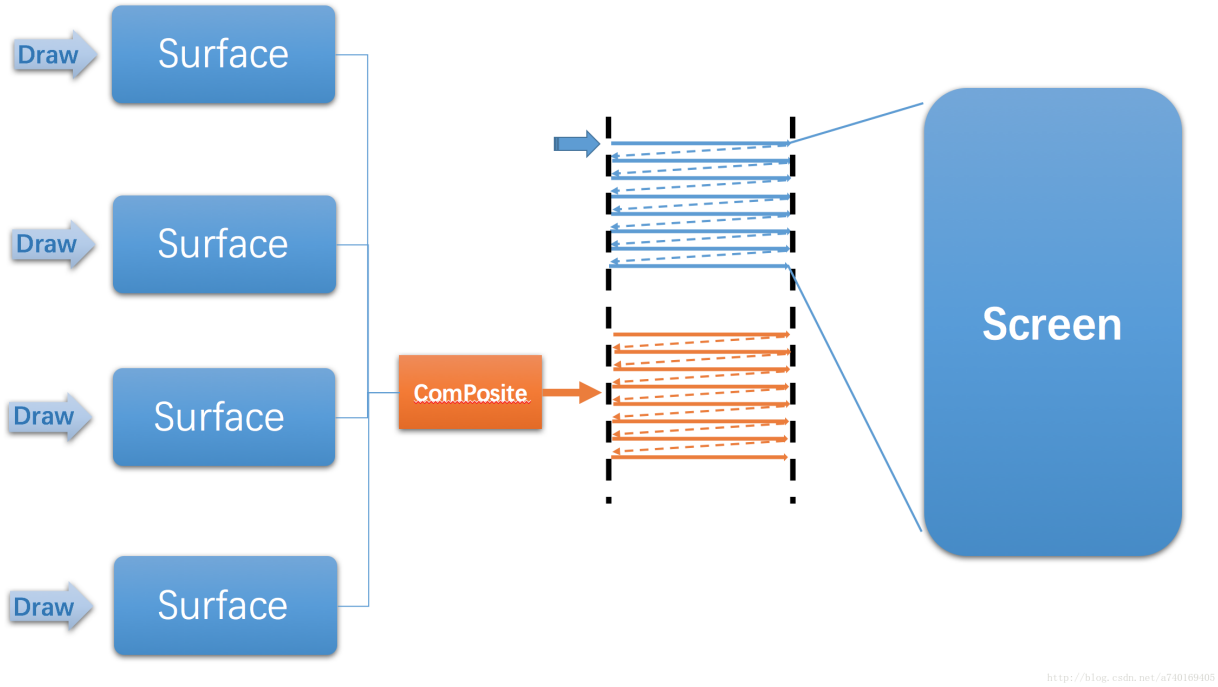
TripleBuffer的理解

我们先理解三个对象应用程序app、SurfaceFlinger、屏幕(Hardware)



