Metal Performance Optimization Techniques

Session 610

Philip Bennett GPU Software Performance
Serhat Tekin GPU Software Developer Technologies

Metal at WWDC

What's New in Metal, Part 1

- Metal in Review
- New Features
- Metal and App Thinning

What's New in Metal, Part 2

- Introducing MetalKit
- Metal Perfomance Shaders

Metal Performance Optimization Techniques

- Metal System Trace Tool
- Metal Best Practices

Metal at WWDC

What's New in Metal, Part 1

- Metal in Review
- New Features
- Metal and App Thinning

What's New in Metal, Part 2

- Introducing MetalKit
- Metal Perfomance Shaders

Metal Performance Optimization Techniques

- Metal System Trace Tool
- Metal Best Practices

Metal at WWDC

What's New in Metal, Part 1

- Metal in Review
- New Features
- Metal and App Thinning

What's New in Metal, Part 2

- Introducing MetalKit
- Metal Perfomance Shaders

Metal Performance Optimization Techniques

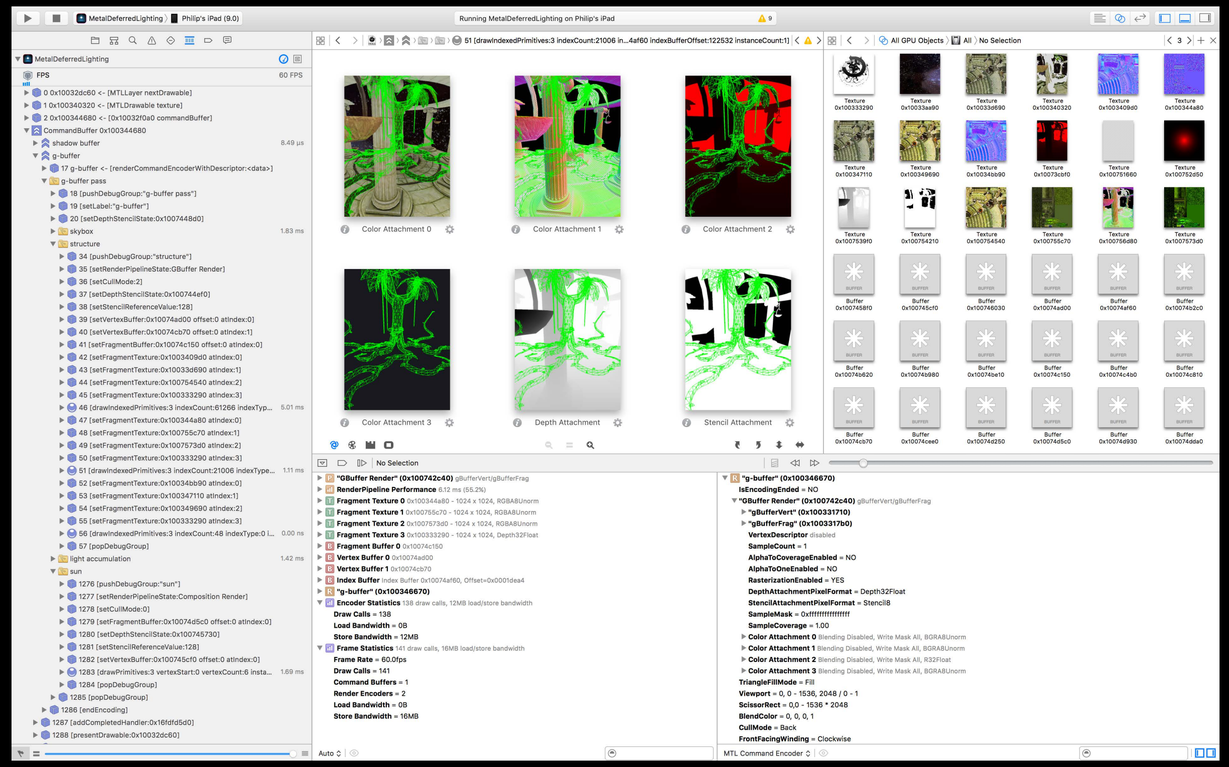
- Metal System Trace Tool
- Metal Best Practices

Metal Debugging and Profiling Tools


XCODE

Metal Tools in Xcode

- Visual frame debugger
- Resource and state viewer
- Shader profiling
- Shader edit-and-continue
- Debug mode for Metal framework
- Integrated offline Metal compiler



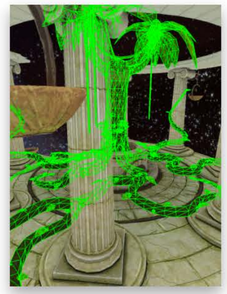
MetalDeferredLighting 〉 Philip's iPad (9.0)

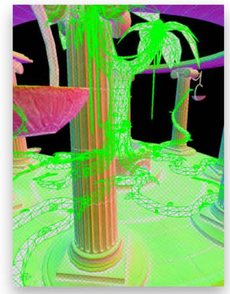
Running MetalDeferredLighting on Philip's iPad 9

51 [drawIndexedPrimitives:3 indexCount:21006 in...4af60 indexBufferOffset:122532 instanceCount:1]

All GPU Objects 〉 All 〉 No Selection

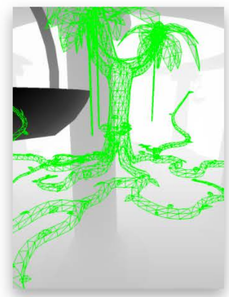
- MetalDeferredLighting
  - FPS — 60 FPS
  - 0 0x10032dc60 <- [MTLLayer nextDrawable]
  - 1 0x100340320 <- [MTLDrawable texture]
  - 2 0x100344680 <- [0x10032f0a0 commandBuffer]
  - CommandBuffer 0x100344680
    - shadow buffer — 8.49 µs
    - g-buffer
      - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
      - g-buffer pass
        - 18 [pushDebugGroup:"g-buffer pass"]
        - 19 [setLabel:"g-buffer"]
        - 20 [setDepthStencilState:0x1007448d0]
        - skybox — 1.83 ms
        - structure
          - 34 [pushDebugGroup:"structure"]
          - 35 [setRenderPipelineState:GBuffer Render]
          - 36 [setCullMode:2]
          - 37 [setDepthStencilState:0x100744ef0]
          - 38 [setStencilReferenceValue:128]
          - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
          - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
          - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0]
          - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
          - 43 [setFragmentTexture:0x10033d690 atIndex:1]
          - 44 [setFragmentTexture:0x100754540 atIndex:2]
          - 45 [setFragmentTexture:0x100333290 atIndex:3]
          - 46 [drawIndexedPrimitives:3 indexCount:61266 indexTyp... — 5.01 ms
          - 47 [setFragmentTexture:0x100344a80 atIndex:0]
          - 48 [setFragmentTexture:0x100755c70 atIndex:1]
          - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
          - 50 [setFragmentTexture:0x100333290 atIndex:3]
          - 51 [drawIndexedPrimitives:3 indexCount:21006 indexType... — 1.11 ms
          - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
          - 53 [setFragmentTexture:0x100347110 atIndex:1]
          - 54 [setFragmentTexture:0x100349690 atIndex:2]
          - 55 [setFragmentTexture:0x100333290 atIndex:3]
          - 56 [drawIndexedPrimitives:3 indexCount:48 indexType:0 i... — 0.00 ns
          - 57 [popDebugGroup]
        - light accumulation — 1.42 ms
        - sun
          - 1276 [pushDebugGroup:"sun"]
          - 1277 [setRenderPipelineState:Composition Render]
          - 1278 [setCullMode:0]
          - 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atIndex:0]
          - 1280 [setDepthStencilState:0x100745730]
          - 1281 [setStencilReferenceValue:128]
          - 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
          - 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 insta... — 1.69 ms
          - 1284 [popDebugGroup]
        - 1285 [popDebugGroup]
      - 1286 [endEncoding]
    - 1287 [addCompletedHandler:0x16fdfd5d0]
    - 1288 [presentDrawable:0x10032dc60]


Color Attachment 0

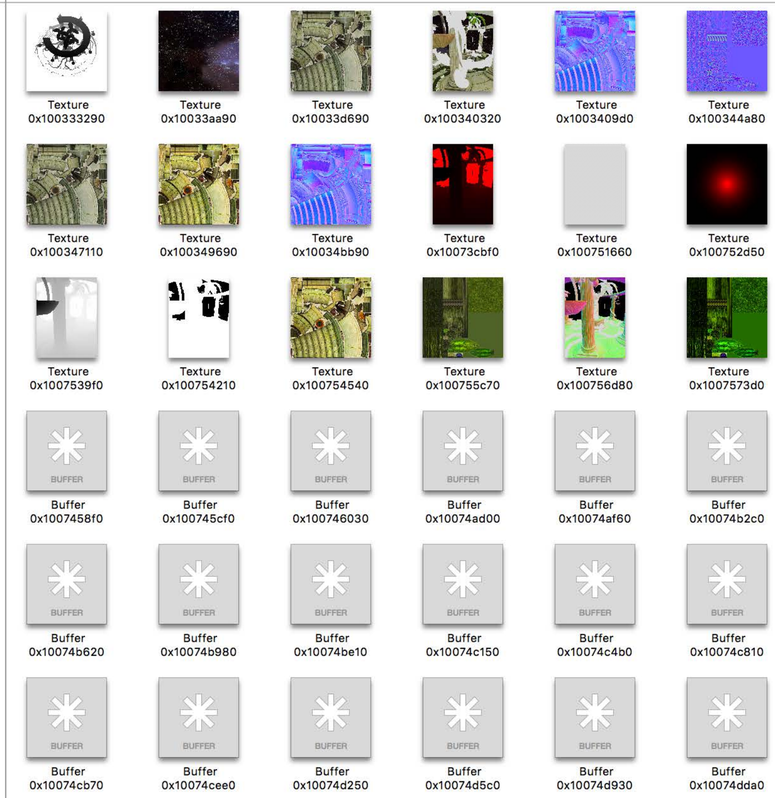

Color Attachment 1


Color Attachment 2


Color Attachment 3


Depth Attachment


Stencil Attachment



No Selection

- "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
- RenderPipeline Performance 6.12 ms (55.2%)
- Fragment Texture 0 0x100344a80 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 1 0x100755c70 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 2 0x1007573d0 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 3 0x100333290 - 1024 x 1024, Depth32Float
- Fragment Buffer 0 0x10074c150
- Vertex Buffer 0 0x10074ad00
- Vertex Buffer 1 0x10074cb70
- Index Buffer Index Buffer 0x10074af60, Offset=0x0001dea4
- "g-buffer" (0x100346670)
- Encoder Statistics 138 draw calls, 12MB load/store bandwidth
  - Draw Calls = 138
  - Load Bandwidth = 0B
  - Store Bandwidth = 12MB
- Frame Statistics 141 draw calls, 16MB load/store bandwidth
  - Frame Rate = 60.0fps
  - Draw Calls = 141
  - Command Buffers = 1
  - Render Encoders = 2
  - Load Bandwidth = 0B
  - Store Bandwidth = 16MB

Auto

- "g-buffer" (0x100346670)
  - IsEncodingEnded = NO
  - "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
    - "gBufferVert" (0x100331710)
    - "gBufferFrag" (0x1003317b0)
    - VertexDescriptor disabled
    - SampleCount = 1
    - AlphaToCoverageEnabled = NO
    - AlphaToOneEnabled = NO
    - RasterizationEnabled = YES
    - DepthAttachmentPixelFormat = Depth32Float
    - StencilAttachmentPixelFormat = Stencil8
    - SampleMask = 0xffffffffffffffff
    - SampleCoverage = 1.00
    - Color Attachment 0 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 1 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 2 Blending Disabled, Write Mask All, R32Float
    - Color Attachment 3 Blending Disabled, Write Mask All, BGRA8Unorm
  - TriangleFillMode = Fill
  - Viewport = 0, 0 - 1536, 2048 / 0 - 1
  - ScissorRect = 0,0 - 1536 * 2048
  - BlendColor = 0, 0, 0, 1
  - CullMode = Back
  - FrontFacingWinding = Clockwise

MTL Command Encoder

MetalDeferredLighting

FPS 60 FPS

- 0 0x10032dc60 <- [MTLLayer nextDrawable]
- 1 0x100340320 <- [MTLDrawable texture]
- 2 0x100344680 <- [0x10032f0a0 commandBuffer]
- CommandBuffer 0x100344680
  - shadow buffer 8.49 µs
  - g-buffer
    - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
    - g-buffer pass
      - 18 [pushDebugGroup:"g-buffer pass"]
      - 19 [setLabel:"g-buffer"]
      - 20 [setDepthStencilState:0x1007448d0]
      - skybox 1.83 ms
      - structure
        - 34 [pushDebugGroup:"structure"]
        - 35 [setRenderPipelineState:GBuffer Render]
        - 36 [setCullMode:2]
        - 37 [setDepthStencilState:0x100744ef0]
        - 38 [setStencilReferenceValue:128]
        - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
        - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
        - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0]
        - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
        - 43 [setFragmentTexture:0x10033d690 atIndex:1]
        - 44 [setFragmentTexture:0x100754540 atIndex:2]
        - 45 [setFragmentTexture:0x100333290 atIndex:3]
        - 46 [drawIndexedPrimitives:3 indexCount:61266 indexTyp... 5.01 ms
        - 47 [setFragmentTexture:0x100344a80 atIndex:0]
        - 48 [setFragmentTexture:0x100755c70 atIndex:1]
        - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
        - 50 [setFragmentTexture:0x100333290 atIndex:3]
        - 51 [drawIndexedPrimitives:3 indexCount:21006 indexType... 1.11 ms
        - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
        - 53 [setFragmentTexture:0x100347110 atIndex:1]

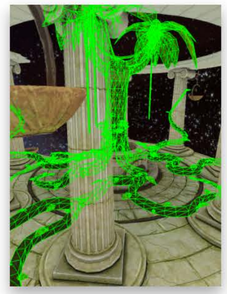
MetalDeferredLighting 〉 Philip's iPad (9.0)

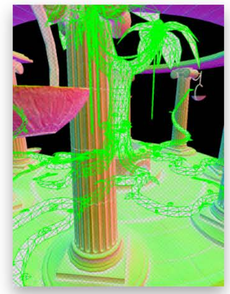
Running MetalDeferredLighting on Philip's iPad 9

