?

- 功能强大的可视化组件TensorBoard
- 相比其他的深度学习框架，文档最全，资源最多
- 部署容易
- 性能、SDK大小、模型大小都满足移动端App的需求

Why Tensorflow?

	Tensorflow	Caffe	Keras	CNTK	mxnet
Github star	96k	23k	28k	14k	13k
Github fork	61k	14k	10k	3k	5k
Owner	Google	BVLC	fchollet	Microsoft	DMLC
Language	Python/C++/...	C++/Python	Python	C++	Python/C++/...
iOS Support	YES	YES	YES	NO	YES
Android Support	YES	YES	YES	NO	YES
Update	Very High	Very High	High	High	Middle

移动端部署

移动端部署

- Tensorflow 的部署

移动端部署

- Tensorflow 的部署
- 从Tensorflow到Tensorflow lite

移动端部署

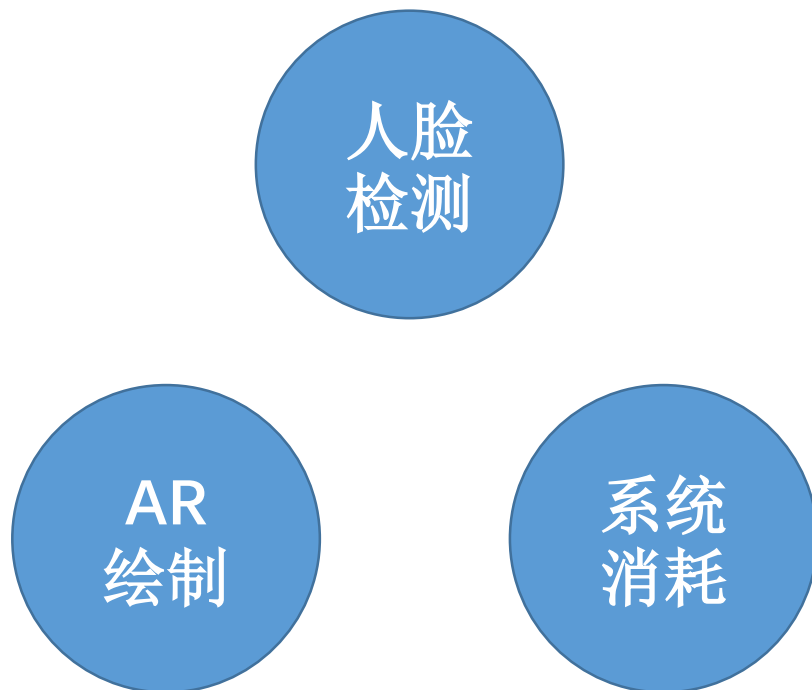
- Tensorflow 的部署
- 从Tensorflow到Tensorflow lite
- 可替换组件化的部署

