Android Application Usage Profiling

- Profiling an Android Application
 - **Gathering Memory Usage Statistics**
 - Log Messages
 - Performance Monitors
 - Data Analysis
 - System Information

Profiling an Android Application

To gather application running statistics on the system, you can profile the system using Android Studio Performance Profiling tools https://develop er.android.com/studio/profile/index.html.

The IDE provides a set of different tools which can effectively be used to profile any application running on a device or an emulator. To profile the memory usage, CPU/GPU usage or Network bandwidth usage, we can use the Android Monitor https://developer.android.com/studio/profile/andr oid-monitor.html.

The Android Monitor helps us in these tasks:

- Log messages, which can be either system or hardware defined.
- Monitor memory, CPU and GPU usage by the application.
- Network bandwidth usage statistics.

To enable Android Monitor you need to take care of some prerequisites and dependencies as specified.

- 1. The device should remain connected to the system via USB cable and the system should be able to detect the device.
- 2. Enable ADB integration by selecting Tools > Android > Enable ADB Integration. Enable ADB Integration should have a check mark next to it in the menu to indicate it's enabled.
- 3. Make sure Android Device Monitor is not running currently.
- 4. In your app, set the "debuggable" property to "true" in the manifest or "build.gradle" file (it's initially set by default).

Now we just need to display the Android Monitor and when we run the application on the device, we can see its statistics be noted on the corresponding screen for "log messages (logcat)" or "performance monitors (monitors)". To view the Android Monitor, you can select it from View > Tool Windows > Android Monitor. Or you can also display Android Monitor by clicking on the Android Monitor button, which is on the bottom of the main window by default or using the shortcut Alt+6. A screenshot of an app being profiled is shown below.



Gathering Memory Usage Statistics

Android Monitor provides us with various tools to check the usage statistics of the android application. Here we will list down some tools and try to understand how these help in understanding the memory usage of the application.

Log Messages

The "logcat" monitor gives us the log messages while the application is deployed and running in real time. It helps us in understanding what is

happening in the system which is very useful for debugging. For example if an application crashes the best way to find out the cause of the crash is to check the last few commands of the logcat. The screenshot below shows the logcat monitor while an app is running.

Android Monitor 🕸 🛓										
Samsung SM-G935F Android 7.0, API 24 🔻	facebook. f8demo (31408) [DEAD] 🔻									
I II¥ logcat Monitors →*	Verbose 🔻 🔍	🗹 Regex	Show only selected application 🔻							
 75 31408-31437/facebook.f8demo E/F8DEM:c 43 31408-31437/facebook.f8demo E/F8DEM:c 34 31408-31437/facebook.f8demo E/F8DEM:c 34 31408-31488/facebook.f8demo D/ViewRootI 44 31408-31408/facebook.f8demo D/ViewRootI 45 31408-31408/facebook.f8demo D/ViewRootI 31408-31408/facebook.f8demo D/ViewRootI 31408-31408/facebook.f8demo D/ViewRootI 31408-31408/facebook.f8demo D/InputTran 13 31408-31491/facebook.f8demo D/InputTran 13 1408-31491/facebook.f8demo D/NemofiLemo 13 1408-31491/facebook.f8demo D/NemofiLemo 31408-31491/facebook.f8demo D/Nali_vins 31408-31437/facebook.f8demo A/native: [33 31408-31437/facebook.f8demo A/libc: Fat 31408-31437/facebook.f8demo A/libc: Fat 	ne. istantiating predictor pl@d7433d6[ClassifyCamera]: MSG_RESIZED_REPORT: ci=Rect(0, 0 - 0, 0) vi=l pl@d7433d6[ClassifyCamera]: MSG_WINOW_FOCUS_CHANGED 1 dManager: Starting input: tba=android.view.inputmethod.EditorInfo@920154. bManager: [IMM] startInputInner - mService.startInputOrWindowGainedFocus iport: Input channel constructed: fd=88 lerer: Initialized EGL, version 1.4 lerer: Swap behavior 1 : STS GLApi : DTS is not allowed for Package : facebook.f8demo /s: EGLint new_window_surface(egl_winsys_display*, void*, EGLSurface, EGL/ gerGlobal: Connecting to camera service given_tensor_fill_op.h:27] Check failed: output->size() == valuessize	=Rect(0, 0 - 0, 0) or=1 44 nm : facebook.f8demo ic=n s LConfig, egl_winsys_surface*	**, egl_color_buffer_format*, EGLBoole							

