移动测试智能化
助力高质量App生态体系

技术专家　字白

阿里云移动云

1 高质量App认证

2 测试现状与痛点

3 智能化实践： Ripper

4 测试效果对比

5 未来智能化方向

高质量App认证合作

阿里云 + 泰尔终端实验室

1 建立App认证标准

2 成立联合实验室

3 测试标准落地

测试现状与痛点

测试现状与痛点

人工测试

- 重复性工作多，测试速度慢
- 需要关注的点多，QA成本高

自动化功能测试

- 变化快，需要经常性修改，维护成本高
- 失败原因往往不是crash，而是脚本或是设置问题

主流测试方法

Monkey

- 动作不可控，覆盖程度低
- 测试耗时长
- 开发定位问题困难

遍历

- 提高检测覆盖度
- 缺少具体的业务场景
- 缺少问题定位

智能化实践：Ripper

丰富的动作支持

双击
长按
连击
手势操作
中文输入
系统事件

精确控制

决策驱动
规则丰富

场景理解

场景判断
连续动作

高测试效率

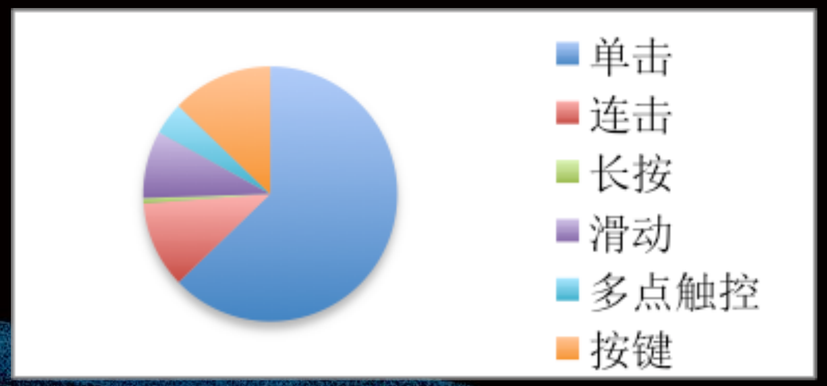
2动作/s

操作便于回溯

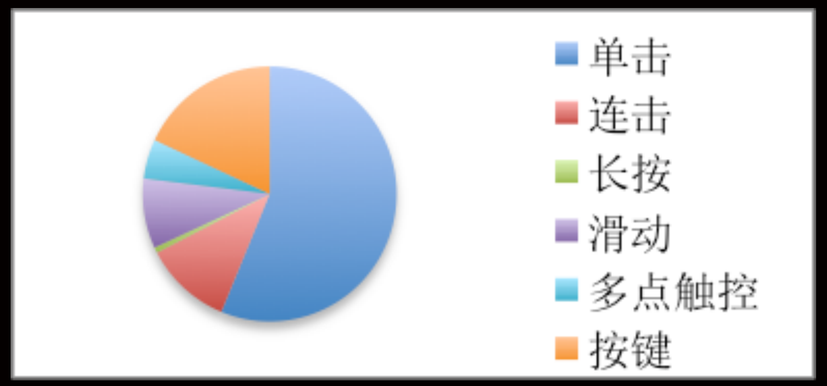
