

LruCache Data Flow Architecture

Comparing file-based loading versus memory-cached retrieval in Android image processing

WITHOUT LruCache

File I/O on every access

11,078 ms

FILE STORAGE
assets/sample.png
880 KB · PNG compressed

read stream

BitmapFactory
decodeStream()
~50-100 ms

decode

Bitmap
953 × 586 ARGB_8888
2.1 MB in memory

ImageProcessor
superResolution2x()
Bicubic interpolation
~10,500 ms

Result Bitmap
1906 × 1172
8.5 MB · 2× upscaled

x 5 iterations

WITH LruCache

Memory-cached source image

10,470 ms

FILE STORAGE
first access only

initial load

LruCache<String, Bitmap>
android.util.LruCache
key: "source_image"
size: maxMemory / 8
< 1 ms access

get()

Bitmap
953 × 586 ARGB_8888
from cache · no I/O

ImageProcessor
superResolution2x()
Bicubic interpolation
~10,470 ms

Result Bitmap
1906 × 1172
8.5 MB · 2× upscaled

x 5 from cache

Benchmark Summary

Without cache: Storage Decode Process = 11,078 ms

With cache: LruCache Process = 10,470 ms

Time saved: 608 ms per iteration

Speedup: 1.06× (6% faster)