Performance Monitors

There are four types of performance monitors which helps us monitor the performance of the app in real time. We can get the statistics for Memory Usage, CPU Usage, Network Usage and GPU Usage with the help of this monitor. They give us tools to record specific additional data in files which can be analysed by using specific analysis tools. The screenshot below shows the performance monitor with each of the four resources being monitored in real time.



Data Analysis

System Information

In the Android Monitor main window, we can find the System Information icon on the side as marked in the screenshot below.



On clicking the icon you will get a list off system information that you can access. These are the various outputs of the "dumpsys" command.

- · Activity Manager State dumpsys activity
- Package Information dumpsys package
- Memory Usage dumpsys meminfo

• A screenshot of the memory usage statistics is shown below. It gives us most of the information we need to know about the amount of memory being used by the application.

Applications Memory Usage (in Kilobytes): 4** MEMLMG0 in pid 15667 (org.teamor/law.dem) ** Friate Dirty Clean Dirty Size Alloc Free Total Dirty Clean Dirty Size Alloc Free Total Dirty Clean Dirty Size Alloc Free Total Dirty Clean Dirty Size Alloc Free Dalvik Whep 15353 105348 28 94 117769 100077 17682 Dalvik Whep 5353 05368 08 94 117769 100077 17682 Dalvik Whep 535 0586 0 94 0 Stack 424 424 0 0 Other dev 5 9 8 764 0 Stack 598 064 0 Stack 598 064 0 Stack 598 064 0 Stack 598 064 0 Stack 3384 33944 0 Other dev 598 0884 0 Other dev 598 0884 0 Other dev 598 064 0 Stack 3384 33944 0 Other dev 598 064 0 Stack 598 064 0 Stack 598 064 0 Stack 598 064 0 Stack 598 064 0 Other dev 598 064 0 Stack 598 064 0 Other dev 598 064 0 Stack 598 0 Stack	CMa	akeLists.txt ×	📀 andro	oid × 🖹	org.tenso	orflow.der	no_2017.	06.19_11.	19.txt ×	💁 Android Manifest.xr	ml × 📓 local.properties ×	🖟 gradle-wrapper.properties × 📀
** MEMINE in pid 15007 [org. tensorflow.demo] **	2											
Native Heap 105455 105455 105455 117767 106017 1766 Dalvik Heap 65201 65168 0 82 17767 10661 7106 Dalvik Heap 105455 105344 424 0 0 100077 17662 Stack 424 424 0 0 0 0 0 0 Ashmem 4 4 0 0 0 0 0 0 Ashmem 4 4 0 0 0 0 0 0 Ashmem 4 4 0 0 0 0 0 0 Ashmem 4 4 0 0 0 0 0 0 Ashmem 4 0 0 0 0 0 0 0 Ashmem 4 0 0 0 0 0 0 0 0 Ashmem 4 0 0 0 0 0 0 0 0 Other map 318 4 0 0 117767 110738 24788 0 0	4 5	** MEMINFO in	Pss	Private	Private	SwapPss						
Dalvik Heap 65201 65168 0 82 1777 1061 7106 11 Stack 424 424 0												
11 Stack 424 424 0 0 13 Other dev 5 0 4 0 13 Other dev 5 0 4 0 15 pk mmp 345 0 0 0 16 pk mmp 345 0 0 0 17 ds mmp 544 590 64 0 18 ds mmp 554 590 64 0 19 dt mmp 138 888 76 9 20 Other mmp 18 4 0 0 21 EGL mtrack 3834 3834 0 0 22 Unknown 379 376 0 18 23 TOTAL 226333 212256 864 247 135527 110738 24788 24 dtrine dtrine dtrine dtrine dtrine dtrine 25 dtrine dtrine dtrine dtrine dtrine 24 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>												
