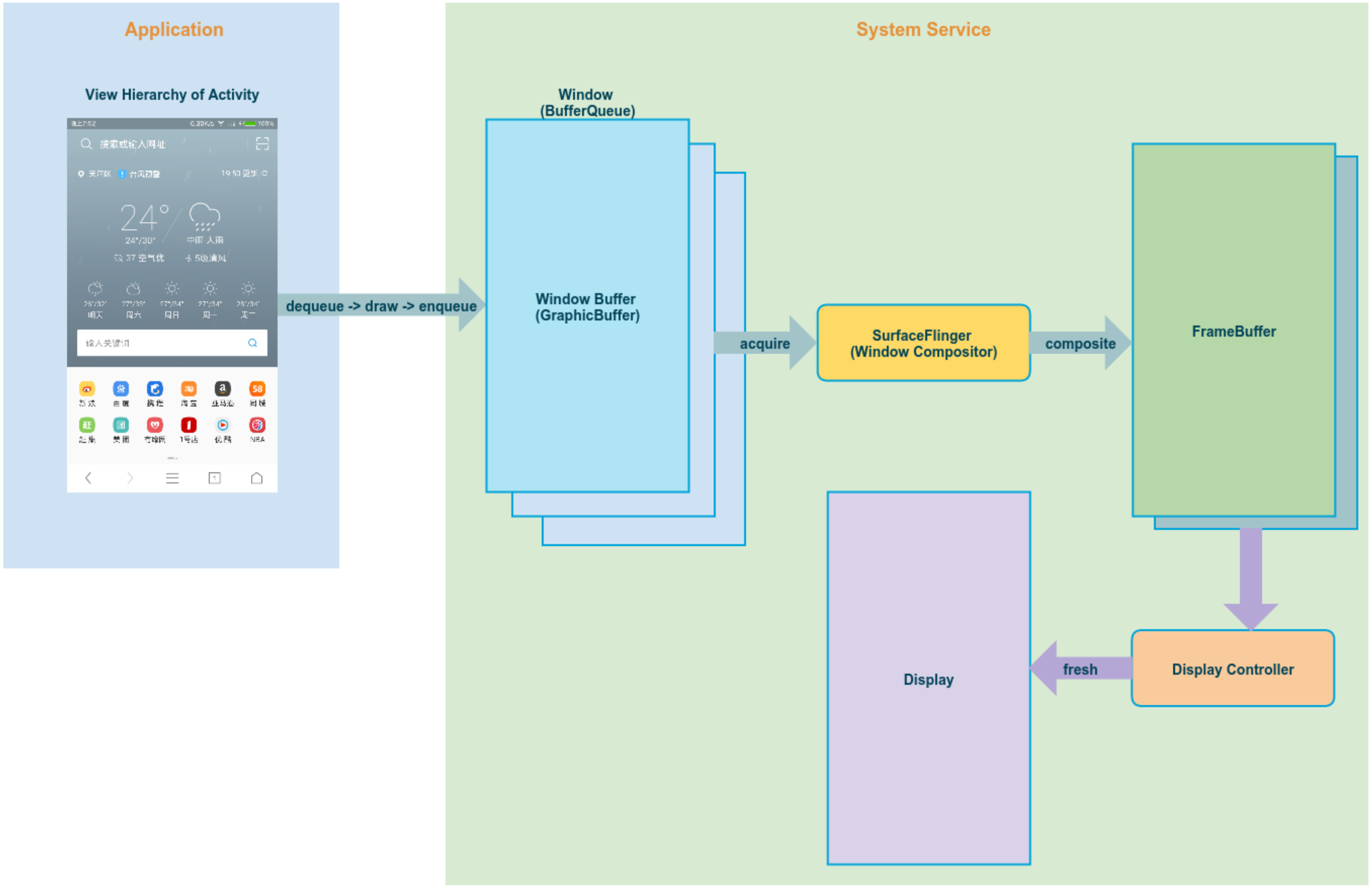
Android 渲染流水线 设计与演化

Roger

yixx@ucweb.com

roger2yi@gmail.com


Application
View Hierarchy of Activity
dequeue -> draw -> enqueue
System Service
Window (BufferQueue)
Window Buffer (GraphicBuffer)
acquire
SurfaceFlinger (Window Compositor)
composite
FrameBuffer
Display Controller
fresh
Display

