Android Workload Suite (AWS): Measure the software stack of mobile devices

Xiao-Feng Li
xiaofeng.li@gmail.com
Oct, 2011

Thanks to Greg Zhu and Ke Chen
Summary

• **Android Workload Suite (AWS)** is an engineering tool for Android software stack measurement
  – It uses the software stack metrics to measure the interaction scenarios

• **AWS** covers the major areas for Android software stack evaluation
  – The key is to map user interactions to system behavior
Agenda

• **User interactions measurement**
• Interaction scenarios definition
  – Case studies
• Android workloads construction
  – Case studies
• Summary
• Information
Optimize User Interaction Systematically

• What we need:
  – A well-established methodology
  – An engineering workload suite
  – An analysis/tuning toolkit
  – Sightings/requests/feedbacks from PECA/IXR, xPGs, developers, users, etc.

• (The methodology details are in another deck)
• (The UXtune toolkit details are in another deck)
User Interactions with Client Device

- A sequence of interactions

- Does the input trigger the target correctly?
- Does the system act responsively?
- Does the graphics transition smoothly?
- Does the object move coherently?
Interaction Measurement Aspects

• User controls device (subject → object)
  1. Accuracy/fuzziness: Range/resolution of inputs that can trigger a correct response
  2. Coherence: Object move delay, difference in move trajectory

• Device reacts to user (object → subject)
  3. Responsiveness: Time between an input delivered to the device response, and to the action finish
  4. Smoothness: Maximal frame time, frame time variance, FPS and frame drop rate
Android Workload Suite (AWS)

• **Goals**
  – Reflect the representative usage of Android client devices
  – Evaluate Performance, Power and User interactions

• **AWS usages**
  – Drive and validate Android optimizations
  – Support comparative and competitive analysis
<table>
<thead>
<tr>
<th>Suite</th>
<th>Workload</th>
<th>#Scenarios</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Media</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Touch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Built-in apps</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task management</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agenda

• User interactions measurement
• **Interaction scenarios definition**
  – Case studies
• Android workloads construction
  – Case studies
• Summary
Understand The Representative Scenarios

• Extensive surveys
  – Feedbacks/inputs from users
  – Public documents from key players
  – Popular applications
  – Form-factor usages (Tablet vs. smart-phone)
  – User interaction life-cycles and software design
# Usage Categories: Market and Built-in Apps

<table>
<thead>
<tr>
<th>Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business &amp; Productivity</td>
<td>Office, Video conference, Payment, LBS, Security...</td>
</tr>
<tr>
<td>Information &amp; Content</td>
<td>Internet access, Video, Music, Gaming, eBooks...</td>
</tr>
<tr>
<td>Communication</td>
<td>Phone, Contacts, SMS, MMS, E-mail, IM, Video phone...</td>
</tr>
<tr>
<td>Basic accessibility</td>
<td>Home screen, App launcher, Setting, Touch, Sensor...</td>
</tr>
</tbody>
</table>
Tablet-specific Apps Characteristics

• Larger screen size than phone
  – More realistic view experience (game, cartoon, 3D)
  – Easier or more controls through touch/sensors or virtual controllers (virtual controller, editor, handwriting)
  – Bigger space to put more contents (news, education, ebook)
  – Support more than one players (game, education)
  – PC-experience web access (browser, info portal)
  – More small utilities apps for daily use (on-screen vs. in-pocket)
Phone-specific Apps Characteristics

• **Phone as handy gadget as a Swiss-knife**
  – Communicator (chat through AV/text/picture)
  – Camera (barcode scanner and photo/video apps)
  – Utility (flashlight, night vision, barcode scanner)
  – Navigation (GPS, compass), music player, Phone

• **Smaller size**
  – Games are cartoon or lightweight-animation based
  – Relatively simple games with simple sensor controls
  – Many accelerometer-based games
    • Shake to operate (vs. gyroscope-based with Tablet)
Form Factor Consideration in Workload Design

• Some scenarios in AWS may only exist in one form factor, e.g.,
  – Status bar vs. system bar
  – Browser: switch window vs. switch tab

• Same scenario in AWS may have two design variants, e.g.,
  – The 2D game workload has more animated sprites in its tablet profile
  – Browser workload use PC web page on tablet, and _can_ use mobile web page on phone
User Scenario Categories

• **User operations**
  – Browsing, gaming, authoring, setting/configuring
    • Touch gestures, and sensors
  – Communications

