

GOSIM WORKSHOP

Makepad Performance
Benchmarking

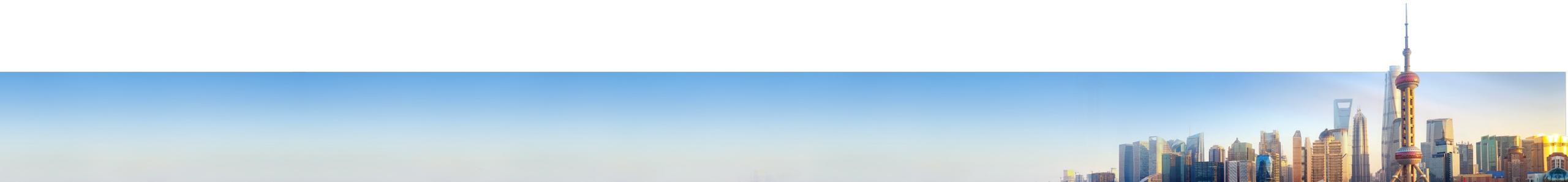
Edward Tan
@guofoo

GOSIM®
GLOBAL OPEN-SOURCE INNOVATION MEETUP



Makepad Performance Benchmarking

1. What To Measure
2. Tools Used
3. What We Tested
4. Results
5. Conclusion



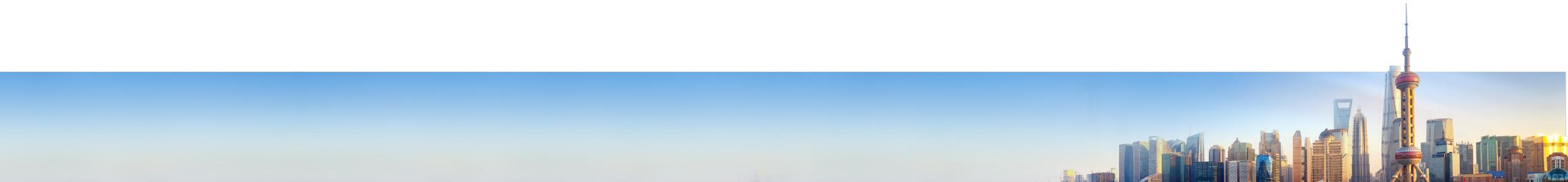
1. What To Measure



Ideal Things to Test

Resources Usage

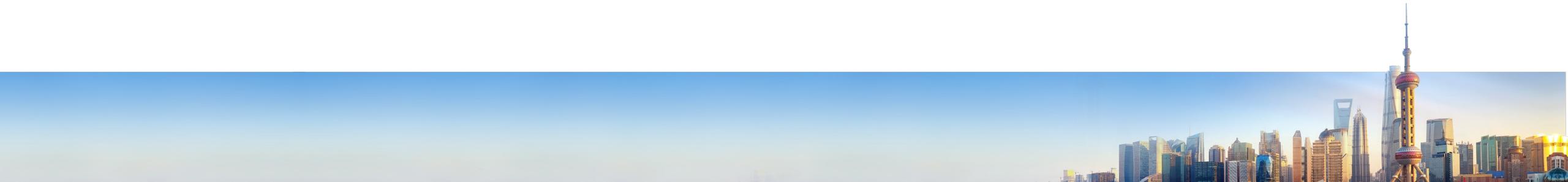
- CPU Usage
 - How many cycles
 - Speed of CPU
- GPU Usage
- Memory Usage
 - Memory Footprint
- Power Consumption



Ideal Things to Test

Responsiveness:

- Startup time – the time between launch of the program up until first render
- Frame rendering time – Time spent by the app to render a single frame
- Input latency – Time between user input and app response
- First content-full paint (FCP)
- Time to interactive (TTI)



2. Tools Used



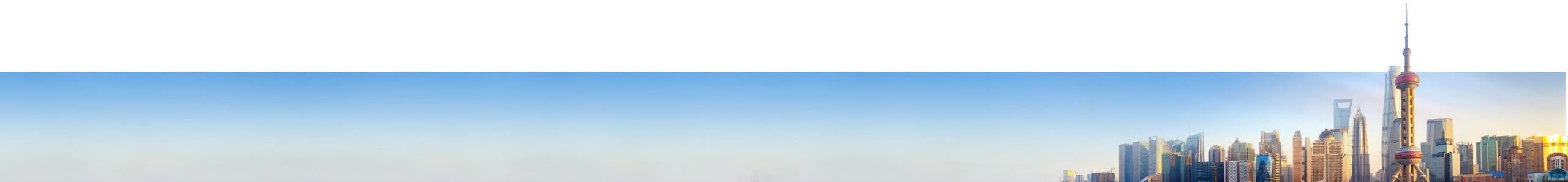
Tools Attempted

- Android AGI
 - Profiling of: GPU, CPU, MEM, Shader Performance, Frame
 - Could not get it working on Samsung Note 9 and Samsung Flip Z
- Snapdragon Profiler
 - Similar to AGI; ran into issues with .NET dependencies on MacOS
- Android Studio Profiler
 - Profiling of CPU and MEM (no GPU)
 - Not able to make Makepad inspectable by making it debug-able or profile-able



Tools Used

- Perfetto UI
 - No installation needed
 - Good for sharing trace recordings
- adb shell dumpsys gfxinfo <name> framestats
 - Should be able to get GPU usage information



Device Used

Samsung SM-F721B (Samsung Galaxy Z Flip 4)

- Running Android 13 API 33
- Chipset: Qualcomm SM8475 Snapdragon 8+ Gen 1 (4 nm)
- CPU: Octa-core (1x3.19 GHz Cortex-X2 & 3x2.75 GHz Cortex-A710 & 4x1.80 GHz Cortex-A510)
- GPU: Adreno 730
- Released: August, 2022

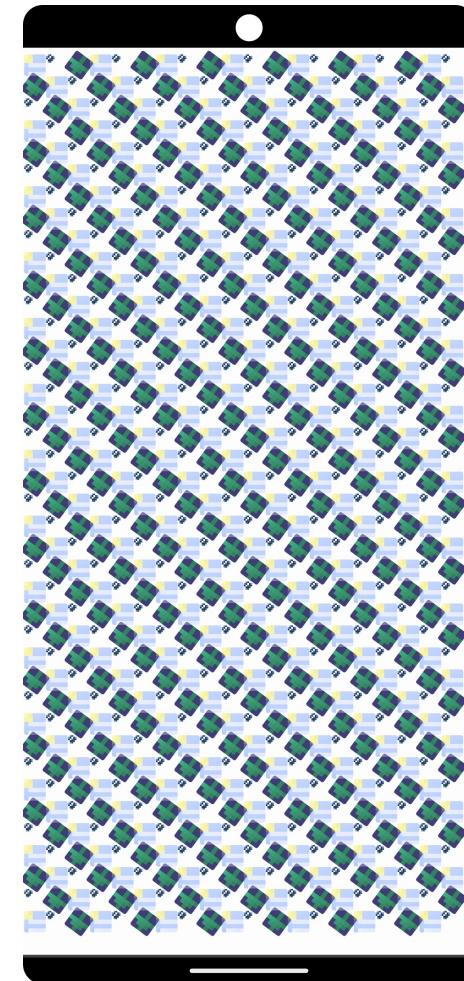


3. What We Tested



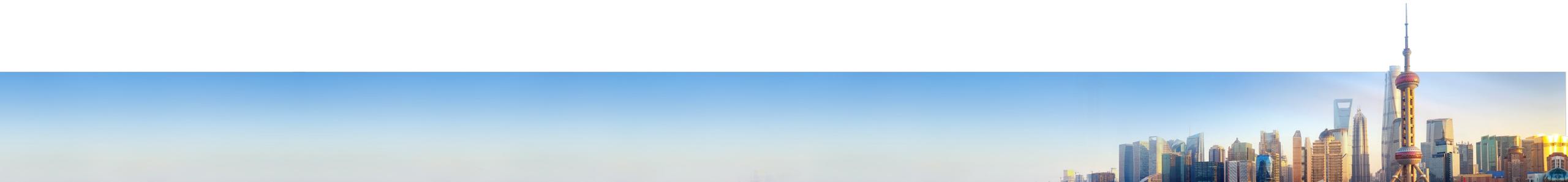
What We Tested

- Image Manipulation test application performs different animation operations on images:
 - Rotation
 - Scaling
 - Fading
- Three 5x5 pixels sized images used for upscaling
- Three 200x200 sized images for downscaling