移动端部署

- Tensorflow 的部署
- 从Tensorflow到Tensorflow lite
- 可替换组件化的部署
- 模型/SDK大小的优化

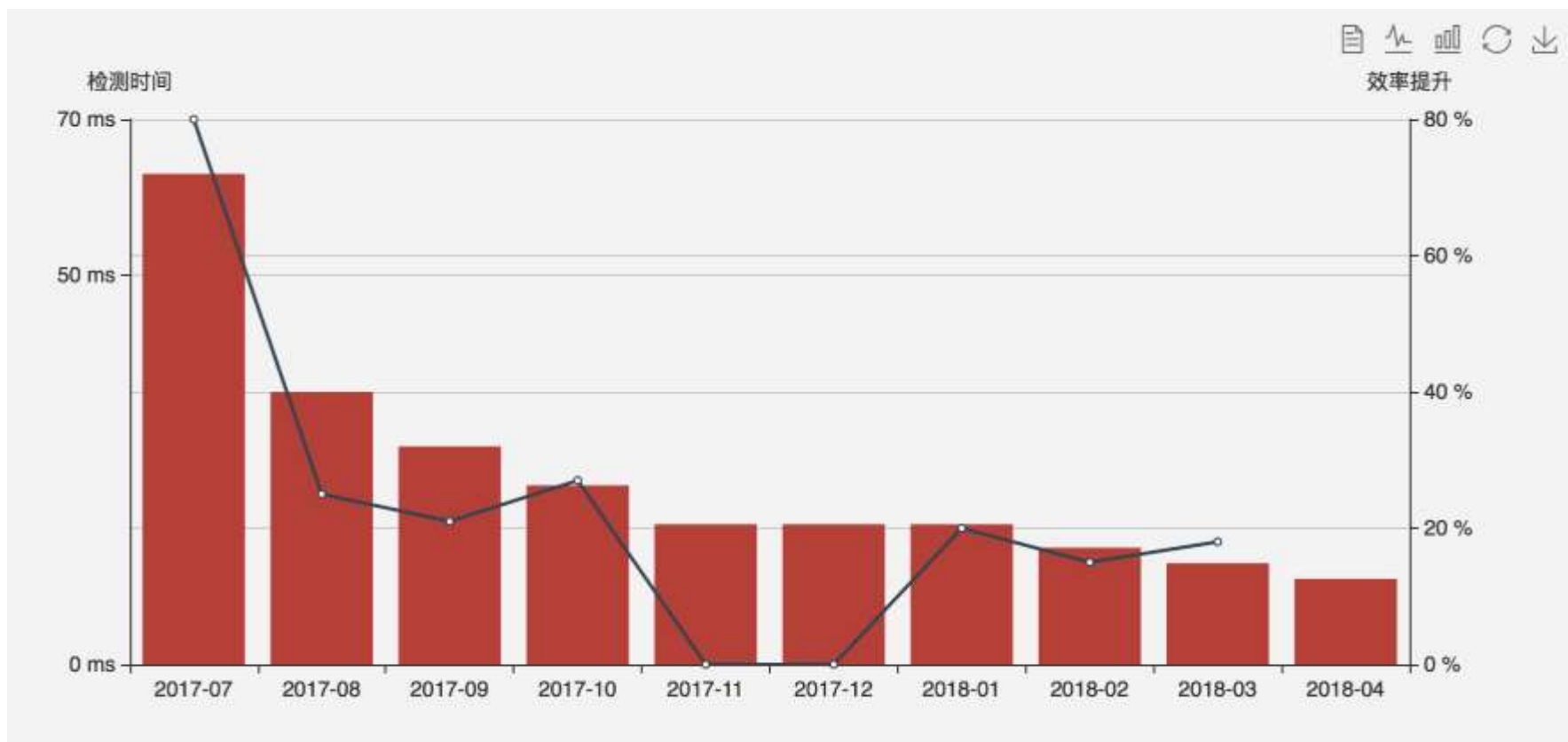
性能优化

性能分析



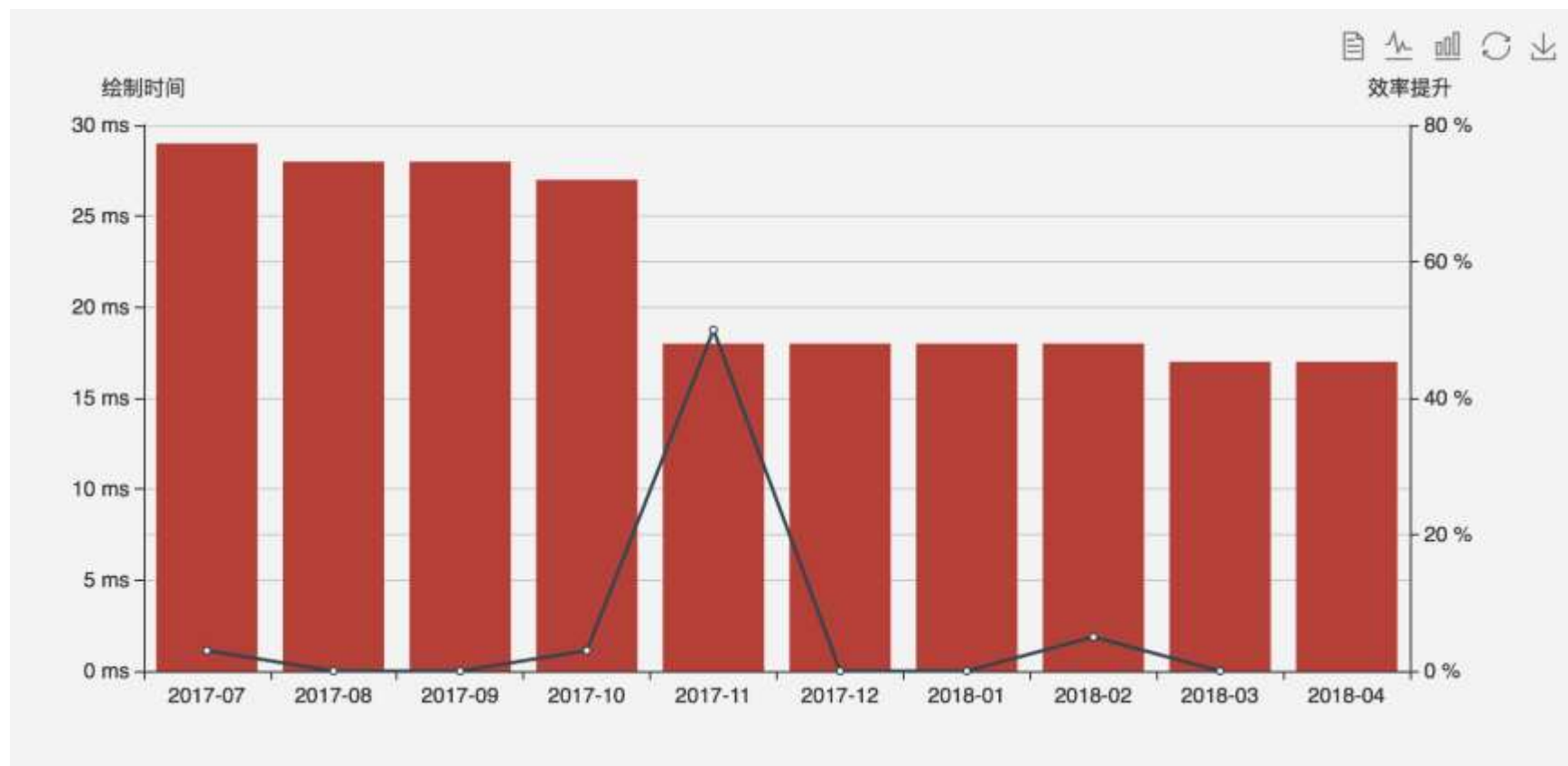
