Comparative Study on Android Kernels

¹Shubham Das, ²Mikhal John, ³Ganesh Rathod, ⁴Pravin Gajghate, ⁵Siddhesh Dongare
Department of Information Technology,
JD College of Engineering, Nagpur, Maharashtra, India

Abstract:- Android is the most popular mobile platform which is based on the Linux kernel. But their devices don't age well compared to the competition. Smartphones phones ship with a stock kernel configured to provide balanced performance. We have conducted experiments on Redmi K20 Pro (codenamed Raphael) with a few different kernels based on linux. We have noticed a variety of different performance and battery performances on different kernels by different developers. Some showed significant performance improvements and stable functioning with a minor increase in device temperature and reduced battery life.

Keywords:- Anykernel3, Android, Kernel.

I. INTRODUCTION

The Android OS is based on the Linux kernel and was first released in 2008. It is an open-source platform backed by Google and served by major hardware and software developers. Android Worldwide market share has risen explosively.

The single most prominent example is the Android ecosystem. The Android open-source Project (AOSP) kernels are derived from Linux kernels (i.e., reused in Android) with many features added for mobile devices. In turn, the AOSP kernels are reused by chipset vendors such as Qualcomm who add additional hardware-specific changes.

We investigate the unique Android kernel ecosystem that is decentralized and fragmented. We mine the patch propagation delays across all layers and locate the bottleneck.

Another method is by making more memory available to run apps faster. Others want to upgrade older models with newer versions of the operating system using custom roms.

II. CUSTOM KERNELS

The kernel has complete control over the system, it's important to choose a suitable custom kernel for your Android device. Custom Kernels are the go to choice among many Android users. That is also a big reason why still some people prefer the Android ecosystem over iOS. Android is a famous operating system that features a lot of custom kernels for almost every phone nowadays. Custom Kernels not only offer security updates, but also various improvements over the Stock Kernel

They have many beneficiaries factors-

- CPU, GPU, Memory and Display overclocking.
- How to control the CPU's cores to boost the performance or save more battery.
- We can manage the CPU governors.
- Change the voltage of the CPU during all possible frequencies.
- Enable USB fast charge.
- Configure the I/O scheduler.
- Have more zRAM.
- Ability to increase RAM.
- Enable/Disable Fsync
- Configure zRAM algorithm.
- Configure TCP algorithm.
- Block Wakelocks.

Custom kernels used here-

LOS Kernel- It is the default kernel which ships with the LOS custom ROM. It is a more balanced based kernel.

Waffles Kernel- It is a modified YAAP custom ROM kernel. It is a more performance based kernel.

Soviet Kernel- It is the default kernel for Syberia OS custom ROM. It is a very heavy performance based kernel with a very poor battery life.

All these kernels are very different from each other having their own merits in each sector. We tested them all over the past years to know how they perform against each other.

III. EXPERIMENT PROCEDURE

➢ *Tools Required-*
These are the tools required for flashing, in other words installing a custom kernel and rooting the device and benchmarking them.

- UBUNTU OS- It is required to compile the kernel.
- ADB/FASTBOOT- It is required to add support for the commands in the terminal.

- RECOVERY- There are a few different types of recovery, like TWRP (Team Win Recovery Project) or OFOX (Orange Fox Recovery).
- MAGISK- It is a method to root the Android device and it gives access to the advanced menus inside the Android's core.
- FRANCO KERNEL MANAGER- It is used to monitor and change the parameters in a kernel.
- ANTUTU- It is a very popular CPU, GPU, RAM, Storage benchmarking tool.
- GEEKBENCH- It is just another app like ANTUTU.
- CPU THROTTLING- It measures the CPU heating by running all its cores at the highest frequency for a specific amount of time.

IV. INSTALLING CUSTOM KERNEL

First, the bootloader of the device has to be unlocked. This enables us to use a custom recovery image and install the kernels and also gain root access.

[Step1: Preparation]

- Install fastboot on Ubuntu/Windows.
- Download TWRP recovery image, kernel zip and Magisk zip
- On the phone enable developer options by tapping build number 5 times.
- In developer options enable OEM UNLOCKING.
- Reboot the phone into FASTBOOT MODE by pressing volume up + power button.
- Connect the phone in fastboot mode to the pc.
- Run the command to unlock the bootloader. # fastboot oem unlock 8. Reboot into fastboot mode.

[Step2: Installing Custom recovery]

- Check if the device is connected using the fastboot command on the terminal. # fastboot devices
- With the recovery file in the current working directory install TWRP image. # fastboot flash recovery twrp.img

[Step3: Installing Kernel and Magisk]

- Using the install option in twrp select the kernel zip and then Magisk zip
- Use the swipe to install option and waitfor both the zips to finish.
- Reboot the phone.