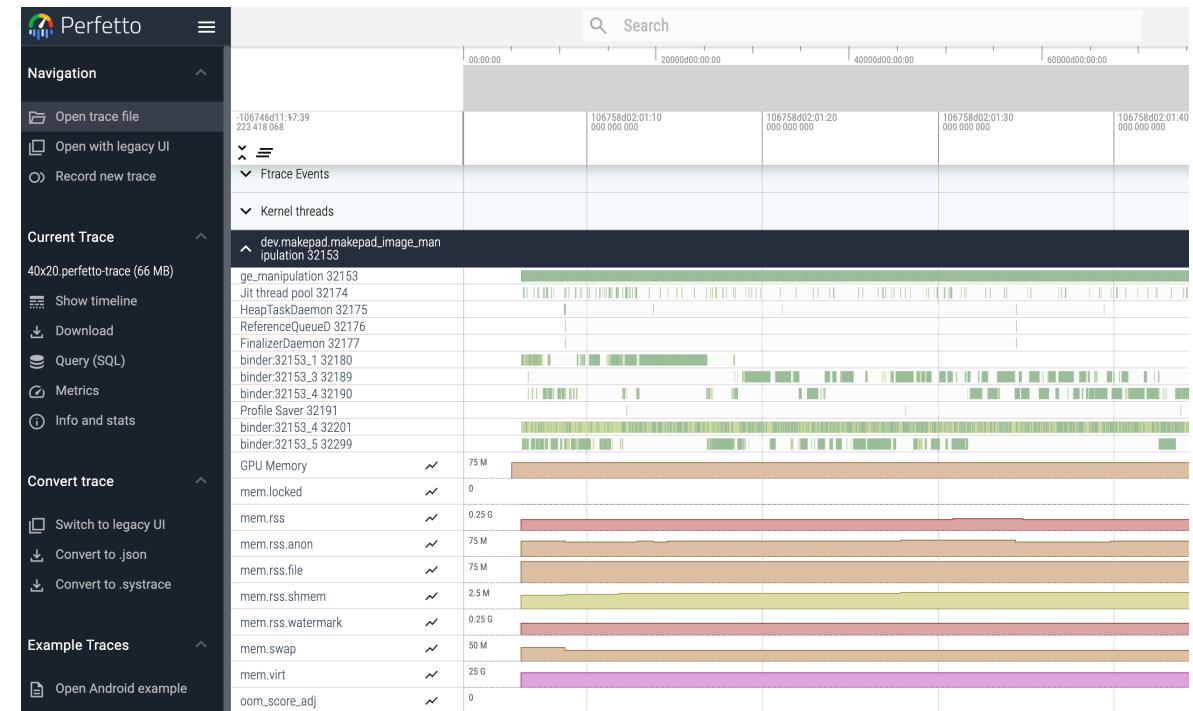
What We Tested

- Wrote same programs using Makepad, Flutter & Jetpack Compose(Native)
- Determine how the performance and responsiveness scale up
 - Run tests on 100 images (10 rows x 10 columns)
 - 200 images
 - 400 images
 - 800 images (20 rows x 40 columns)



Performance

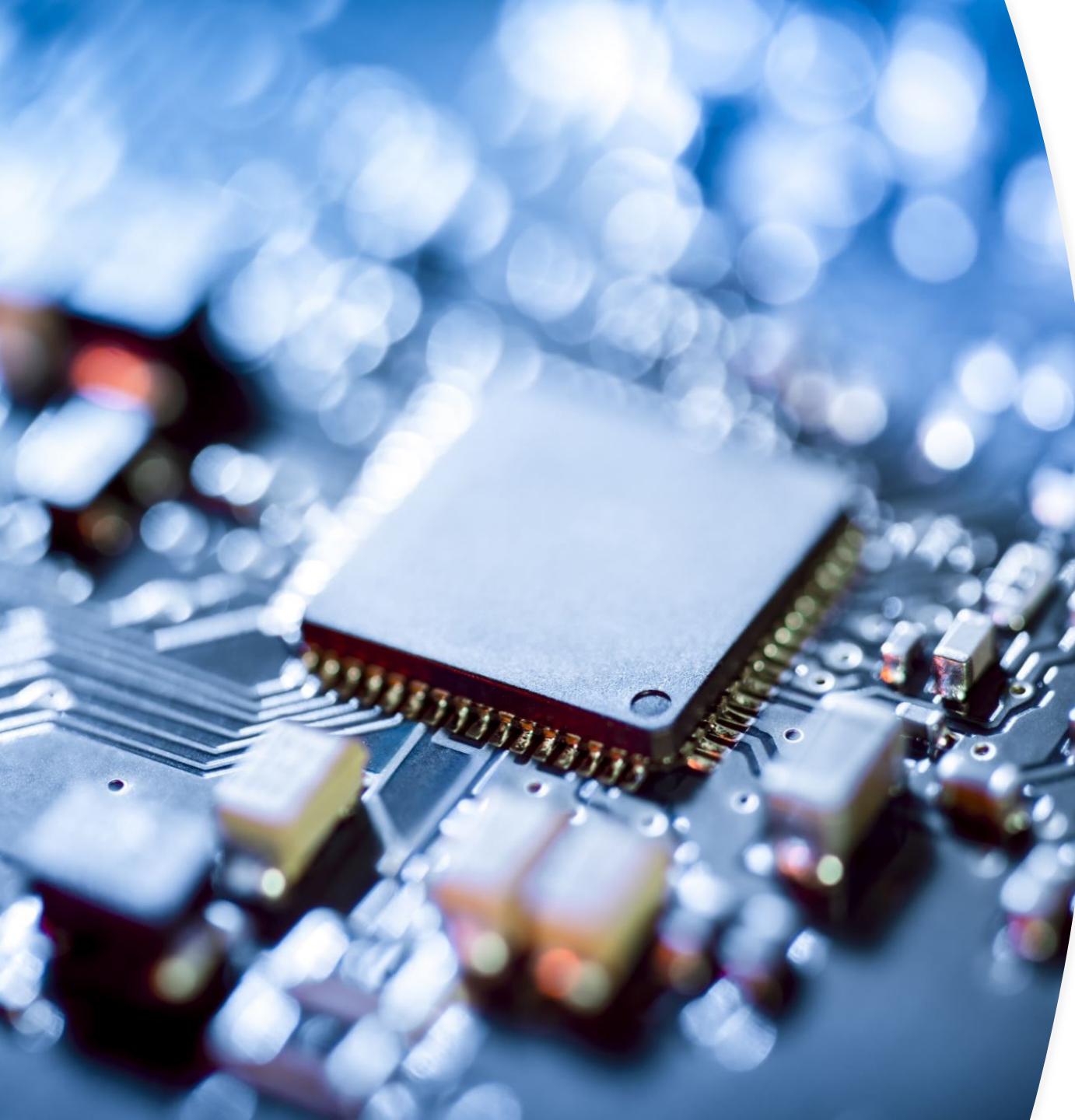
- Tracing results with Perfetto
- GPU, CPU, Memory usage
- Data is from single trace
 - very little difference between multiple runs
- Each recording takes several hours



4. Results



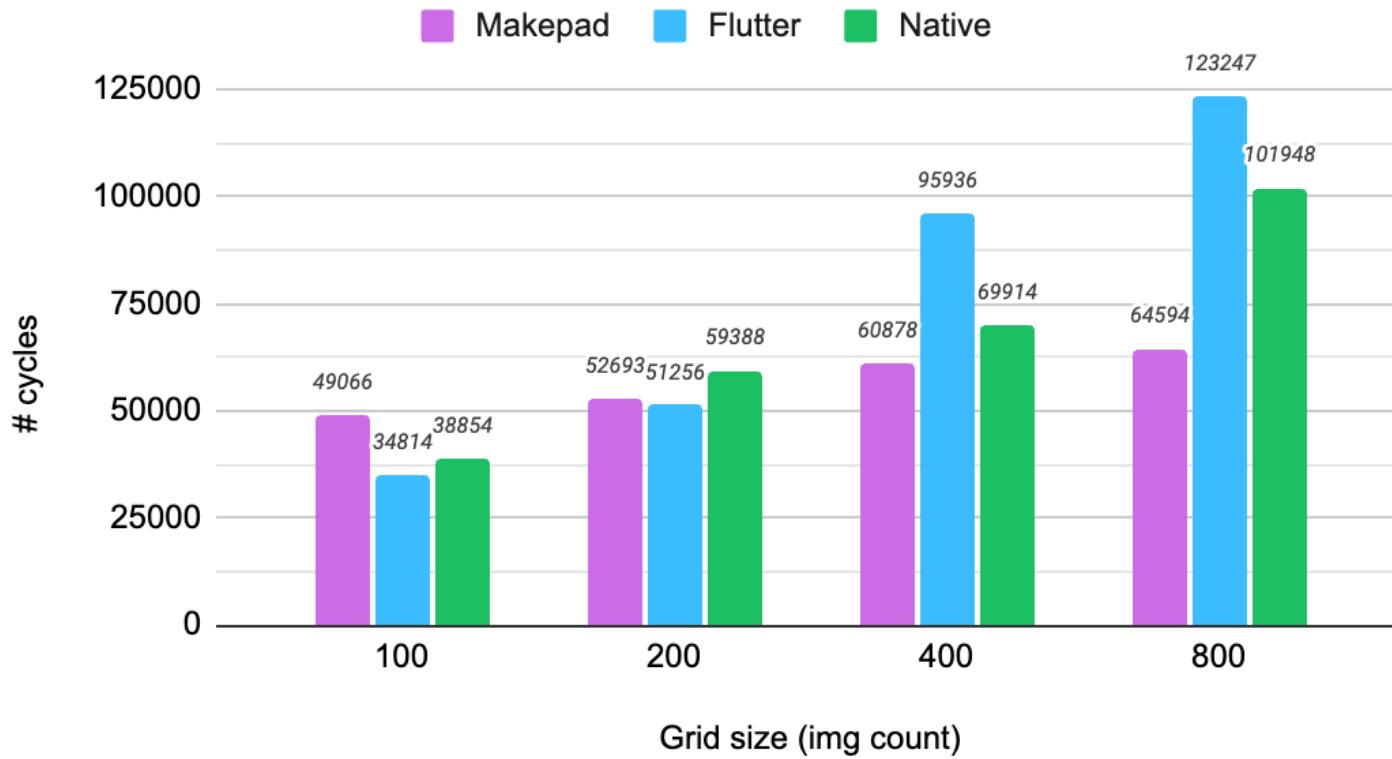
All Results presented are Early Preliminary
and will subject to change as the Makepad
Framework continues to evolve toward its
official 1.0 release.