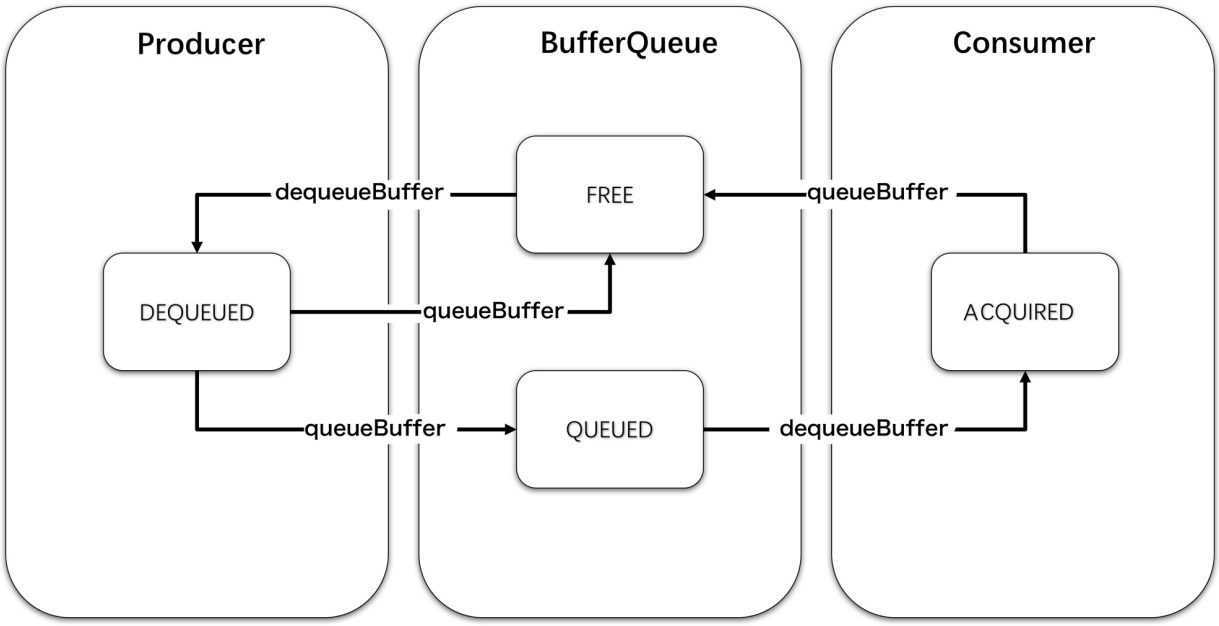
1、Buffer 数据扭转-三者之间的联通媒介

- app->SurfaceFlinger: 通过surface进行通信
- SurfaceFlinger->屏幕: 通过帧缓冲进行通信

SF作为surface合成器，将当前所有要显示的surface对应的BufferQueue中的buffer进行合成，最终提交到帧缓冲区交给屏幕进行显示。

不要把这里的buffer与屏幕使用的帧缓冲搞混，这是两个概念， 某一时刻，多个要显示的surface中的buffer会被合成成一个屏幕帧缓冲

app与surfaceFlinger的交互时机时通过surface内部维护的buffer进行图像交换，surface内部可能存在多个buffer，形成bufferqueue。 这个bufferQueue里面的buffer，我们知道surface实际是surfaceFlinger进程中进行管理的，app获取的surface对象实际都会在SF当中， 这里的buffer会被谁持有并进行什么操作呢?



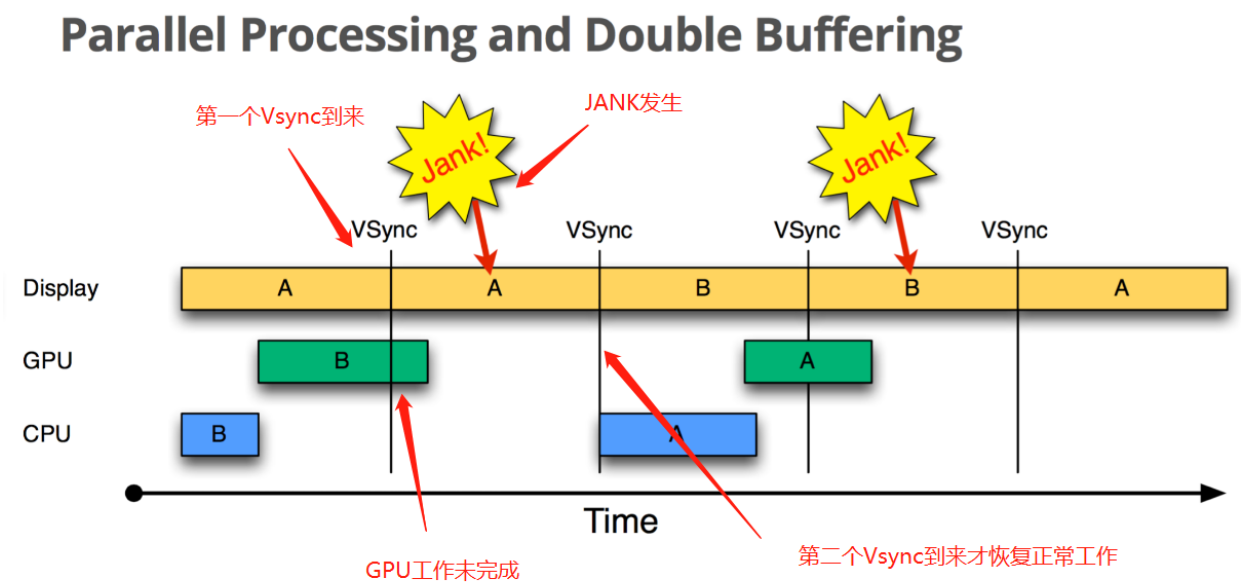
上图对应BufferQueue的生产者消费者模型，生产者一般是上层的APP， 消费者是SF。

