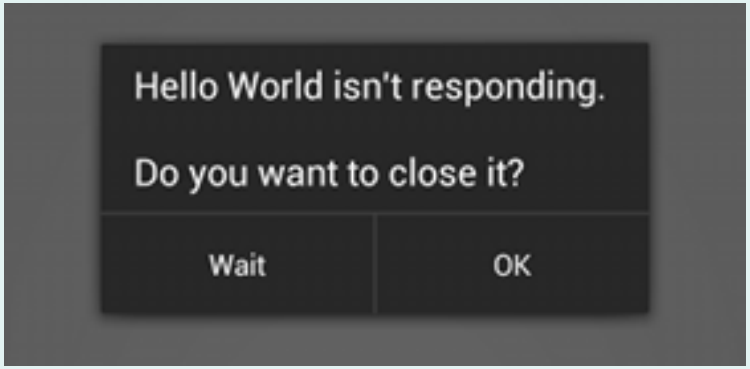
Performance analysis of Android application (a preliminary study)

LU,Yingjun

Huang,Richeng

Introduction

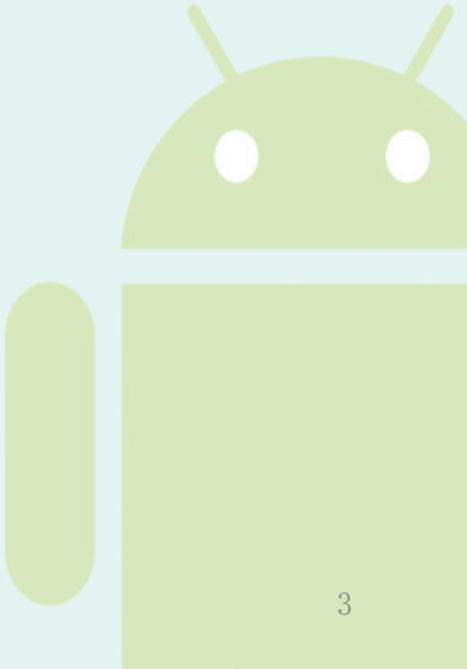
- Many Android applications nowadays still suffer from the slowness problem in varying degrees.



Identify the rudimentary cause

- App cache?
- Device is too old?

➢The design of the application.



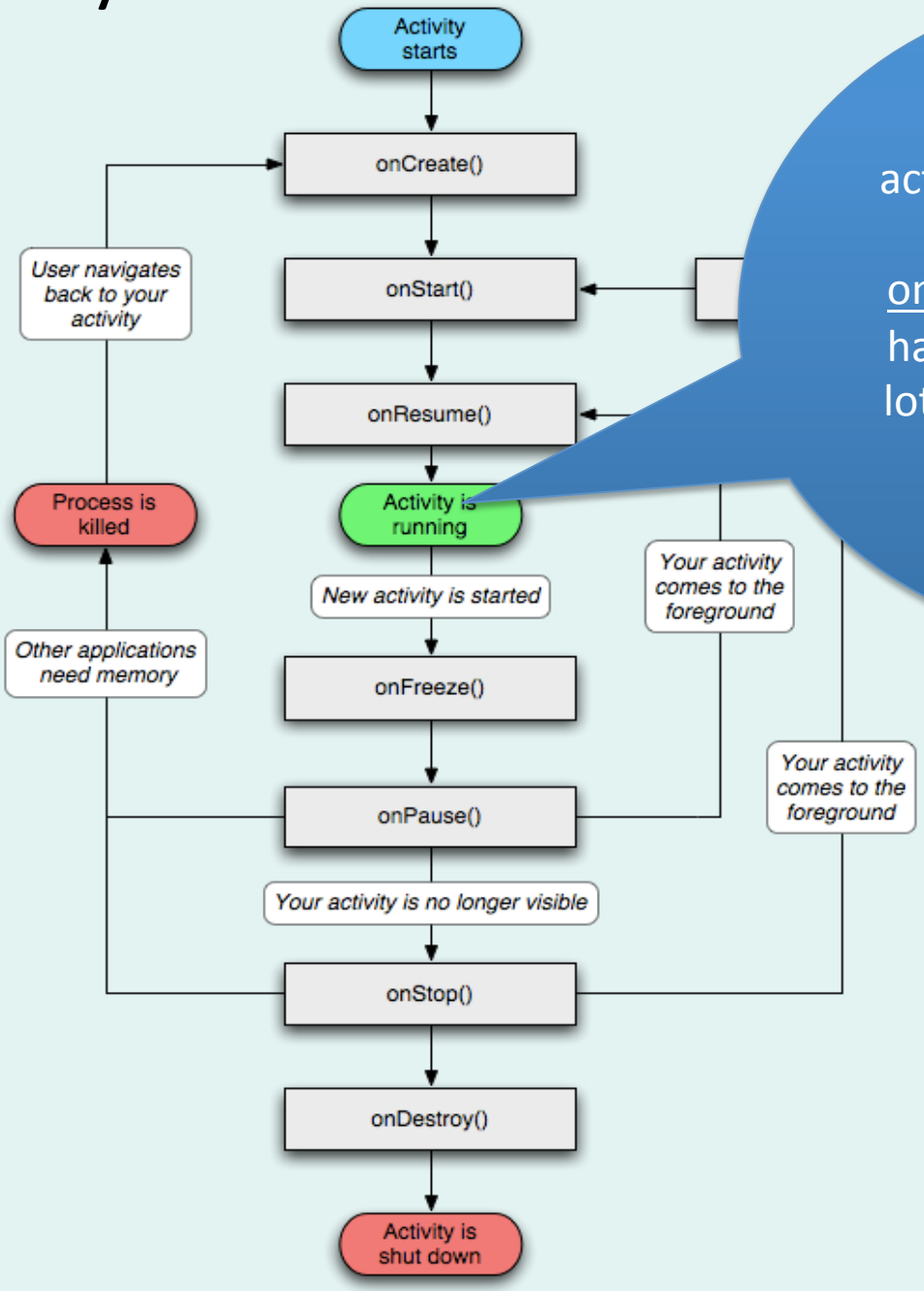
Three Questions

- What is the method that costs most time?*Is there any particular handler that is more likely to be time consuming?*
- Does an application always contain heavy weight methods?
- How would these methods affect the performance of the application?

Background

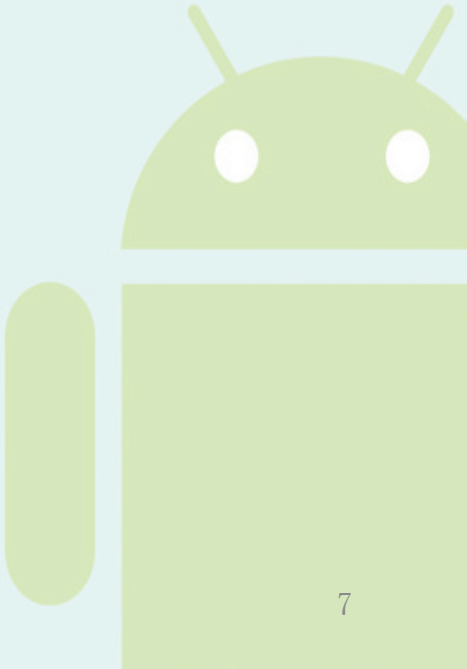
- Application components (4 types)
  - Activities, Services, Content providers and Broadcast receivers
- Component lifecycles (use activity as example)
- Single thread policy
  - By default, all components of the same application run in the same process and thread (called the "main" thread).