性能分析

- 人脸检测



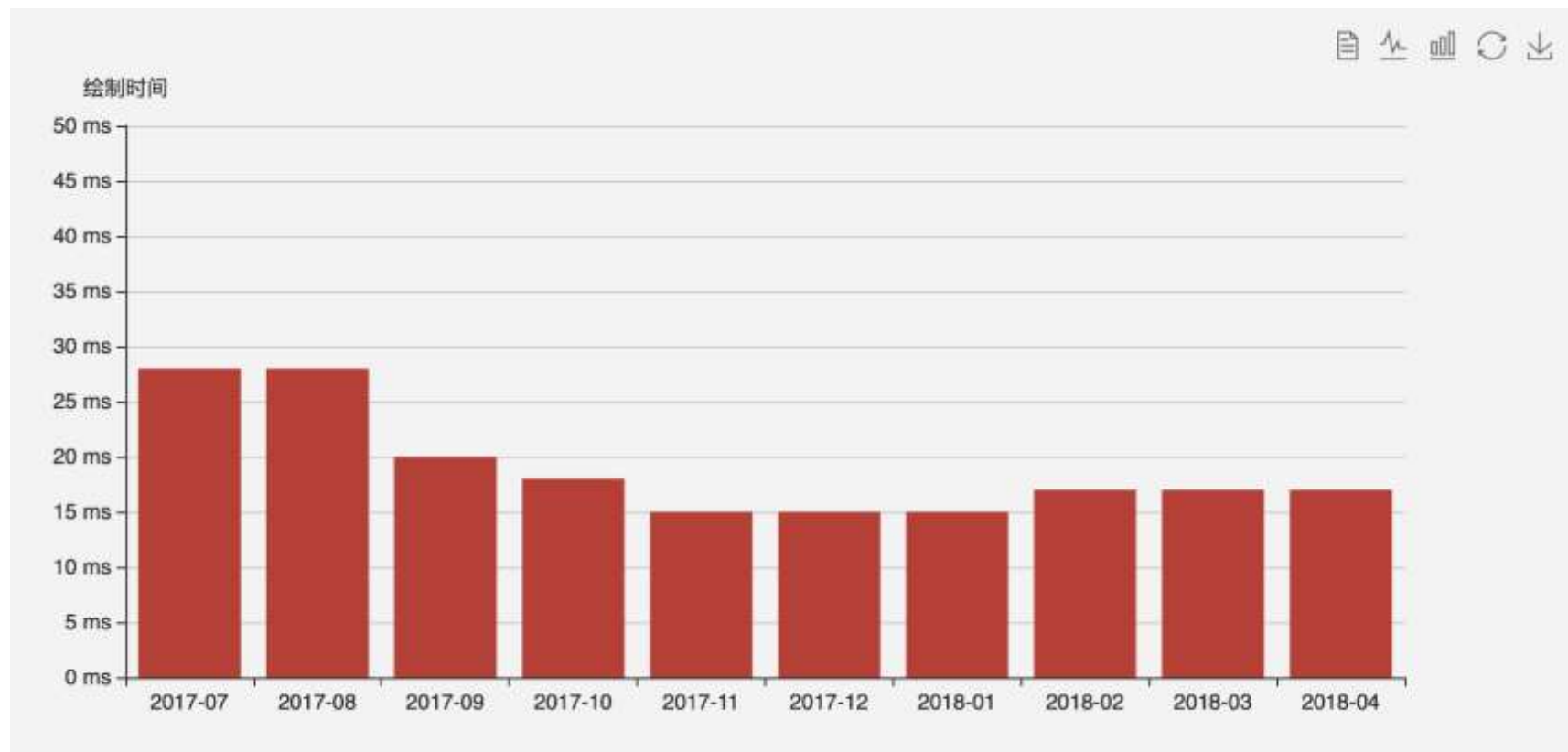
性能分析

- AR绘制



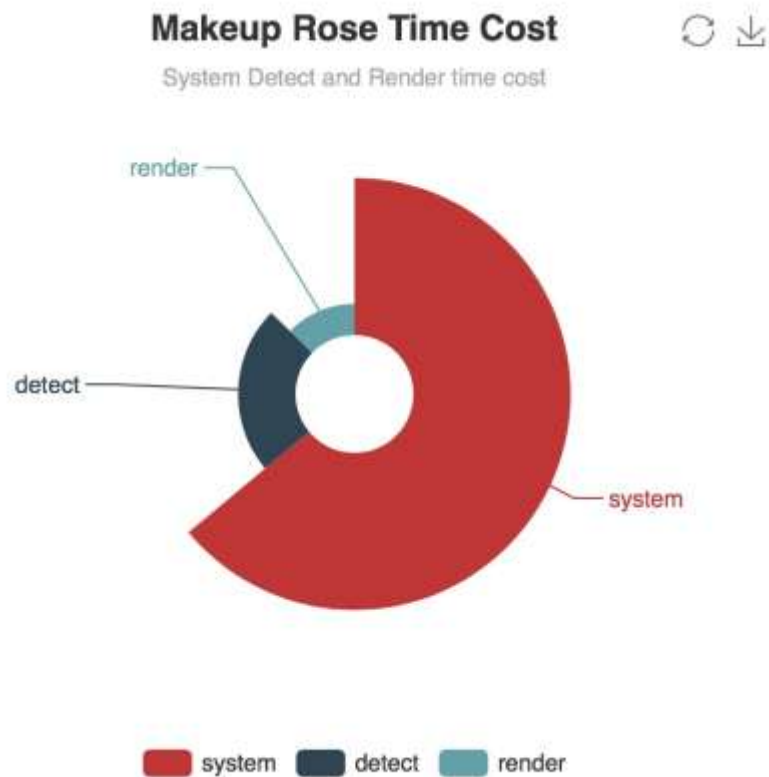
性能分析

- 系统占用



性能分析

- 总体时间消耗



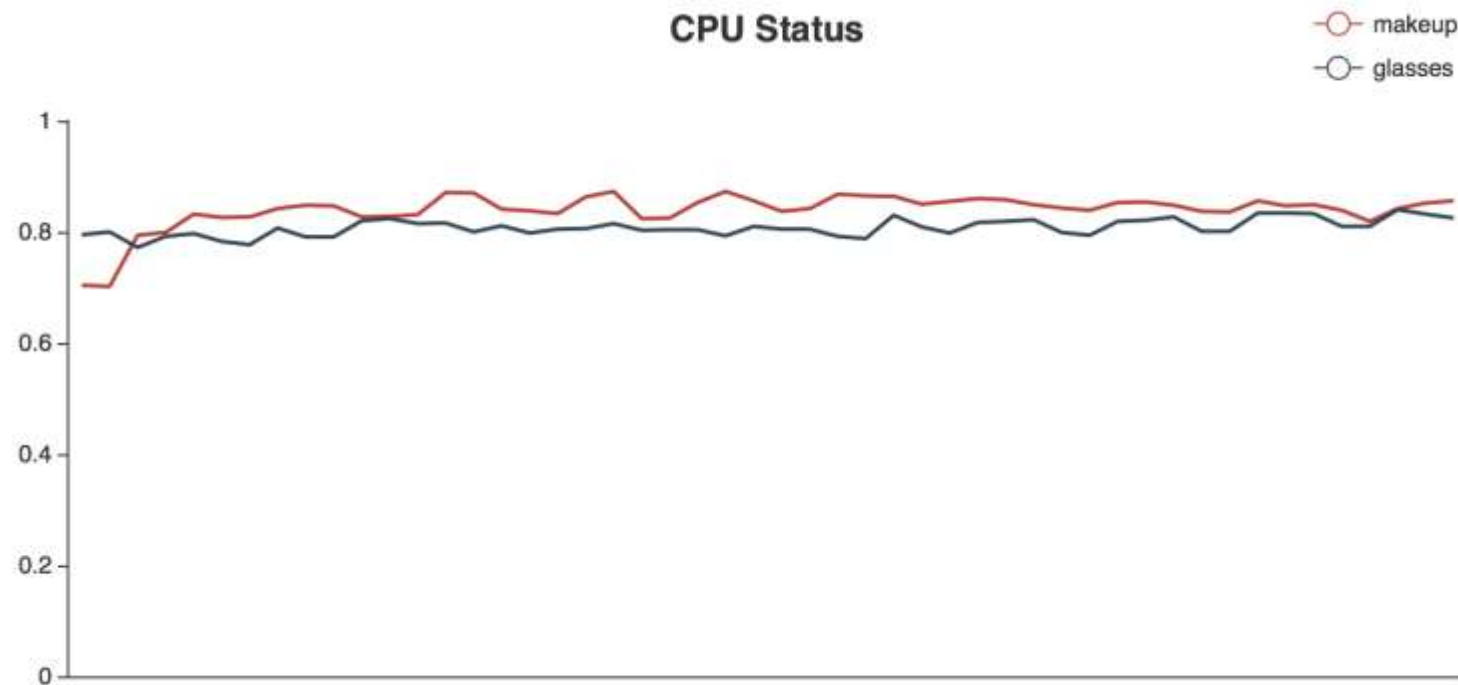