Performance - CPU

- CPU Cycles
- CPU Frequency

CPU Cycles



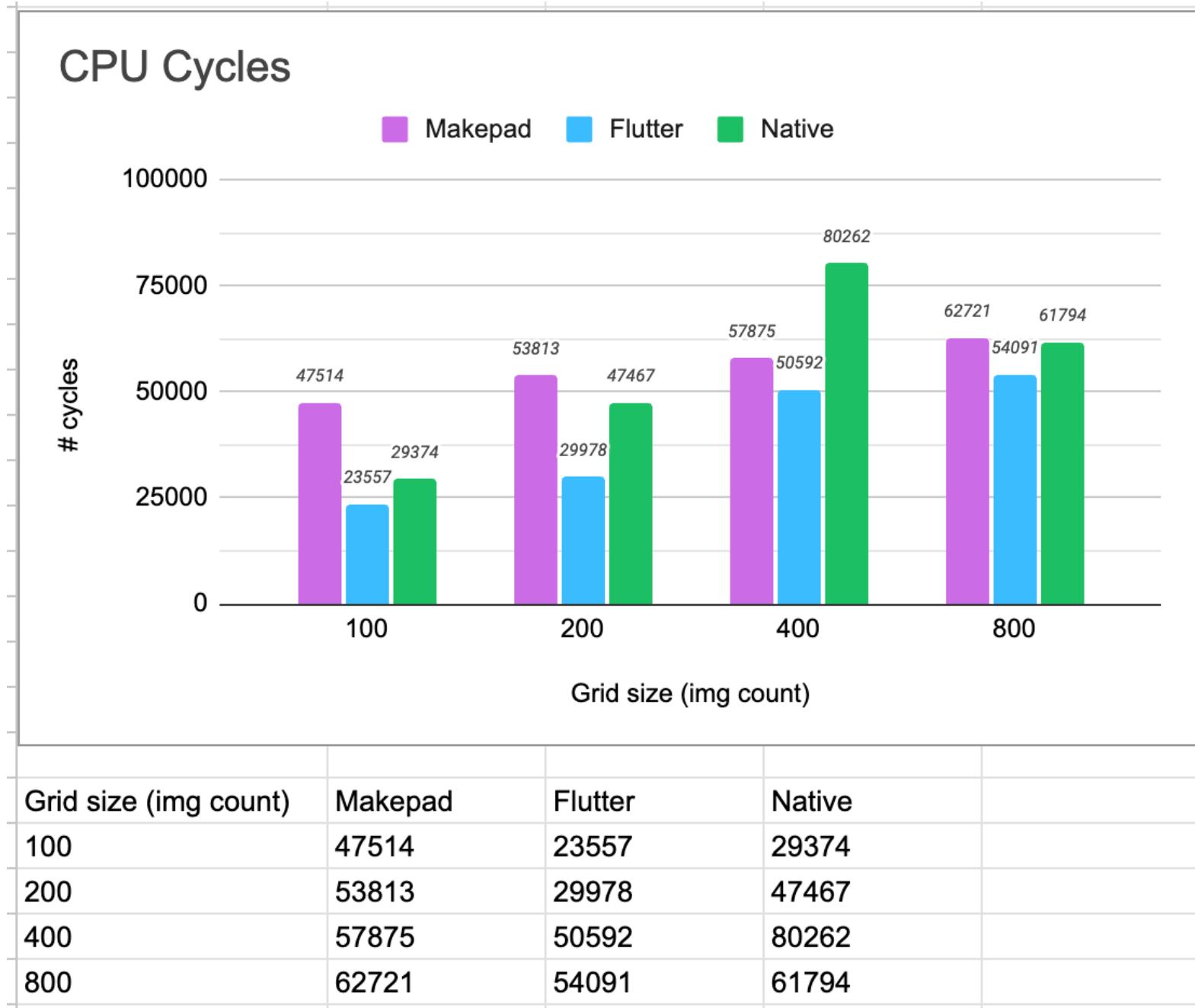
Rendering Performance: Downscaling

- # of Cycles
- Lower means less processing
- How hard the CPU is working

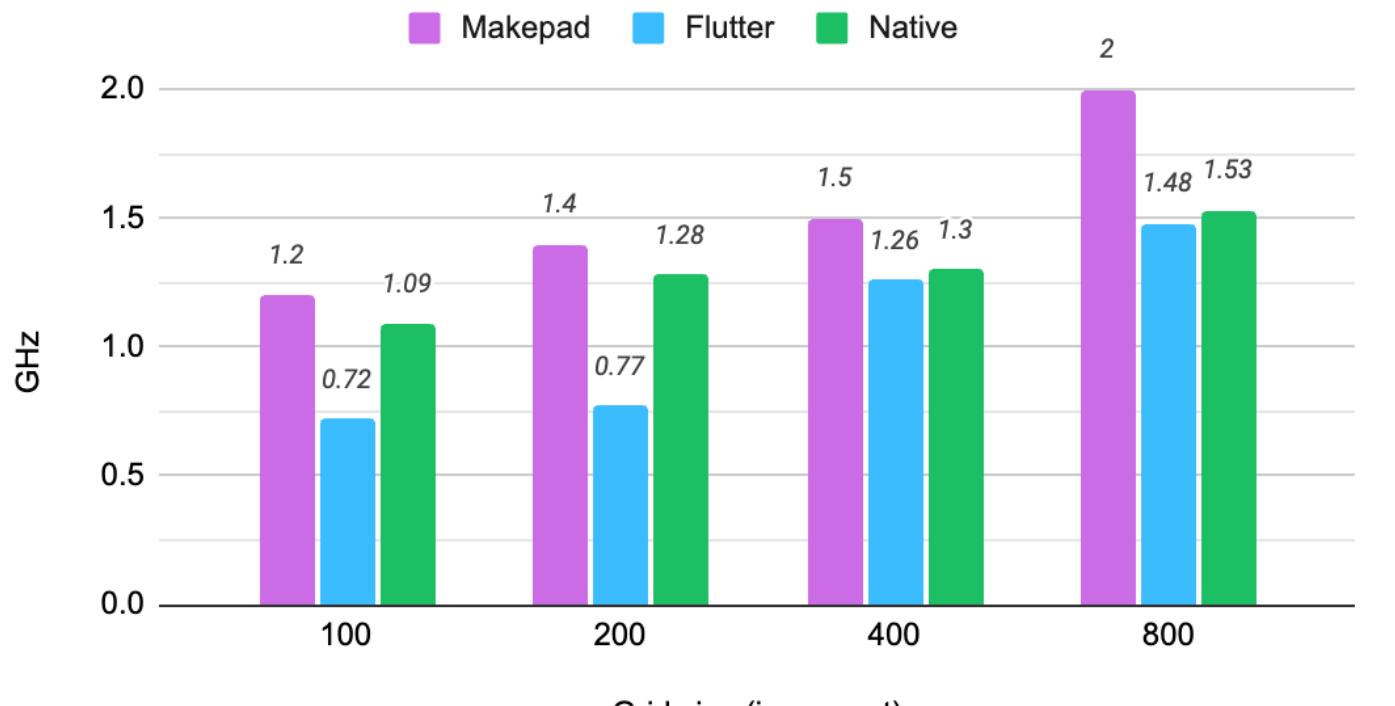
Grid size (img count)	Makepad	Flutter	Native
100	49066	34814	38854
200	52693	51256	59388
400	60878	95936	69914
800	64594	123247	101948