Activity lifecycle



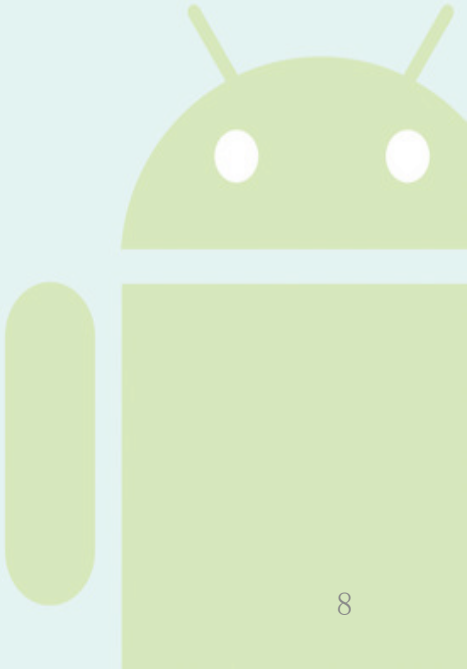
Every time before the activity is ready and visible to user, onCreate(), onStart() and onResume() have been called first --- a lot of waiting time if these three are not well-designed.

- 100 to 200ms is the threshold beyond which users will perceive slowness in an application
  - http://developer.android.com/training/articles/perf-anr.html



- component lifecycle handler
- GUI event handler
- Other methods

- More than 200ms→ Warning

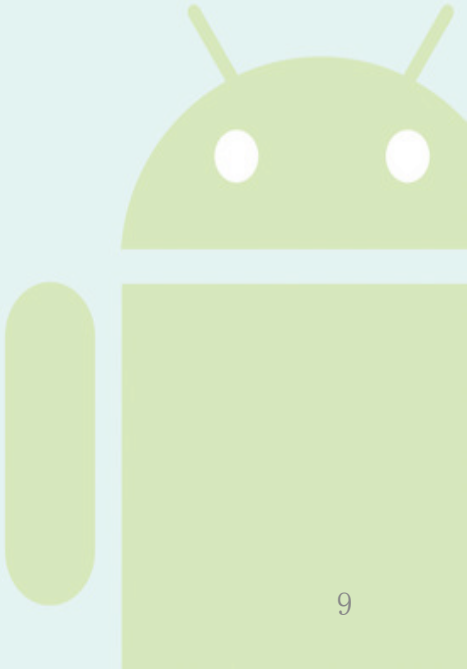


Ultimate goal

- Develop a tool:

Input : Any Android application

Output: Identify possible methods that have potential threat of causing slowness

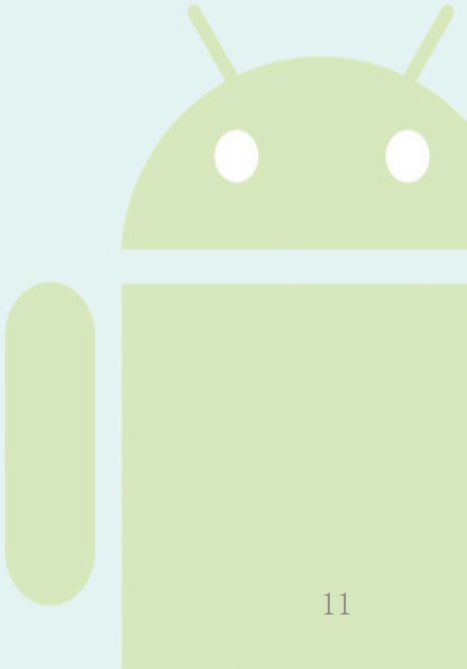


Project Plan

- a)Automatic Instrumentaion for Android Application
  - Soot ?
  - Manually add codes ?
- b)Random Testing For Android Application
  - Robotium ?
  - Monkey ?
- c)Log the Information
- d)Offline Analysis

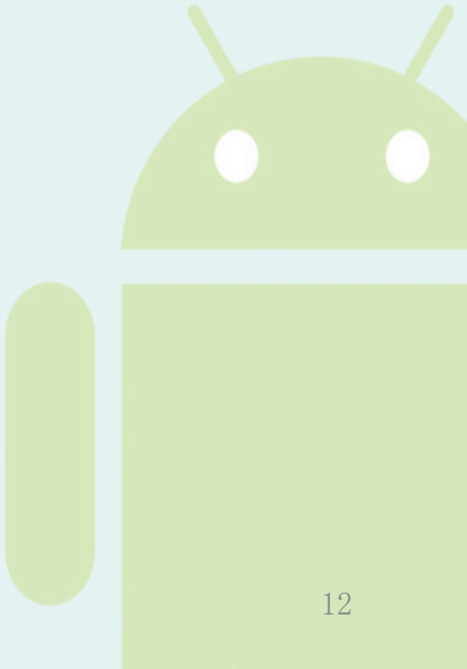
Problems with Soot

- Limited documentation
- A number of problems reported by other users.
- Try to follow some blogs,but fail
- Schedule limitation



Manually add codes

- Log the running duration of every component lifecycle handler and GUI event handler,and other methods.
- Log thread info



Program code

- Initialize CSV writer

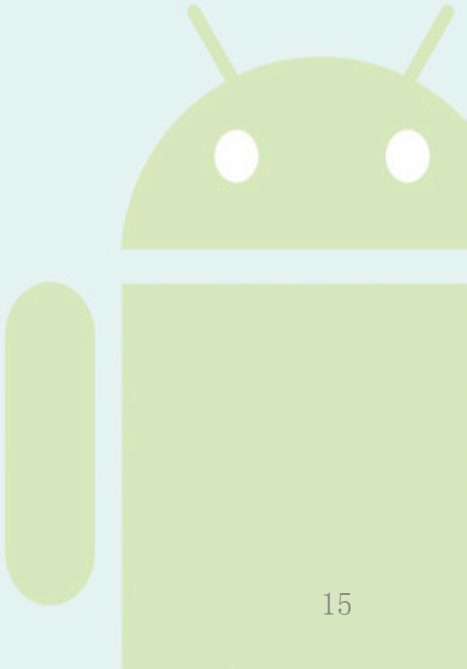
```
static
{
    //LOG
    try
    {
    File root = Environment.getExternalStorageDirectory();
    File gpxfile = new File(root, "FileName.csv");
    if(!gpxfile.exists())
    {
        gpxfile.createNewFile();
    }
    FileWriter fwriter = new FileWriter(gpxfile);
    writer = new CSVWriter(fwriter);
    }

        catch (IOException e)
        {
            e.printStackTrace(); //error
        }
}
```

- Compute Consuming Time

```
@Override
public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  long start = System.currentTimeMillis();
  //Code Body
  long end = System.currentTimeMillis();
  entries = (this.getClass().getName()+"#"+"onCreate"+ "#"
          +Thread.currentThread().getId()+"#"+ (end-start)).split("#");
  writer.writeNext(entries);
   try {
      writer.flush();
  } catch (IOException e) {
      e.printStackTrace();
  }
}
```

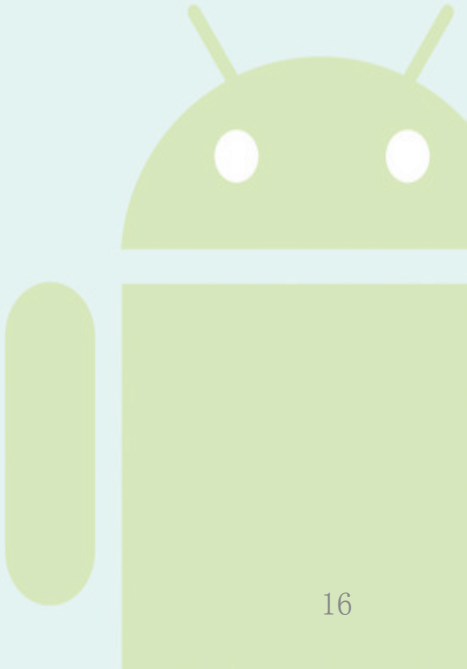
Random Testing

- First Try:Robotium
  - Robotium is a powerful tool.
  - However, when aiming at generating test for arbitary application, robotium is not suitable.