测试

测试

- 功能测试

测试

- 性能测试

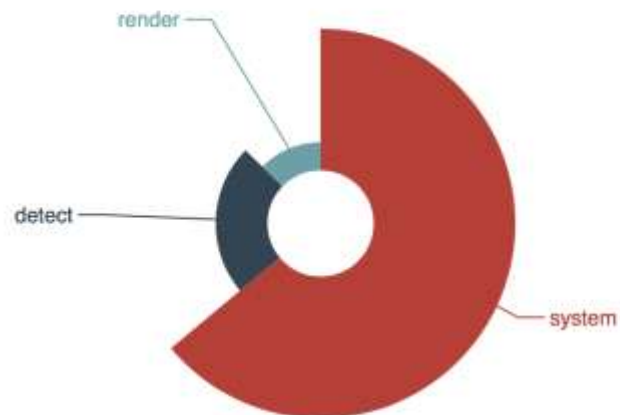


测试

- 性能测试

Makeup Rose Time Cost

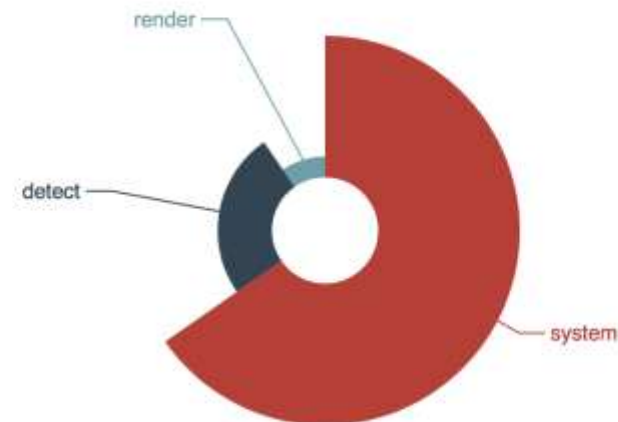
System Detect and Render time cost