[Step4: Kernel Manager]
- Install Kernel Manager of any preference. We are using the Franco kernel Manager from the Play Store.

V. RESULT

Kernel Name	CPU Score	GPU Score	Memory Score	UX Score	Overall Score
LOS	96607	123418	49159	56029	325213
YAAP	98473	127027	505086	58398	334406
Soviet	10389	140131	52613	56355	352918
Stock	51767	69977	40908	30402	193054

Table -1: Antutu Benchmark scores for K20 Pro

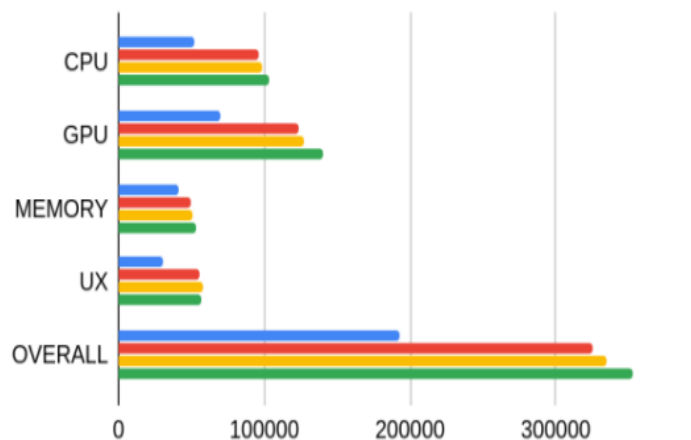The stock kernel has the lowest performance scores compared to the custom kernels as expected.

Chart -1: Antutu Benchmark scores for K20 Pro
Blue Graph- Stock kernel
Red Graph- LOS kernel
Yellow Graph- YAAP Kernel
Green Graph- Soviet Kernel

Kernel Name	Single Score	Multi Score
Stock	276	1098
LOS	522	2223
YAAP	523	2258
Soviet	520	2290

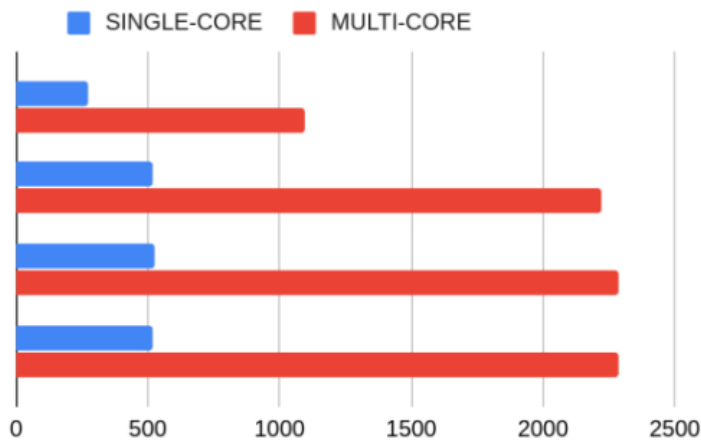**Table -2: Geekbench Scores for K20 Pro**

Chart -2: Geekbench Scores for K20 Pro
1. Stock Kernel
2. LOS
3. YAAP
4. Soviet

VI. CONCLUSION

From the above results it is clear that stock kernels are not focused for high performance and lean towards efficiency and limited features. The custom kernels unlock a lot of features like overclocking the display refresh rate considering the fact that if the hardware supports them. For performance evaluation, we have collected benchmarks for different kernels for key Android system components such as Multithreading, Task scheduling, Binder, Storage, File System, CPU utilization, GPU utilization, Ram usage, etc. . Our test results show that the average system performance of speedups up to 10-30%.

REFERENCES

[1]. P. Yuan, Y. Guo, X. Chen and H. Mei, "Device-Specific Linux Kernel Optimization for Android Smartphones," 2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Bamberg, 2018

[2]. L. Corral, A. B. Georgiev, A. Janes and S. Kofler, "EnergyAware Performance Evaluation of Android Custom Kernels," 2015 IEEE/ACM 4th International Workshop on Green and Sustainable Software, Florence, 2015

[3]. Joo, B.-G & Kim, S.-M. (2012). A user's experience in optimizing smartphone performance using overclocking and memory cleaning techniques

[4]. Google Developers (2020) Android Platform. Available at: https://developer.android.com/guide/platform

[5]. Akhilnarang (2020) Illusion Kernel. Available at: https://forum.xda-developers.com/pocof1/development/derp-kernel-v1-0-gcc-7-4-9-133- t3856615

[6]. Arter97 (2020) Arter97 Kernel. Available at: https://forum.xda-developers.com/pocof1/development/arter97-kernel-poco-f1-t3919127

[7]. Negron (2020) LawRun Kernel. Available at: https://forum.xda-developers.com/pocof1/development/kernel-lawrun-kernel-v3-t4037729.