

Fair Performance Analysis (FPA) - Linux Vs Android

Motivation:

An FPA between Linux and Android was needed to address the below items,

- Which OS better in Performance (CPU, Memory, Responsiveness, Statistics etc.) ?
- Which OS can suit our requirements better?
- Have credible data to prove the decision.

What is Fair Performance Analysis (FPA):

An FPA is an analysis performed/measured/debugged using a fair environment to both candidates Linux & Android.

There are numerous tools / packages / frameworks available to measure performance (ANtutu, GpuPrime, nvidia-smi etc) . Although, very limited amount of them work for both and also behave the same for both.

There are various use cases which can be used for analysis although we choose a fairly common and simple use case that's predictable, measurable and stable for both candidates.

What FPA is used for Linux Vs Android:

The below elements form the common foundation of the FPA for both candidates.

Element	Linux	Android
Hardware - SOC	Raspberry Pi 3B	Raspberry Pi 3B
OS Name	Raspbian Buster	Android Lineage OS 16
Linux Kernel	4.19.58-v7l+	4.19.102-v7
Total CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU	< - - Same
Total RAM	1GB RAM	< - - Same
Display	1 HDMI - HD 1280 x 720	< - - Same
UC1: Use Case Boot	Boot Time from Power on to Full Desktop Access	< - - Same
UC2: After Boot Statistics	CPU, Memory load after full boot	< - - Same
UC3 : Use Case Application	Launch and app that can load a compressed image from disk and render it to the display (fullscreen). The app is written in Java (Android), Flutter (Android) and Qt (Linux)	< - - Same
UC4: Time taken for UC3	Launch UC1 and measure time elapsed from launch to full execution.	< - - Same
UC5: App Execution Statistics	CPU, Memory load , SystemCall Context Switches during UC3	< - - Same

FPA Results

UC1: Use Case Boot :

	Linux	Android
Boot Time	~ 20 Seconds	~ 70 Seconds

UC2: After Boot Statistics

Conditions:

Wifi Driver - OFF

Default Apps - NONE

Network - ETH ONLY

	Linux	Android
CPU Usage Avg after Boot	2 %	3 %
RAM Usage Avg after Boot	128 MiB	128 MiB
Default Process Count	130	168

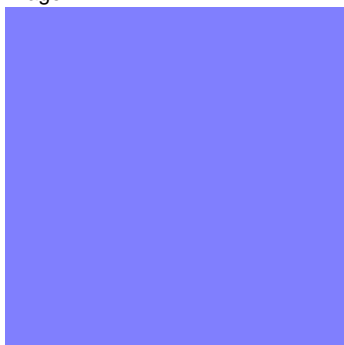
UC3 : Use Case Application

Linux Qt App : <https://github.com/bensinghbeno/design-engine/tree/master/projects/qt/imagedisplay>

Android Java App : https://github.com/bensinghbeno/design-engine/tree/master/projects/android/native_ImageDisplay

Android Flutter App : https://github.com/bensinghbeno/design-engine/tree/master/projects/android/flutter_ImageDisplay

Image :



UC4: Time taken for UC3

	Linux - Qt App	Android - Java App	Android - Flutter App
Time for Full launch	2 Seconds	4 Seconds	5 Seconds