动作记录
状态回溯

动作类型分布均衡

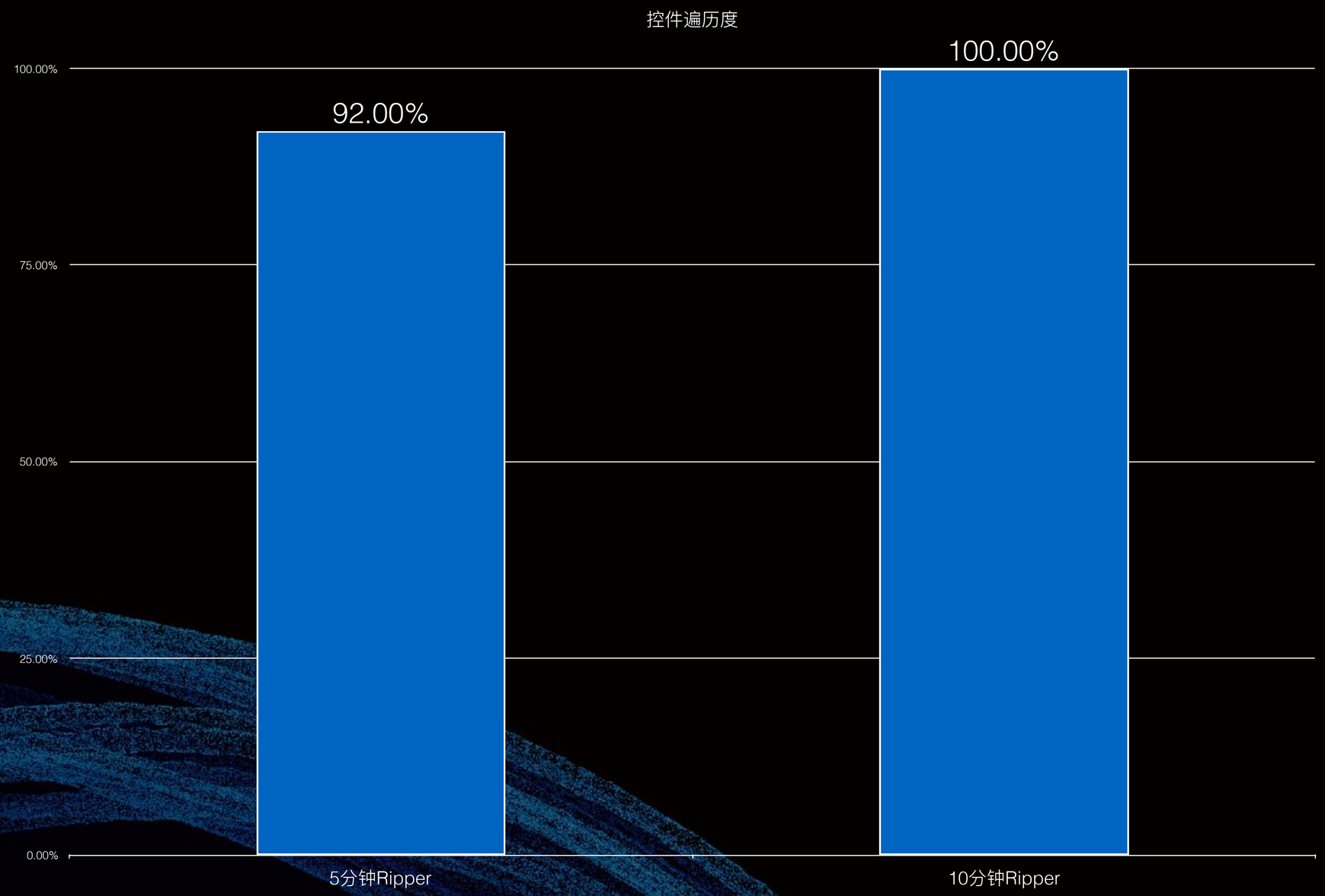
Ripper 5分钟



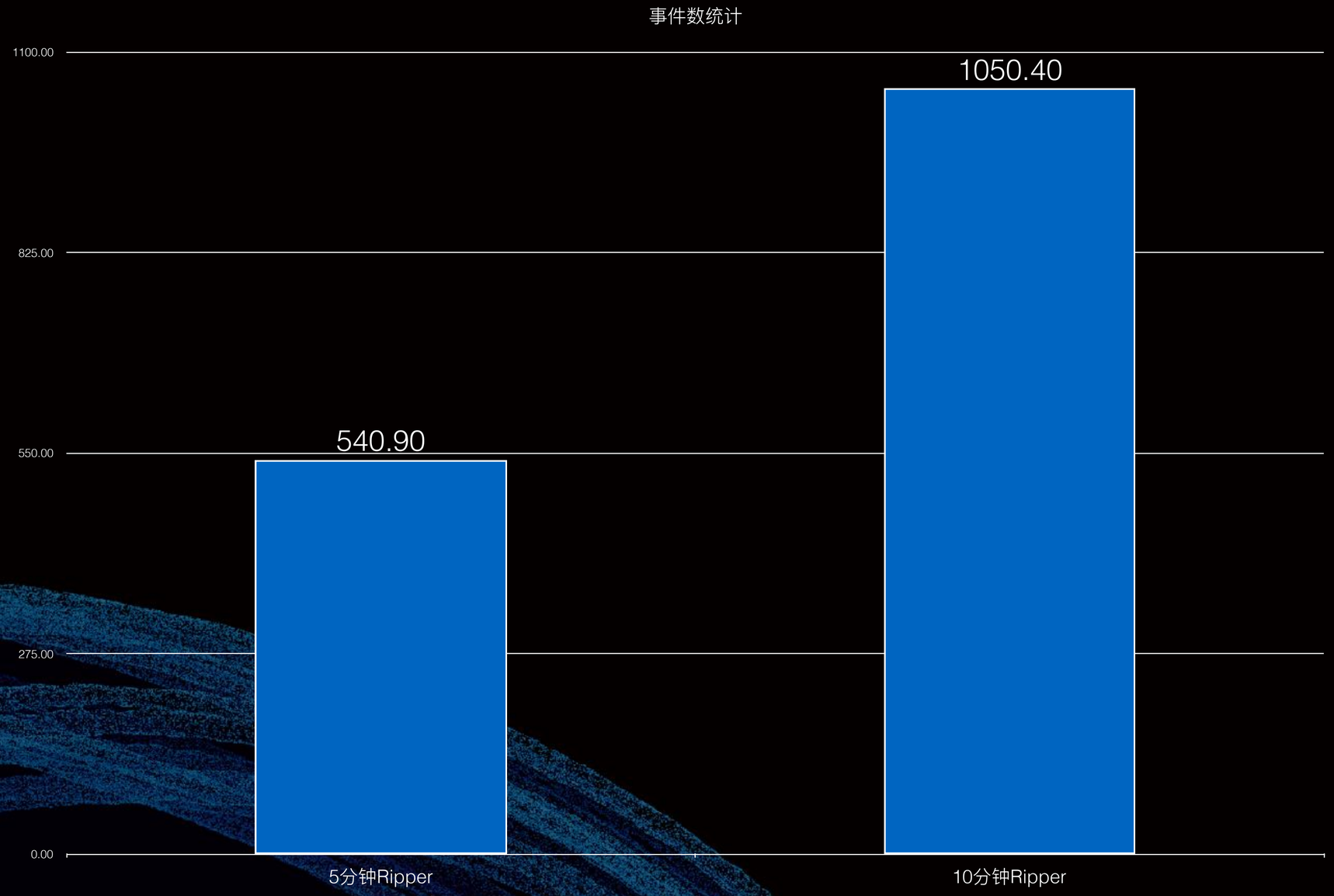
Ripper 10分钟



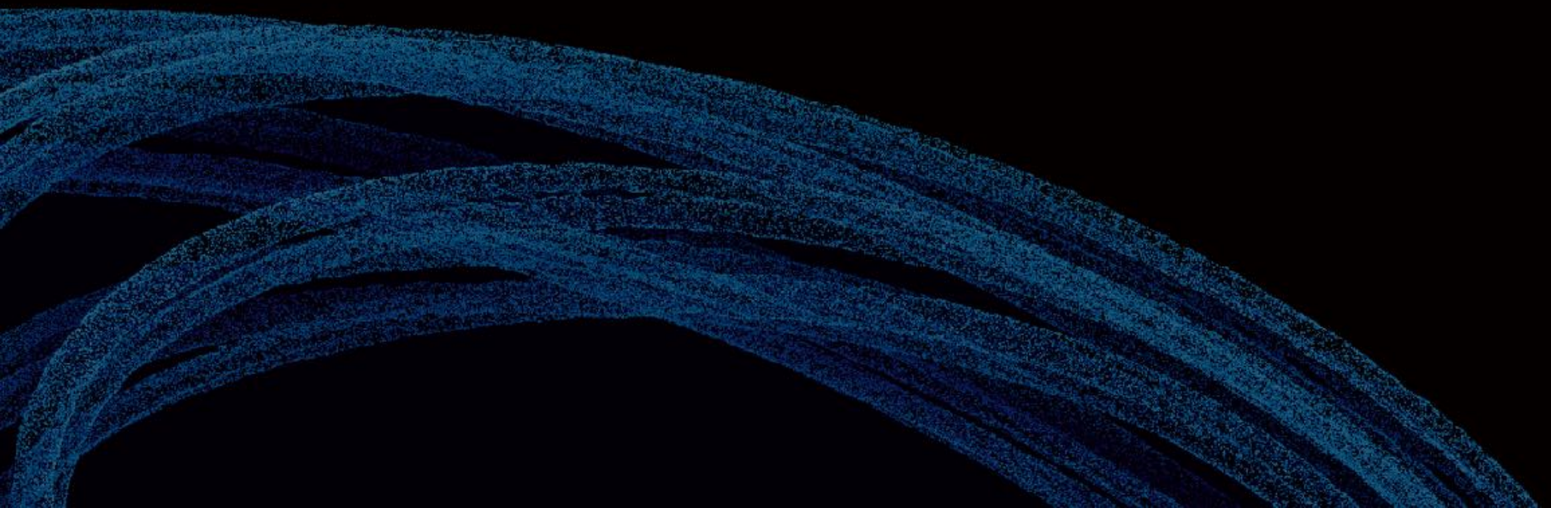
阿里云移动云


控件遍历度
100.00%
92.00%
100.00%
75.00%
50.00%
25.00%
0.00%
控件遍历度
5分钟Ripper
10分钟Ripper