Rendering Performance: Upscaling

- # of Cycles
- Lower means less processing
- How hard the CPU is working



CPU Frequency



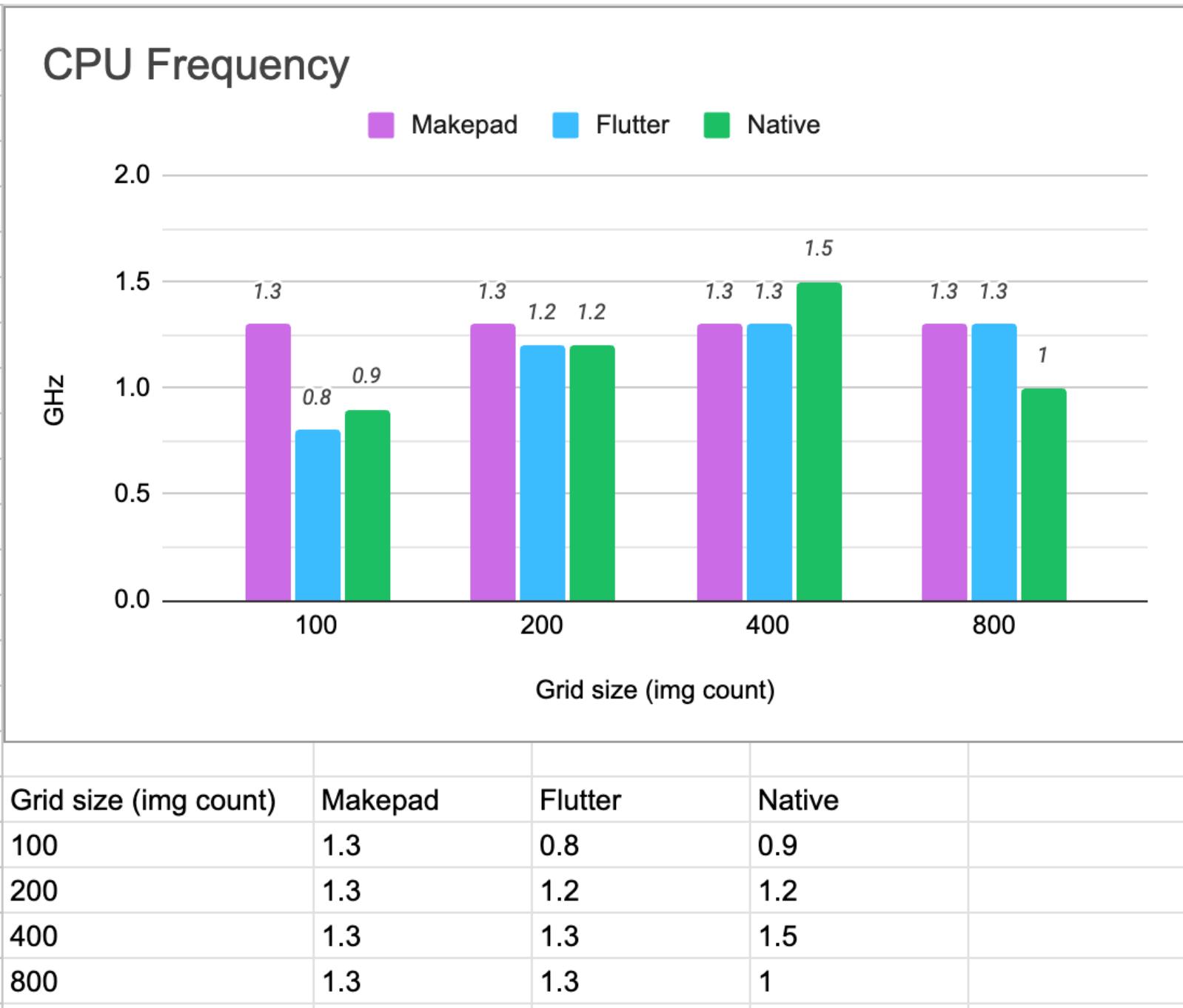
Rendering Performance: Downscaling

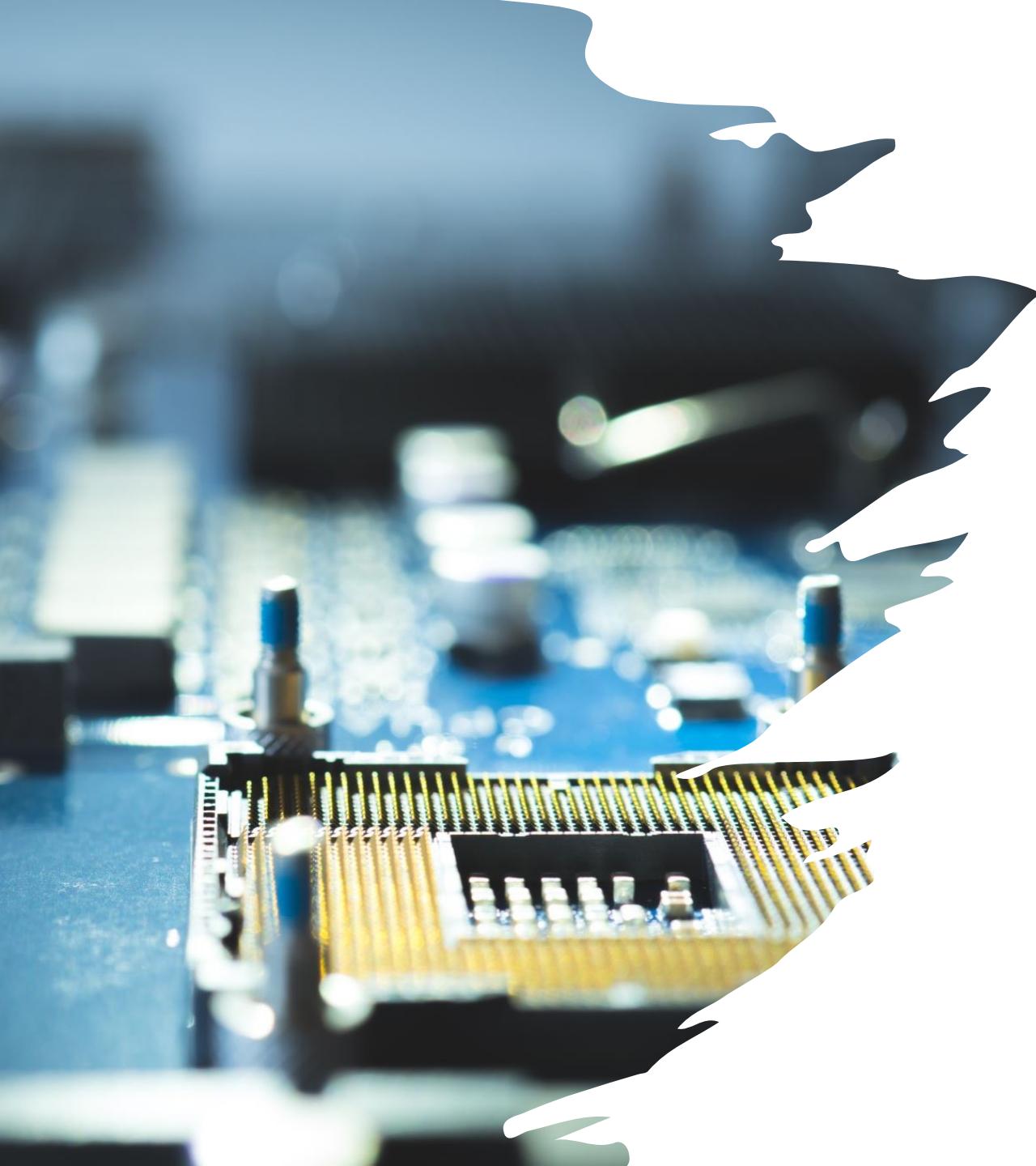
- GHz
- How fast the CPU was running

Grid size (img count)	Makepad	Flutter	Native
100	1.2	0.72	1.09
200	1.4	0.77	1.28
400	1.5	1.26	1.3
800	2.0	1.48	1.53