Monkey

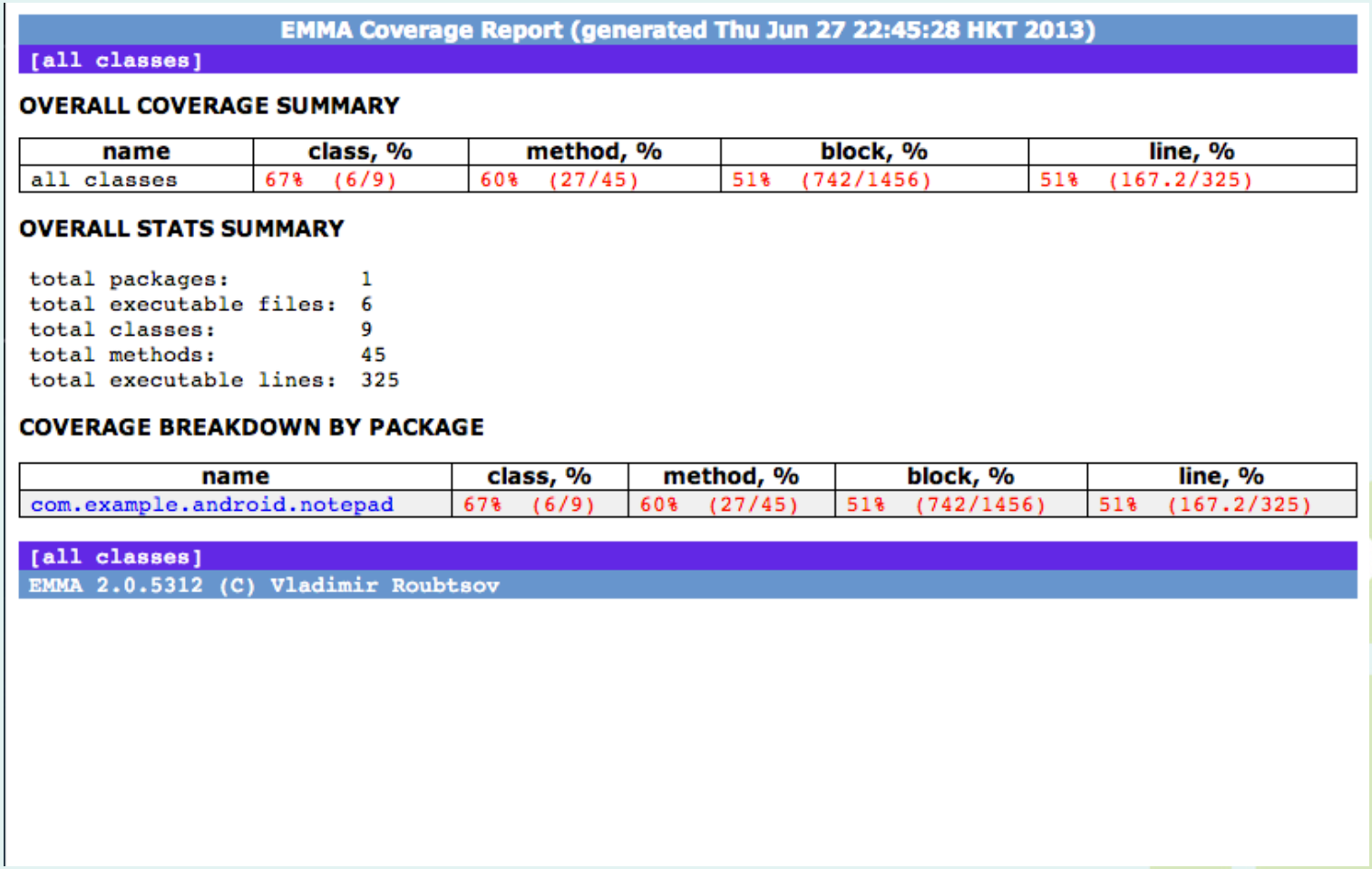
- Monkey is a tool that sends random events to the device.
- The more events it send, the higher coverage the test can reach.



Test Code

```
public class ContactAdderTest extends
                ActivityInstrumentationTestCase2<ContactAdder> {

        private int NUM_EVENTS = 1000;

        public ContactAdderTest() {
                super("com.example.android.contactmanager", ContactAdder.class);
        }

        @Override
        protected void setUp() throws Exception {
                super.setUp();
                setActivityInitialTouchMode(false);

        }

        public void testMonkeyEvents() {
                Display display = getActivity().getWindowManager().getDefaultDisplay();
                Instrumentation inst = getInstrumentation();
                PackageManager pm = getActivity().getPackageManager();

                Monkey monkey = new Monkey(display,
                                "com.example.android.contactmanager", inst, pm);

                // Generate and fire a random event.
                for (int i = 0; i < NUM_EVENTS; i++) {
                        monkey.nextRandomEvent();
                }

        }
}
```

Coverage Report



EMMA Coverage Report (generated Thu Jun 27 22:45:28 HKT 2013)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	67% (6/9)	60% (27/45)	51% (742/1456)	51% (167.2/325)

OVERALL STATS SUMMARY

```
total packages:          1
total executable files:  6
total classes:           9
total methods:           45
total executable lines:  325
```

COVERAGE BREAKDOWN BY PACKAGE

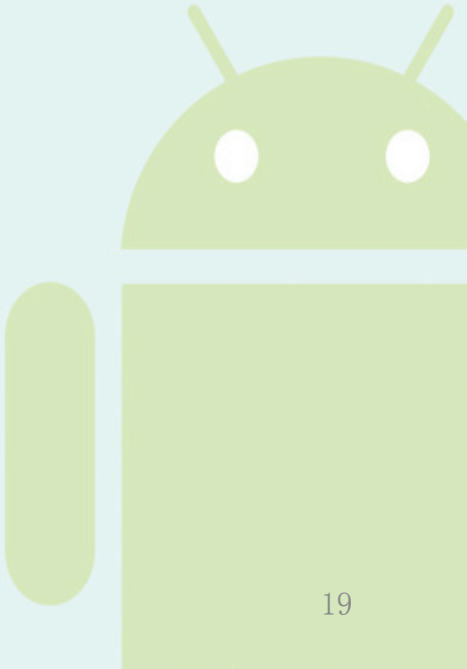
name	class, %	method, %	block, %	line, %
com.example.android.notepad	67% (6/9)	60% (27/45)	51% (742/1456)	51% (167.2/325)

[all classes]

