Automated Performance Analysis for Android Applications

--- A tool that helps developers improve the performance of their apps

LU,Yingjun Wang,Xiaofang Xie,Min

Advised by

Prof. Shing-chi Cheung

Introduction

1.5 billion+ new apps downloads per month

1 million+ new Android devices are activated every day

But the app's quality is difficult to assure.

App Crash (functional)

Low quality → User complaints → Market failure

How we can help?

Bug Pattern Finding

Automated Bug Detection

An automated performance analysis TOOL!

What we have done..

After survey of developers' forum, Android documentation and research paper, we discovered 8 bug patterns. Then we implemented a scalable and efficient tool to detect bug patterns.

Inputs

Android app

.apk

Java bytecode

.class

Performance Analyzer

Outputs

Analysis Report

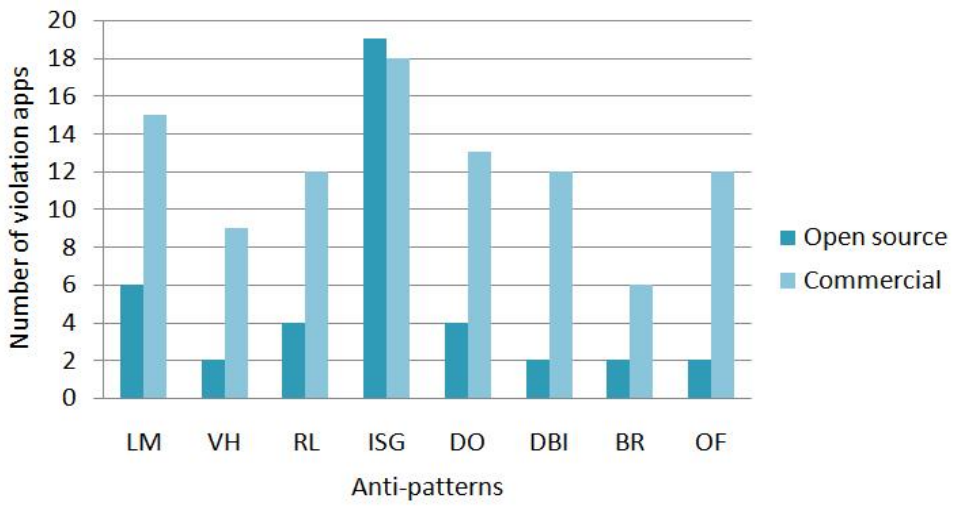
- Detected bugs
- Suggestion

Pattern set

Lengthy operations in Main thread	View Holder	Resource Leakage	Internal setter / getter
Drawing operations	Direct/ Batch Insertion	Bitmap images resize	Operations of FLING

Result

41 real-world apps (22 commercial apps and 19 open sources) are tested.
Here shows the number of apps that contain the eight anti-patterns.



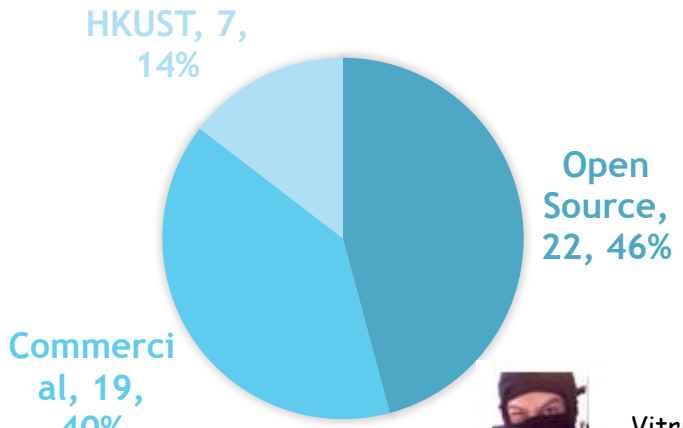
The results from the report are checked manually.
The average accuracy of our analyzer is 81.48%.

Anti-pattern	Num of violations	Num of violations checked	Precision
LM - Lengthy operations in Main thread	281	43	90.3%
VH - View Holder	33	33	84.8%
RL - Resource Leakage	56	56	49%
ISG - Internal setter / getter	1478	60	80%
DO - Drawing operations	61	30	100%
DBI - Direct/ Batch Insertion	14	14	85.7%
BR - Bitmap images resize	16	16	94%
OF - Operations of FLING	35	35	68%

Feedback From developers

Zhao Guanlun (Undergraduate in HKUST)

…this kind of code analysis would be a GREAT feature …Reminding the developer when such a mistake is made would definitely be awesome.



Vitriolix from Story Maker

Jing from Flipboard

…it's awesome to connect with bright developers this way.

…We'll definitely fix this and any others you find …finds like this will surely help.

Limitation & Conclusion

Implementation
- Bytecode generation, usage of WALA.

Static analysis tool
- Runtime information may help.

Algorithm designed
- More reasonable assumption can be developed.

During the final year project, we :
- Study and collect various anti-patterns。
- Implement an analysis tool.
- Report suggestion to real developers and obtain many positive feedback.