12 Ashmem 4 4 0 0 14 .so mmap 10250 388 764 44 14 .so mmap 10250 388 764 44 15 .apk mmap 34 0 0 16 .ttf mmap 54 0 24 0 17 .dex mmap 528 0 84 0 18 .oat mmap 128 0 0 0 18 .oat map 2928 0 804 0 18 .oat map 328 808 76 9 18 .oat map 339 306 0 1 19 .ntrack 3394 306 0 1 10 Intrack 3394 306 1 1 1001 TOTAL 22633 212256 6664 1 1 11 TOTAL 22633 10534 1 1 1 12 Stack: 424 1 1 1 13 TotAL: 2633 Activitis: </td <td></td>												
13 Other dev 5 0 4 0 15 apk map 345 0 0 0 15 apk map 345 0 0 0 17 apk map 54 0 24 0 17 dex map 564 500 64 0 18 art map 1358 888 76 9 20 Other map 18 4 0 0 21 EGL mtrack 3334 0 0 22 Unknown 379 37 0 18 23 TOTAL 226333 212256 8684 247 135527 110738 24788 24 Differing art map 15534 art map 15534 art map												
14 .50 mmap 10250 388 764 44 16 so mmap 10250 388 764 4 16 so mmap 54 0 0 0 16 so mmap 54 0 24 0 17 so mmap 1358 888 76 9 20 0ther mmap 18 4 0 0 21 0so mmap 1358 888 76 9 20 0ther mmap 18 4 0 0 21 Unknown 379 376 0 18 22 Unknown 379 376 0 18 23 TOTAL 226333 212256 8684 247 135527 24 0 0 0 0 0 0 24 0 13527 110738 24788 25 0 58 66132 247 135527 26 Frivate 16142 25334 24788												
15 .apk map 345 0 0 16 .ift map 54 0 24 0 17 .dex map 564 500 64 0 18 .oat map 564 500 64 0 19 .art map 1358 888 76 9 0 Other map 138 4 0 0 21 Other map 138 4 0 0 22 Introve 379 376 0 18 23 TOTAL 226333 212256 8684 247 135527 24 Dava Heap: 105122												
17 .dex mmap 564 500 64 0 19 .art mmap 1358 888 76 9 0 0 ther mmap 138 4 0 0 21 0 ther mmap 38344 38394 0 0 22 0 ther mmap 373 37 0 18 23 10TAL 226333 212256 8684 247 135527 24 10TAL 226333 212256 8684 247 135527 110738 24788 25 App Summary 26 Java Heap: 16532 27 Sistem: 5383 38 Origits 9 ViewRootImpl: 1 39 Views: 9 ViewRootImpl: 1 40 AppContexts: 3 Activities: 1 41 Assets: 3 Activities: 1 1												
18 .oat mmap 228 0 804 0 19 .art mmap 1358 888 76 9 10 Other mmap 18 4 0 0 11 EGL attrack 38344 3834 0 0 12 Unknown 379 376 0 18 12 TOTAL 226333 212256 8684 247 135527 13 App Summary												
19 .art mmap 158 888 76 9 21 Cher mmap 18 4 0 0 21 Columnary 38304 38304 0 0 22 TOTAL 226333 212256 8684 247 135527 110738 24788 24 Total 226333 212256 8684 247 135527 110738 24788 24 Total 226333 212256 8684 247 135527 110738 24788 25 Java Heap: 66132 5384 5384 5384 5384 33 Private Other: 1232 5384 547 5384 33 Private Other: 1232 5354 547 33 Private Other: 1232 5354 547 343 System: 5393 107AL SWAP PSS: 247 344 Assets: 3 AssetManagers: 2 455 ViewS: 9 ViewRootImpl: 1 464 Assets: 3 AssetManagers: 2												
20 Other map: 18 4 0 0 21 EGL mtrack \$3834 3834 3834 0 0 22 Unknown 379 376 0 18 23 TOTAL 226333 212256 8684 247 135527 110738 24788 24 App Summary												