2.x

- Draw Views by CPU (via Skia)
- Composite Window by GPU (via GLES)
- Some Soc support overlay

3.x

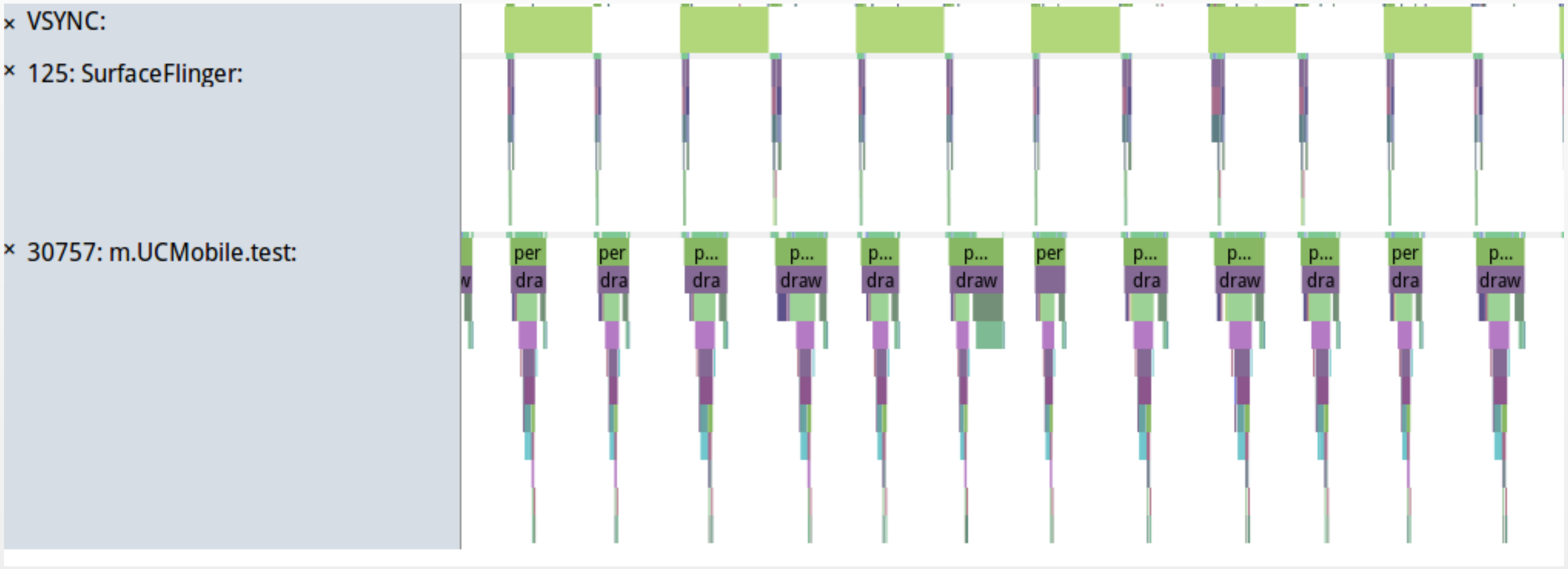
- Draw Views by both CPU/GPU
  - **getDisplayList** - Record DisplayList in View.onDraw
  - **drawDisplayList** - OpenGLRenderer turn DisplayList into GL drawing commands, issue to GPU
  - Support hardware layer of View, usually used in View animation
- Composite Window by Display Controller instead of GPU (via Hardware Composer)