Rendering Performance: Upscaling

- GHz
- How fast the CPU was running

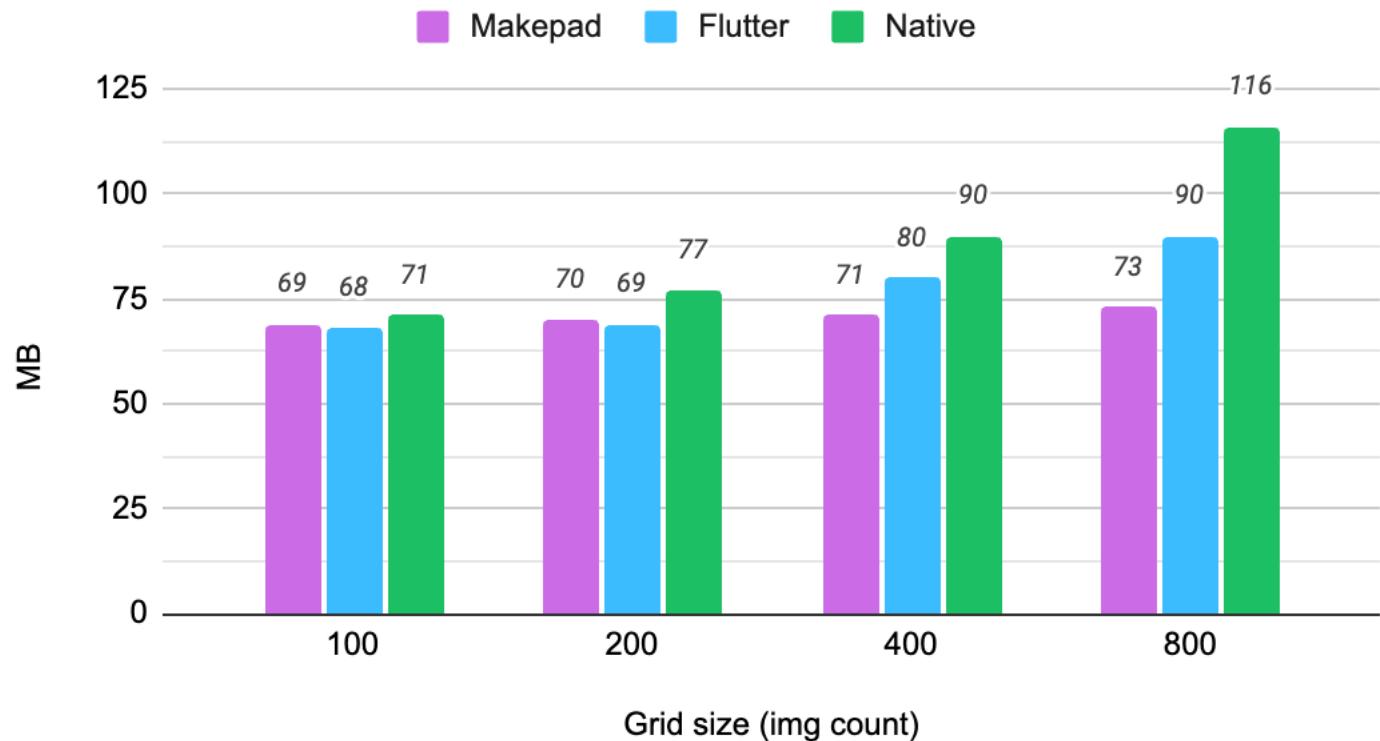




Performance - Memory

- Memory Average
- Memory Max

Memory AVG



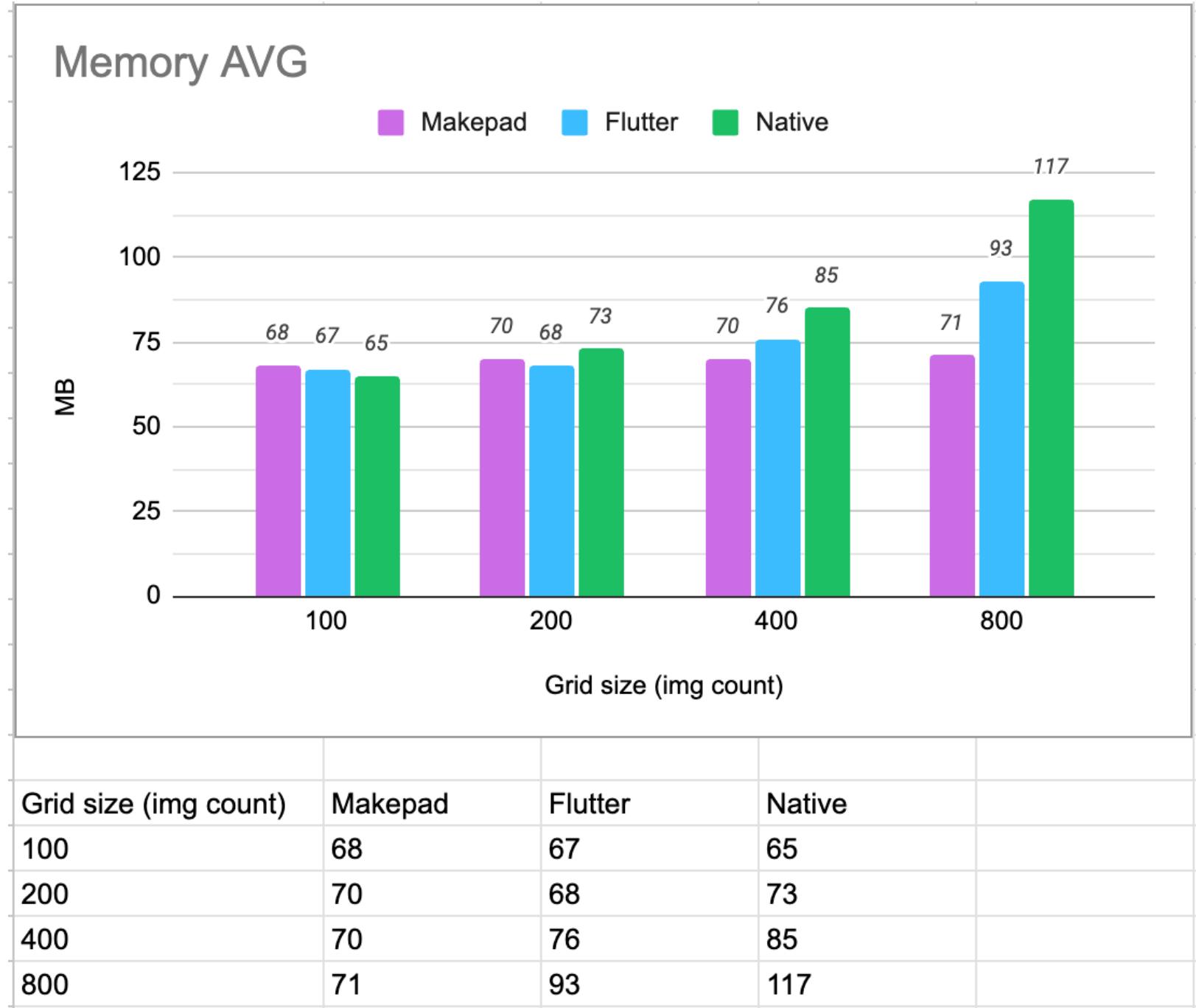
Rendering Performance: Downscaling

- MBs
- Average memory used

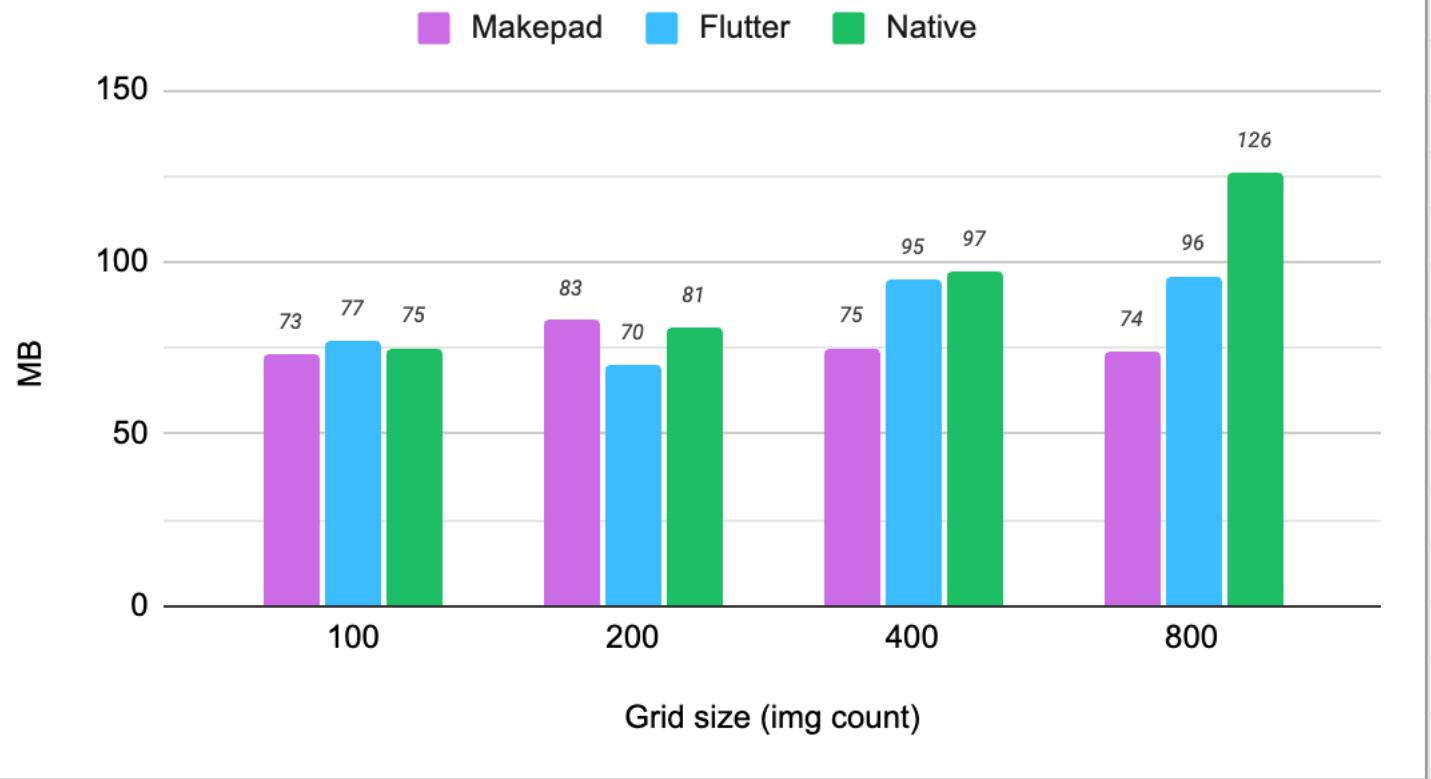
Grid size (img count)	Makepad	Flutter	Native
100	69	68	71
200	70	69	77
400	71	80	90
800	73	90	116