21 GGL mtrack 38394 9 0 22 Unknown 379 376 0 18 24												
22 Unknown 379 376 0 18 23 OTAL 226333 212256 8684 247 135527 110738 24788 25 App Summary												
App Summary Private Page: 66132 Native Heap: 105384 Stack: 424 Graphics: 38364 Private Other: 1232 System: 5393 Objects 0 AppContexts: 3 AppContexts: 4 AppContexts: 0 Parcel memory: 5 Parcel memory: 5 Parcel memory: 6				376	0	18						
App Summary 26 Pss(KB) 27 28 Java Heap: 66132 29 Native Heap: 105384 20 Code: 9464 31 Stack: 424 32 Graphics: 38304 33 Private Other: 1232 34 System: 5393 35 TOTAL: 226333 36 TOTAL: 226333 37 Objects 38 Piexes: 9 39 Views: 9 20 Assets: 3 30 Assets: 3 31 Assets: 3 32 Objects 33 Parcel memory: 5 34 Parcel memory: 5 35 OpenstS Sockets: 0 43 Parcel memory: 5 44 Death Recipients: 20 45 MEMORY_USED: 0 46 MEMORY_USED: 0 47 MEMORY_USED: 0 48 MEMORY_USED: 0		TOTAL	226333	212256	8684	247	135527	110738	24788			
26 Java Heap: 66132 27												
22		App Summary										
Bava Heap: 66132 Pava Heap: 105384 Code: 9464 Stack: 424 Graphics: 38044 Private Other: 1232 System: 5393 TOTAL: 226333 TOTAL SWAP PSS: 247 Objects 0 Views: 9 ViewRootImpl: 1 AppContexts: 3 Activities: 1 AppContexts: 3 Assetts: 20 Parcel memory: 5 Parcel count: 21 Death Recipients: 2 OpenSSL Sockets: 0 MEMORY USED: 0 MALLOC_SIZE: 0												
38 Code: 9464 31 Stack: 424 32 Graphics: 38304 33 Private Other: 1232 34 System: 5393 35 TOTAL: 226333 36 TOTAL: 226333 37 Objects 0 38 Objects: 3 39 Views: 9 ViewRootImpl: 40 AppContexts: 3 41 Assets: 3 42 Local Binders: 11 43 Parcel memory: 5 44 Death Recipients: 2 45 OpenSSL Sockets: 0 46 SOL 4 47 MEMORY USED: 0 48 PAGECACHE_OVERFLOW: 0		Jav	/a Heap:									
31 Stack: 4.24 32 Graphics: 38304 33 Private Other: 1232 34 System: 5393 35 TOTAL: 226333 36 TOTAL: 226333 37 TOTAL: 226333 38 Objects 39 Views: 9 40 AppContexts: 3 41 Assets: 3 42 Local Binders: 11 43 Parcel memory: 15 44 Death Recipients: 2 45 46 50L 47 48 49		Nati										
32 Graphics: 38304 33 Private Other: 1232 34 System: 5393 35 Jornal: 226333 36 TOTAL: 226333 37 TOTAL: 226333 38 Objects 39 Views: 9 40 AppContexts: 3 41 Assets: 3 42 Local Binders: 11 43 Parcel memory: 5 44 Death Recipients: 2 45 OpenSSL Sockets: 0 46 SOL 47 MEMORY USED: 0 48 PAGECACHE_OVERFLOW: 0 MALLOC_SIZE: 0												
33 Private other: 1232 34 System: 5393 35 TOTAL: 226333 TOTAL SWAP PSS: 247 36 TOTAL: 226333 TOTAL SWAP PSS: 247 37 Objects 0 0 Activities: 1 38 Objects 3 Activities: 1 40 AppContexts: 3 Activities: 1 41 Assets: 3 AssetManagers: 20 42 Local Binders: 11 Proxy Binders: 20 43 Parcet memory: 5 Parcet count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 45 6 6 6 6 46 AppCeCACHE_OVERFLOW: 0 MALLOC_SIZE: 0												
34 System: 5393 35 TOTAL: 226333 TOTAL SWAP PSS: 247 36 TOTAL: 226333 TOTAL SWAP PSS: 247 37												