阿里云移动云


事件数统计
1100.00
825.00
550.00
275.00
0.00
动作数
540.90
1050.40
5分钟Ripper
10分钟Ripper

场景理解



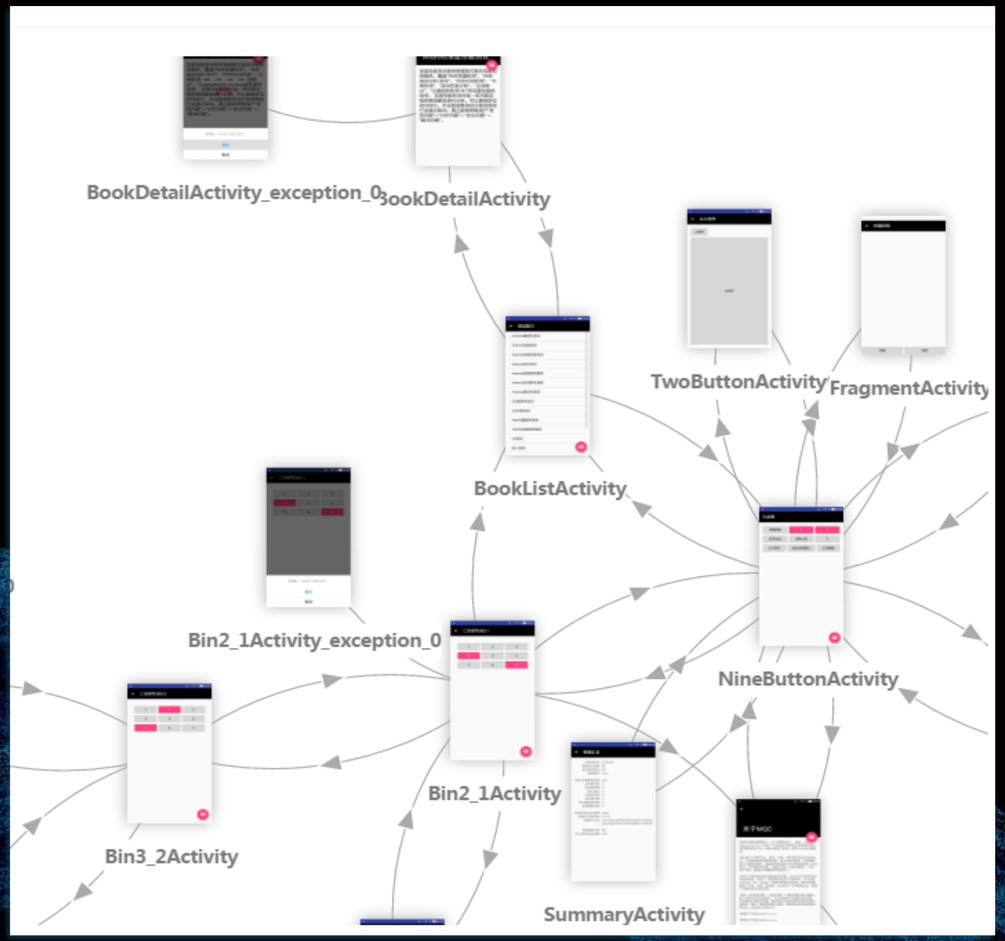
弱场景识别

- 控件（语义、类型）
- 弹窗

复杂场景识别

- 缩放，连击
- 视频切换，方向切换
- 登录，注册，反馈
- 自由探索

问题的复现与定位


BookDetailActivity_exception_0
BookDetailActivity
TwoButtonActivity
FragmentActivity
BookListActivity
Bin2_1Activity_exception_0
NineButtonActivity
Bin2_1Activity
Bin3_2Activity
SummaryActivity


问题复现步骤
cn.cmgame.demo-.MainActivity -> android.content.activitynotfoundexception: no activity found to handle in...
from
左到右的动作
点击坐标:
(569,458)
热门推荐
更多游戏
我的汤姆...
王者荣耀
汤姆猫跑...
快乐点点消
火力前线
疯狂动物园
僵尸榨汁机
猪猪侠机...
退出游戏
返回
应用程序错误
很抱歉，“Demo测试”已停止运行。
threadid=1: thread exiting with uncaught exception (group=0x41ffb898)
FATAL EXCEPTION: main
android.content.ActivityNotFoundException: No Activity found to handle Intent { act=android.int
at android.app.Instrumentation.checkStartActivityResult(Instrumentation.java:1632)
at android.app.Instrumentation.execStartActivity(Instrumentation.java:1424)
at android.app.Activity.startActivityForResult(Activity.java:3390)