Rendering Performance: Upscaling

- MBs
- Average memory used



Memory MAX

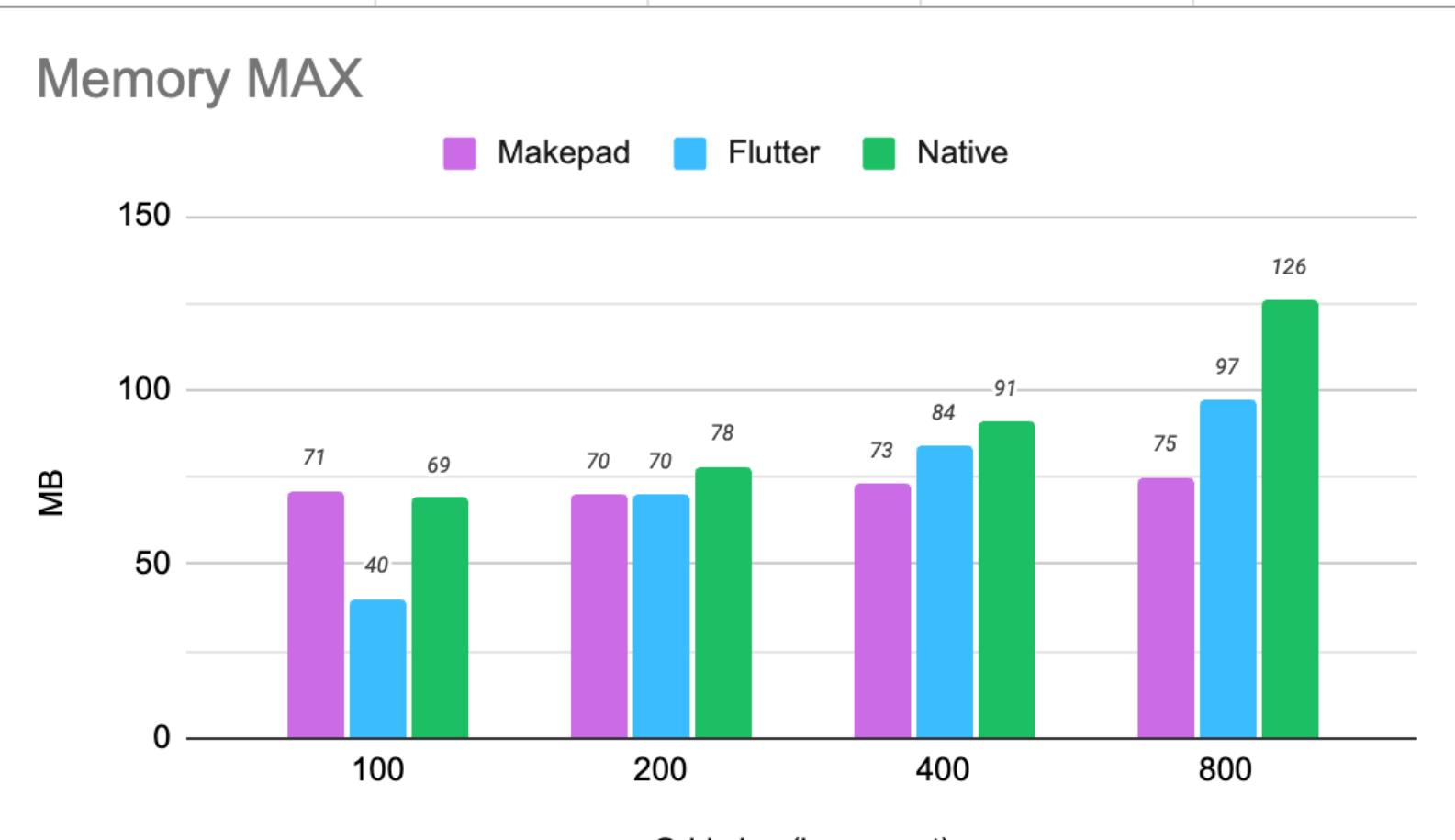


Rendering Performance: Downscaling

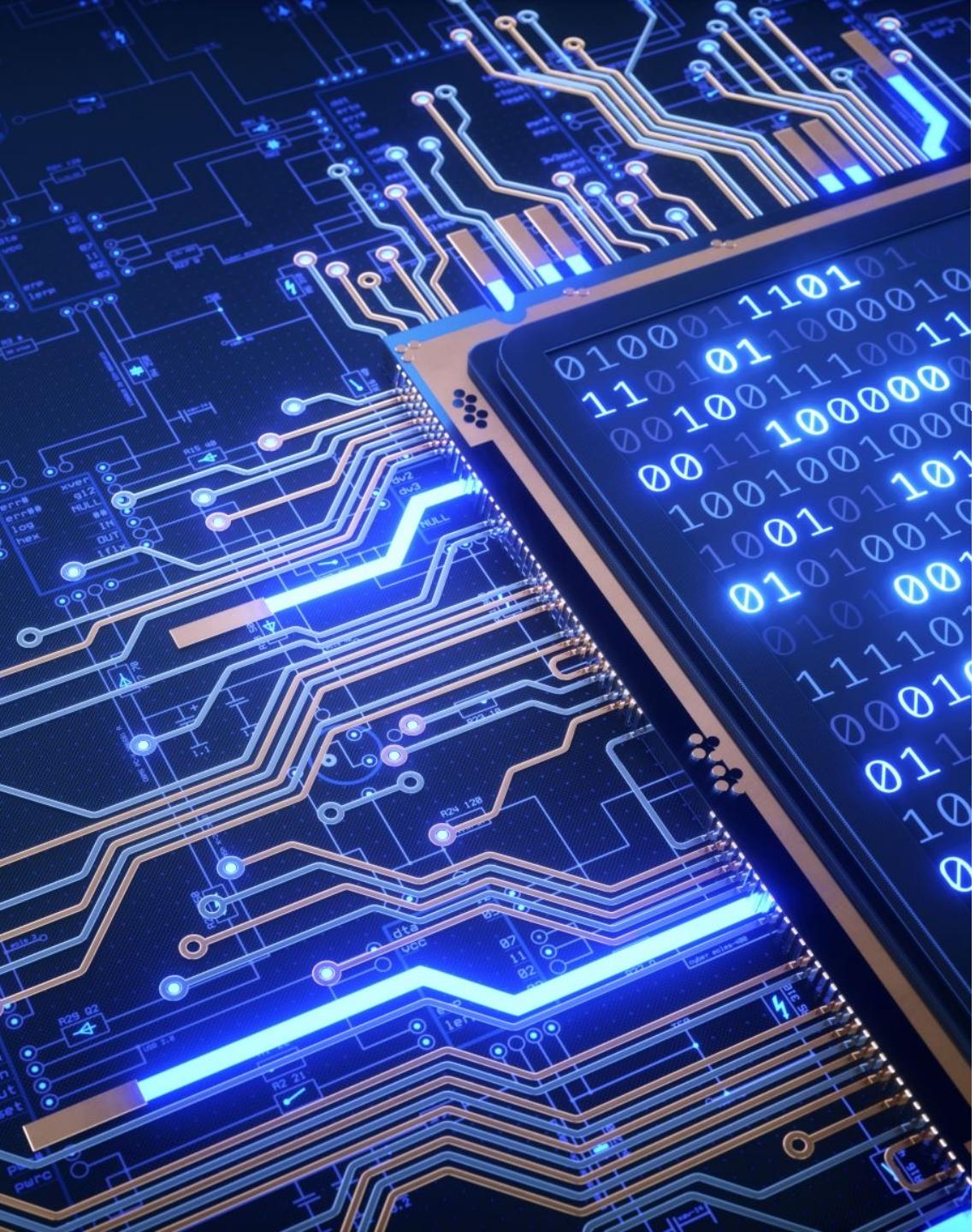
- MBs
- Max memory used – usually during startup

Rendering Performance: Upscaling

- MBs
- Max memory used – usually during startup

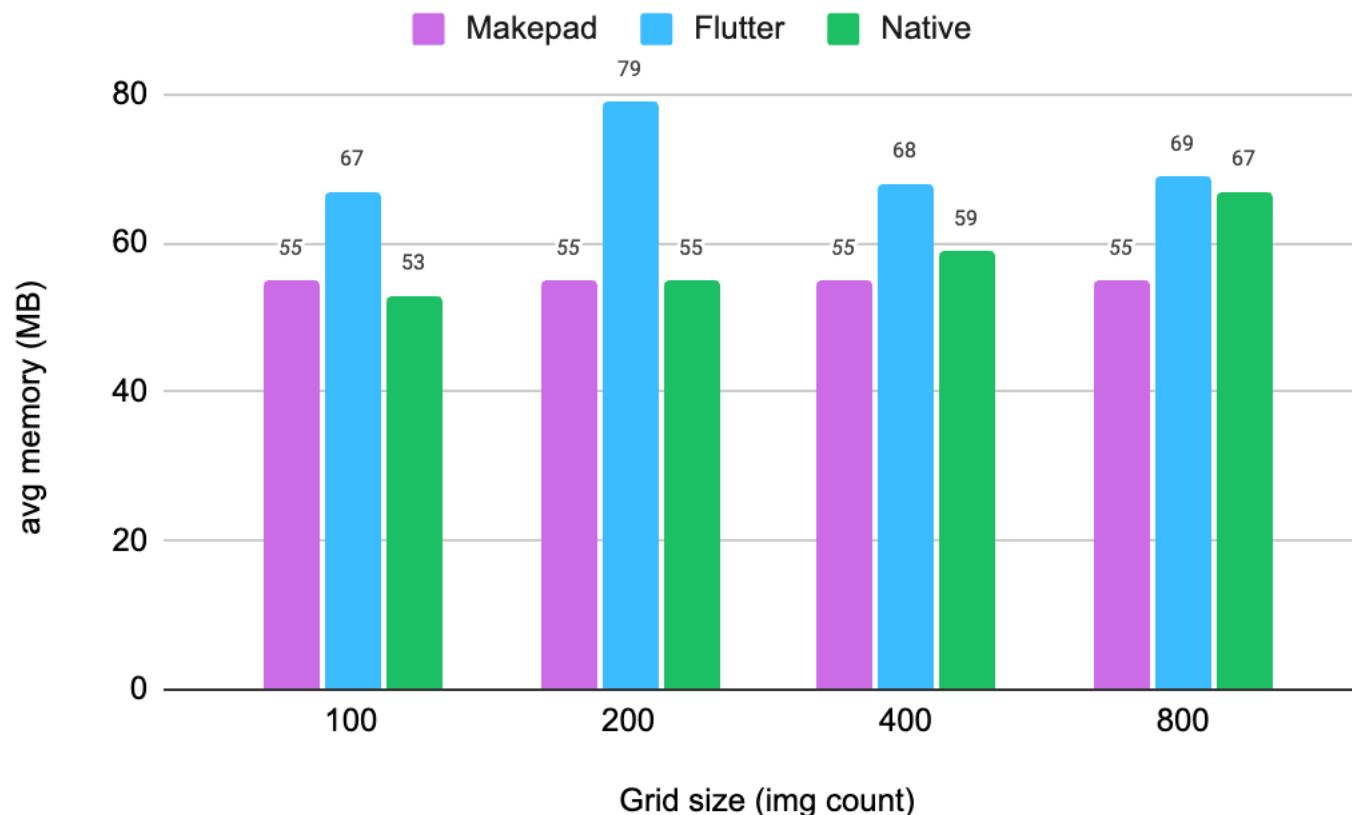


Grid size (img count)	Makepad	Flutter	Native
100	71	40	69
200	70	70	78
400	73	84	91
800	75	97	126



