

The performance data for the mobile application running on an Android device with a 1-core CPU and 3 GB of RAM reveals several areas for potential optimization. Below is a detailed analysis of the key performance metrics, resource utilization, and actionable insights to enhance the application's efficiency.

- 1. Application CPU Usage:** The average CPU usage of the application is 45.87%, with a maximum of 173.2%. Given that the device has a single-core CPU, the maximum utilization can be 100%. The observed maximum usage of 173.2% indicates that the application is likely causing the CPU to be overburdened, leading to potential performance bottlenecks. To address this, developers should profile the application to identify and optimize CPU-intensive operations. Techniques such as offloading heavy computations to background threads, optimizing algorithms, and reducing the frequency of resource-intensive tasks can help lower CPU usage.
- 2. Device CPU Usage:** The device's overall CPU usage averages at 171.53%, with a maximum of 383.0%. This high utilization suggests that the device is consistently under heavy load, which can degrade the user experience. Developers should ensure that the application is not contributing excessively to this load by optimizing background services, reducing wake locks, and minimizing the use of high-priority threads.
- 3. Threads:** The application uses an average of 86.76 threads, with a maximum of 105.0. While multithreading can improve performance, excessive thread creation can lead to context switching overhead and increased memory usage. Developers should review the threading model to ensure that threads are being used efficiently and consider using thread pools to manage thread creation and reuse.
- 4. App Memory PSS Usage:** The application's average memory usage is 179.15 MB, with a maximum of 230.52 MB. Given the device's 3 GB of RAM, this usage is within acceptable limits. However, developers should still aim to optimize memory usage by identifying and fixing memory leaks, using efficient data structures, and releasing unused resources promptly.
- 5. Device Memory PSS Usage:** The device's overall memory usage averages at 1516.05 MB, with a maximum of 1571.5 MB. This indicates that the device has sufficient memory headroom, and the application is not causing significant memory pressure.
- 6. FPS (Frames Per Second):** The average FPS is 18.95, with a minimum of 1.0 and a maximum of 60.0. The low average FPS indicates that the application is not rendering smoothly, which can lead to a poor user experience. Developers should profile the rendering pipeline to identify bottlenecks, optimize the use of graphics resources, and ensure that the UI is being updated efficiently. Techniques such as reducing the complexity of UI elements, using hardware-accelerated rendering, and minimizing overdraw can help

improve FPS.

7. Energy Score: The average energy score is 32.55, with a maximum of 996.24.

The high maximum score suggests that there are periods when the application consumes a significant amount of energy, which can drain the device's battery quickly. Developers should focus on optimizing energy consumption by reducing the frequency of background tasks, minimizing the use of high-power components (e.g., GPS, network), and optimizing the use of wake locks and alarms.

8. Network Download and Upload: The average network download is 0.03 MB, with a maximum of 2.4 MB, and the average network upload is 0.0 MB, with a maximum of 0.1 MB. The low network usage indicates that the application is not heavily reliant on network operations. However, developers should still ensure that network requests are batched, use efficient data formats, and implement caching strategies to minimize network usage further.

In summary, the key areas for optimization include reducing CPU usage, improving FPS, and optimizing energy consumption. By addressing these areas, developers can enhance the performance and user experience of the mobile application on the specified Android device.