at android.app.Activity.startActivityForResult(Activity.java:3351)
at android.app.Activity.startActivity(Activity.java:3587)
at android.app.Activity.startActivity(Activity.java:3555)
at cn.cmgame.marketing.contacor.cancel_page.BannerViewController$2.onClick(BannerViewController
at android.view.View.performClick(View.java:4240)
at android.view.View$PerformClick.run(View.java:17740)

问题解决方案

详细说明

1. 示例一：

```
java.lang.RuntimeException: Unable to start activity ComponentInfo{*}:
android.support.v4.app.Fragment$InstantiationException: *: make sure class name exists, is public, and has an empty
constructor that is public
```

当系统因为内存不足杀死非前台进程，用户又将被系统杀掉的非前台应用带回前台，如果这个时候有UI是呈现在Fragment中，那么会因为restore造成fragment需要通过反射实例对象，从而将之前save的状态还原。因此不应该改写构造函数，而应该使用静态方法比如newInstance来构造有参Fragment对象并返回。

代码示例：

步骤一：编写有参静态创建对象函数

```
public static final AlertFragment newInstance(int title, String message)
{
    AlertFragment f = new AlertFragment();
    Bundle bdl = new Bundle(2);
    bdl.putInt(EXTRA_TITLE, title);
    bdl.putString(EXTRA_MESSAGE, message);
    f.setArguments(bdl);
```