system detect render

Glasses Rose Time Cost

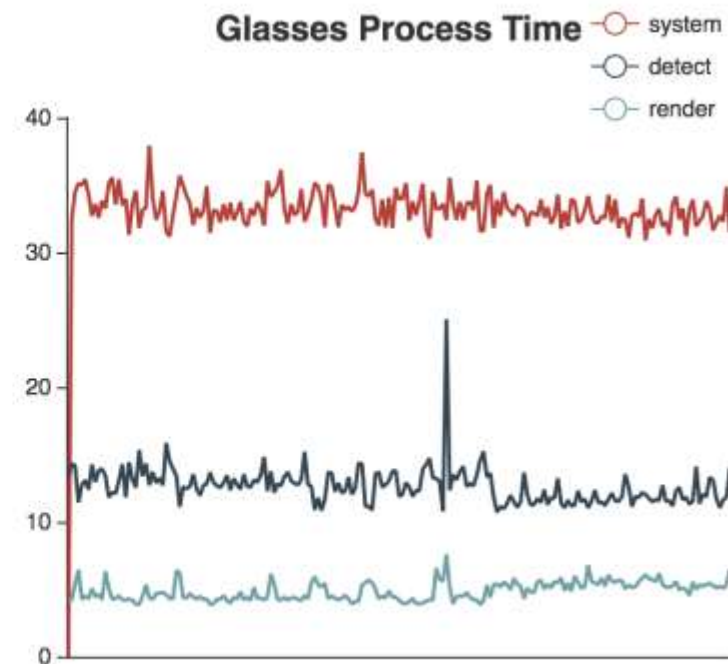
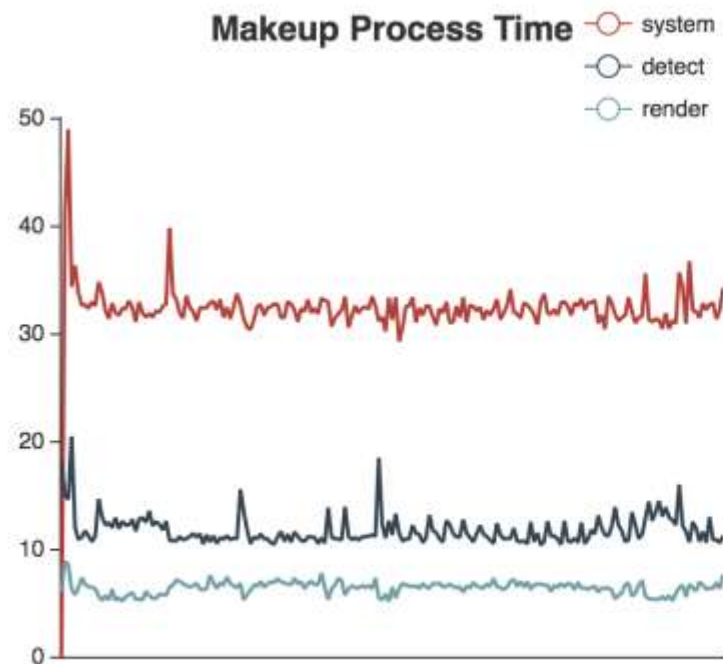
System Detect and Render time cost