51 [drawIndexedPrimitives:3 indexCount:21006 in...4af60 indexBufferOffset:122532 instanceCount:1]

All GPU Objects 〉 All 〉 No Selection

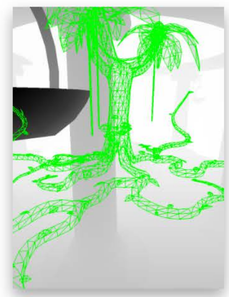
- MetalDeferredLighting
  - FPS — 60 FPS
  - 0 0x10032dc60 <- [MTLLayer nextDrawable]
  - 1 0x100340320 <- [MTLDrawable texture]
  - 2 0x100344680 <- [0x10032f0a0 commandBuffer]
  - CommandBuffer 0x100344680
    - shadow buffer — 8.49 µs
    - g-buffer
      - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
      - g-buffer pass
        - 18 [pushDebugGroup:"g-buffer pass"]
        - 19 [setLabel:"g-buffer"]
        - 20 [setDepthStencilState:0x1007448d0]
        - skybox — 1.83 ms
        - structure
          - 34 [pushDebugGroup:"structure"]
          - 35 [setRenderPipelineState:GBuffer Render]
          - 36 [setCullMode:2]
          - 37 [setDepthStencilState:0x100744ef0]
          - 38 [setStencilReferenceValue:128]
          - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
          - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
          - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0]
          - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
          - 43 [setFragmentTexture:0x10033d690 atIndex:1]
          - 44 [setFragmentTexture:0x100754540 atIndex:2]
          - 45 [setFragmentTexture:0x100333290 atIndex:3]
          - 46 [drawIndexedPrimitives:3 indexCount:61266 indexTyp... — 5.01 ms
          - 47 [setFragmentTexture:0x100344a80 atIndex:0]
          - 48 [setFragmentTexture:0x100755c70 atIndex:1]
          - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
          - 50 [setFragmentTexture:0x100333290 atIndex:3]
          - 51 [drawIndexedPrimitives:3 indexCount:21006 indexType... — 1.11 ms
          - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
          - 53 [setFragmentTexture:0x100347110 atIndex:1]
          - 54 [setFragmentTexture:0x100349690 atIndex:2]
          - 55 [setFragmentTexture:0x100333290 atIndex:3]
          - 56 [drawIndexedPrimitives:3 indexCount:48 indexType:0 i... — 0.00 ns
          - 57 [popDebugGroup]
      - light accumulation — 1.42 ms
      - sun
        - 1276 [pushDebugGroup:"sun"]
        - 1277 [setRenderPipelineState:Composition Render]
        - 1278 [setCullMode:0]
        - 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atIndex:0]
        - 1280 [setDepthStencilState:0x100745730]
        - 1281 [setStencilReferenceValue:128]
        - 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
        - 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 insta... — 1.69 ms
        - 1284 [popDebugGroup]
      - 1285 [popDebugGroup]
    - 1286 [endEncoding]
  - 1287 [addCompletedHandler:0x16fdfd5d0]
  - 1288 [presentDrawable:0x10032dc60]


Color Attachment 0

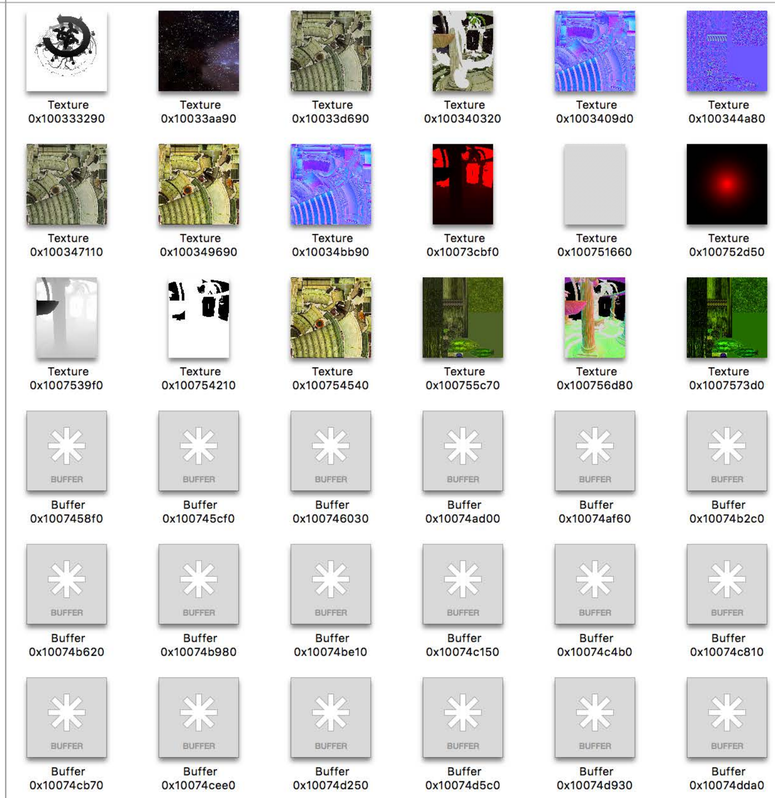

Color Attachment 1


Color Attachment 2


Color Attachment 3


Depth Attachment


Stencil Attachment



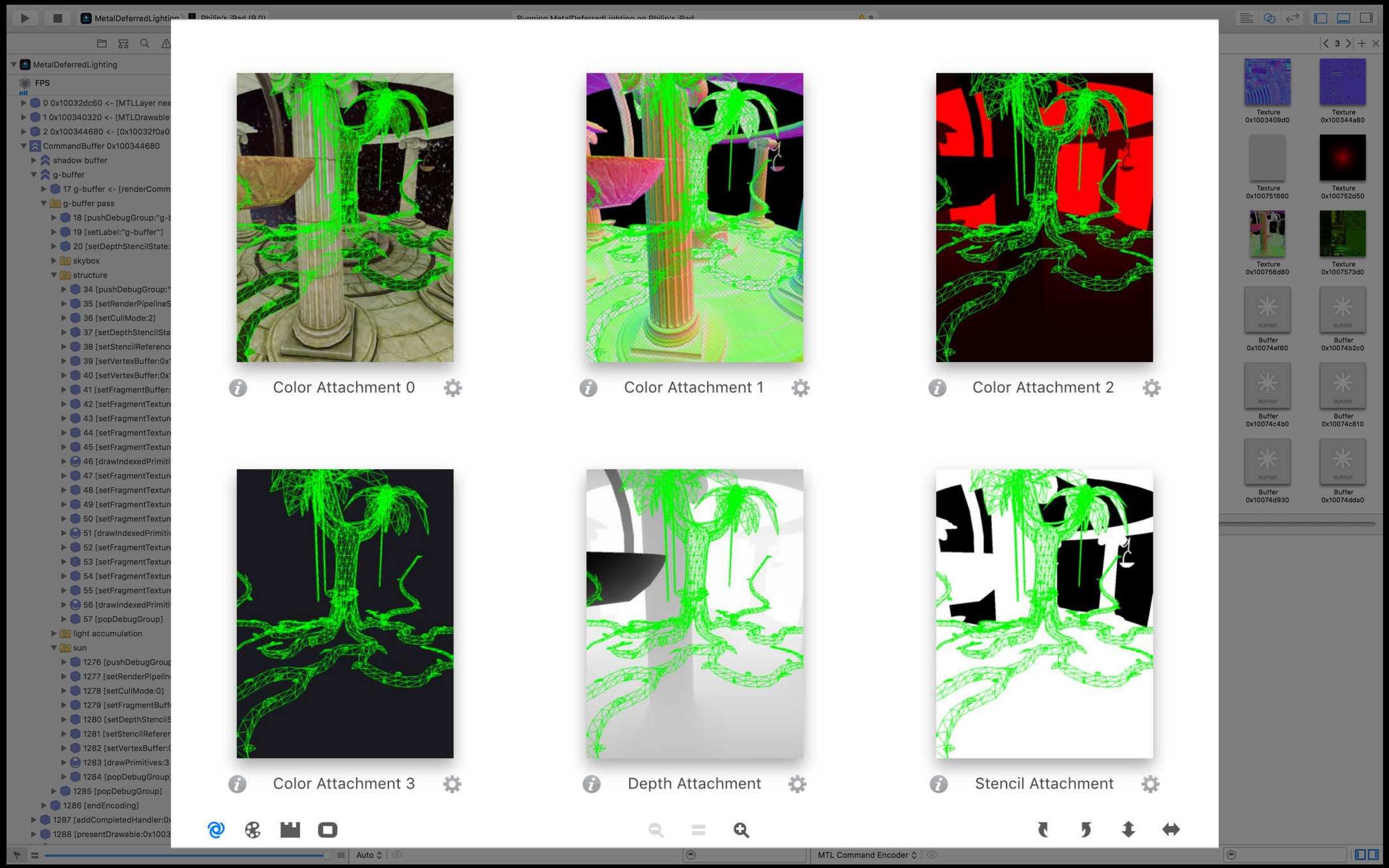
No Selection

- "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
- RenderPipeline Performance 6.12 ms (55.2%)
- Fragment Texture 0 0x100344a80 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 1 0x100755c70 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 2 0x1007573d0 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 3 0x100333290 - 1024 x 1024, Depth32Float
- Fragment Buffer 0 0x10074c150
- Vertex Buffer 0 0x10074ad00
- Vertex Buffer 1 0x10074cb70
- Index Buffer Index Buffer 0x10074af60, Offset=0x0001dea4
- "g-buffer" (0x100346670)
- Encoder Statistics 138 draw calls, 12MB load/store bandwidth
  - Draw Calls = 138
  - Load Bandwidth = 0B
  - Store Bandwidth = 12MB
- Frame Statistics 141 draw calls, 16MB load/store bandwidth
  - Frame Rate = 60.0fps
  - Draw Calls = 141
  - Command Buffers = 1
  - Render Encoders = 2
  - Load Bandwidth = 0B
  - Store Bandwidth = 16MB

Auto

- "g-buffer" (0x100346670)
  - IsEncodingEnded = NO
  - "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
    - "gBufferVert" (0x100331710)
    - "gBufferFrag" (0x1003317b0)
    - VertexDescriptor disabled
    - SampleCount = 1
    - AlphaToCoverageEnabled = NO
    - AlphaToOneEnabled = NO
    - RasterizationEnabled = YES
    - DepthAttachmentPixelFormat = Depth32Float
    - StencilAttachmentPixelFormat = Stencil8
    - SampleMask = 0xffffffffffffffff
    - SampleCoverage = 1.00
    - Color Attachment 0 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 1 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 2 Blending Disabled, Write Mask All, R32Float
    - Color Attachment 3 Blending Disabled, Write Mask All, BGRA8Unorm
  - TriangleFillMode = Fill
  - Viewport = 0, 0 - 1536, 2048 / 0 - 1
  - ScissorRect = 0,0 - 1536 * 2048
  - BlendColor = 0, 0, 0, 1
  - CullMode = Back
  - FrontFacingWinding = Clockwise

MTL Command Encoder


MetalDeferredLighting
FPS
0 0x10032dc60 <- [MTLLayer nex
1 0x100340320 <- [MTLDrawable
2 0x100344680 <- [0x10032f0a0
CommandBuffer 0x100344680
shadow buffer
g-buffer
17 g-buffer <- [renderComm
g-buffer pass
18 [pushDebugGroup:"g-b
19 [setLabel:"g-buffer"]
20 [setDepthStencilState:
skybox
structure
34 [pushDebugGroup:"
35 [setRenderPipelineS
36 [setCullMode:2]
37 [setDepthStencilSta
38 [setStencilReference
39 [setVertexBuffer:0x
40 [setVertexBuffer:0x
41 [setFragmentBuffer:
42 [setFragmentTexture
43 [setFragmentTexture
44 [setFragmentTexture
45 [setFragmentTexture
46 [drawIndexedPrimiti
47 [setFragmentTexture
48 [setFragmentTexture
49 [setFragmentTexture
50 [setFragmentTexture
51 [drawIndexedPrimiti
52 [setFragmentTexture
53 [setFragmentTexture
54 [setFragmentTexture
55 [setFragmentTexture
56 [drawIndexedPrimiti
57 [popDebugGroup]
light accumulation
sun
1276 [pushDebugGroup
1277 [setRenderPipelin
1278 [setCullMode:0]
1279 [setFragmentBuff
1280 [setDepthStencilS
1281 [setStencilReferen
1282 [setVertexBuffer:
1283 [drawPrimitives:3
1284 [popDebugGroup]
1285 [popDebugGroup]
1286 [endEncoding]
1287 [addCompletedHandler:0x
1288 [presentDrawable:0x1003
Color Attachment 0
Color Attachment 1
Color Attachment 2
Color Attachment 3
Depth Attachment
Stencil Attachment
Texture 0x1003409d0
Texture 0x100344a80
Texture 0x100751660
Texture 0x100752d50
Texture 0x100756d80
Texture 0x1007573d0
Buffer 0x10074af60
Buffer 0x10074b2c0
Buffer 0x10074c4b0
Buffer 0x10074c810
Buffer 0x10074d930
Buffer 0x10074dda0
Auto
MTL Command Encoder

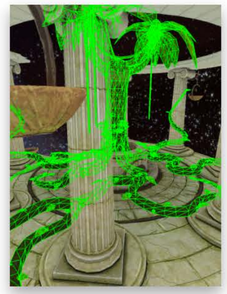
MetalDeferredLighting 〉 Philip's iPad (9.0)

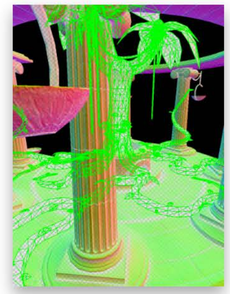
Running MetalDeferredLighting on Philip's iPad 9

51 [drawIndexedPrimitives:3 indexCount:21006 in...4af60 indexBufferOffset:122532 instanceCount:1]