EMMA 2.0.5312 (C) Vladimir Roubtsov

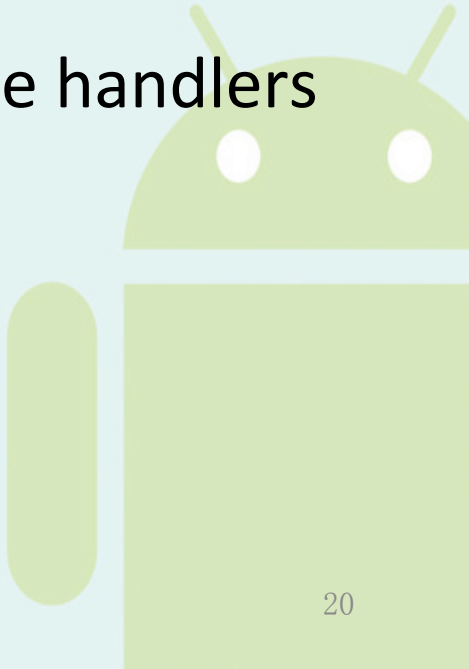
Result

- random events : 100 to 100000.
- Coverage :25% to 55%



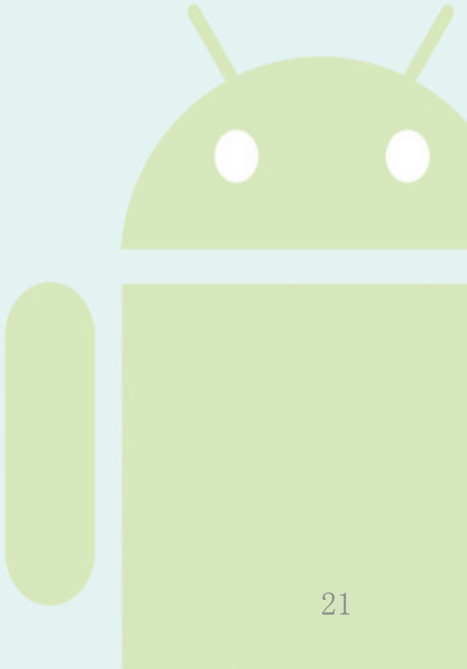
Approaches as a whole

- Dynamic analysis tool
  - Random testing engine (based on Monkey)
  - Runtime profiler (by instrumentation)
- Offline analysis
  - Identify heavy weight handlers (lifecycle handlers and UI event handlers)