Performance - GPU

GPU



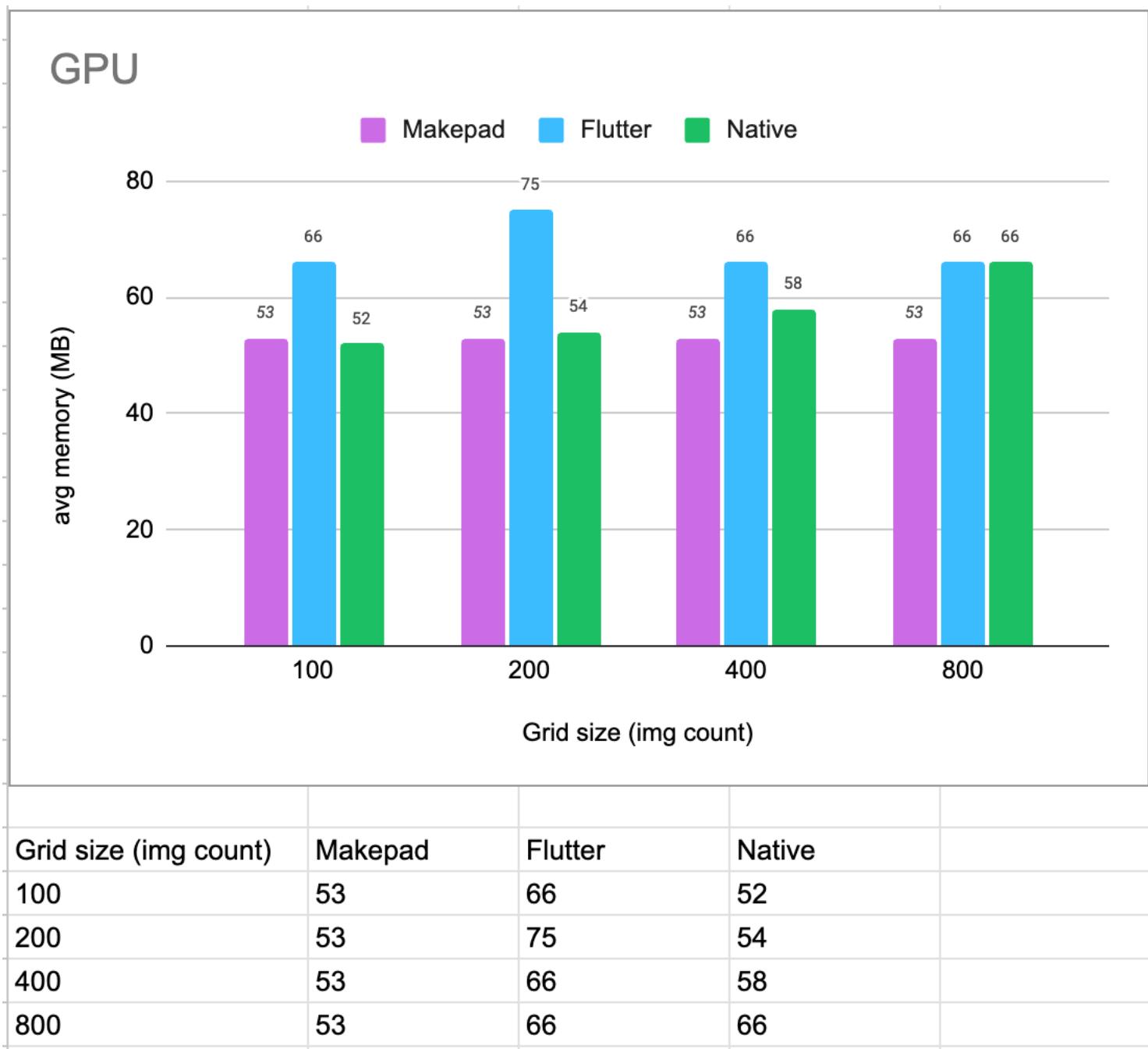
Rendering Performance: Downscaling

- Average Memory (MB)
- How much data sent to GPU

Grid size (img count)	Makepad	Flutter	Native
100	55	67	53
200	55	79	55
400	55	68	59
800	55	69	67

Rendering Performance: Upscaling

- Average Memory (MB)
- How much data sent to GPU

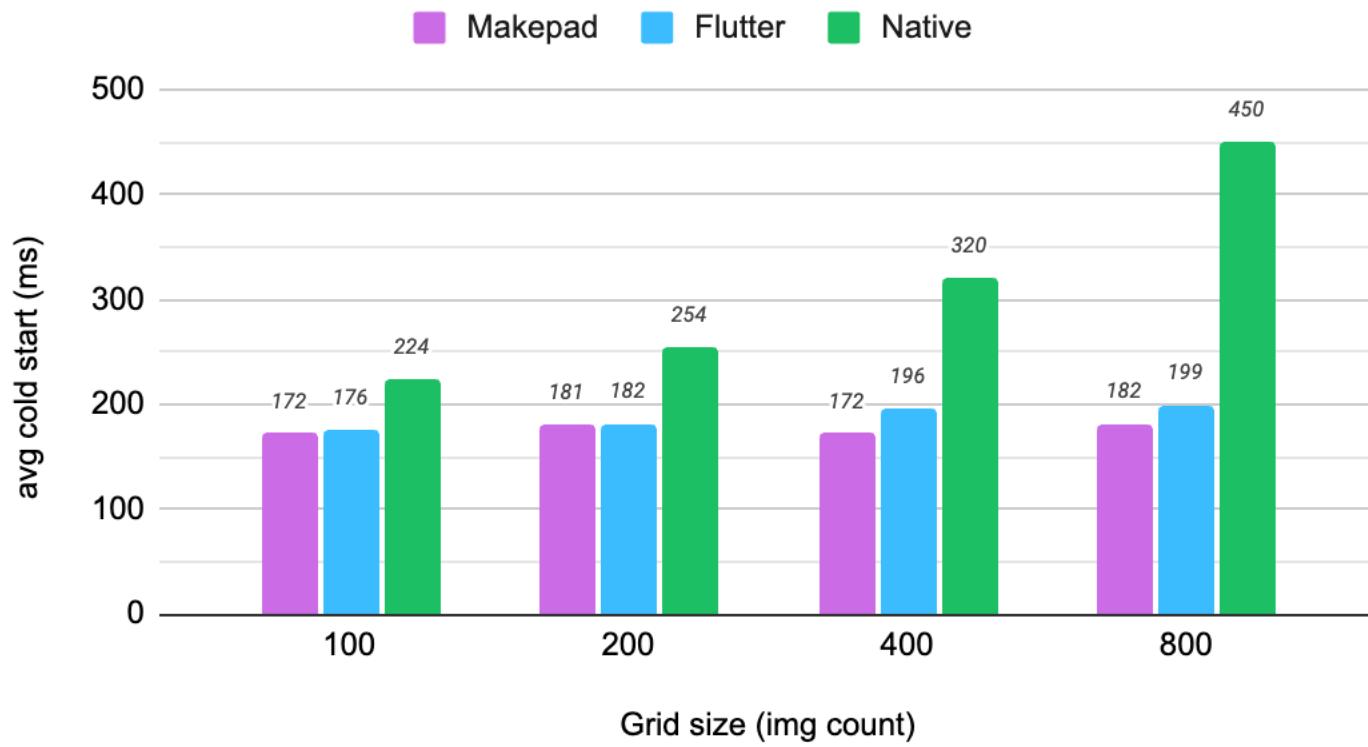




Responsiveness – Startup Time

- Initial startup time – from launch command to first render command
- Does not consider time for all application content to be displayed
 - Since Makepad does not use the Android UI rendering system (similar to Flutter)
- It may not indicate full user-perceived startup time
- Cold start times used

Startup Time



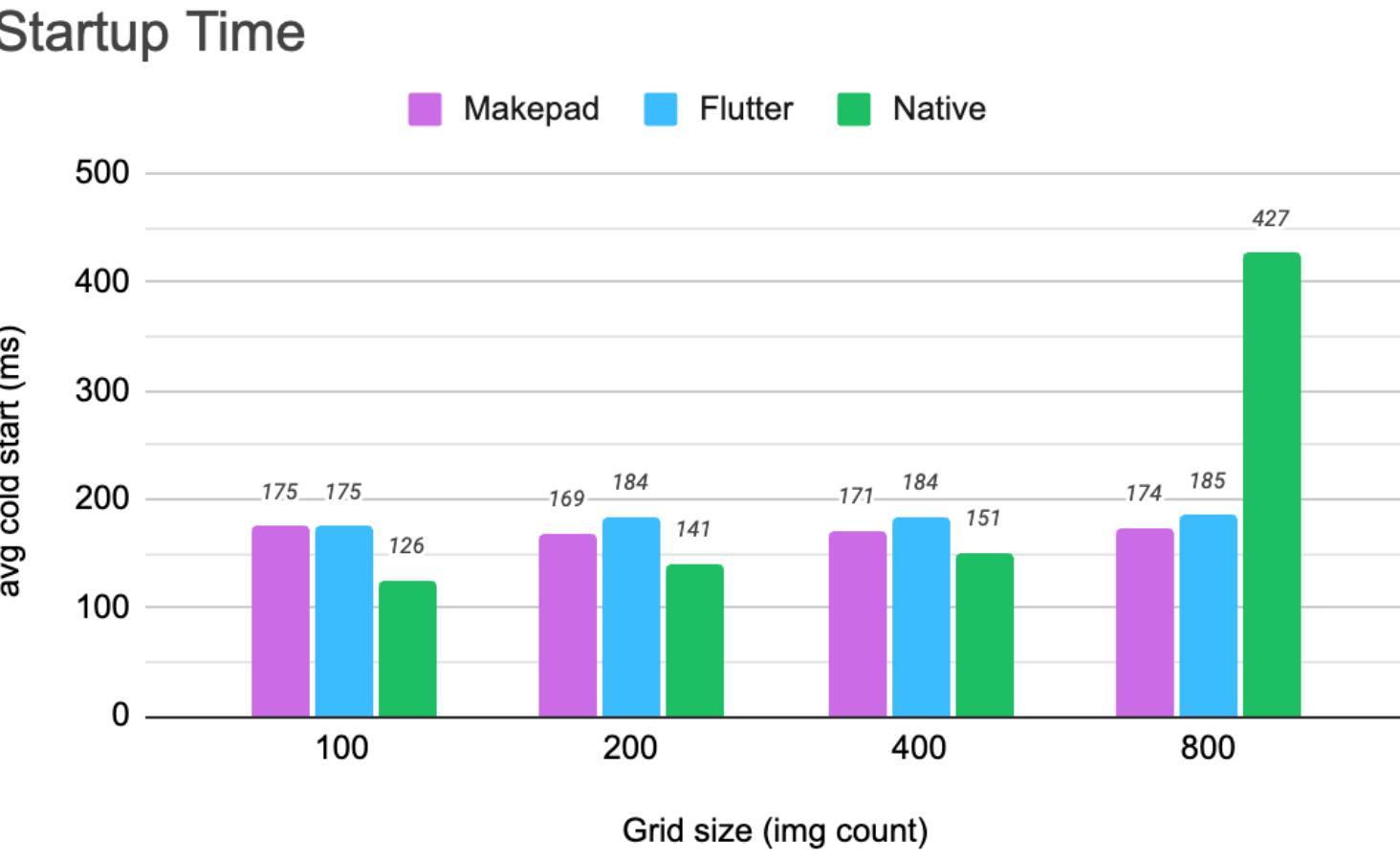
Rendering Performance: Downscaling

- Average time in milliseconds (ms)
- Time from launch to first render
- Lower the faster

Grid size (img count)	Makepad	Flutter	Native
100	172	176	224
200	181	182	254
400	172	196	320
800	182	199	450