在Android系统中，app作为生产者首先将FREE状态的Buffer通过dequeueBuffer获取后， buffer转变为DEQUQUED状态，最终经过CPU与GPU的处理工作后，通过queueBuffer接口将buffer归还给BufferQueue，并成为Queued状态，SF通过dequeueBuffer获取buffer， buffer扭转为ACQUIED状态， 最终SF合成消费结束后归还Buffer，buffer重回FREE状态。

- buffer被app层获取时【DEQUEUED】：
  - CPU：
    1、Measure
    2、Layout
    3、Draw
    4、Polygons Texture: 生成多边形和纹理 发送给GPU。
  - GPU：Rasterization对多边形和纹理进行栅格化操作。
    栅格化是指将矢量图形（如线条、曲线、文字等）转化为像素阵列（栅格图像）的过程。在计算机图形学和图像处理中，栅格化是非常常见的操作。在栅格化过程中，需要将矢量图形中的每个点映射到栅格图像中的像素位置，并根据颜色、透明度等属性为每个像素赋值。这个过程可以通过算法实现，最常见的算法是扫描线算法和边界填充算法。栅格化后的图像可以方便地进行存储、显示和处理，是计算机图形学和图像处理的基础。
- Buffer在SF中的时候【ACQUIRED】：
  用于与其他Surface中的buffer一起合成帧缓冲。