All GPU Objects 〉 All 〉 No Selection

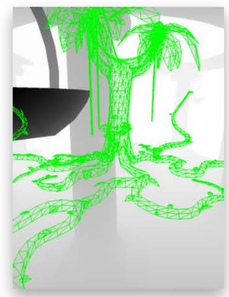
- MetalDeferredLighting
  - FPS — 60 FPS
  - 0 0x10032dc60 <- [MTLLayer nextDrawable]
  - 1 0x100340320 <- [MTLDrawable texture]
  - 2 0x100344680 <- [0x10032f0a0 commandBuffer]
  - CommandBuffer 0x100344680
    - shadow buffer — 8.49 µs
    - g-buffer
      - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
      - g-buffer pass
        - 18 [pushDebugGroup:"g-buffer pass"]
        - 19 [setLabel:"g-buffer"]
        - 20 [setDepthStencilState:0x1007448d0]
        - skybox — 1.83 ms
        - structure
          - 34 [pushDebugGroup:"structure"]
          - 35 [setRenderPipelineState:GBuffer Render]
          - 36 [setCullMode:2]
          - 37 [setDepthStencilState:0x100744ef0]
          - 38 [setStencilReferenceValue:128]
          - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
          - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
          - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0]
          - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
          - 43 [setFragmentTexture:0x10033d690 atIndex:1]
          - 44 [setFragmentTexture:0x100754540 atIndex:2]
          - 45 [setFragmentTexture:0x100333290 atIndex:3]
          - 46 [drawIndexedPrimitives:3 indexCount:61266 indexTyp... — 5.01 ms
          - 47 [setFragmentTexture:0x100344a80 atIndex:0]
          - 48 [setFragmentTexture:0x100755c70 atIndex:1]
          - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
          - 50 [setFragmentTexture:0x100333290 atIndex:3]
          - 51 [drawIndexedPrimitives:3 indexCount:21006 indexType... — 1.11 ms
          - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
          - 53 [setFragmentTexture:0x100347110 atIndex:1]
          - 54 [setFragmentTexture:0x100349690 atIndex:2]
          - 55 [setFragmentTexture:0x100333290 atIndex:3]
          - 56 [drawIndexedPrimitives:3 indexCount:48 indexType:0 i... — 0.00 ns
          - 57 [popDebugGroup]
        - light accumulation — 1.42 ms
        - sun
          - 1276 [pushDebugGroup:"sun"]
          - 1277 [setRenderPipelineState:Composition Render]
          - 1278 [setCullMode:0]
          - 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atIndex:0]
          - 1280 [setDepthStencilState:0x100745730]
          - 1281 [setStencilReferenceValue:128]
          - 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
          - 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 insta... — 1.69 ms
          - 1284 [popDebugGroup]
        - 1285 [popDebugGroup]
      - 1286 [endEncoding]
    - 1287 [addCompletedHandler:0x16fdfd5d0]
    - 1288 [presentDrawable:0x10032dc60]


Color Attachment 0

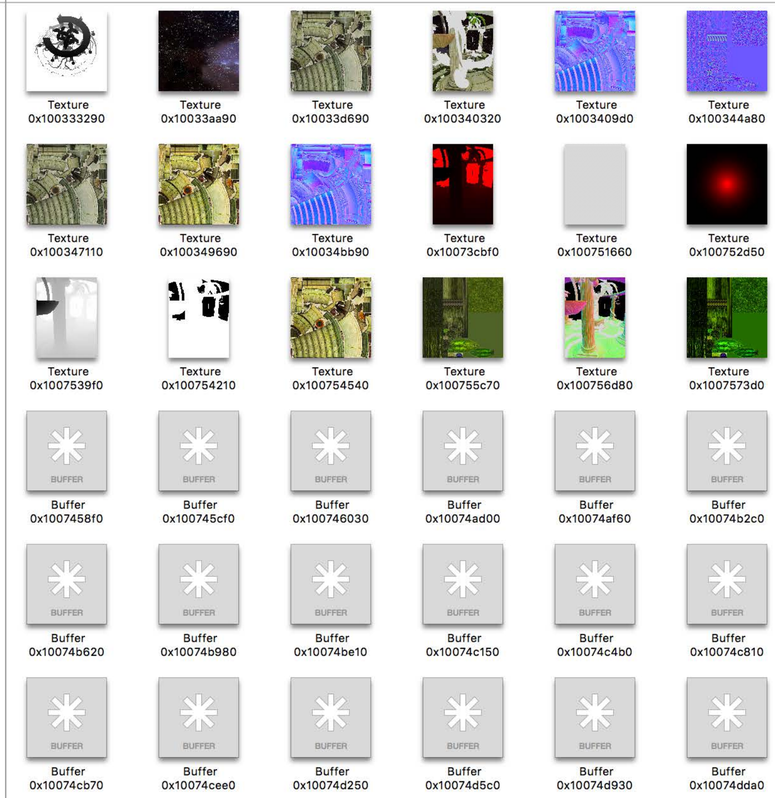

Color Attachment 1


Color Attachment 2


Color Attachment 3


Depth Attachment


Stencil Attachment



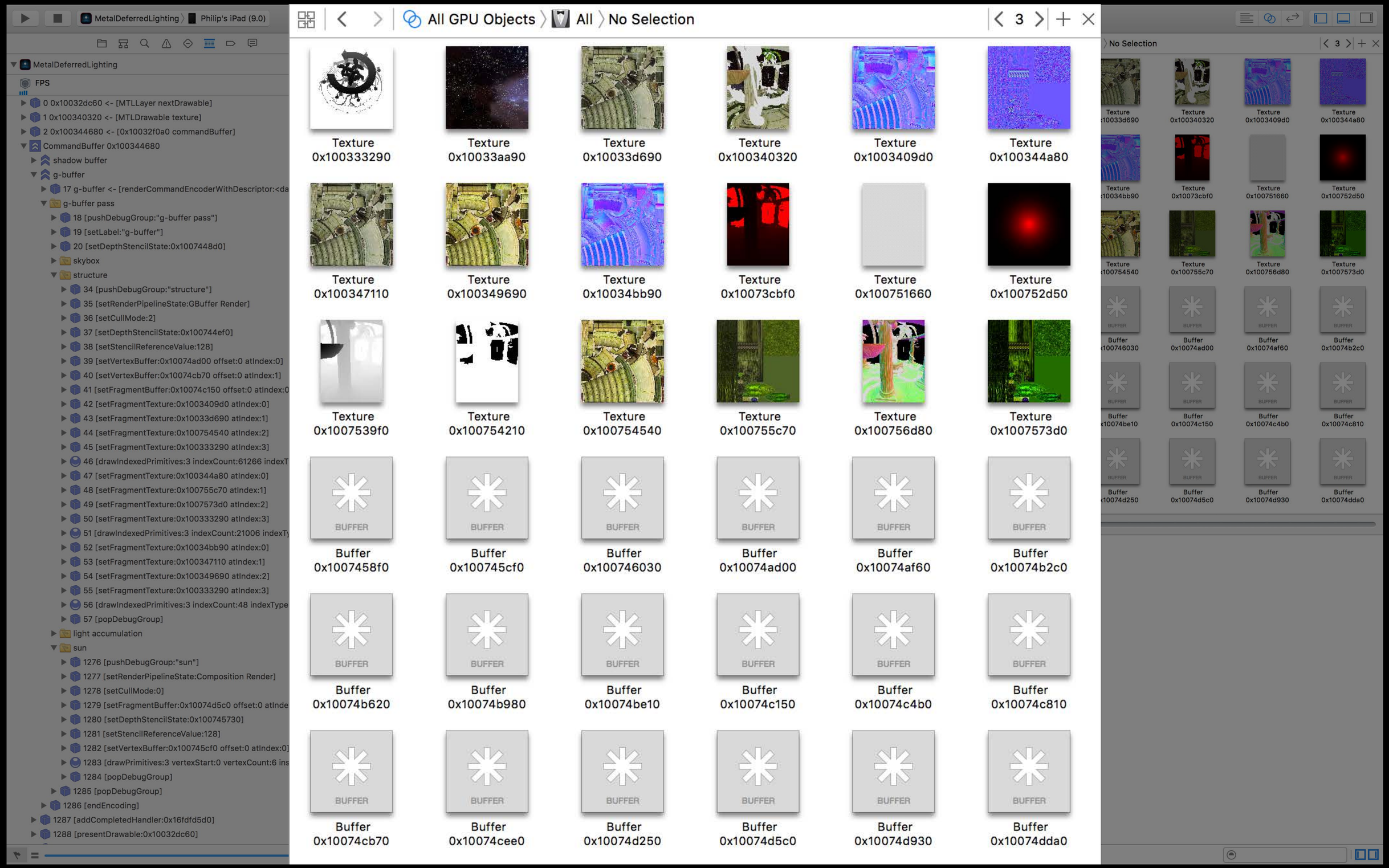
No Selection

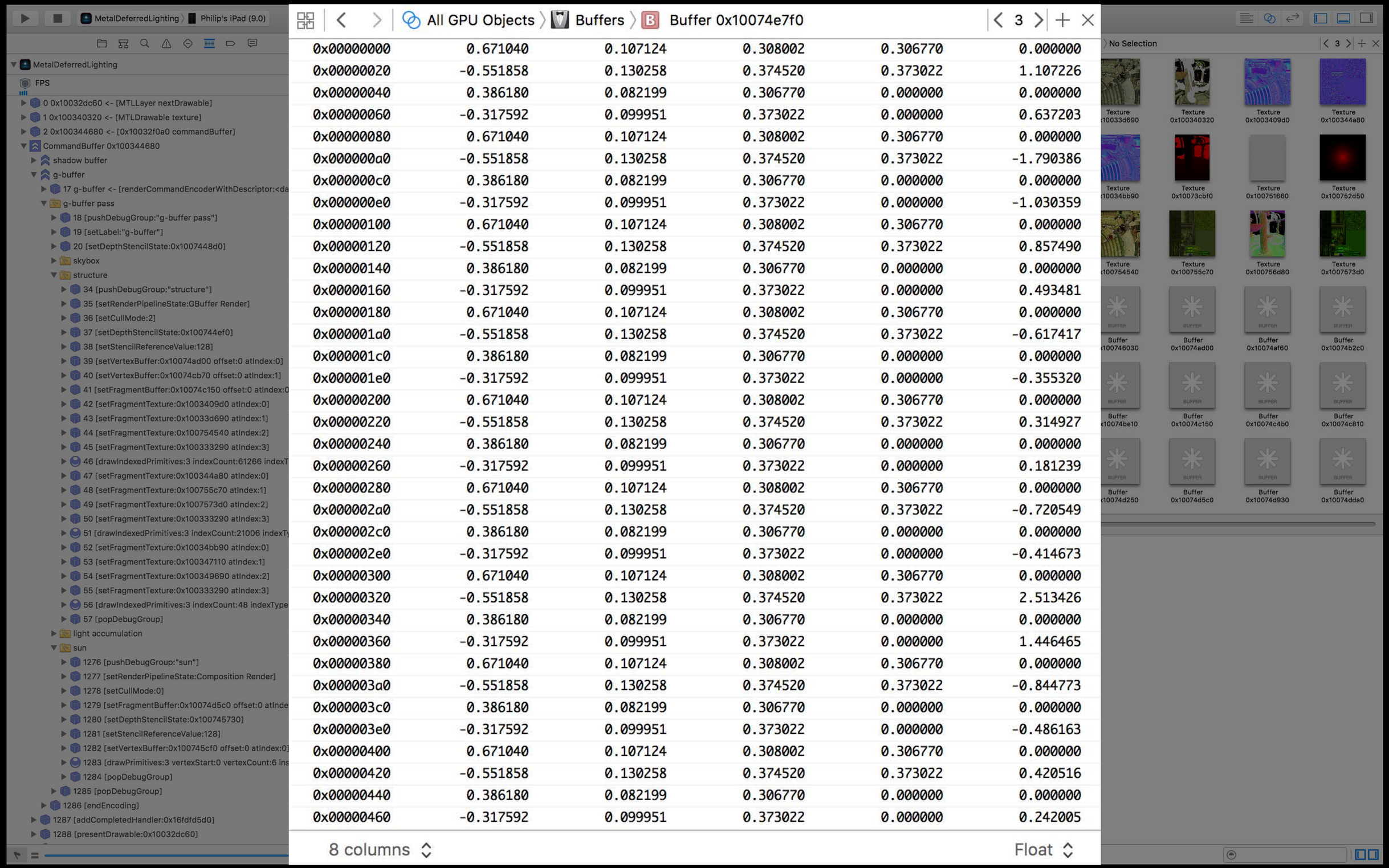
- "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
- RenderPipeline Performance 6.12 ms (55.2%)
- Fragment Texture 0 0x100344a80 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 1 0x100755c70 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 2 0x1007573d0 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 3 0x100333290 - 1024 x 1024, Depth32Float
- Fragment Buffer 0 0x10074c150
- Vertex Buffer 0 0x10074ad00
- Vertex Buffer 1 0x10074cb70
- Index Buffer Index Buffer 0x10074af60, Offset=0x0001dea4
- "g-buffer" (0x100346670)
- Encoder Statistics 138 draw calls, 12MB load/store bandwidth
  - Draw Calls = 138
  - Load Bandwidth = 0B
  - Store Bandwidth = 12MB
- Frame Statistics 141 draw calls, 16MB load/store bandwidth
  - Frame Rate = 60.0fps
  - Draw Calls = 141
  - Command Buffers = 1
  - Render Encoders = 2
  - Load Bandwidth = 0B
  - Store Bandwidth = 16MB

Auto

- "g-buffer" (0x100346670)
  - IsEncodingEnded = NO
  - "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
    - "gBufferVert" (0x100331710)
    - "gBufferFrag" (0x1003317b0)
    - VertexDescriptor disabled
    - SampleCount = 1
    - AlphaToCoverageEnabled = NO
    - AlphaToOneEnabled = NO
    - RasterizationEnabled = YES
    - DepthAttachmentPixelFormat = Depth32Float
    - StencilAttachmentPixelFormat = Stencil8
    - SampleMask = 0xffffffffffffffff
    - SampleCoverage = 1.00
    - Color Attachment 0 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 1 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 2 Blending Disabled, Write Mask All, R32Float
    - Color Attachment 3 Blending Disabled, Write Mask All, BGRA8Unorm
  - TriangleFillMode = Fill
  - Viewport = 0, 0 - 1536, 2048 / 0 - 1
  - ScissorRect = 0,0 - 1536 * 2048
  - BlendColor = 0, 0, 0, 1
  - CullMode = Back
  - FrontFacingWinding = Clockwise