• **Loading and rendering**
  – Loading:
    • Web page, eBook, image
  – Rendering:
    • Web page, HTML5, eBook, media, 2D/3D

• **Task management**
  – App launch, Task switch
  – Multi tasking (Parallel execution)
Primary Metrics for User Scenarios

• **User operations**
  – Browsing, gaming, authoring, setting, communication
  – **Responsiveness, smoothness, coherency, accuracy**

• **Loading and rendering**
  – Web/HTML5, eBook, media, image, 2D/3D
  – **Responsiveness** (loading time, rendering capability), smoothness, coherency, accuracy

• **Task management**
  – App launch, Task switch, Multi tasking
  – **Responsiveness** (time to launch/exit), smoothness, coherency, accuracy
Agenda

• User interactions measurement
• Interaction scenarios definition
  – Case studies
• Android workloads construction
  – Case studies
• Summary
Example of Interaction Lifecycle - Browser

- User operations
- Loading and rendering
- Task management

User interaction lifecycle is composed with three types of scenarios:

1. Launch browser (loading time)
2. Input URL (responsiveness)
3. Webpage loading (loading time)
4. Read webpage
5. Scroll/Fling/Zoom webpage (responsiveness, smoothness)
6. Exit browser (loading time)

- Open new tab (responsiveness)
- Switch tab (responsiveness)
Example of Interaction Lifecycle - Video Player

- Touch thumbnail to Play (startup time)
- Seek forward/backward while playing (seek response time)
- Exit player (unloading time)

- Normal playback (Smoothness, dropped frames)
- Pause/Resume (resume response time)
- Play next video clip (switch response time)
Agenda

• User interactions measurement
• Interaction scenarios definition
  – Case studies
• Android workloads construction
  – Case studies
• Summary
Interaction Measurement Criteria

- Measure the critical path of user interactions in software stack

- Criteria
  - **Perceivable** (PECA/IXR has the UX perceptual model)
  - **Measureable** (by different teams)
  - **Repeatable** (in multiple measurements)
  - **Comparable** (between different measured systems)
  - **Reasonable** (about the causality)
  - **Verifiable** (for an optimization)
  - **Automatable** (largely unattended, not strictly)
Workloads Construction

• Key is to map user interactions to system behavior
  – Purpose is to assist software optimization instead of simulating user behavior

• Kinds of workloads
  – **Standalone workload**: Run as full workload and give results
  – **Micro workload**: Stress certain execution paths of the stack
  – **Measurement tool**: Allow manual operation and get metrics
  – **Scenario driver of built-in app**: only give inputs and extract metrics
Kinds of Workloads

1. Standalone workload
   - Input → Activity 1 → Service 1 → Service 2 → Activity 2 → Output

2. Micro workload
   - Input → Activity 1 → Service 1 → Service 2 → Activity 2 → Output

3. Measurement tool
   - Input → Activity 1 → Service 1 → Service 2 → Activity 2 → Output

4. Scenario driver
   - Input → Activity 1 → Service 1 → Service 2 → Activity 2 → Output

2011-11-23
Challenges in Workload Construction

• How to measure response time of user inputs?
• How to measure smoothness?
• How to measure drag coherence?
• How to make the results repeatable?
• How to make the workload comparable across platforms?
• Etc.
Challenge 1: Response Time Measurement

- **Software latency is our optimization focus**
  - Software latency is around x100ms
  - Touch sampling rate is typically 200HZ (5ms interval)
Challenge 2: Smoothness Measurement

Notice the followings:
- Max frame time
- #frames > 30ms
- Frame time variance
- FPS
Challenge 3: Drag Coherence Measurement

Distances\[k\] = \{Touch\[i\].pos – Draw\[k\].pos \mid Touch\[i\].t\leq Draw[k+1].t AND Touch\[i\].t>Draw[k].t\}

Coherency = \text{Max}(\{\text{Max}(\text{Distances}[k]) \mid k=0,\ldots,N\})
Challenge4: Repeatable Results

- Use Input-Gesture tool to generate standard touch gestures for inputs
- Ensure the generated gestures are comparable across different platforms

Events of same gesture on Device X

- 1000000000 3 48 1
- 1000000010 3 53 3284
- 1000000020 3 54 2747
- 1000000030 0 2 0
- 1000000040 0 0 0
- 1000005000 3 48 1
- 1000005010 3 53 3284
- 1000005020 3 54 2735