2、双缓冲

先看看双缓冲：



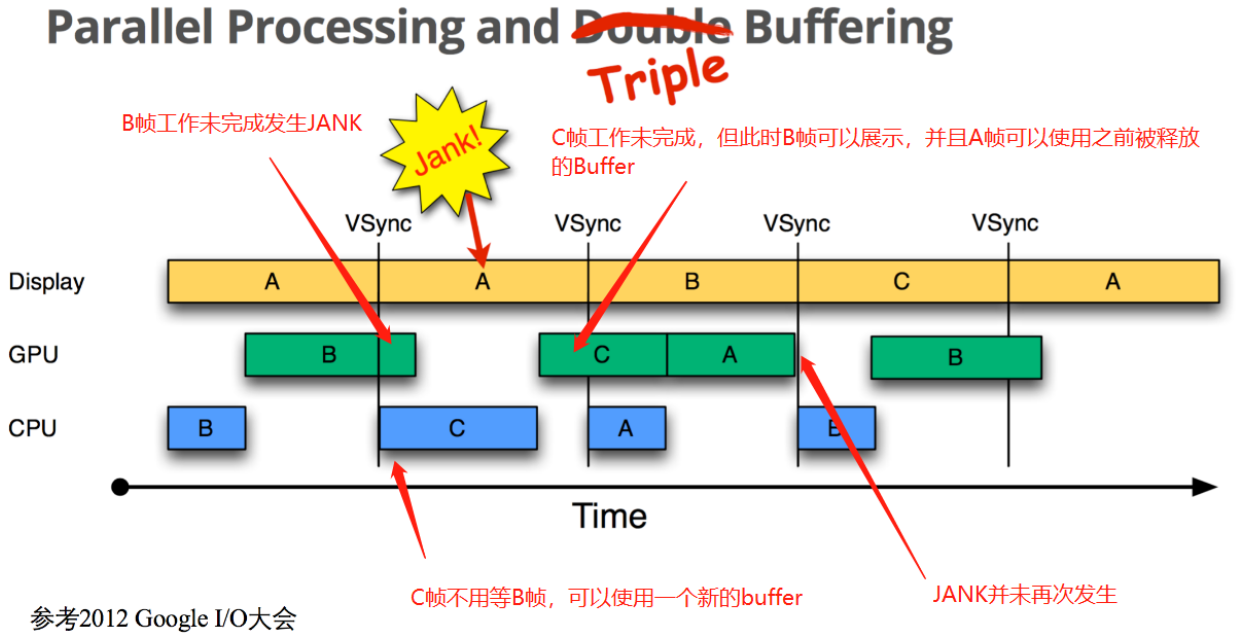
参考2012 Google I/O大会

我们看上图中第一个VSYNC(表姐)到来时，GPU对B帧的工作未完成，此时Surface中的BufferQueue中的buffer在【DEQUEUED】状态，gpu正在对其操作。 此时还会有另一个buffer在【ACQUIRED】状态，SF正在对其进行合成屏幕帧缓冲， 那么此次Vsync到来时，CPU并没有办法进行下一帧的操作。 那么此时JANK发生。 直到第二个VSYNC才恢复正常。**但是假如第二帧的绘制也超过16.6ms, 那么jank又会发生。**

这就是双缓冲的局限性

3、三缓冲

为了解决上述的问题，google引入了三缓冲，增加多了一个buffer， 其实就是在bufferQueue中，最多可以存在多个buffer。



参考2012 Google I/O大会

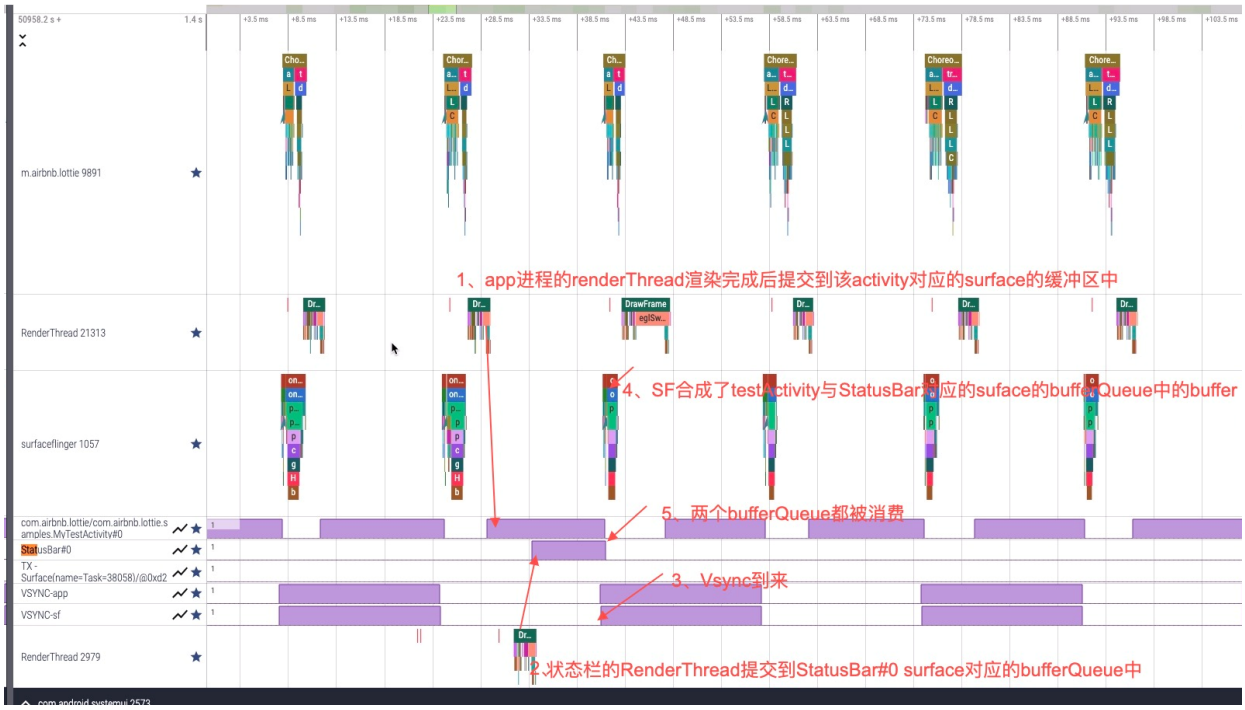
根据上图可以看到，由于多出一个buffer，当JANK发生是，下一帧可以提前开始工作，尽量减少后面继续发生JANK的可能性。