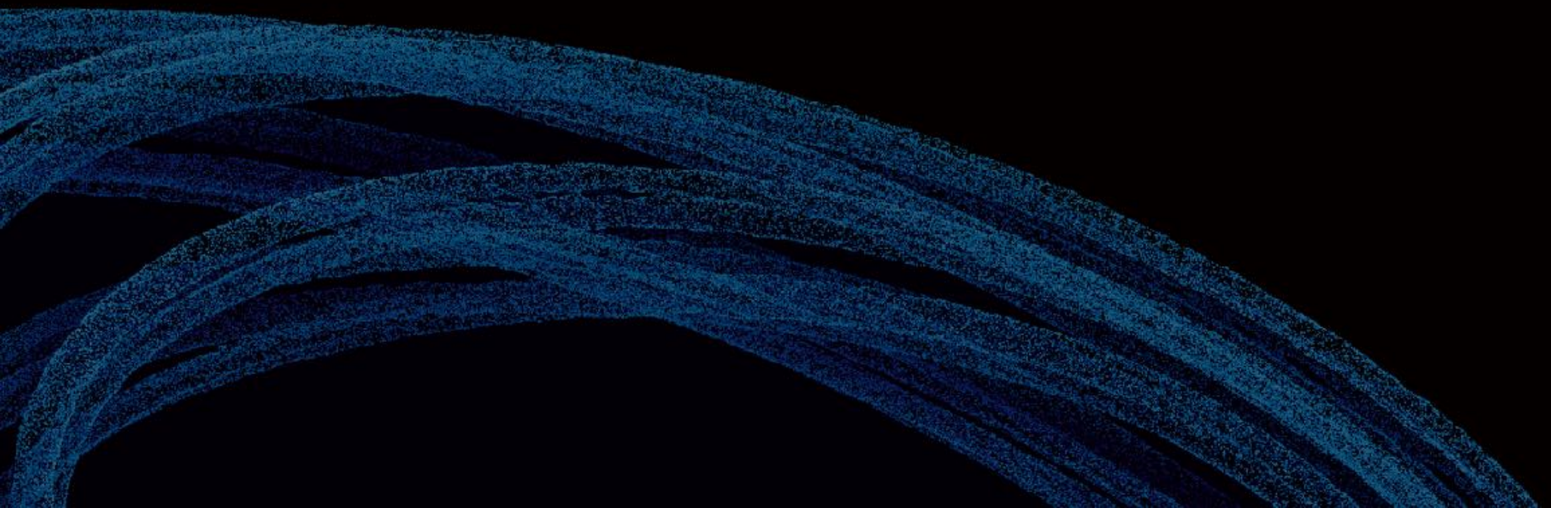
Ripper is Powerful !