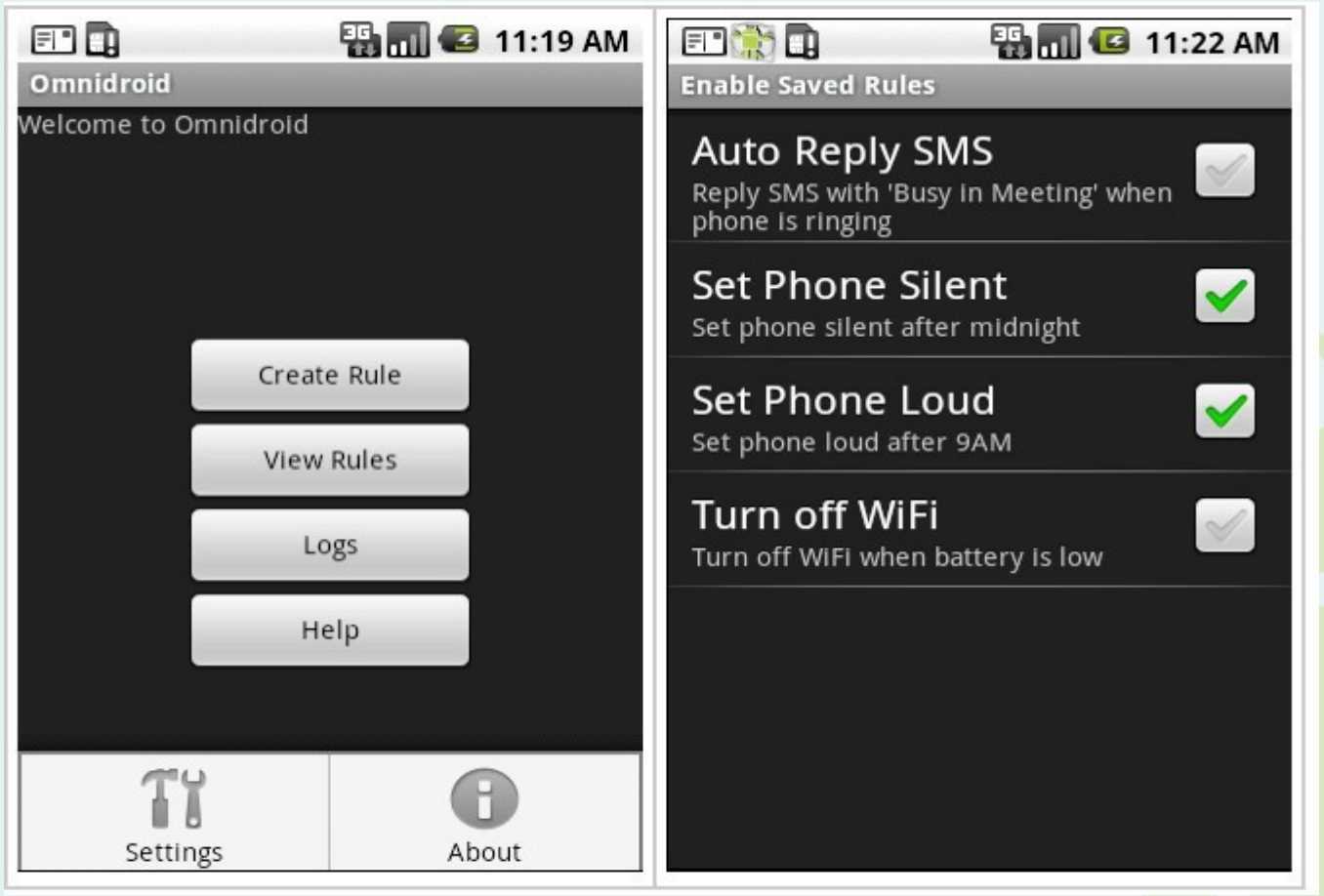
App example

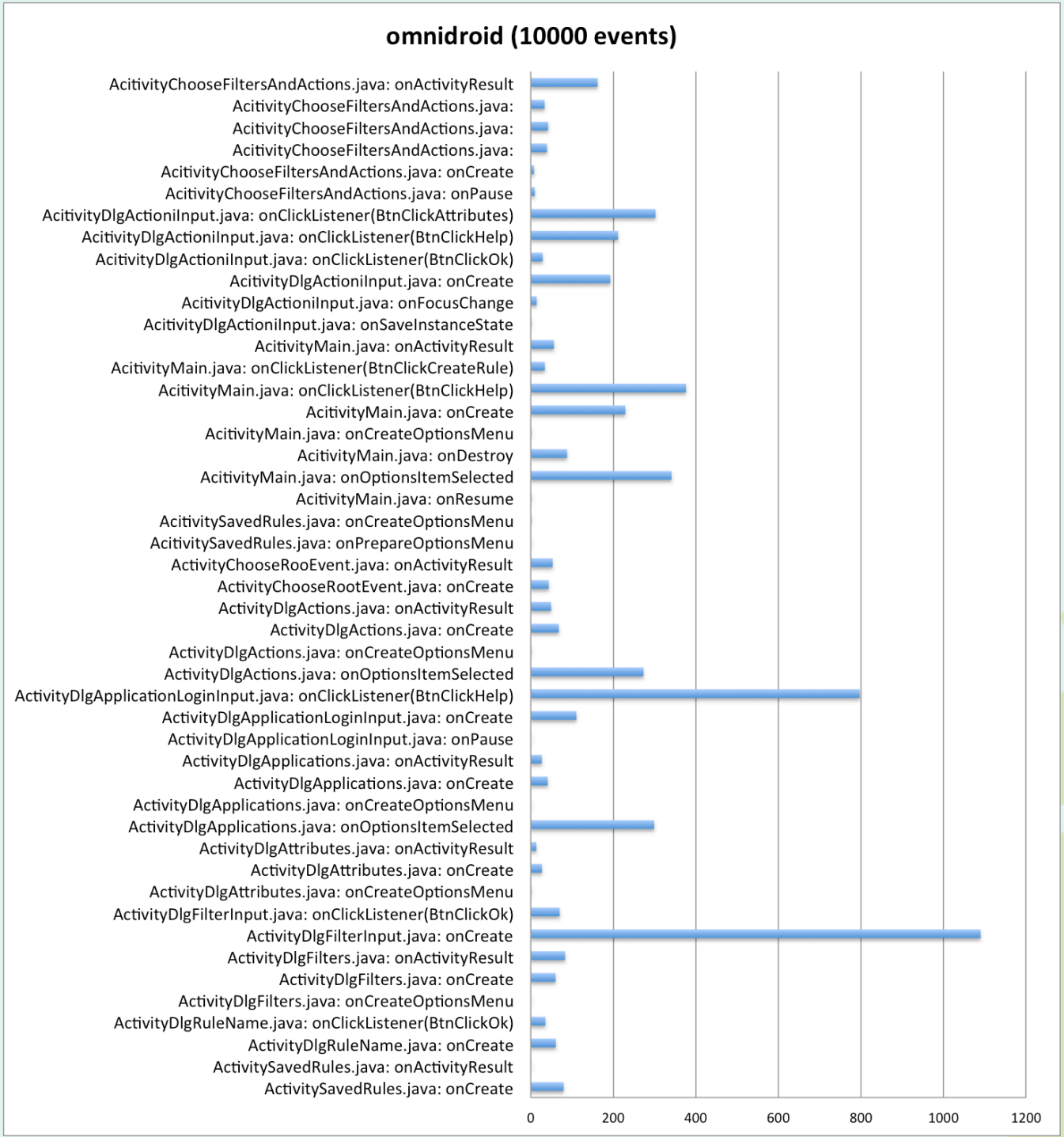
- Omnidroid
  - App category: Efficiency
  - App downloads: 1,000 - 5,000
  - Size: 6902 lines
- Opensudoku
  - App category: Brain & Puzzle
  - App downloads: 1,000,000 - 5,000,000
  - Size: 3813 lines