MTL Command Encoder


MetalDeferredLighting
Philip's iPad (9.0)
All GPU Objects
All
No Selection
3
MetalDeferredLighting
FPS
0 0x10032dc60 <- [MTLLayer nextDrawable]
1 0x100340320 <- [MTLDrawable texture]
2 0x100344680 <- [0x10032f0a0 commandBuffer]
CommandBuffer 0x100344680
shadow buffer
g-buffer
17 g-buffer <- [renderCommandEncoderWithDescriptor:<da
g-buffer pass
18 [pushDebugGroup:"g-buffer pass"]
19 [setLabel:"g-buffer"]
20 [setDepthStencilState:0x1007448d0]
skybox
structure
34 [pushDebugGroup:"structure"]
35 [setRenderPipelineState:GBuffer Render]
36 [setCullMode:2]
37 [setDepthStencilState:0x100744ef0]
38 [setStencilReferenceValue:128]
39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0
42 [setFragmentTexture:0x1003409d0 atIndex:0]
43 [setFragmentTexture:0x10033d690 atIndex:1]
44 [setFragmentTexture:0x100754540 atIndex:2]
45 [setFragmentTexture:0x100333290 atIndex:3]
46 [drawIndexedPrimitives:3 indexCount:61266 indexT
47 [setFragmentTexture:0x100344a80 atIndex:0]
48 [setFragmentTexture:0x100755c70 atIndex:1]
49 [setFragmentTexture:0x1007573d0 atIndex:2]
50 [setFragmentTexture:0x100333290 atIndex:3]
51 [drawIndexedPrimitives:3 indexCount:21006 indexTy
52 [setFragmentTexture:0x10034bb90 atIndex:0]
53 [setFragmentTexture:0x100347110 atIndex:1]
54 [setFragmentTexture:0x100349690 atIndex:2]
55 [setFragmentTexture:0x100333290 atIndex:3]
56 [drawIndexedPrimitives:3 indexCount:48 indexType
57 [popDebugGroup]
light accumulation
sun
1276 [pushDebugGroup:"sun"]
1277 [setRenderPipelineState:Composition Render]
1278 [setCullMode:0]
1279 [setFragmentBuffer:0x10074d5c0 offset:0 atInde
1280 [setDepthStencilState:0x100745730]
1281 [setStencilReferenceValue:128]
1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 ins
1284 [popDebugGroup]
1285 [popDebugGroup]
1286 [endEncoding]
1287 [addCompletedHandler:0x16fdfd5d0]
1288 [presentDrawable:0x10032dc60]
Texture 0x100333290
Texture 0x10033aa90
Texture 0x10033d690
Texture 0x100340320
Texture 0x1003409d0
Texture 0x100344a80
Texture 0x100347110
Texture 0x100349690
Texture 0x10034bb90
Texture 0x10073cbf0
Texture 0x100751660
Texture 0x100752d50
Texture 0x1007539f0
Texture 0x100754210
Texture 0x100754540
Texture 0x100755c70
Texture 0x100756d80
Texture 0x1007573d0
Buffer 0x1007458f0
Buffer 0x100745cf0
Buffer 0x100746030
Buffer 0x10074ad00
Buffer 0x10074af60
Buffer 0x10074b2c0
Buffer 0x10074b620
Buffer 0x10074b980
Buffer 0x10074be10
Buffer 0x10074c150
Buffer 0x10074c4b0
Buffer 0x10074c810
Buffer 0x10074cb70
Buffer 0x10074cee0
Buffer 0x10074d250
Buffer 0x10074d5c0
Buffer 0x10074d930
Buffer 0x10074dda0



0x00000000	0.671040	0.107124	0.308002	0.306770	0.000000
0x00000020	-0.551858	0.130258	0.374520	0.373022	1.107226
0x00000040	0.386180	0.082199	0.306770	0.000000	0.000000
0x00000060	-0.317592	0.099951	0.373022	0.000000	0.637203
0x00000080	0.671040	0.107124	0.308002	0.306770	0.000000
0x000000a0	-0.551858	0.130258	0.374520	0.373022	-1.790386
0x000000c0	0.386180	0.082199	0.306770	0.000000	0.000000
0x000000e0	-0.317592	0.099951	0.373022	0.000000	-1.030359
0x00000100	0.671040	0.107124	0.308002	0.306770	0.000000
0x00000120	-0.551858	0.130258	0.374520	0.373022	0.857490
0x00000140	0.386180	0.082199	0.306770	0.000000	0.000000
0x00000160	-0.317592	0.099951	0.373022	0.000000	0.493481
0x00000180	0.671040	0.107124	0.308002	0.306770	0.000000
0x000001a0	-0.551858	0.130258	0.374520	0.373022	-0.617417
0x000001c0	0.386180	0.082199	0.306770	0.000000	0.000000
0x000001e0	-0.317592	0.099951	0.373022	0.000000	-0.355320
0x00000200	0.671040	0.107124	0.308002	0.306770	0.000000
0x00000220	-0.551858	0.130258	0.374520	0.373022	0.314927
0x00000240	0.386180	0.082199	0.306770	0.000000	0.000000
0x00000260	-0.317592	0.099951	0.373022	0.000000	0.181239
0x00000280	0.671040	0.107124	0.308002	0.306770	0.000000
0x000002a0	-0.551858	0.130258	0.374520	0.373022	-0.720549
0x000002c0	0.386180	0.082199	0.306770	0.000000	0.000000
0x000002e0	-0.317592	0.099951	0.373022	0.000000	-0.414673
0x00000300	0.671040	0.107124	0.308002	0.306770	0.000000
0x00000320	-0.551858	0.130258	0.374520	0.373022	2.513426
0x00000340	0.386180	0.082199	0.306770	0.000000	0.000000
0x00000360	-0.317592	0.099951	0.373022	0.000000	1.446465
0x00000380	0.671040	0.107124	0.308002	0.306770	0.000000
0x000003a0	-0.551858	0.130258	0.374520	0.373022	-0.844773
0x000003c0	0.386180	0.082199	0.306770	0.000000	0.000000
0x000003e0	-0.317592	0.099951	0.373022	0.000000	-0.486163
0x00000400	0.671040	0.107124	0.308002	0.306770	0.000000
0x00000420	-0.551858	0.130258	0.374520	0.373022	0.420516
0x00000440	0.386180	0.082199	0.306770	0.000000	0.000000
0x00000460	-0.317592	0.099951	0.373022	0.000000	0.242005

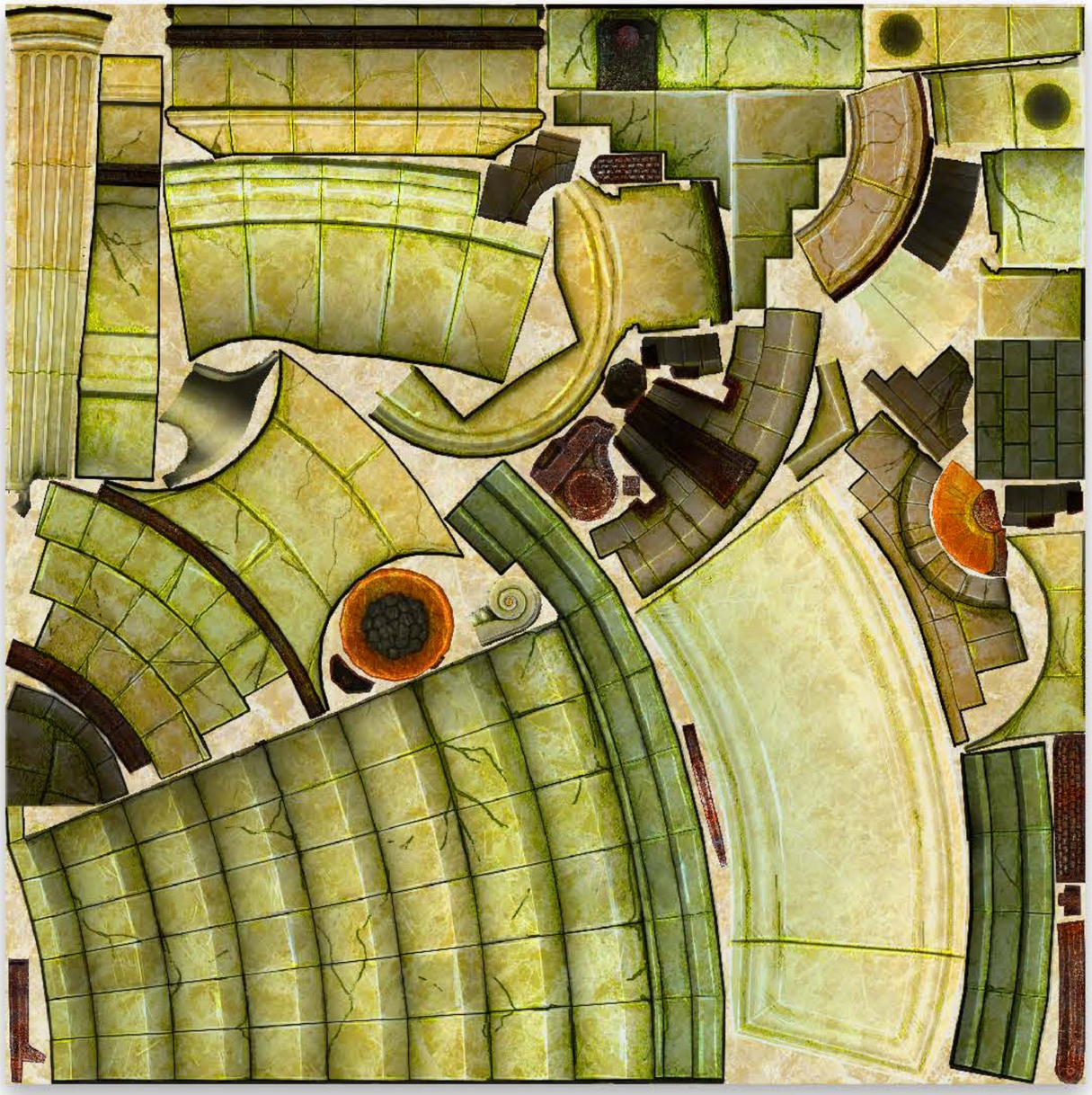
MetalDeferredLighting 〉 Philip's iPad (9.0)

All GPU Objects 〉 All 〉 Texture 0x100754540

3

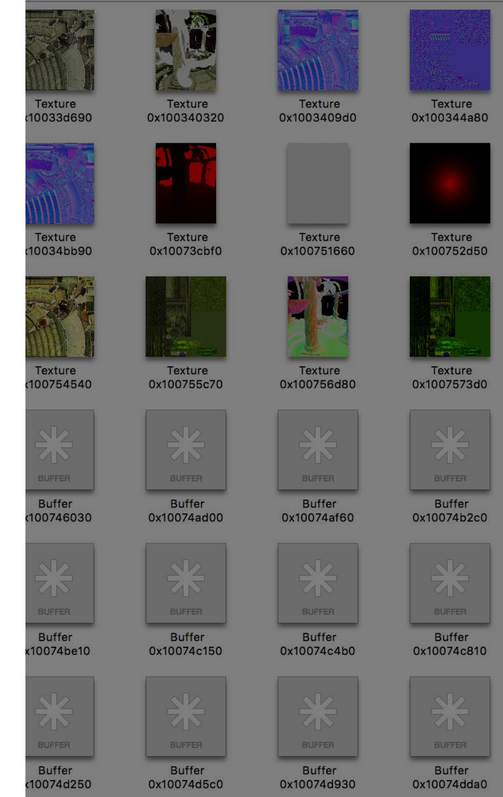
No Selection

- MetalDeferredLighting
  - FPS
  - 0 0x10032dc60 <- [MTLLayer nextDrawable]
  - 1 0x100340320 <- [MTLDrawable texture]
  - 2 0x100344680 <- [0x10032f0a0 commandBuffer]
  - CommandBuffer 0x100344680
    - shadow buffer
    - g-buffer
      - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<da
      - g-buffer pass
        - 18 [pushDebugGroup:"g-buffer pass"]
        - 19 [setLabel:"g-buffer"]
        - 20 [setDepthStencilState:0x1007448d0]
        - skybox
        - structure
          - 34 [pushDebugGroup:"structure"]
          - 35 [setRenderPipelineState:GBuffer Render]
          - 36 [setCullMode:2]
          - 37 [setDepthStencilState:0x100744ef0]
          - 38 [setStencilReferenceValue:128]
          - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
          - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
          - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0
          - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
          - 43 [setFragmentTexture:0x10033d690 atIndex:1]
          - 44 [setFragmentTexture:0x100754540 atIndex:2]
          - 45 [setFragmentTexture:0x100333290 atIndex:3]
          - 46 [drawIndexedPrimitives:3 indexCount:61266 indexT
          - 47 [setFragmentTexture:0x100344a80 atIndex:0]
          - 48 [setFragmentTexture:0x100755c70 atIndex:1]
          - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
          - 50 [setFragmentTexture:0x100333290 atIndex:3]
          - 51 [drawIndexedPrimitives:3 indexCount:21006 indexTy
          - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
          - 53 [setFragmentTexture:0x100347110 atIndex:1]
          - 54 [setFragmentTexture:0x100349690 atIndex:2]
          - 55 [setFragmentTexture:0x100333290 atIndex:3]
          - 56 [drawIndexedPrimitives:3 indexCount:48 indexType
          - 57 [popDebugGroup]
        - light accumulation
        - sun
          - 1276 [pushDebugGroup:"sun"]
          - 1277 [setRenderPipelineState:Composition Render]
          - 1278 [setCullMode:0]
          - 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atInde
          - 1280 [setDepthStencilState:0x100745730]
          - 1281 [setStencilReferenceValue:128]
          - 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
          - 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 ins
          - 1284 [popDebugGroup]
        - 1285 [popDebugGroup]
      - 1286 [endEncoding]
    - 1287 [addCompletedHandler:0x16fdfd5d0]
    - 1288 [presentDrawable:0x10032dc60]