35 TOTAL: 226333 TOTAL SWAP PSS: 247 36 Objects 37 38 Objects 39 Views: 9 ViewRootImpl: 1 40 AppContexts: 3 Activities: 1 41 Assets: 3 AssetManagers: 2 42 Local Binders: 11 Proxy Binders: 20 43 Parcel memory: 5 Parcel count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 45 Total Sockets: 0 46 MEMORY_USED: 0 47 MEMORY_USED: 0 49 MALLOC_SIZE: 0		1111000										
37 Objects 38 Objects 39 Views: 9 ViewRootImpl: 1 40 AppContexts: 3 Activities: 1 41 Assets: 3 Assets: 2 42 Local Binders: 11 Proxy Binders: 20 43 Parcel memory: 5 Parcel count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 45 SOL 47 MEMORY USED: 0 48 PAGECACHE_OVERFLOW: 0 MALLOC_SIZE: 0	35		-		τοτο		. 2	47				
38 Objects 39 Views: 9 ViewRootImpl: 1 40 AppContexts: 3 Activities: 1 41 Assets: 3 AssetManagers: 2 42 Local Binders: 11 Proxy Binders: 20 43 Parcel memory: 5 Parcel count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 45			. OTAL.	220333	1014	C 30AC 733	. 2					
40 AppContexts: 3 Activities: 1 41 Asset: 3 AssetManagers: 2 42 Local Binders: 1 Proxy Binders: 20 43 Parcel memory: 5 Parcel count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 45 SOL	38	Objects										
41 Assets: 3 AssetManagers: 2 42 Local Binders: 11 Proxy Binders: 20 43 Parcel memory: 5 Parcel count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 45												
42 Local Binders: 11 Proxy Binders: 20 43 Parcel memory: 5 Parcel count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 45 SQL 4 PAGECACHE_OVERFLOW: 0 MALLOC_SIZE: 0		АррСо										
43 Parcel memory: 5 Parcel count: 21 44 Death Recipients: 2 OpenSSL Sockets: 0 46 SQL 47 MEMORY USED: 0 48 PAGECACHE_OVERFLOW: 0 MALLOC_SIZE: 0		Local H										
44 Death Recipients: 2 OpenSSL Sockets: θ 45 45 46 SOL 47 MEMORY_USED: θ 48 PAGECACHE_OVERFLOW: θ 49 MALLOC_SIZE: θ												
46 SQL 47 MEMORY_USED: θ 48 PAGECACHE_OVERFLOW: θ MALLOC_SIZE: θ 49												
47 ΜΕΝΟRY_USED: θ 48 PAGECACHE_OVERFLOW: θ MALLOC_SIZE: θ 49												
48 PAGECACHE_OVERFLOW: θ MALLOC_SIZE: θ												
49					м			0				
		CAUCCACHE_0	ER LON.	9	n	ACCOC_5120						

- Memory Usage over Time dumpsys procstats
- Graphics State dumpsys gfxinfo

Some of the other Data Analysis tools which can be viewed using Android Monitor are as listed.

1. HPROF Viewer and Analyzer: This tools dumps the Java Heap to an HPROF file using the Memory Monitor. It's icon can be found, as one of the three icons, on the memory monitor just beside the play/pause button. The output of the dump can be viewed in the HPROF

- Viewer which gives a nice class hierarchical view of the heap dump.
 Allocation Tracker: This tool can also be found on the Memory Monitor which is used to track down all the memory allocation data in the app. It displays each method responsible for the allocation along with the size and the number of instances.
 Method Trace
- 4. GPU Debugger