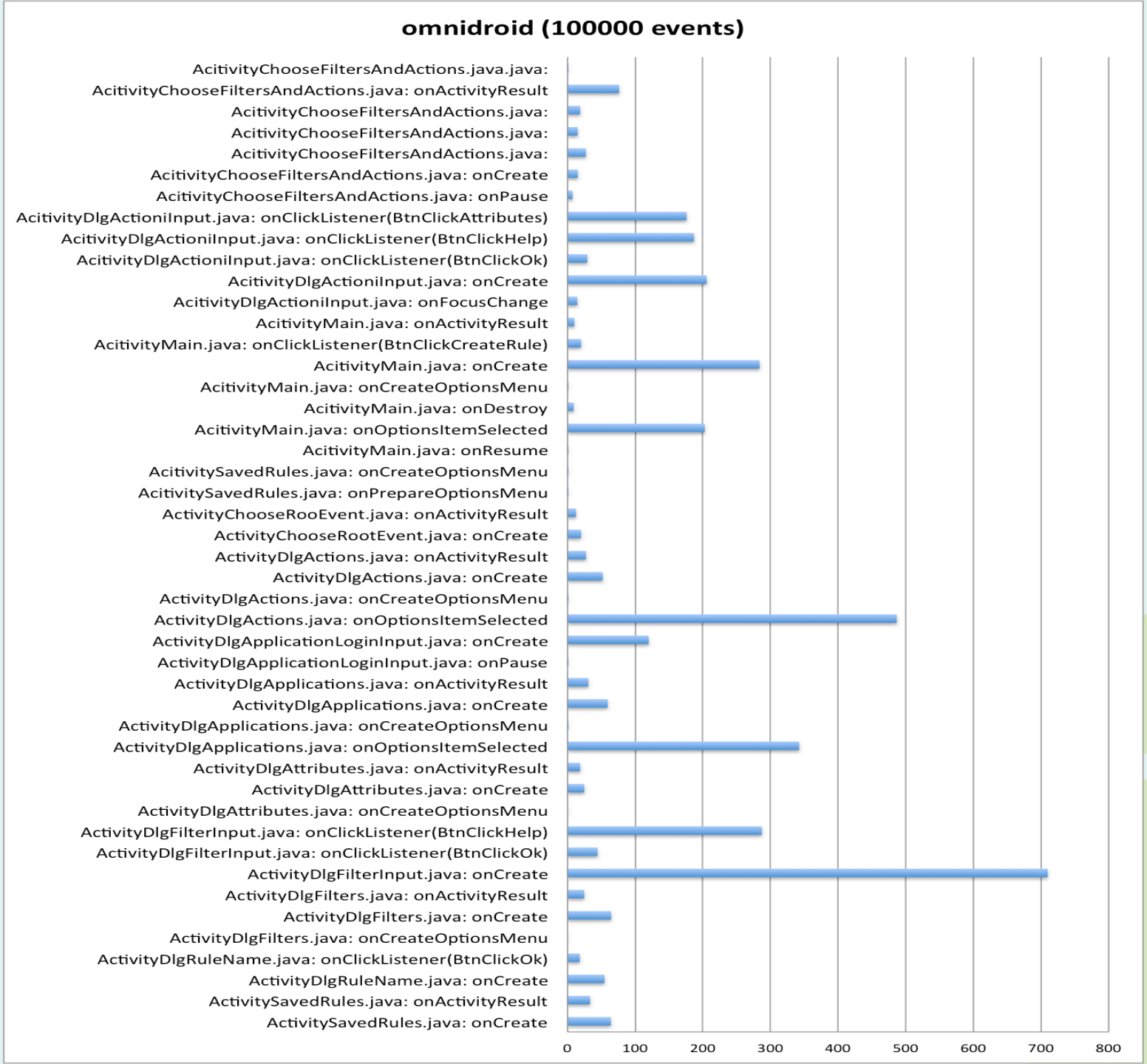


Figures and tables

- Omnidroid




omnidroid (10000 events)
AcitivityChooseFiltersAndActions.java: onActivityResult
AcitivityChooseFiltersAndActions.java:
AcitivityChooseFiltersAndActions.java:
AcitivityChooseFiltersAndActions.java:
AcitivityChooseFiltersAndActions.java: onCreate
AcitivityChooseFiltersAndActions.java: onPause
AcitivityDlgActioniInput.java: onClickListener(BtnClickAttributes)
AcitivityDlgActioniInput.java: onClickListener(BtnClickHelp)
AcitivityDlgActioniInput.java: onClickListener(BtnClickOk)
AcitivityDlgActioniInput.java: onCreate
AcitivityDlgActioniInput.java: onFocusChange
AcitivityDlgActioniInput.java: onSaveInstanceState
AcitivityMain.java: onActivityResult
AcitivityMain.java: onClickListener(BtnClickCreateRule)
AcitivityMain.java: onClickListener(BtnClickHelp)
AcitivityMain.java: onCreate
AcitivityMain.java: onCreateOptionsMenu
AcitivityMain.java: onDestroy
AcitivityMain.java: onOptionsItemSelected
AcitivityMain.java: onResume
AcitivitySavedRules.java: onCreateOptionsMenu
AcitivitySavedRules.java: onPrepareOptionsMenu
ActivityChooseRooEvent.java: onActivityResult
ActivityChooseRootEvent.java: onCreate
ActivityDlgActions.java: onActivityResult
ActivityDlgActions.java: onCreate
ActivityDlgActions.java: onCreateOptionsMenu
ActivityDlgActions.java: onOptionsItemSelected
ActivityDlgApplicationLoginInput.java: onClickListener(BtnClickHelp)
ActivityDlgApplicationLoginInput.java: onCreate
ActivityDlgApplicationLoginInput.java: onPause
ActivityDlgApplications.java: onActivityResult
ActivityDlgApplications.java: onCreate
ActivityDlgApplications.java: onCreateOptionsMenu
ActivityDlgApplications.java: onOptionsItemSelected
ActivityDlgAttributes.java: onActivityResult
ActivityDlgAttributes.java: onCreate
ActivityDlgAttributes.java: onCreateOptionsMenu
ActivityDlgFilterInput.java: onClickListener(BtnClickOk)
ActivityDlgFilterInput.java: onCreate
ActivityDlgFilters.java: onActivityResult
ActivityDlgFilters.java: onCreate
ActivityDlgFilters.java: onCreateOptionsMenu
ActivityDlgRuleName.java: onClickListener(BtnClickOk)
ActivityDlgRuleName.java: onCreate
ActivitySavedRules.java: onActivityResult
ActivitySavedRules.java: onCreate
0
200
400
600
800
1000
1200


omnidroid (100000 events)
AcitivityChooseFiltersAndActions.java.java:
AcitivityChooseFiltersAndActions.java: onActivityResult
AcitivityChooseFiltersAndActions.java:
AcitivityChooseFiltersAndActions.java:
AcitivityChooseFiltersAndActions.java:
AcitivityChooseFiltersAndActions.java: onCreate
AcitivityChooseFiltersAndActions.java: onPause
AcitivityDlgActioniInput.java: onClickListener(BtnClickAttributes)
AcitivityDlgActioniInput.java: onClickListener(BtnClickHelp)
AcitivityDlgActioniInput.java: onClickListener(BtnClickOk)
AcitivityDlgActioniInput.java: onCreate
AcitivityDlgActioniInput.java: onFocusChange
AcitivityMain.java: onActivityResult
AcitivityMain.java: onClickListener(BtnClickCreateRule)
AcitivityMain.java: onCreate
AcitivityMain.java: onCreateOptionsMenu
AcitivityMain.java: onDestroy
AcitivityMain.java: onOptionsItemSelected
AcitivityMain.java: onResume
AcitivitySavedRules.java: onCreateOptionsMenu
AcitivitySavedRules.java: onPrepareOptionsMenu
ActivityChooseRooEvent.java: onActivityResult
ActivityChooseRootEvent.java: onCreate
ActivityDlgActions.java: onActivityResult
ActivityDlgActions.java: onCreate
ActivityDlgActions.java: onCreateOptionsMenu
ActivityDlgActions.java: onOptionsItemSelected
ActivityDlgApplicationLoginInput.java: onCreate
ActivityDlgApplicationLoginInput.java: onPause
ActivityDlgApplications.java: onActivityResult
ActivityDlgApplications.java: onCreate
ActivityDlgApplications.java: onCreateOptionsMenu
ActivityDlgApplications.java: onOptionsItemSelected
ActivityDlgAttributes.java: onActivityResult
ActivityDlgAttributes.java: onCreate
ActivityDlgAttributes.java: onCreateOptionsMenu
ActivityDlgFilterInput.java: onClickListener(BtnClickHelp)
ActivityDlgFilterInput.java: onClickListener(BtnClickOk)
ActivityDlgFilterInput.java: onCreate
ActivityDlgFilters.java: onActivityResult
ActivityDlgFilters.java: onCreate
ActivityDlgFilters.java: onCreateOptionsMenu
ActivityDlgRuleName.java: onClickListener(BtnClickOk)
ActivityDlgRuleName.java: onCreate
ActivitySavedRules.java: onActivityResult
ActivitySavedRules.java: onCreate
0
100
200
300
400
500
600
700
800

EMMA Coverage Report (generated Sat Jul 27 00:52:02 HKT 2013)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	66% (159/241)	46% (628/1362)	41% (13488/33022)	40% (2785.7/6902)

OVERALL STATS SUMMARY

total packages: 15
total executable files: 154
total classes: 241
total methods: 1362
total executable lines: 6902

Line, %
40% (2785.7/6902)