Texture 0x100754540

Level 0



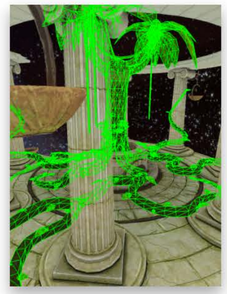
MetalDeferredLighting 〉 Philip's iPad (9.0)

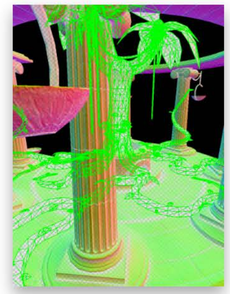
Running MetalDeferredLighting on Philip's iPad 9

51 [drawIndexedPrimitives:3 indexCount:21006 in...4af60 indexBufferOffset:122532 instanceCount:1]

All GPU Objects 〉 All 〉 No Selection

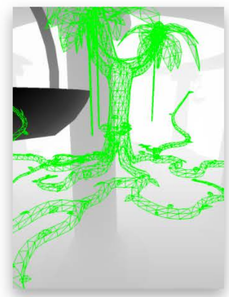
- MetalDeferredLighting
  - FPS — 60 FPS
  - 0 0x10032dc60 <- [MTLLayer nextDrawable]
  - 1 0x100340320 <- [MTLDrawable texture]
  - 2 0x100344680 <- [0x10032f0a0 commandBuffer]
  - CommandBuffer 0x100344680
    - shadow buffer — 8.49 µs
    - g-buffer
      - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
      - g-buffer pass
        - 18 [pushDebugGroup:"g-buffer pass"]
        - 19 [setLabel:"g-buffer"]
        - 20 [setDepthStencilState:0x1007448d0]
        - skybox — 1.83 ms
        - structure
          - 34 [pushDebugGroup:"structure"]
          - 35 [setRenderPipelineState:GBuffer Render]
          - 36 [setCullMode:2]
          - 37 [setDepthStencilState:0x100744ef0]
          - 38 [setStencilReferenceValue:128]
          - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
          - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
          - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0]
          - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
          - 43 [setFragmentTexture:0x10033d690 atIndex:1]
          - 44 [setFragmentTexture:0x100754540 atIndex:2]
          - 45 [setFragmentTexture:0x100333290 atIndex:3]
          - 46 [drawIndexedPrimitives:3 indexCount:61266 indexTyp... — 5.01 ms
          - 47 [setFragmentTexture:0x100344a80 atIndex:0]
          - 48 [setFragmentTexture:0x100755c70 atIndex:1]
          - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
          - 50 [setFragmentTexture:0x100333290 atIndex:3]
          - 51 [drawIndexedPrimitives:3 indexCount:21006 indexType... — 1.11 ms
          - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
          - 53 [setFragmentTexture:0x100347110 atIndex:1]
          - 54 [setFragmentTexture:0x100349690 atIndex:2]
          - 55 [setFragmentTexture:0x100333290 atIndex:3]
          - 56 [drawIndexedPrimitives:3 indexCount:48 indexType:0 i... — 0.00 ns
          - 57 [popDebugGroup]
        - light accumulation — 1.42 ms
        - sun
          - 1276 [pushDebugGroup:"sun"]
          - 1277 [setRenderPipelineState:Composition Render]
          - 1278 [setCullMode:0]
          - 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atIndex:0]
          - 1280 [setDepthStencilState:0x100745730]
          - 1281 [setStencilReferenceValue:128]
          - 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
          - 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 insta... — 1.69 ms
          - 1284 [popDebugGroup]
        - 1285 [popDebugGroup]
      - 1286 [endEncoding]
    - 1287 [addCompletedHandler:0x16fdfd5d0]
    - 1288 [presentDrawable:0x10032dc60]


Color Attachment 0

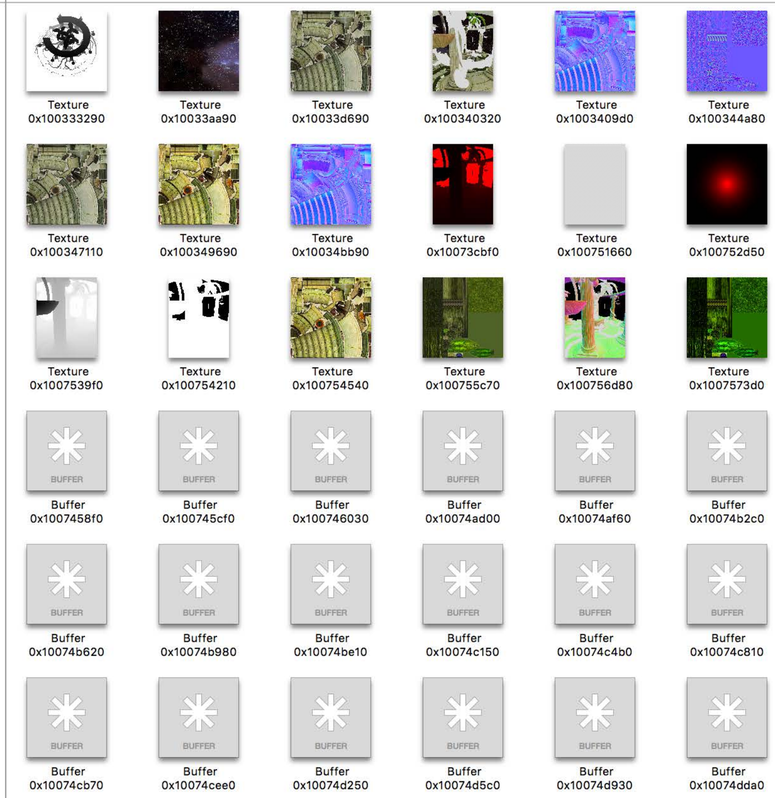

Color Attachment 1


Color Attachment 2


Color Attachment 3


Depth Attachment


Stencil Attachment



No Selection

- "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
- RenderPipeline Performance 6.12 ms (55.2%)
- Fragment Texture 0 0x100344a80 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 1 0x100755c70 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 2 0x1007573d0 - 1024 x 1024, RGBA8Unorm
- Fragment Texture 3 0x100333290 - 1024 x 1024, Depth32Float
- Fragment Buffer 0 0x10074c150
- Vertex Buffer 0 0x10074ad00
- Vertex Buffer 1 0x10074cb70
- Index Buffer Index Buffer 0x10074af60, Offset=0x0001dea4
- "g-buffer" (0x100346670)
- Encoder Statistics 138 draw calls, 12MB load/store bandwidth
  - Draw Calls = 138
  - Load Bandwidth = 0B
  - Store Bandwidth = 12MB
- Frame Statistics 141 draw calls, 16MB load/store bandwidth
  - Frame Rate = 60.0fps
  - Draw Calls = 141
  - Command Buffers = 1
  - Render Encoders = 2
  - Load Bandwidth = 0B
  - Store Bandwidth = 16MB

Auto

- "g-buffer" (0x100346670)
  - IsEncodingEnded = NO
  - "GBuffer Render" (0x100742c40) gBufferVert/gBufferFrag
    - "gBufferVert" (0x100331710)
    - "gBufferFrag" (0x1003317b0)
    - VertexDescriptor disabled
    - SampleCount = 1
    - AlphaToCoverageEnabled = NO
    - AlphaToOneEnabled = NO
    - RasterizationEnabled = YES
    - DepthAttachmentPixelFormat = Depth32Float
    - StencilAttachmentPixelFormat = Stencil8
    - SampleMask = 0xffffffffffffffff
    - SampleCoverage = 1.00
    - Color Attachment 0 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 1 Blending Disabled, Write Mask All, BGRA8Unorm
    - Color Attachment 2 Blending Disabled, Write Mask All, R32Float
    - Color Attachment 3 Blending Disabled, Write Mask All, BGRA8Unorm
  - TriangleFillMode = Fill
  - Viewport = 0, 0 - 1536, 2048 / 0 - 1
  - ScissorRect = 0,0 - 1536 * 2048
  - BlendColor = 0, 0, 0, 1
  - CullMode = Back
  - FrontFacingWinding = Clockwise

MTL Command Encoder

MetalDeferredLighting 〉 Philip's iPad (9.0)

Running MetalDeferredLighting on Philip's iPad

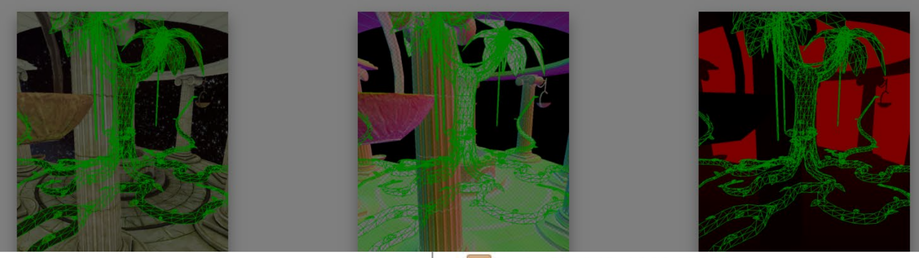
51 [drawIndexedPrimitives:3 indexCount:21006 in...4af60 indexBufferOffset:122532 instanceCount:1]

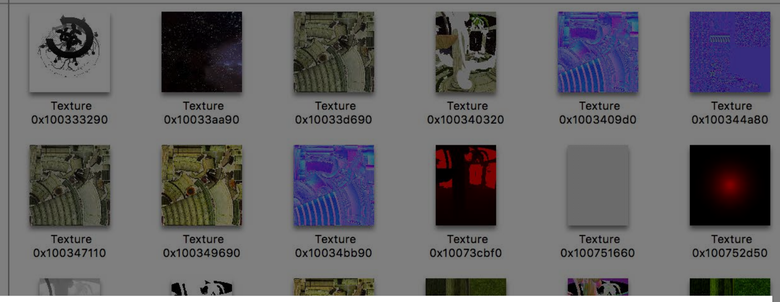
All GPU Objects 〉 All 〉 No Selection

MetalDeferredLighting

FPS 60 FPS

- 0 0x10032dc60 <- [MTLLayer nextDrawable]
- 1 0x100340320 <- [MTLDrawable texture]
- 2 0x100344680 <- [0x10032f0a0 commandBuffer]
- CommandBuffer 0x100344680
  - shadow buffer 8.49 µs
  - g-buffer
    - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
    - g-buffer pass
      - 18 [pushDebugGroup:"g-buffer pass"]





- **"GBuffer Render" (0x100742c40)** gBufferVert/gBufferFrag
- **RenderPipeline Performance** 6.12 ms (55.2%)
- **Fragment Texture 0** 0x100344a80 - 1024 x 1024, RGBA8Unorm
- **Fragment Texture 1** 0x100755c70 - 1024 x 1024, RGBA8Unorm
- **Fragment Texture 2** 0x1007573d0 - 1024 x 1024, RGBA8Unorm
- **Fragment Texture 3** 0x100333290 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10074c150
- **Vertex Buffer 0** 0x10074ad00
- **Vertex Buffer 1** 0x10074cb70
- **Index Buffer** Index Buffer 0x10074af60, Offset=0x0001dea4
- **"g-buffer" (0x100346670)**
- **Encoder Statistics** 138 draw calls, 12MB load/store bandwidth
  - **Draw Calls** = 138
  - **Load Bandwidth** = 0B
  - **Store Bandwidth** = 12MB
- **Frame Statistics** 141 draw calls, 16MB load/store bandwidth
  - **Frame Rate** = 60.0fps
  - **Draw Calls** = 141
  - **Command Buffers** = 1
  - **Render Encoders** = 2
  - **Load Bandwidth** = 0B
  - **Store Bandwidth** = 16MB

Auto

- **"g-buffer" (0x100346670)**
  - **IsEncodingEnded** = NO
  - **"GBuffer Render" (0x100742c40)** gBufferVert/gBufferFrag
    - **"gBufferVert" (0x100331710)**
    - **"gBufferFrag" (0x1003317b0)**
    - **VertexDescriptor** disabled
    - **SampleCount** = 1
    - **AlphaToCoverageEnabled** = NO
    - **AlphaToOneEnabled** = NO
    - **RasterizationEnabled** = YES
    - **DepthAttachmentPixelFormat** = Depth32Float
    - **StencilAttachmentPixelFormat** = Stencil8
    - **SampleMask** = 0xffffffffffffffff
    - **SampleCoverage** = 1.00
    - **Color Attachment 0** Blending Disabled, Write Mask All, BGRA8Unorm
    - **Color Attachment 1** Blending Disabled, Write Mask All, BGRA8Unorm
    - **Color Attachment 2** Blending Disabled, Write Mask All, R32Float
    - **Color Attachment 3** Blending Disabled, Write Mask All, BGRA8Unorm
  - **TriangleFillMode** = Fill
  - **Viewport** = 0, 0 - 1536, 2048 / 0 - 1
  - **ScissorRect** = 0,0 - 1536 * 2048
  - **BlendColor** = 0, 0, 0, 1
  - **CullMode** = Back
  - **FrontFacingWinding** = Clockwise

MTL Command Encoder

- 1276 [pushDebugGroup:"sun"]
- 1277 [setRenderPipelineState:Composition Render]
- 1278 [setCullMode:0]
- 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atIndex:0]
- 1280 [setDepthStencilState:0x100745730]
- 1281 [setStencilReferenceValue:128]
- 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
- 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 insta... 1.69 ms
- 1284 [popDebugGroup]
- 1285 [popDebugGroup]
- 1286 [endEncoding]
- 1287 [addCompletedHandler:0x16fdfd5d0]
- 1288 [presentDrawable:0x10032dc60]

- **Index Buffer** Index Buffer 0x10074af60, Offset=0x0001dea4
- **"g-buffer" (0x100346670)**
- **Encoder Statistics** 138 draw calls, 12MB load/store bandwidth
  - **Draw Calls** = 138
  - **Load Bandwidth** = 0B
  - **Store Bandwidth** = 12MB
- **Frame Statistics** 141 draw calls, 16MB load/store bandwidth
  - **Frame Rate** = 60.0fps
  - **Draw Calls** = 141
  - **Command Buffers** = 1
  - **Render Encoders** = 2
  - **Load Bandwidth** = 0B
  - **Store Bandwidth** = 16MB