Events of same gesture on Device Y

- 1000000000 3 48 1
- 1000000010 3 53 1810
- 1000000020 3 54 1515
- 1000000030 0 2 0
- 1000000040 0 0 0
- 1000005000 3 48 1
- 1000005010 3 53 1810
- 1000005020 3 54 1508

2011-11-23
Challenge 5: Comparable Across Platforms

• For example, browser workloads
  – Different platforms may have different built-in browsers

• Depending on the measurement purpose
  – If for rendering engine comparison, use standard contents (web pages or Javascripts)
  – If for app operation comparison, use “scenario driver” generated by input-Gestures
  – If for framework comparison, build a “standalone browser” and install to target platforms
Agenda

• User interactions measurement

• Interaction scenarios definition
  – Case studies

• Android workloads construction
  – Case studies

• Summary
Workload Construction Case Studies

- Browser scroll scenario
Browser Scroll Scenario

1. Finger starts

2. Content starts to move

3. Finger moves, content moves

4. Finger releases
Measurement for Scroll

• **Response time**
  – How fast the content start to follow the finger

• **Drag lag distance**
  – How far the content movement lags behind finger

• **Smoothness**
  – How smooth the browser animates the scroll
Software Stack Internals in Scroll

Input raw events

Event1 ➔ ... ➔ EventM ➔ EventN ➔ ... ➔ EventX ➔ EventY

Browser events

ACTION DOWN ➔ ACTION MOVE ➔ ACTION MOVE ➔ ACTION MOVE ➔ ...

Browser drawing

Frame1 ➔ ...

Detects Scroll Gesture
Response Time Measurement

Input raw events

Event1 → ... → EventM → EventN → ... → EventX → EventY

Browser events

ACTION DOWN → ACTION MOVE → ACTION MOVE → ACTION MOVE → ACTIONS

Browser drawing

Frame1 → ... → Detects Scroll Gesture

Response Time

First event send time

First frame drawn time

$\Delta x^2 + \Delta y^2 > mTouchSlotSquare$ (\(\Delta x, \Delta y\): offset from ACTION_DOWN)
Smoothness Measurement

Input raw events

... → EventX → EventY → EventZ

Browser events

... → ACTION MOVE → ACTION MOVE → ACTION UP

Browser drawing

First Frame → ... → Frame m → Frame n → ... → Last frame

Time

T1

T2
Drag Lag Measurement

Distances[k] = \{Touch[i].pos – Draw[k].pos | Touch[i].t<=Draw[k+1].t AND Touch[i].t>Draw[k].t\}

Coherency = Max(\{Max(Distances[k]) | k=0,...,N\})
Results Repeatability

• Standard scroll gesture set generated by the Input-Gestures tool
  – Scroll up 20 times, down 20 times
  – Events are transformed for different devices
Workload Usage

- Support built-in and self-built browser
- Support scenario selection
- Support user input webpage address
Detailed Results Archive

- Result Files - /data/local/tmp/XXX_result.txt
  - Record data of each gesture
- Frame interval, maximum LTF, #LTFs

--- Workload Result of Scenario Scroll ---

Frame Intervals:
0  77  85  79  79  77  74  76  74  72  74  73  108  74  72  74  72  71  71  69  72  72  73  70  69  69  69  69  68  66  68  70  68  65
Response Time: 130
Average FPS: 13.66120218578235
Number of Long Time Frames: 35
Longest Time Frame: 108

Frame Intervals:
0  61  60  60  59  60  60  59  60  61  59  60  60  59  61  61  60  60  60  64
Response Time: 140
Average FPS: 16.605166051660516
Number of Long Time Frames: 18
Longest Time Frame: 64

Frame Intervals:
0  58  58  60  60  61  62  62  64  63  64  65  65  64  64  64  64  64  64  64
Response Time: 130
Average FPS: 15.908312866076174
Number of Long Time Frames: 18
Longest Time Frame: 66

Frame Intervals:
0  60  58  60  60  59  60  59  59  58  60  58  60  59  59  59  59  59  59  59  59  59
Response Time: 130
Average FPS: 16.90391439074733
Number of Long Time Frames: 19
Longest Time Frame: 60
Agenda

• User interactions measurement
• Interaction scenarios definition
  – Case studies
• Android workloads construction
  – Case studies
• Summary
Summary

• **Android Workload Suite (AWS)** is an engineering tool for Android software stack measurement
  – It uses the software stack metrics to measure the interaction scenarios

• **AWS covers the major areas for Android software stack evaluation**
  – The key is to map user interactions to system behavior