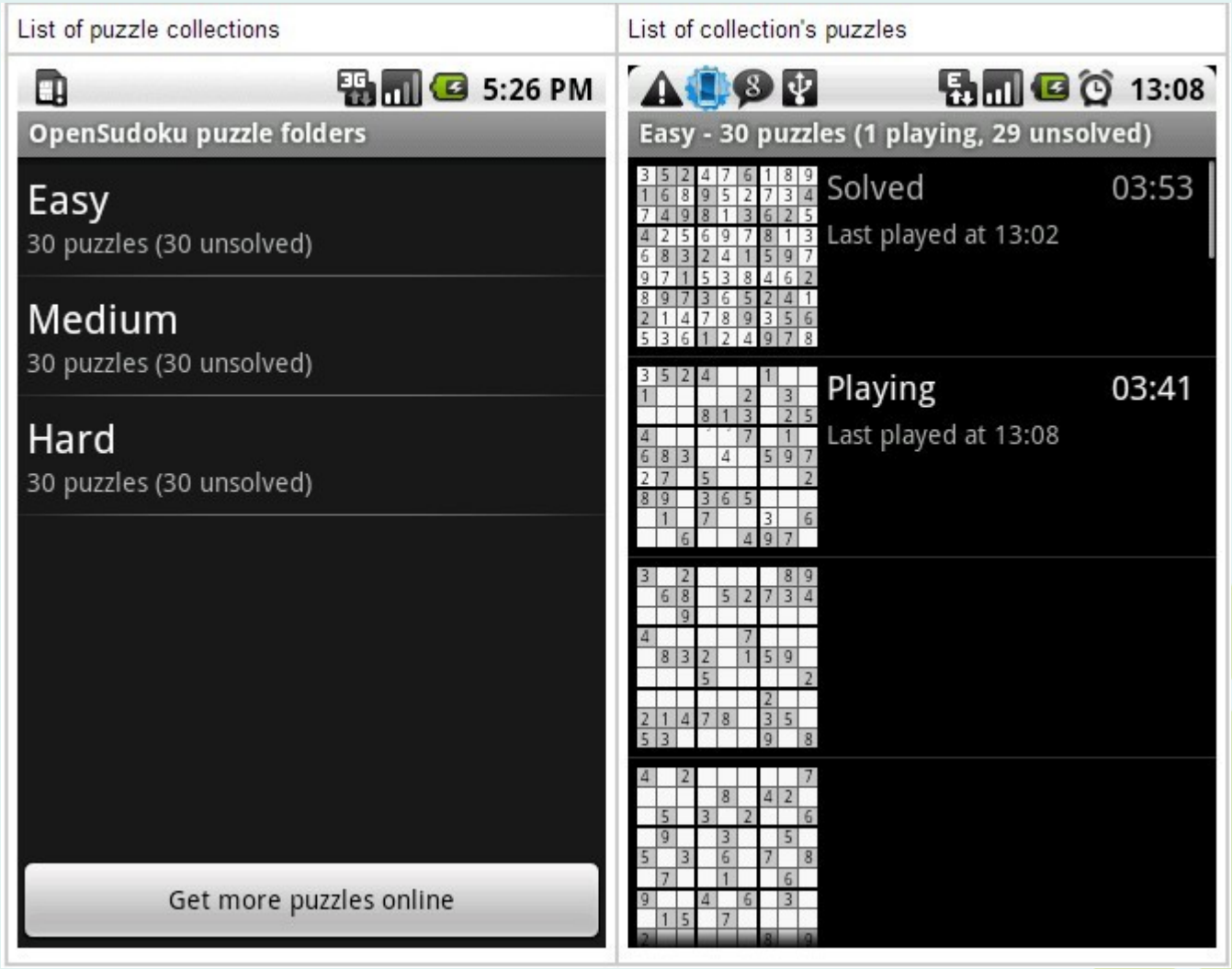
COVERAGE BREAKDOWN BY PACKAGE

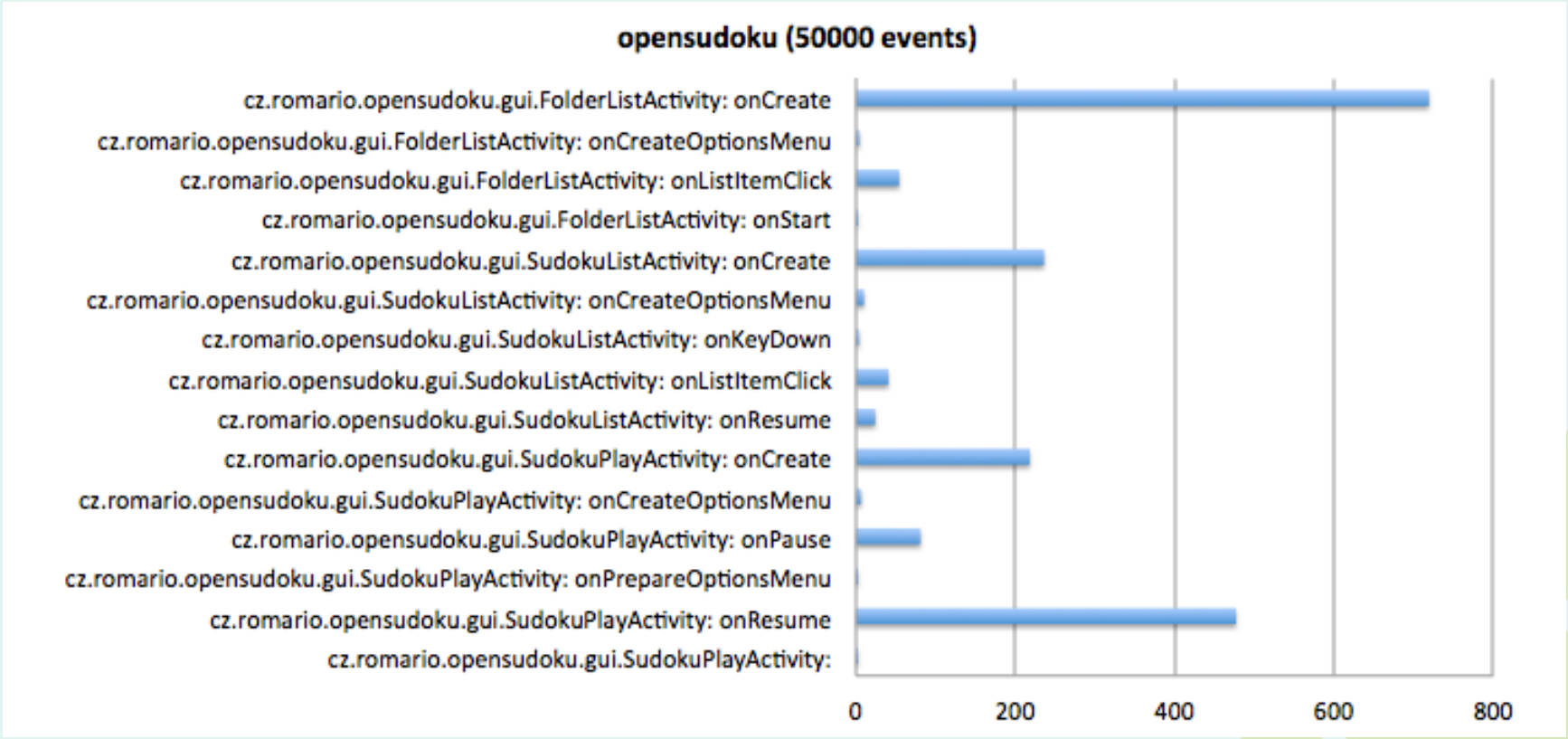
name	class, %	method, %	block, %	line, %
edu.nyu.cs.omnidroid.app.controller.actions	0% (0/14)	0% (0/45)	0% (0/831)	0% (0/176)
edu.nyu.cs.omnidroid.app.controller.external.actions	0% (0/9)	0% (0/40)	0% (0/1000)	0% (0/245)
edu.nyu.cs.omnidroid.app.controller.events	18% (2/11)	9% (2/22)	4% (12/310)	6% (4/67)
edu.nyu.cs.omnidroid.app.controller.util	50% (4/8)	20% (11/56)	12% (84/715)	14% (24.8/177)
edu.nyu.cs.omnidroid.app.model.db	83% (19/23)	34% (77/228)	23% (1537/6700)	20% (248.6/1257)
edu.nyu.cs.omnidroid.app.controller.datatypes	68% (15/22)	31% (44/144)	30% (745/2478)	25% (108.8/434)
edu.nyu.cs.omnidroid.app.controller.bkgservice	50% (1/2)	60% (3/5)	35% (34/98)	32% (8/25)
edu.nyu.cs.omnidroid.app.controller	80% (8/10)	50% (25/50)	45% (620/1387)	46% (141.5/306)
edu.nyu.cs.omnidroid.app.controller.external.helper.telephony	100% (5/5)	34% (10/29)	46% (81/175)	46% (24.3/53)
edu.nyu.cs.omnidroid.app.model	88% (14/16)	49% (72/146)	47% (2488/5245)	51% (534.8/1050)
edu.nyu.cs.omnidroid.app.view.simple.viewitem	83% (10/12)	60% (40/67)	49% (660/1344)	54% (150.4/281)
edu.nyu.cs.omnidroid.app.controller.external.attributes	91% (10/11)	58% (35/60)	53% (459/870)	52% (103.2/199)
edu.nyu.cs.omnidroid.app.view.simple	70% (57/82)	65% (250/387)	56% (5832/10476)	53% (1235.6/2327)
edu.nyu.cs.omnidroid.app.view.simple.model	92% (11/12)	73% (52/71)	64% (465/725)	66% (126/190)
edu.nyu.cs.omnidroid.app.view.simple.factoryui	75% (3/4)	58% (7/12)	71% (471/668)	66% (75.8/115)

[all classes]