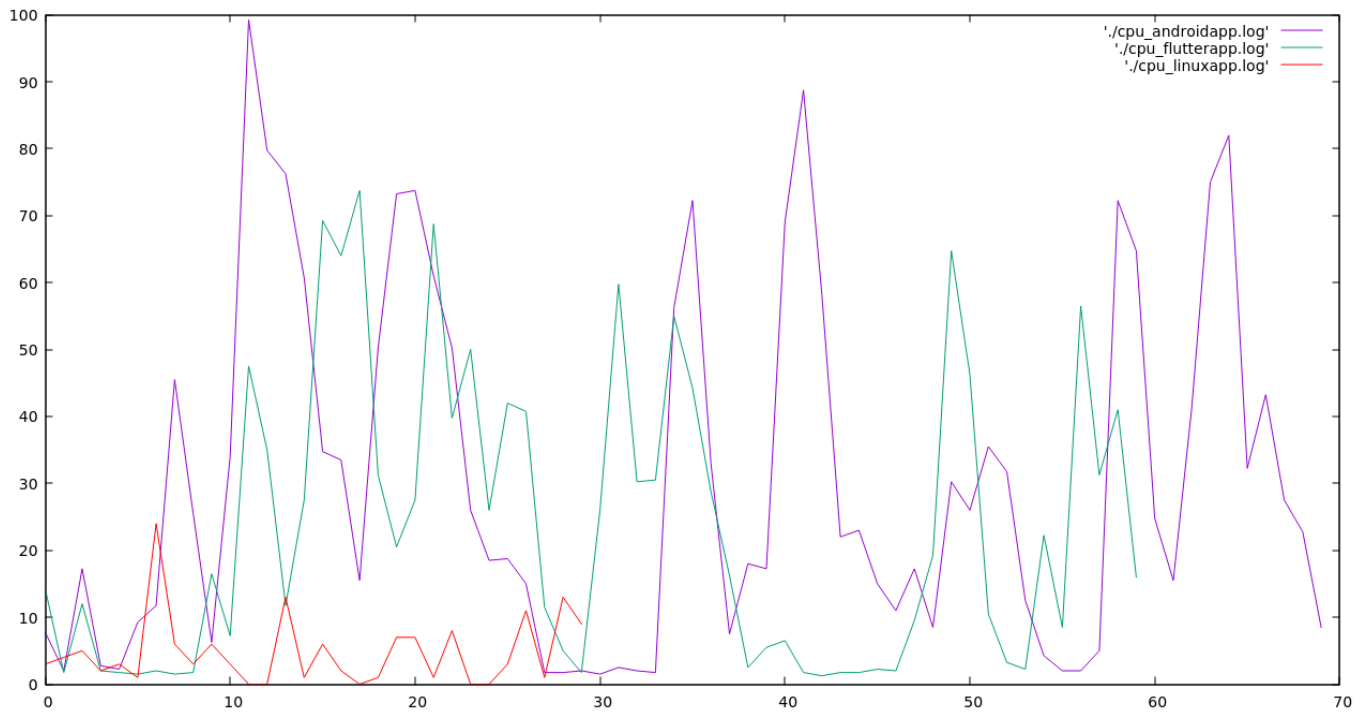
UC5: App Execution Statistics

CPU - Plot :

Java & Flutter App on Android , Qt App on Linux

The app is launched 3 times successively and hence the 3 cpu peaks / Dips.

Cpu usage is captured using **iostat** tool.

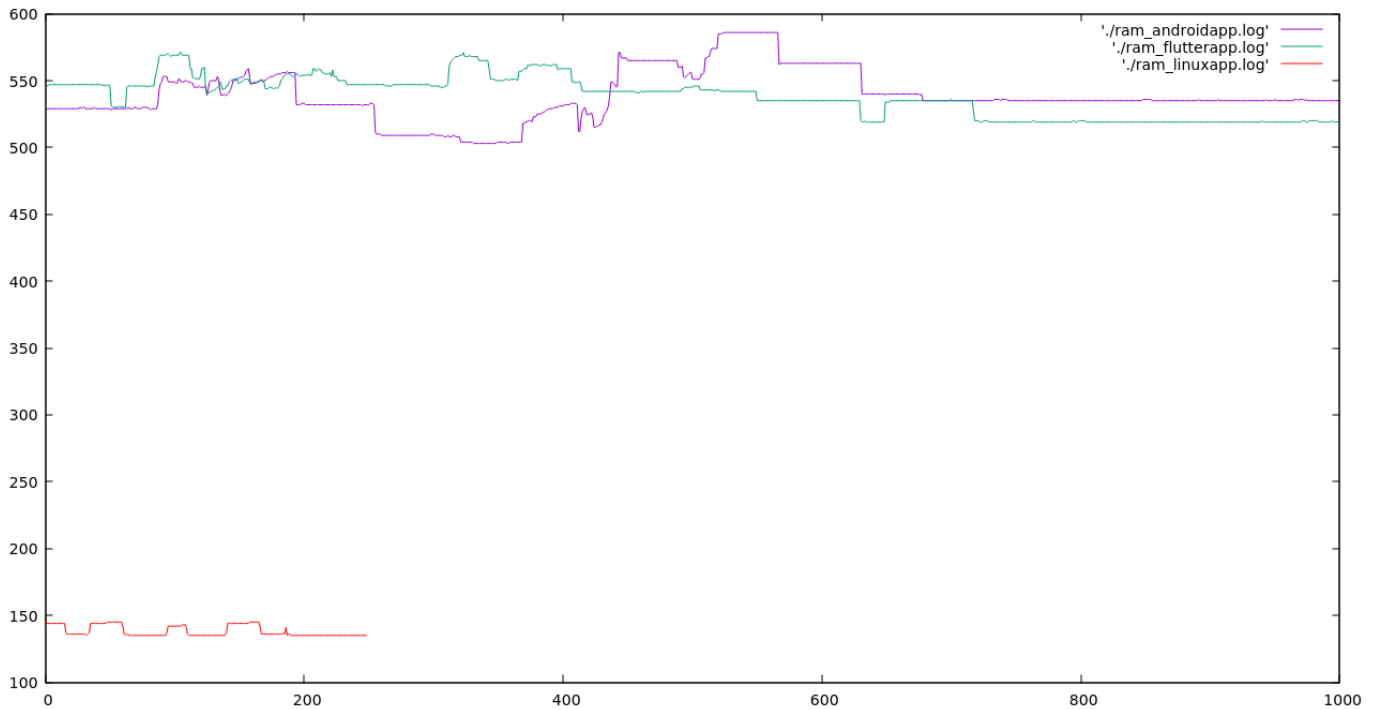


Ram Plot : (MebiBytes)

Java & Flutter App on Android , Qt App on Linux

The app is launched 3 times successively and hence the 3 ram peaks / Dips.

Ram values are captured using **free** tool.



System Calls :

System calls are captured using **strace** tool.

	Linux - Qt App	Android - Java App	Android - Flutter App
System Call Count	5945	24461	52890

Conclusion

	Linux - Qt App	Android - Java App	Android - Flutter App
CPU Usage	LOW ~ 15 %	HIGHEST ~ 65 %	HIGH ~ 50 %
RAM Usage	LOW ~ 10 MiB	HIGHEST ~ 70 MiB	MEDIUM ~ 10 MiB
System Call Count	LOW	MEDIUM	HIGH

Data & Measurements :



androidVsLinux_consolidated.7z