Auto

- **RasterizationEnabled** = YES
- **DepthAttachmentPixelFormat** = Depth32Float
- **StencilAttachmentPixelFormat** = Stencil8
- **SampleMask** = 0xffffffffffffffff
- **SampleCoverage** = 1.00
- **Color Attachment 0** Blending Disabled, Write Mask All, BGRA8Unorm
- **Color Attachment 1** Blending Disabled, Write Mask All, BGRA8Unorm
- **Color Attachment 2** Blending Disabled, Write Mask All, R32Float
- **Color Attachment 3** Blending Disabled, Write Mask All, BGRA8Unorm
- **TriangleFillMode** = Fill
- **Viewport** = 0, 0 - 1536, 2048 / 0 - 1
- **ScissorRect** = 0,0 - 1536 * 2048
- **BlendColor** = 0, 0, 0, 1
- **CullMode** = Back
- **FrontFacingWinding** = Clockwise

MTL Command Encoder

MetalDeferredLighting 〉 Philip's iPad (9.0)

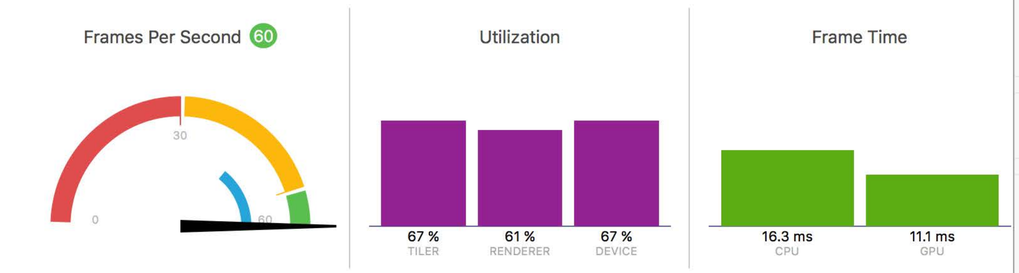
Running MetalDeferredLighting on Philip's iPad 9

GPU Report

lightFrag(VertexOutput in [[...r(0)]], FragOutput gBuffers)

- MetalDeferredLighting
  - FPS — 60 FPS
  - 0 0x10032dc60 <- [MTLLayer nextDrawable]
  - 1 0x100340320 <- [MTLDrawable texture]
  - 2 0x100344680 <- [0x10032f0a0 commandBuffer]
  - CommandBuffer 0x100344680
    - shadow buffer — 8.49 µs
    - g-buffer
      - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
      - g-buffer pass
        - 18 [pushDebugGroup:"g-buffer pass"]
        - 19 [setLabel:"g-buffer"]
        - 20 [setDepthStencilState:0x1007448d0]
        - skybox — 1.83 ms
        - structure
          - 34 [pushDebugGroup:"structure"]
          - 35 [setRenderPipelineState:GBuffer Render]
          - 36 [setCullMode:2]
          - 37 [setDepthStencilState:0x100744ef0]
          - 38 [setStencilReferenceValue:128]
          - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
          - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
          - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0]
          - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
          - 43 [setFragmentTexture:0x10033d690 atIndex:1]
          - 44 [setFragmentTexture:0x100754540 atIndex:2]
          - 45 [setFragmentTexture:0x100333290 atIndex:3]
          - 46 [drawIndexedPrimitives:3 indexCount:61266 indexTyp... — 5.01 ms
          - 47 [setFragmentTexture:0x100344a80 atIndex:0]
          - 48 [setFragmentTexture:0x100755c70 atIndex:1]
          - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
          - 50 [setFragmentTexture:0x100333290 atIndex:3]
          - 51 [drawIndexedPrimitives:3 indexCount:21006 indexType... — 1.11 ms
          - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
          - 53 [setFragmentTexture:0x100347110 atIndex:1]
          - 54 [setFragmentTexture:0x100349690 atIndex:2]
          - 55 [setFragmentTexture:0x100333290 atIndex:3]
          - 56 [drawIndexedPrimitives:3 indexCount:48 indexType:0 i... — 0.00 ns
          - 57 [popDebugGroup]
        - light accumulation — 1.42 ms
        - sun
          - 1276 [pushDebugGroup:"sun"]
          - 1277 [setRenderPipelineState:Composition Render]
          - 1278 [setCullMode:0]
          - 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atIndex:0]
          - 1280 [setDepthStencilState:0x100745730]
          - 1281 [setStencilReferenceValue:128]
          - 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
          - 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 insta... — 1.69 ms
          - 1284 [popDebugGroup]
        - 1285 [popDebugGroup]
      - 1286 [endEncoding]
    - 1287 [addCompletedHandler:0x16fdfd5d0]
    - 1288 [presentDrawable:0x10032dc60]

GPU



Pipeline Performance

Current ms

Program	Frame %	Current ms
Render Pipeline "GBuffer Render"	55.2%	6.12
46 [drawIndexedPrimitives:3 indexCount:61266 indexT...	45.2%	5.01
51 [drawIndexedPrimitives:3 indexCount:21006 indexTy...	10.0%	1.11
56 [drawIndexedPrimitives:3 indexCount:48 indexType:...	0.0%	0.00
Render Pipeline "Skybox Render"	16.6%	1.83
Render Pipeline "Composition Render"	15.3%	1.69
Render Pipeline "Light Color Render"	12.8%	1.42
Render Pipeline "Shadow Render"	0.1%	0.01
Total		11.07

```
38      output.v_view = (matrices->mvMatrix * tempPosition).xyz;                    0.0%
39
40      return output;                                                              0.0%
41  }
42
43  fragment FragOutput lightFrag(VertexOutput in [[stage_in]],
44                                constant LightFragmentInputs *lightData [[buffer(0)]],
45                                FragOutput gBuffers)
46  {                                                                               5.0%
47      float3 n_s = gBuffers.normal.rgb;
48
49      float scene_z = gBuffers.depth;
50
51      float3 n = n_s * 2.0 - 1.0;                                                 5.2%
52
53      // Derive the view-space position of the scene fragment in the G-buffer
54      // Since the light primitive and the G-buffer were rendered with the same view-
            projection matrix,
55      // we can treat the view-space position of the current light primitive fragment as a
            ray from the origin,
56      // and derive the view-space position of the scene by projecting along the ray with
            (scene_z / v_view.z).
57      // Our scene view-space position is also the view-vector to the scene fragment.
58      float3 v = in.v_view * (scene_z / in.v_view.z);                             11.2%
59
60      // Now, we have everything we need to calculate our view-space lighting vectors.
61      float3 l = (lightData->view_light_position.xyz - v);                        1.7%
62      float n_ls = dot(n, n);                                                     2.4%
63      float v_ls = dot(v, v);                                                     3.6%
64      float l_ls = dot(l, l);                                                     3.9%
65      float3 h = (l * rsqrt(l_ls / v_ls) - v);                                    20.0%
66      float h_ls = dot(h, h);                                                     4.5%
67      float nl = dot(n, l) * rsqrt(n_ls * l_ls);                                  12.8%
68      float nh = dot(n, h) * rsqrt(n_ls * h_ls);                                  7.2%
69      float d_atten = sqrt(l_ls);
70      float atten = fmax(1.0 - d_atten / lightData->light_color_radius.w, 0.0);   1.1%
71      float diffuse = fmax(nl, 0.0) * atten;                                      2.5%
72
73      float4 light = gBuffers.light;
74      light.rgb += lightData->light_color_radius.xyz * diffuse;
75      light.a += pow(fmax(nh, 0.0), 32.0) * step(0.0, nl) * atten * 1.0001;       7.4%
76
77      FragOutput output;
78      output.albedo = gBuffers.albedo;
79      output.normal = gBuffers.normal;
80      output.depth = gBuffers.depth;
81      output.light = light;
82
83      return output;                                                              11.2%
```

No Selection

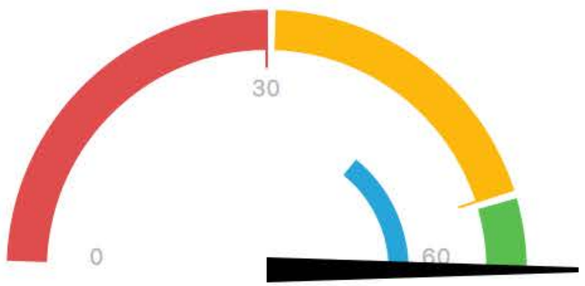
Auto

MTL Command Encoder

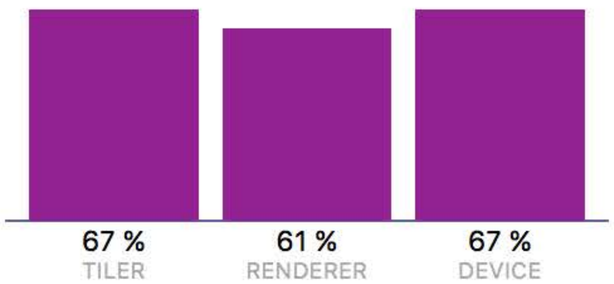
GPU Report

GPU

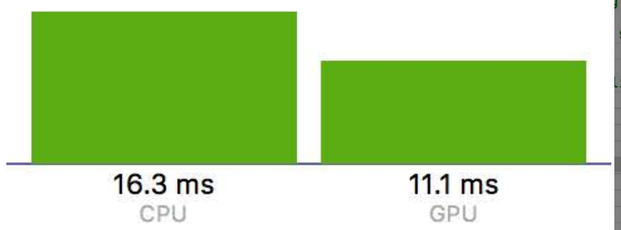
Frames Per Second 60



Utilization



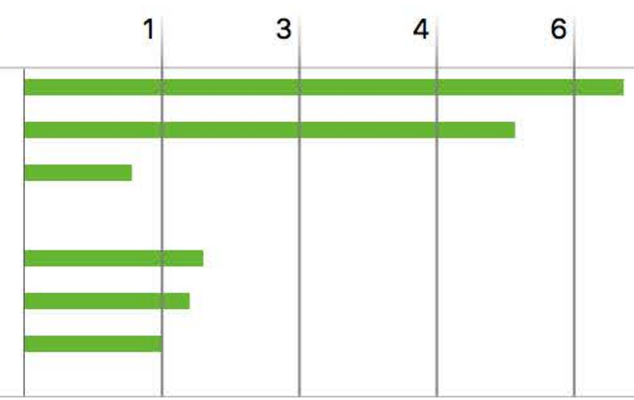
Frame Time



Pipeline Performance

Current ms

Program	Frame %	Current ms
Render Pipeline "GBuffer Render"	55.2%	6.12
46 [drawIndexedPrimitives:3 indexCount:61266 indexT...	45.2%	5.01
51 [drawIndexedPrimitives:3 indexCount:21006 indexTy...	10.0%	1.11
56 [drawIndexedPrimitives:3 indexCount:48 indexType:...	0.0%	0.00
Render Pipeline "Skybox Render"	16.6%	1.83
Render Pipeline "Composition Render"	15.3%	1.69
Render Pipeline "Light Color Render"	12.8%	1.42
Render Pipeline "Shadow Render"	0.1%	0.01
Total		11.07



MetalDeferredLighting 〉 Philip's iPad (9.0)

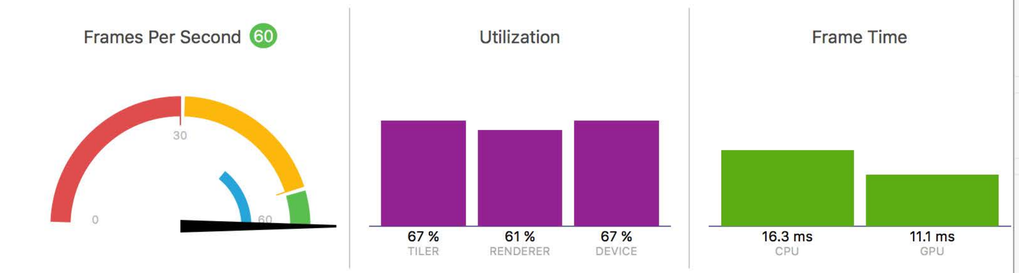
Running MetalDeferredLighting on Philip's iPad ⚠ 9

GPU Report

lightFrag(VertexOutput in [[...r(0)]], FragOutput gBuffers)

- MetalDeferredLighting
  - FPS — 60 FPS
  - 0 0x10032dc60 <- [MTLLayer nextDrawable]
  - 1 0x100340320 <- [MTLDrawable texture]
  - 2 0x100344680 <- [0x10032f0a0 commandBuffer]
  - CommandBuffer 0x100344680
    - shadow buffer — 8.49 µs
    - g-buffer
      - 17 g-buffer <- [renderCommandEncoderWithDescriptor:<data>]
      - g-buffer pass
        - 18 [pushDebugGroup:"g-buffer pass"]
        - 19 [setLabel:"g-buffer"]
        - 20 [setDepthStencilState:0x1007448d0]
        - skybox — 1.83 ms
        - structure
          - 34 [pushDebugGroup:"structure"]
          - 35 [setRenderPipelineState:GBuffer Render]
          - 36 [setCullMode:2]
          - 37 [setDepthStencilState:0x100744ef0]
          - 38 [setStencilReferenceValue:128]
          - 39 [setVertexBuffer:0x10074ad00 offset:0 atIndex:0]
          - 40 [setVertexBuffer:0x10074cb70 offset:0 atIndex:1]
          - 41 [setFragmentBuffer:0x10074c150 offset:0 atIndex:0]
          - 42 [setFragmentTexture:0x1003409d0 atIndex:0]
          - 43 [setFragmentTexture:0x10033d690 atIndex:1]
          - 44 [setFragmentTexture:0x100754540 atIndex:2]
          - 45 [setFragmentTexture:0x100333290 atIndex:3]
          - 46 [drawIndexedPrimitives:3 indexCount:61266 indexTyp... — 5.01 ms
          - 47 [setFragmentTexture:0x100344a80 atIndex:0]
          - 48 [setFragmentTexture:0x100755c70 atIndex:1]
          - 49 [setFragmentTexture:0x1007573d0 atIndex:2]
          - 50 [setFragmentTexture:0x100333290 atIndex:3]
          - 51 [drawIndexedPrimitives:3 indexCount:21006 indexType... — 1.11 ms
          - 52 [setFragmentTexture:0x10034bb90 atIndex:0]
          - 53 [setFragmentTexture:0x100347110 atIndex:1]
          - 54 [setFragmentTexture:0x100349690 atIndex:2]
          - 55 [setFragmentTexture:0x100333290 atIndex:3]
          - 56 [drawIndexedPrimitives:3 indexCount:48 indexType:0 i... — 0.00 ns
          - 57 [popDebugGroup]
        - light accumulation — 1.42 ms
        - sun
          - 1276 [pushDebugGroup:"sun"]
          - 1277 [setRenderPipelineState:Composition Render]
          - 1278 [setCullMode:0]
          - 1279 [setFragmentBuffer:0x10074d5c0 offset:0 atIndex:0]
          - 1280 [setDepthStencilState:0x100745730]
          - 1281 [setStencilReferenceValue:128]
          - 1282 [setVertexBuffer:0x100745cf0 offset:0 atIndex:0]
          - 1283 [drawPrimitives:3 vertexStart:0 vertexCount:6 insta... — 1.69 ms
          - 1284 [popDebugGroup]
        - 1285 [popDebugGroup]
    - 1286 [endEncoding]
  - 1287 [addCompletedHandler:0x16fdfd5d0]
  - 1288 [presentDrawable:0x10032dc60]