EMMA 2.0.5312 (C) Vladimir Roubtsov

- Opensudoku


List of puzzle collections
5:26 PM
OpenSudoku puzzle folders
Easy
30 puzzles (30 unsolved)
Medium
30 puzzles (30 unsolved)
Hard
30 puzzles (30 unsolved)
Get more puzzles online
List of collection's puzzles
13:08
Easy - 30 puzzles (1 playing, 29 unsolved)
Solved
03:53
Last played at 13:02
Playing
03:41
Last played at 13:08


opensudoku (50000 events)
cz.romario.opensudoku.gui.FolderListActivity: onCreate
cz.romario.opensudoku.gui.FolderListActivity: onCreateOptionsMenu
cz.romario.opensudoku.gui.FolderListActivity: onListItemClick
cz.romario.opensudoku.gui.FolderListActivity: onStart
cz.romario.opensudoku.gui.SudokuListActivity: onCreate
cz.romario.opensudoku.gui.SudokuListActivity: onCreateOptionsMenu
cz.romario.opensudoku.gui.SudokuListActivity: onKeyDown
cz.romario.opensudoku.gui.SudokuListActivity: onListItemClick
cz.romario.opensudoku.gui.SudokuListActivity: onResume
cz.romario.opensudoku.gui.SudokuPlayActivity: onCreate
cz.romario.opensudoku.gui.SudokuPlayActivity: onCreateOptionsMenu
cz.romario.opensudoku.gui.SudokuPlayActivity: onPause
cz.romario.opensudoku.gui.SudokuPlayActivity: onPrepareOptionsMenu
cz.romario.opensudoku.gui.SudokuPlayActivity: onResume
cz.romario.opensudoku.gui.SudokuPlayActivity:
0
200
400
600
800

EMMA Coverage Report (generated Thu Aug 01 00:24:50 HKT 2013)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	58% (69/118)	56% (388/690)	47% (9033/19037)	49% (1876.1/3813)

OVERALL STATS SUMMARY

total packages: 8
total executable files: 54
total classes: 118
total methods: 690
total executable lines: 3813

Line, %
49% (1876.1/3813)

COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
cz.romario.opensudoku.gui.exporting	0% (0/4)	0% (0/11)	0% (0/444)	0% (0/90)
cz.romario.opensudoku.gui.importing	0% (0/4)	0% (0/23)	0% (0/771)	0% (0/189)
cz.romario.opensudoku.utils	33% (1/3)	25% (2/8)	21% (25/119)	20% (6/30)
cz.romario.opensudoku.db	33% (2/6)	27% (9/33)	24% (409/1688)	29% (89.9/311)
cz.romario.opensudoku.game.command	56% (5/9)	51% (24/47)	32% (287/898)	35% (77/220)
cz.romario.opensudoku.gui	47% (28/59)	44% (135/305)	44% (4211/9540)	43% (835.5/1928)
cz.romario.opensudoku.game	100% (6/6)	73% (74/101)	53% (1308/2470)	72% (314.8/435)
cz.romario.opensudoku.gui.inputmethod	100% (27/27)	89% (144/162)	90% (2793/3107)	91% (552.8/610)

Analysis

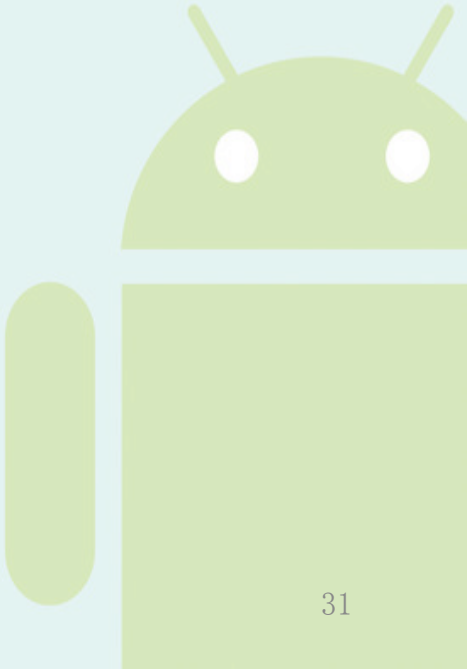
- Omnidroid:
  - 9 methods spend near or more than 200ms to finish.
  - Most time consuming method:OnCreate in ActivityDlgFilterInput needs on average 700 ms.
  - All recorded methods run in one thread

Analysis

- OpenSudoku:
  - 4 methods spend near or more than 200ms to finish.
  - Most time consuming method:OnCreate in FolderListActivity needs more than 700 ms.
  - All recorded methods run in one thread
- User may fail to do any operation as lifecycle or GUI events during that 700ms.

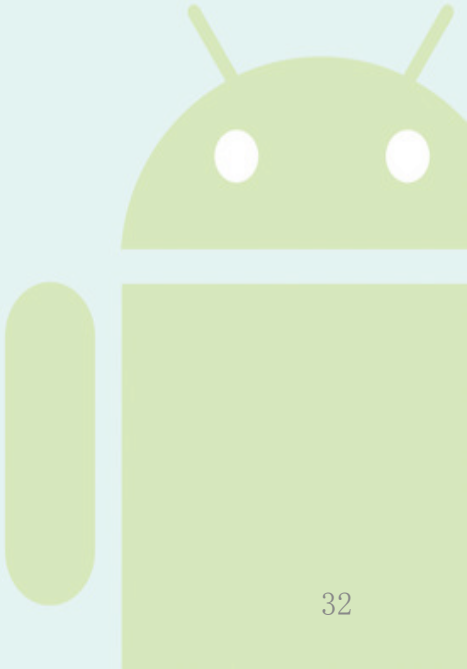
Threats to validity

- Emulator and real device
- Universality(Sample is too small)
- Overhead of instrumented code



Difficulties

- Automatic Instrumentation
- Random testing
- Subject building

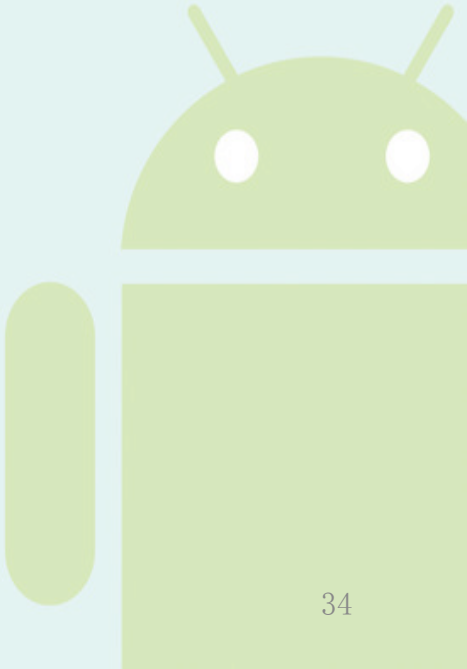


Conclusion

- Component lifecycle handlers and GUI event handlers that require near or more than 200ms are successfully detected in every application in the project.
- The most time consuming handler in every application are identified.
- The applications suffer from potential threat of slowness in those methods.

Conclusion

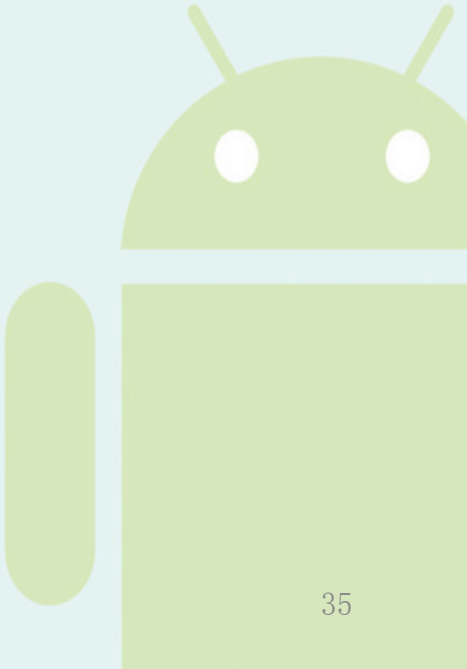
- The onCreate method has a higher possibility to be most time consuming.



Improvement on Random Testing?

- Discuss how statement coverage increases with the number of generated events.
  - Random testing difficult to achieve high coverage

    Targeted event sequence generation to touch certain code
- Interpret running time data



Thank you.