可以看到三缓冲并不是完全解决了JANK，而是在JANK发生时降低持续jank的可能性。

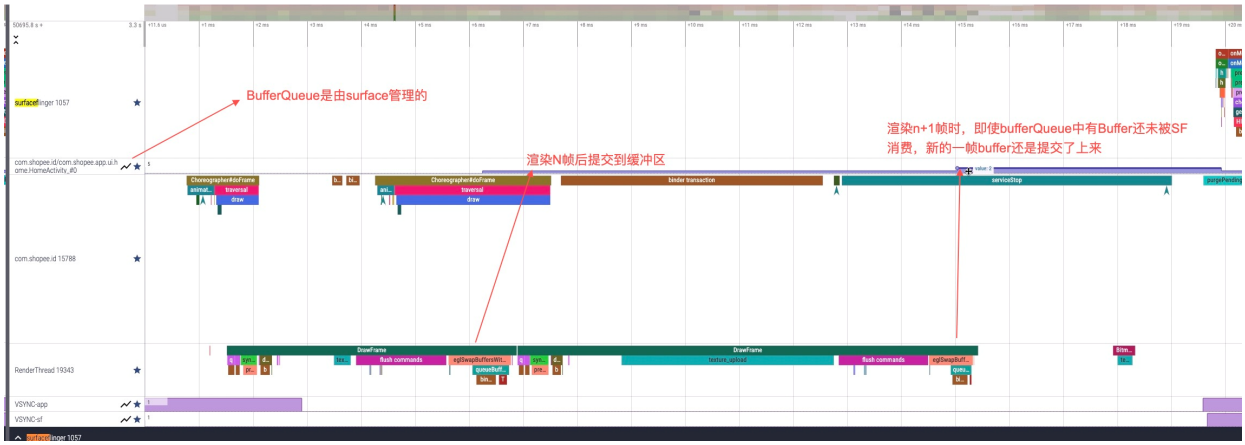
4、Systrace实际验证三缓冲的工作：

以下是一个正常情况下工作的bufferQueue中的生产者消费者模式：

- StatusBar#0： 显示的是状态栏对应的surface当中，状态为【QUEUED】的buffer数量。
  【QUEUED】状态代表CPU与GPU工作完成后通过queueBuffer提交给BufferQueue，等待SF消费，消费后【QUEUED】状态的buffer数量会-1。
- MyTestActivity#0： 显示的是MyTestActivity对应的surface当中，状态为【QUEUED】的buffer数量。



以下是三缓冲发挥作用时的case：



可以看到在HomeActivity对应的Surface的BuffeQueue当中，出现了两个【QUEUED】状态的buffer。 在双缓冲的情况下，我们知道为【QUEUED】状态的buffer数量是不超过1的。

双缓冲情况下两个buffer的情况有以下几种情况：

- 【FREE】【FREE】
- 【DEQUEUED】【ACQUIRD】
- 【QUEUED】【FREE】
- 【QUEUED】【ACQUIRD】

DONE