GPU



Pipeline Performance

Current ms

Program	Frame %	Current ms
Render Pipeline "GBuffer Render"	55.2%	6.12
46 [drawIndexedPrimitives:3 indexCount:61266 indexT...	45.2%	5.01
51 [drawIndexedPrimitives:3 indexCount:21006 indexTy...	10.0%	1.11
56 [drawIndexedPrimitives:3 indexCount:48 indexType:...	0.0%	0.00
Render Pipeline "Skybox Render"	16.6%	1.83
Render Pipeline "Composition Render"	15.3%	1.69
Render Pipeline "Light Color Render"	12.8%	1.42
Render Pipeline "Shadow Render"	0.1%	0.01
Total		11.07

```
    output.v_view = (matrices->mvMatrix * tempPosition).xyz;                    0.0%

    return output;                                                              0.0%
}

fragment FragOutput lightFrag(VertexOutput in [[stage_in]],
                              constant LightFragmentInputs *lightData [[buffer(0)]],
                              FragOutput gBuffers)
{                                                                               5.0%
    float3 n_s = gBuffers.normal.rgb;

    float scene_z = gBuffers.depth;

    float3 n = n_s * 2.0 - 1.0;                                                 5.2%

    // Derive the view-space position of the scene fragment in the G-buffer
    // Since the light primitive and the G-buffer were rendered with the same view-
        projection matrix,
    // we can treat the view-space position of the current light primitive fragment as a
        ray from the origin,
    // and derive the view-space position of the scene by projecting along the ray with
        (scene_z / v_view.z).
    // Our scene view-space position is also the view-vector to the scene fragment.
    float3 v = in.v_view * (scene_z / in.v_view.z);                             11.2%

    // Now, we have everything we need to calculate our view-space lighting vectors.
    float3 l = (lightData->view_light_position.xyz - v);                        1.7%
    float n_ls = dot(n, n);                                                     2.4%
    float v_ls = dot(v, v);                                                     3.6%
    float l_ls = dot(l, l);                                                     3.9%
    float3 h = (l * rsqrt(l_ls / v_ls) - v);                                    20.0%
    float h_ls = dot(h, h);                                                     4.5%
    float nl = dot(n, l) * rsqrt(n_ls * l_ls);                                  12.8%
    float nh = dot(n, h) * rsqrt(n_ls * h_ls);                                  7.2%
    float d_atten = sqrt(l_ls);
    float atten = fmax(1.0 - d_atten / lightData->light_color_radius.w, 0.0);   1.1%
    float diffuse = fmax(nl, 0.0) * atten;                                      2.5%

    float4 light = gBuffers.light;
    light.rgb += lightData->light_color_radius.xyz * diffuse;
    light.a += pow(fmax(nh, 0.0), 32.0) * step(0.0, nl) * atten * 1.0001;       7.4%

    FragOutput output;
    output.albedo = gBuffers.albedo;
    output.normal = gBuffers.normal;
    output.depth = gBuffers.depth;
    output.light = light;

    return output;                                                              11.2%
```

No Selection

Auto

MTL Command Encoder

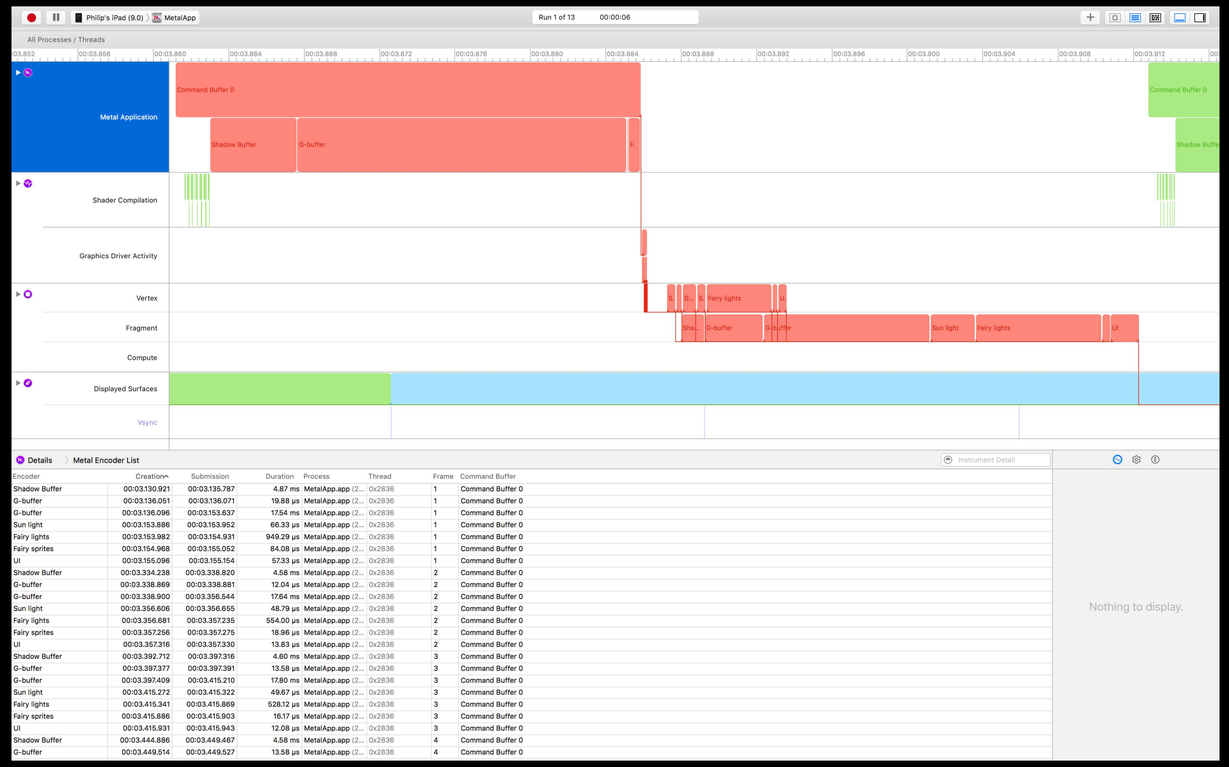
```
    output.v_view = (matrices->mvMatrix * tempPosition).xyz;                    0.0%

    return output;                                                              0.0%
}

fragment FragOutput lightFrag(VertexOutput in [[stage_in]],
                              constant LightFragmentInputs *lightData [[buffer(0)]],
                              FragOutput gBuffers)
{                                                                               5.0%
    float3 n_s = gBuffers.normal.rgb;

    float scene_z = gBuffers.depth;

    float3 n = n_s * 2.0 - 1.0;                                                 5.2%

    // Derive the view-space position of the scene fragment in the G-buffer
    // Since the light primitive and the G-buffer were rendered with the same view-projection matrix,
    // we can treat the view-space position of the current light primitive fragment as a ray from the origin,
    // and derive the view-space position of the scene by projecting along the ray with (scene_z / v_view.z).
    // Our scene view-space position is also the view-vector to the scene fragment.
    float3 v = in.v_view * (scene_z / in.v_view.z);                             11.2%

    // Now, we have everything we need to calculate our view-space lighting vectors.
    float3 l = (lightData->view_light_position.xyz - v);                        1.7%
    float n_ls = dot(n, n);                                                     2.4%
    float v_ls = dot(v, v);                                                     3.6%
    float l_ls = dot(l, l);                                                     3.9%
    float3 h = (l * rsqrt(l_ls / v_ls) - v);                                    20.0%
    float h_ls = dot(h, h);                                                     4.5%
    float nl = dot(n, l) * rsqrt(n_ls * l_ls);                                  12.8%
    float nh = dot(n, h) * rsqrt(n_ls * h_ls);                                  7.2%
    float d_atten = sqrt(l_ls);
    float atten = fmax(1.0 - d_atten / lightData->light_color_radius.w, 0.0);   1.1%
    float diffuse = fmax(nl, 0.0) * atten;                                      2.5%

    float4 light = gBuffers.light;
    light.rgb += lightData->light_color_radius.xyz * diffuse;
    light.a += pow(fmax(nh, 0.0), 32.0) * step(0.0, nl) * atten * 1.0001;       7.4%

    FragOutput output;
    output.albedo = gBuffers.albedo;
    output.normal = gBuffers.normal;
    output.depth = gBuffers.depth;
    output.light = light;

    return output;                                                              11.2%
```

NEW

Metal Tools in Instruments

Metal System Trace

NEW



Encoder	Creation	Submission	Duration	Process	Thread	Frame	Command Buffer
Shadow Buffer	00:03.130.921	00:03.135.787	4.87 ms	MetalApp.app (2...	0x2836	1	Command Buffer 0
G-buffer	00:03.136.051	00:03.136.071	19.88 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
G-buffer	00:03.136.096	00:03.153.637	17.54 ms	MetalApp.app (2...	0x2836	1	Command Buffer 0
Sun light	00:03.153.886	00:03.153.952	66.33 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Fairy lights	00:03.153.982	00:03.154.931	949.29 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Fairy sprites	00:03.154.968	00:03.155.052	84.08 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
UI	00:03.155.096	00:03.155.154	57.33 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Shadow Buffer	00:03.334.238	00:03.338.820	4.58 ms	MetalApp.app (2...	0x2836	2	Command Buffer 0
G-buffer	00:03.338.869	00:03.338.881	12.04 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
G-buffer	00:03.338.900	00:03.356.544	17.64 ms	MetalApp.app (2...	0x2836	2	Command Buffer 0
Sun light	00:03.356.606	00:03.356.655	48.79 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Fairy lights	00:03.356.681	00:03.357.235	554.00 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Fairy sprites	00:03.357.256	00:03.357.275	18.96 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
UI	00:03.357.316	00:03.357.330	13.83 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Shadow Buffer	00:03.392.712	00:03.397.316	4.60 ms	MetalApp.app (2...	0x2836	3	Command Buffer 0
G-buffer	00:03.397.377	00:03.397.391	13.58 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
G-buffer	00:03.397.409	00:03.415.210	17.80 ms	MetalApp.app (2...	0x2836	3	Command Buffer 0
Sun light	00:03.415.272	00:03.415.322	49.67 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Fairy lights	00:03.415.341	00:03.415.869	528.12 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Fairy sprites	00:03.415.886	00:03.415.903	16.17 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
UI	00:03.415.931	00:03.415.943	12.08 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Shadow Buffer	00:03.444.886	00:03.449.467	4.58 ms	MetalApp.app (2...	0x2836	4	Command Buffer 0
G-buffer	00:03.449.514	00:03.449.527	13.58 µs	MetalApp.app (2...	0x2836	4	Command Buffer 0