并且

无休止运行

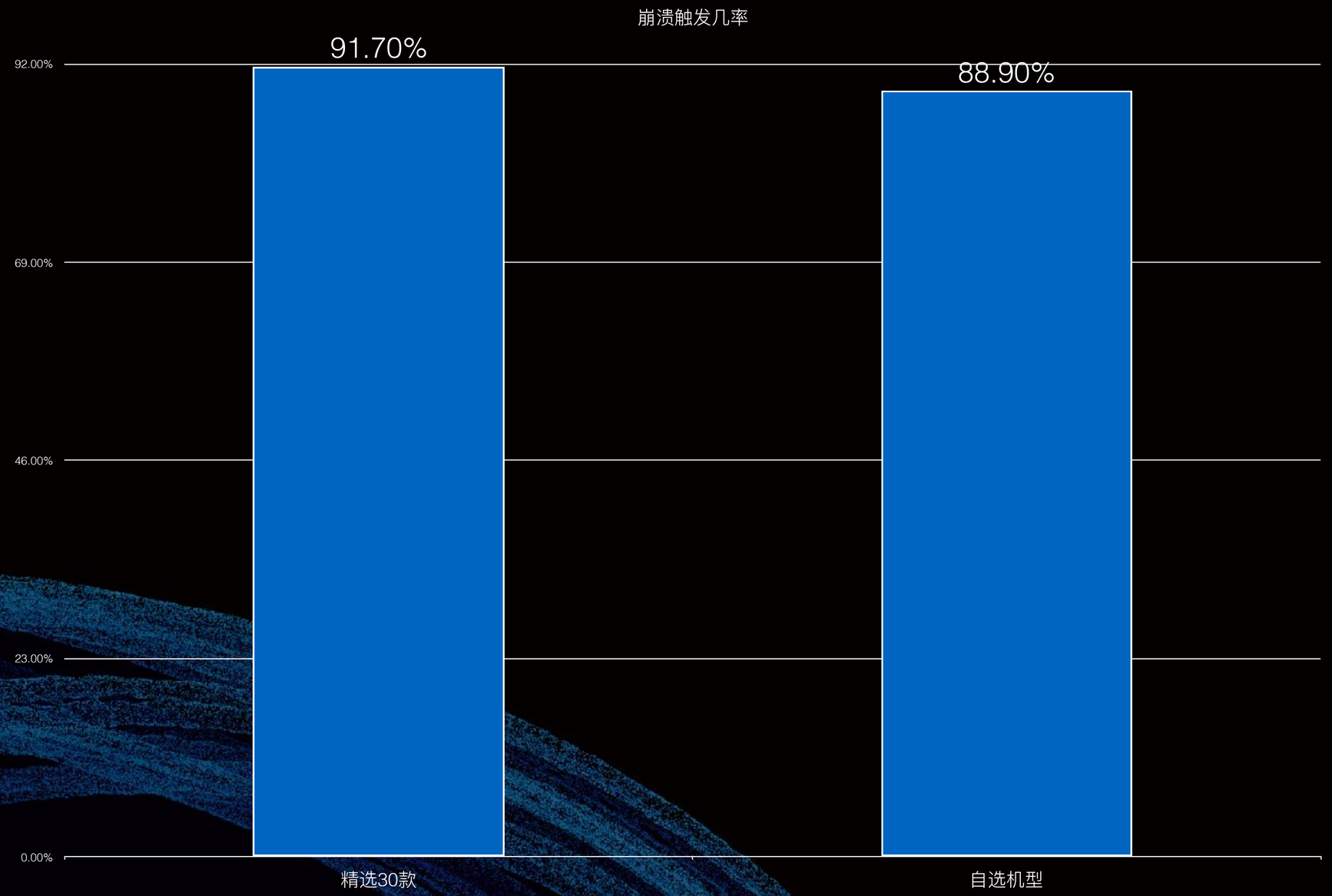
600+手机并行

检测效果



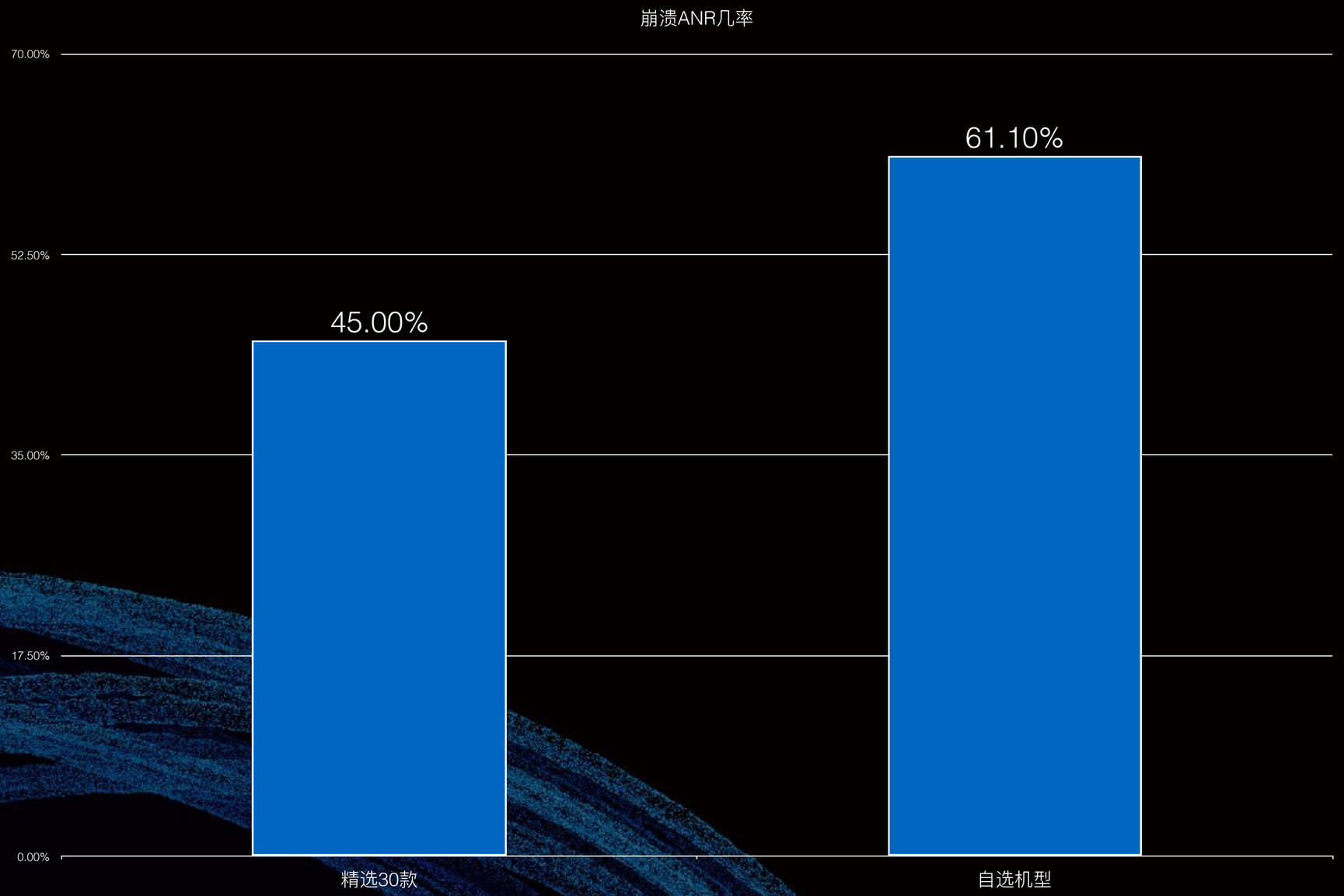
阿里云移动云

检测应用崩溃



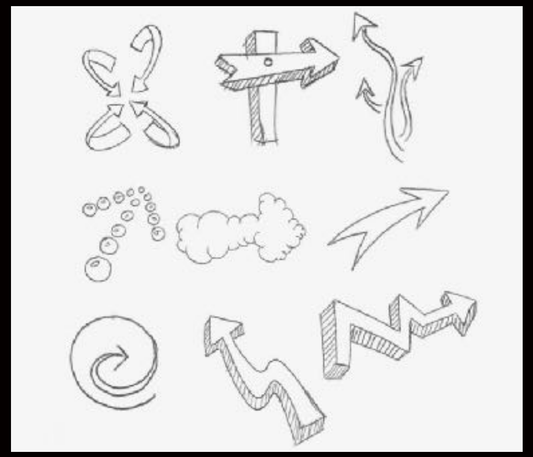
阿里云移动云

检测应用卡死



未来方向

Ripper的未来方向



理解复杂场景

UI问题自检测

自我学习机制

Ripper集成认证标准

将认证标准录入Ripper大脑，形成通用规则

- 兼容性认证规则
- 稳定性认证规则
- 性能认证规则
- UI 认证规则

阿里云移动云

杨志（花名：字白）
阿里巴巴技术专家

钉钉扫码加入移动云大
客户接待厅