system detect render

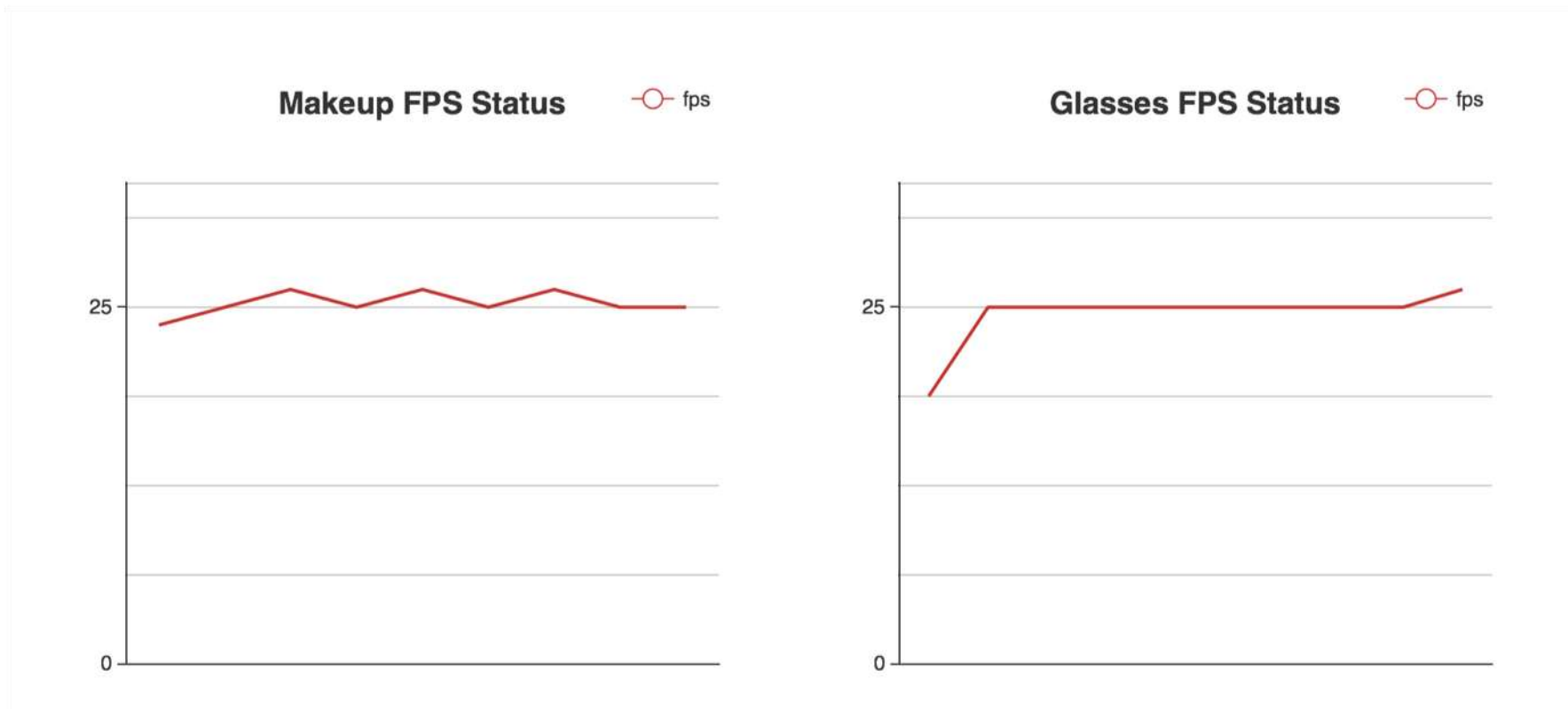
测试

- 性能测试



测试

- 性能测试



测试

- 性能测试

SDK size analysis

	Origin			Compressed			Compressed Rate		
	executable	resource	total	executable	resource	total	executable	resource	total
unintegrate	195.0KB	462.4KB	657.5KB	48.2KB	340.9KB	389.1KB	75.3%	26.3%	40.8%
integrated	7.5MB	3.3MB	10.8MB	2.9MB	1.5MB	4.4MB	61.1%	54.4%	59.1%
delta	7.3MB	2.8MB	10.1MB	2.9MB	1.2MB	4.0MB	60.8%	58.9%	60.2%

自动化测试

为什么要做自动化测试？

为什么要做自动化测试？

- 避免人工测试产生的错误

为什么要做自动化测试？

- 避免人工测试产生的错误
- 记录每一次测试的原始数据，确保测试结果可以重现，以便追踪问题

为什么要做自动化测试？

- 避免人工测试产生的错误
- 记录每一次测试的原始数据，确保测试结果可以重现，以便追踪问题
- 提供更多的原始数据进行分析为下一步优化提供更多参考

为什么要做自动化测试？

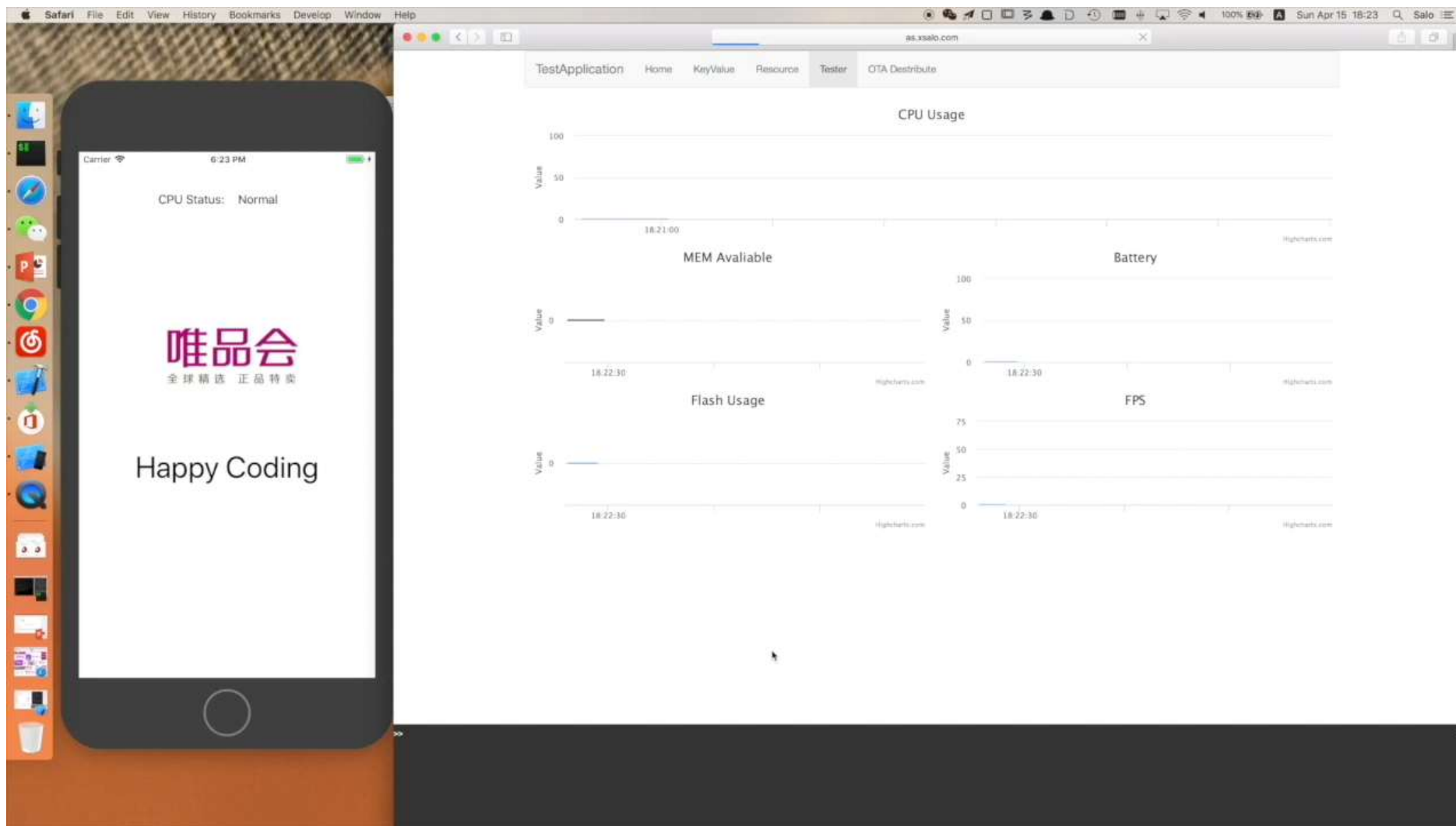
- 避免人工测试产生的错误
- 记录每一次测试的原始数据，确保测试结果可以重现，以便追踪问题
- 提供更多的原始数据进行分析为下一步优化提供更多参考
- 效率提升