4.x - Project Butter

- VSync
- Triple buffers
- Android sync framework

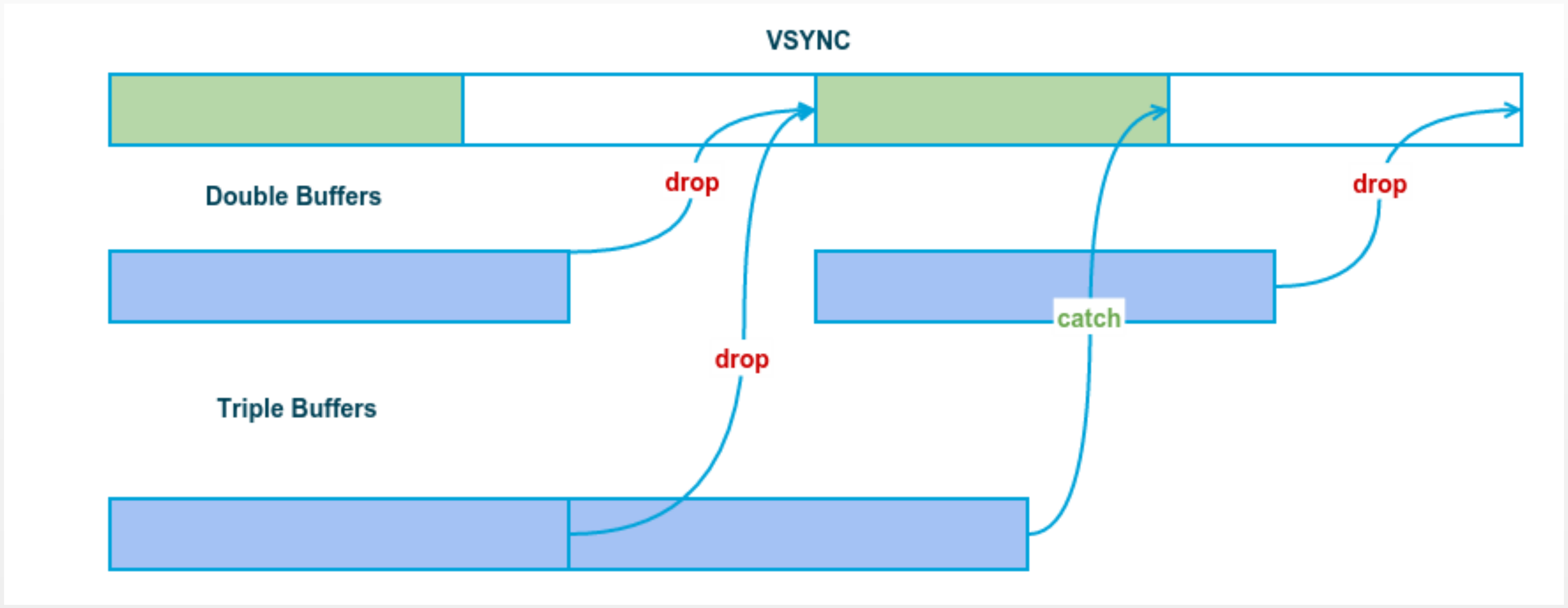
VSync

- VSYNC synchronizes certain events to the refresh cycle of the display
  - Applications start drawing on a VSYNC boundary
  - SurfaceFlinger composites on a VSYNC boundary


× VSYNC:
× 125: SurfaceFlinger:
× 30757: m.UCMobile.test:
per
dra
p...
draw

Triple buffers

- Slove the issue introduce by Hardware Composer
- Avoid drop frames continuously


VSYNC
Double Buffers
drop
drop
Triple Buffers
drop
catch

Android Sync Framework

- Application can just issue GL drawing commands to GPU， then enqueue the buffer, then **RETURN** before rendering completes
- The buffer is accompanied by a fence that signals when the contents are ready
- SF or HWC can wait on the fence until signaled

5.x

- Individual gpu thread - Render Thread
  - getDisplayList in main thread, but drawDisplayList in render thread, offload the main thread
  - getDisplayList and drawDisplayList are async, can run parallelized
  - Some view animations run solely by render thread

Evolve

- CPU -> GPU -> Display Controller
- Async and parallelized
- Offload main thread of application

Golden Rules for Butter Graphics

- Whatever you can do in another thread, then do it in another thread
- Whatever you must do in main thread, then do it fast
- Always profiling, it is your most dear friend

Reference

- https://source.android.com/devices/graphics/index.html
- http://blog.csdn.net/rogeryi/article/details/8724233

The End

Thank you for your listening

Yours Sincerely, Roger