Rendering Performance: Upscaling

- Average time in milliseconds (ms)
- Time from launch to first render
- Lower the faster



Grid size (img count)	Makepad	Flutter	Native
100	175	175	126
200	169	184	141
400	171	184	151
800	174	185	427

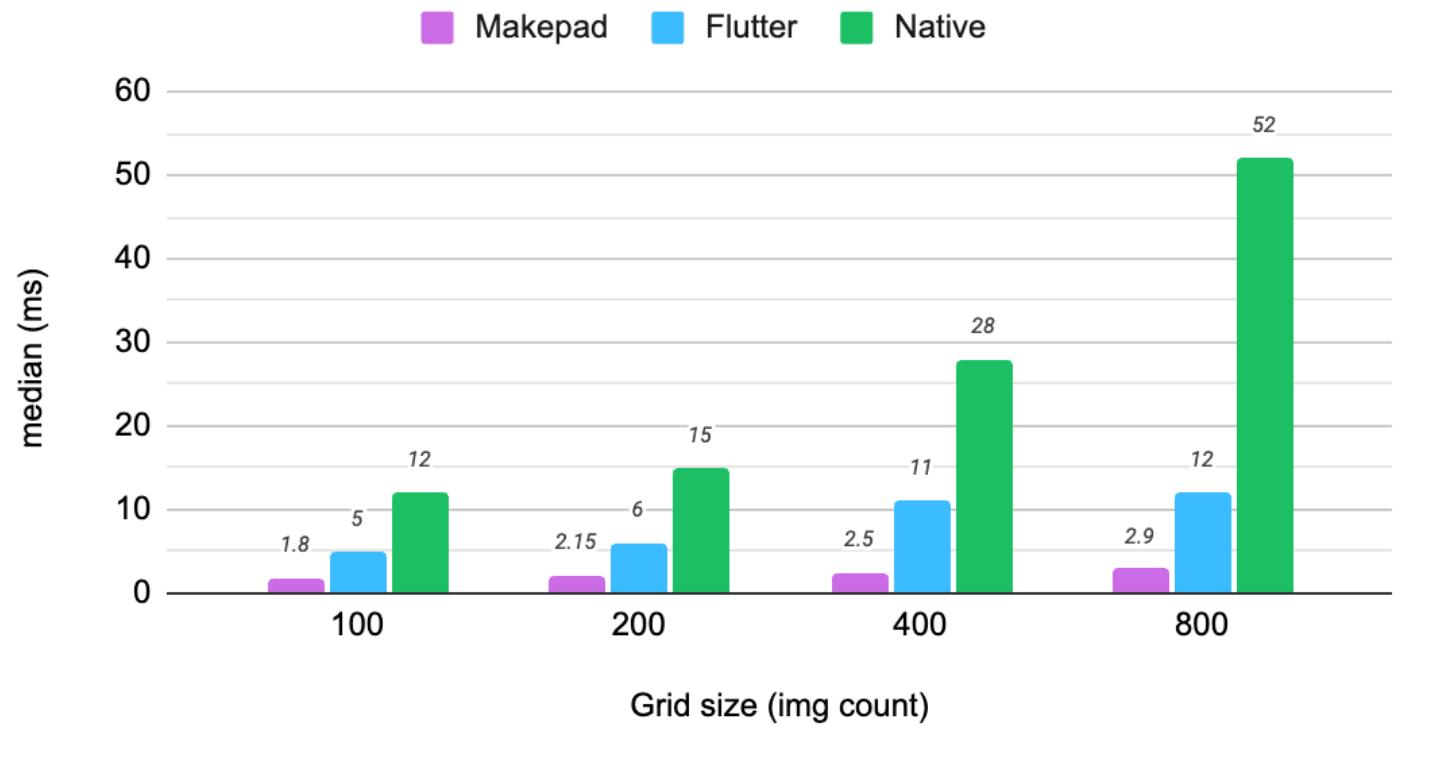


Responsiveness – Frame Time

- Frame Rate(FPS) is a misleading comparison
- Makepad & Native rely on Android's onDraw function
 - FPS is decided by the system and not by the UI kit
- Flutter owns their own render loop and has its own logic for determining the draw frequency
- Instead of FPS, we measure Frame Time
- Use `adb shell dumpsys gfxinfo <process> framestats`
 - Native & Flutter
- Makepad: added platform level code to measure time elapsed between draw calls

Frame Time

(time to draw each frame)



Rendering Performance: Downscaling

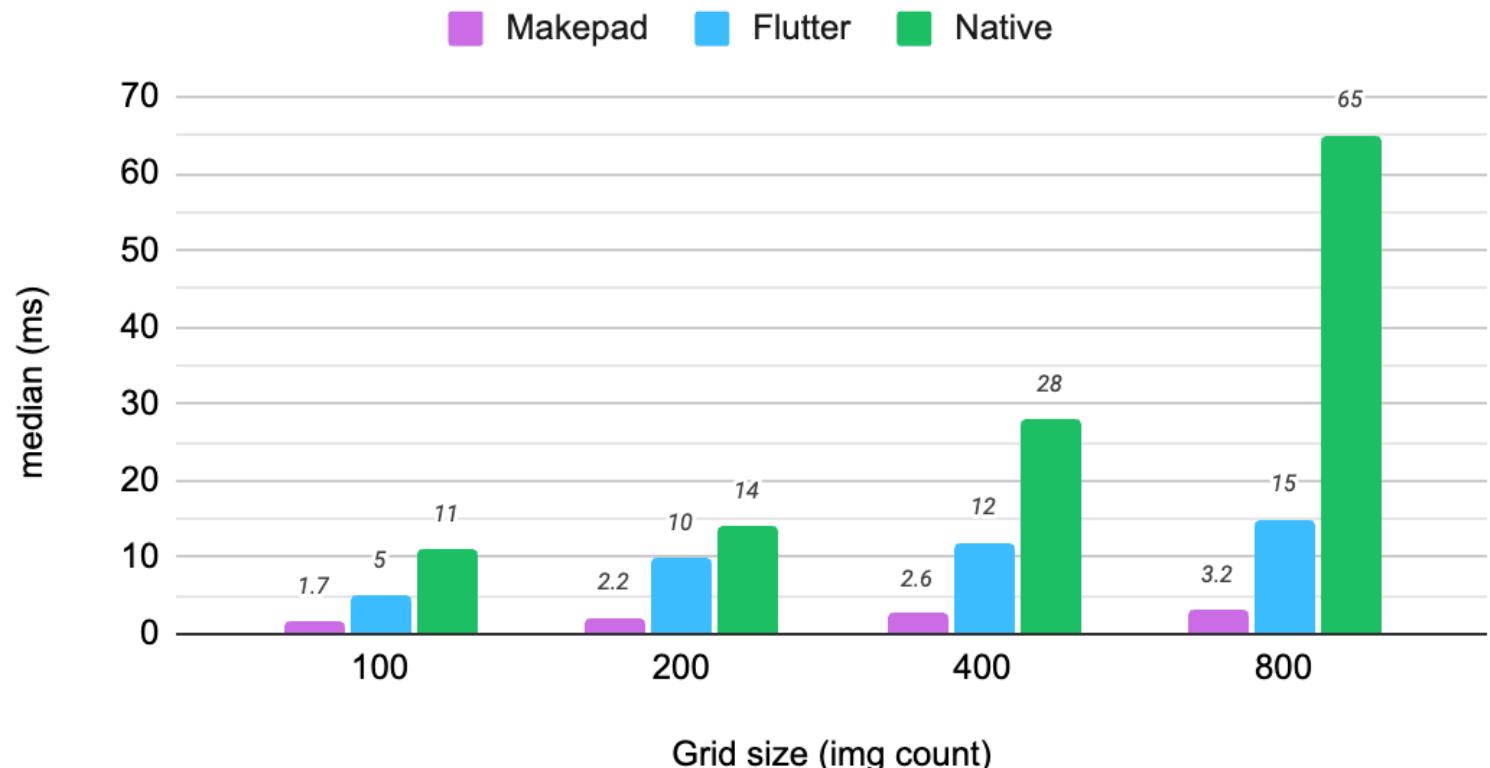
- Median time in milliseconds (ms)
- Time to draw each frame
- Lower the faster

Rendering Performance: Upscaling

- Median time in milliseconds (ms)
- Time to draw each frame
- Lower the faster

Frame Time

(time to draw each frame)



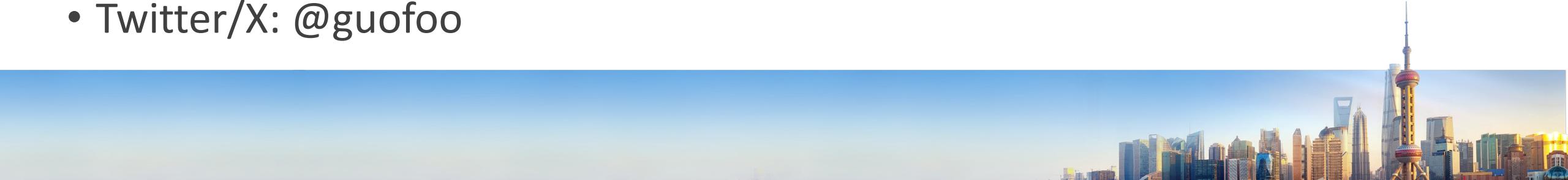
Grid size (img count)	Makepad	Flutter	Native
100	1.7	5	11
200	2.2	10	14
400	2.6	12	28
800	3.2	15	65

4. Conclusion



Conclusion

- These are very preliminary results – A snapshot during development
- Makepad's overall performance is on-par or better in many respects compared to the current industry standards
 - Especially for a super small team of developers
- Pull Requests are welcome
- Join the Makepad discord channel discussions
- https://github.com/project-robius/makepad_image_manipulation
- Twitter/X: @guofoo



Demo Videos





THANK YOU



Thank You