测试工具

- Farseer

测试工具

- Farseer
查看



测试工具

- Farseer
调用

The screenshot displays a Mac desktop environment. On the left, a mobile application is simulated on a virtual device, showing a screen with the text 'CPU Status: Normal', the '唯品会' (Vipshop) logo, and 'Happy Coding'. The desktop background features a grid of system monitoring graphs including 'CPU Usage', 'MEM Available', and 'Battery'. A terminal window in the foreground shows the following commands and their outputs:

```

>> exec UIApplication.sharedApplication
<UIApplication: 0x7fe749f00210>
>> exec UIApplication.sharedApplication.keyWindow
<UIWindow: 0x7fe749a8b4b0; frame = (0 0; 414 736); autoresize = W+H; gestureRecognizers = <NSArray: 0x60c000251850>; layer = <UIWindowLayer: 0x7fe749a24cf0>
>> exec UIApplication.sharedApplication.keyWindow.rootViewController
<ViewController: 0x7fe749a24cf0>
>>
  
```

At the bottom right, a small inset image shows a person's hands typing on a laptop keyboard.

测试工具

- Farseer

监测

The screenshot displays the Farseer tool interface. On the left, a simulated mobile device shows the Weipinhui (唯品会) app with the text 'Happy Coding'. On the right, a browser window shows system monitoring graphs for CPU Usage, MEM Available, and Battery. Below the graphs, a terminal window shows the following commands and their outputs:

```
<ViewController: 0x7fe749a24cf0>  
>> exec UIApplication.sharedApplication.keyWindow.rootViewController.logoView  
<UIImageView: 0x7fe747417d90; frame = (87 220; 240 128); autoresize = RM+BM; userInter  
>> exec UIApplication.sharedApplication.keyWindow.rootViewController.logoView.rotate  
ok  
>> exec UIApplication.sharedApplication.keyWindow.rootViewController.logoView.pause  
ok  
>>
```

At the bottom of the terminal, there is a small inset image showing a person's hands typing on a laptop keyboard.

测试工具

- Farseer

修改

The screenshot shows a mobile application interface on the left and a web-based testing tool on the right. The mobile app displays 'CPU Status: Normal', the '唯品会' (Vipshop) logo, and the text 'Happy Coding'. The web tool shows a 'CPU Usage' graph with a spike at 18:22:30. Below the graphs is a terminal window with the following code:

```

ok
>> exec UIApplication.sharedApplication.keyWindow.rootViewController.logoView.cpuTest
object '<UIImageView: 0x7fe747417d90; frame = (87 220; 240 128); autoresize = RM+BM; us
>> exec UIApplication.sharedApplication.keyWindow.rootViewController.cpuTest
ok
>> exec UIApplication.sharedApplication.keyWindow.rootViewController.cpuTestFinish
ok
>>
    
```

At the bottom right, there is a small inset image showing a person's hands typing on a laptop keyboard.

测试工具

- Farseer
- Template Loader 动态参数更新

测试工具

- Farseer
- Template Loader 动态参数更新
- Macarons

测试工具-Macarons

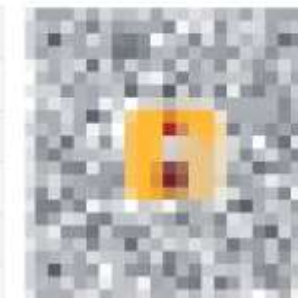
- 自动化流程测试

测试工具-Macarons

- 自动化流程测试
- 性能参数分析

VSARSDK-VSRISDK_SPEC 93 - Auto Test Report

begin date	platform	bundle id
2018-03-28 23:18:23	IOS	[REDACTED]
version	sdk size	camera fps MK / GL / SP
1.0 (1)	4.0MB	11.1 / 11.3 / 24.0
avg render time (ms) MK / GL	avg detect time (ms) MK / GL	avg cpu (%) MK / GL
0 / 0	57 / 55	119.0 / 120.5



Download QR Code

Git info

Container	
name	ARSDK
URL	[REDACTED]
hash	2ffe3253b3ba2553fb7167aadc3b1b5e122119e3
commit date	2018-03-28 14:57:53
committer	[REDACTED]

测试工具-Macarons

- 自动化流程测试
- 性能参数分析
- 对比历史

SDK size analysis

	Origin			Compressed			Compressed Rate		
	executable	resource	total	executable	resource	total	executable	resource	total
unintegrate	195.0KB [→]	462.4KB [→]	657.5KB [→]	48.2KB [→]	340.8KB [→]	389.0KB [→]	75.3% [→]	26.3% [→]	40.8% [→]
integrated	7.5MB [→]	3.3MB [→]	10.8MB [→]	2.9MB [→]	1.5MB [→]	4.4MB [→]	61.1% [→]	54.4% [→]	59.1% [→]
delta	7.3MB [→]	2.8MB [→]	10.1MB [→]	2.9MB [→]	1.2MB [→]	4.0MB [→]	60.8% [→]	58.9% [→]	60.2% [→]

Performce Record in VSRItest - iPhone 6

device name	device id	device model	device os	test at	test time	specified camera fps
VSRItest	[REDACTED]	iPhone 6	11.2.6	2018-03-28 23:20:14	27s	24.0

Makeup

avg render time (ms)	avg detect time (ms)	avg cpu usage (%)	camera fps
0 [100% ↘]	57 [383% ↗]	119.0 [42% ↗]	11.1 [44% ↘]

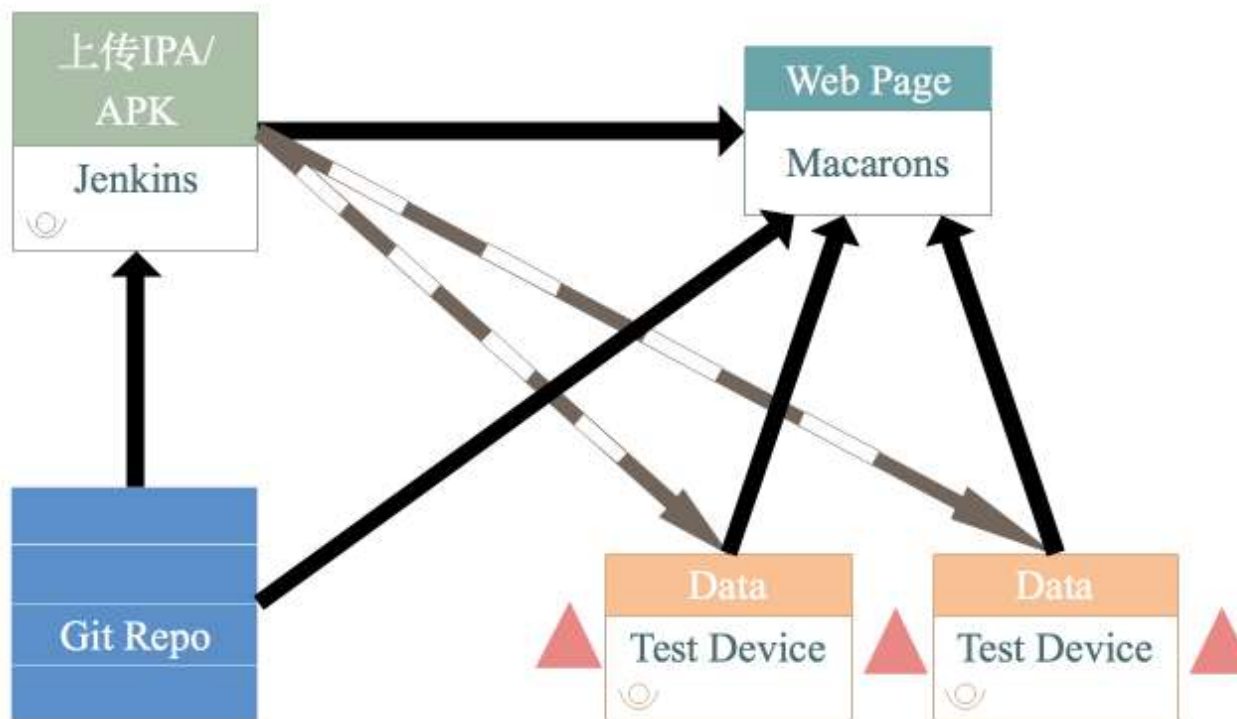
3D Glasses

avg render time (ms)	avg detect time (ms)	avg cpu usage (%)	camera fps
0 [100% ↘]	55 [337% ↗]	120.5 [49% ↗]	11.3 [43% ↘]

测试工具-Macarons

- 自动化流程测试
- 性能参数分析
- 对比历史
- 异常测试结论提醒

测试工具-Macarons



我们在做的一切都是为了让开发这件事变得更高效率、更有乐趣

全球精选，正品特卖 — 唯品会



携程技术中心

THANK YOU!

Q&A

本PPT来自携程技术沙龙
更多信息可搜索关注“携程技术中心”微信公号