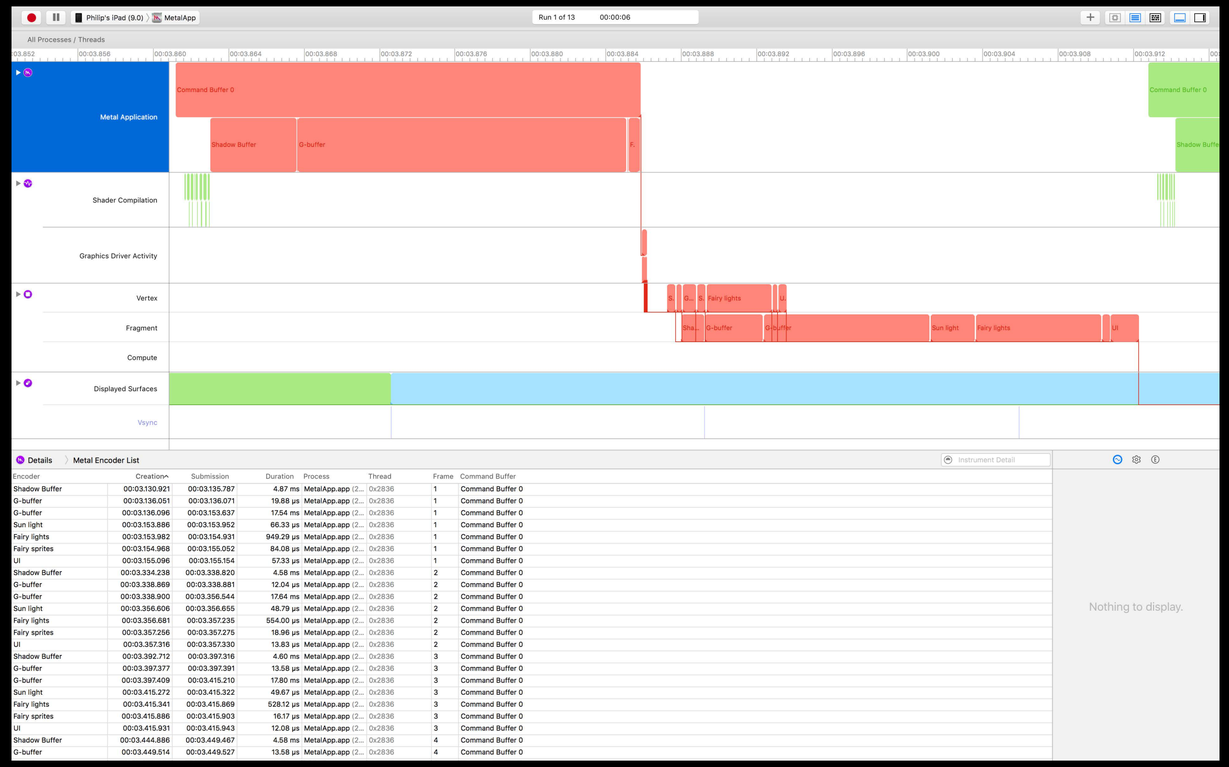
Metal Tools in Instruments

Metal System Trace

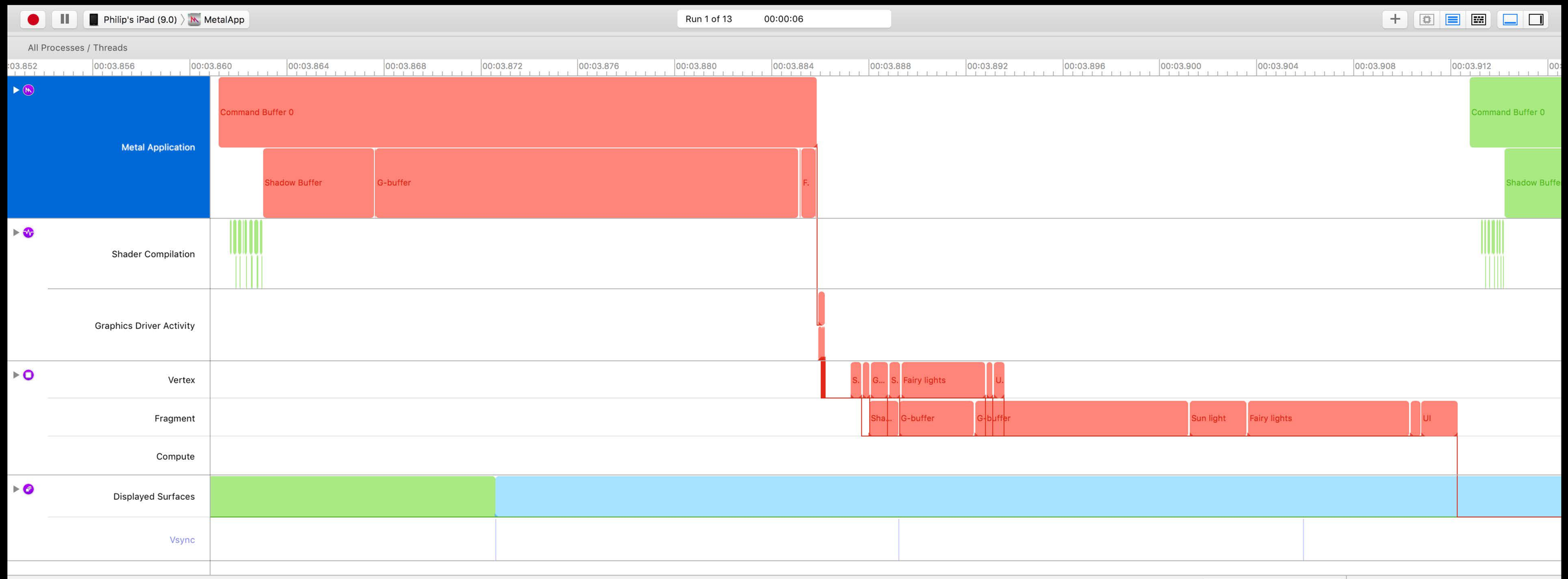
NEW

New instrument for iOS 9

Profile Metal apps across CPU and GPU



Encoder	Creation	Submission	Duration	Process	Thread	Frame	Command Buffer
Shadow Buffer	00:03.130.921	00:03.135.787	4.87 ms	MetalApp.app (2...	0x2836	1	Command Buffer 0
G-buffer	00:03.136.051	00:03.136.071	19.88 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
G-buffer	00:03.136.096	00:03.153.637	17.54 ms	MetalApp.app (2...	0x2836	1	Command Buffer 0
Sun light	00:03.153.886	00:03.153.952	66.33 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Fairy lights	00:03.153.982	00:03.154.931	949.29 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Fairy sprites	00:03.154.968	00:03.155.052	84.08 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
UI	00:03.155.096	00:03.155.154	57.33 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Shadow Buffer	00:03.334.238	00:03.338.820	4.58 ms	MetalApp.app (2...	0x2836	2	Command Buffer 0
G-buffer	00:03.338.869	00:03.338.881	12.04 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
G-buffer	00:03.338.900	00:03.356.544	17.64 ms	MetalApp.app (2...	0x2836	2	Command Buffer 0
Sun light	00:03.356.606	00:03.356.655	48.79 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Fairy lights	00:03.356.681	00:03.357.235	554.00 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Fairy sprites	00:03.357.256	00:03.357.275	18.96 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
UI	00:03.357.316	00:03.357.330	13.83 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Shadow Buffer	00:03.392.712	00:03.397.316	4.60 ms	MetalApp.app (2...	0x2836	3	Command Buffer 0
G-buffer	00:03.397.377	00:03.397.391	13.58 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
G-buffer	00:03.397.409	00:03.415.210	17.80 ms	MetalApp.app (2...	0x2836	3	Command Buffer 0
Sun light	00:03.415.272	00:03.415.322	49.67 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Fairy lights	00:03.415.341	00:03.415.869	528.12 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Fairy sprites	00:03.415.886	00:03.415.903	16.17 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
UI	00:03.415.931	00:03.415.943	12.08 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Shadow Buffer	00:03.444.886	00:03.449.467	4.58 ms	MetalApp.app (2...	0x2836	4	Command Buffer 0
G-buffer	00:03.449.514	00:03.449.527	13.58 µs	MetalApp.app (2...	0x2836	4	Command Buffer 0

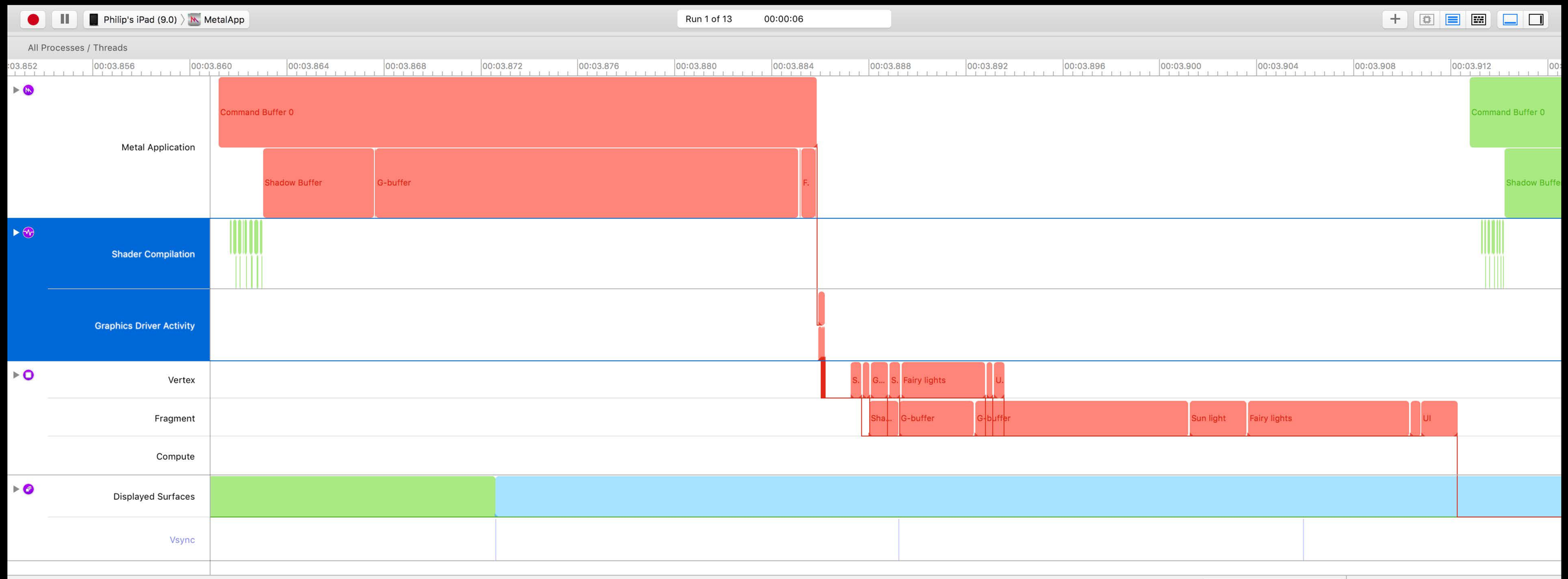


Details 〉 Metal Encoder List

Instrument Detail

Encoder	Creation	Submission	Duration	Process	Thread	Frame	Command Buffer
Shadow Buffer	00:03.130.921	00:03.135.787	4.87 ms	MetalApp.app (2...	0x2836	1	Command Buffer 0
G-buffer	00:03.136.051	00:03.136.071	19.88 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
G-buffer	00:03.136.096	00:03.153.637	17.54 ms	MetalApp.app (2...	0x2836	1	Command Buffer 0
Sun light	00:03.153.886	00:03.153.952	66.33 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Fairy lights	00:03.153.982	00:03.154.931	949.29 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Fairy sprites	00:03.154.968	00:03.155.052	84.08 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
UI	00:03.155.096	00:03.155.154	57.33 µs	MetalApp.app (2...	0x2836	1	Command Buffer 0
Shadow Buffer	00:03.334.238	00:03.338.820	4.58 ms	MetalApp.app (2...	0x2836	2	Command Buffer 0
G-buffer	00:03.338.869	00:03.338.881	12.04 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
G-buffer	00:03.338.900	00:03.356.544	17.64 ms	MetalApp.app (2...	0x2836	2	Command Buffer 0
Sun light	00:03.356.606	00:03.356.655	48.79 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Fairy lights	00:03.356.681	00:03.357.235	554.00 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Fairy sprites	00:03.357.256	00:03.357.275	18.96 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
UI	00:03.357.316	00:03.357.330	13.83 µs	MetalApp.app (2...	0x2836	2	Command Buffer 0
Shadow Buffer	00:03.392.712	00:03.397.316	4.60 ms	MetalApp.app (2...	0x2836	3	Command Buffer 0
G-buffer	00:03.397.377	00:03.397.391	13.58 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
G-buffer	00:03.397.409	00:03.415.210	17.80 ms	MetalApp.app (2...	0x2836	3	Command Buffer 0
Sun light	00:03.415.272	00:03.415.322	49.67 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Fairy lights	00:03.415.341	00:03.415.869	528.12 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Fairy sprites	00:03.415.886	00:03.415.903	16.17 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
UI	00:03.415.931	00:03.415.943	12.08 µs	MetalApp.app (2...	0x2836	3	Command Buffer 0
Shadow Buffer	00:03.444.886	00:03.449.467	4.58 ms	MetalApp.app (2...	0x2836	4	Command Buffer 0
G-buffer	00:03.449.514	00:03.449.527	13.58 µs	MetalApp.app (2...	0x2836	4	Command Buffer 0

Nothing to display.

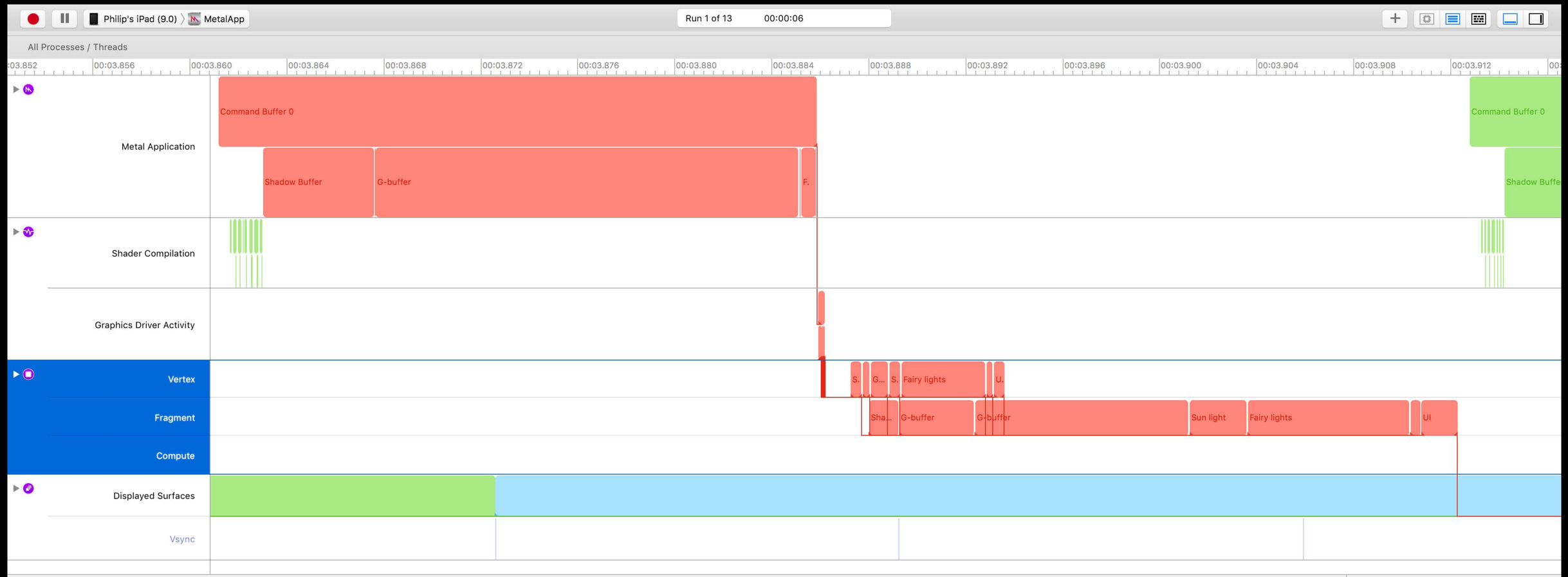


Details 〉 Graphics Driver Events

Instrument Detail

Source	Event	Timestamp	Notes	Process
Framebuffer	Has depth-stencil load/st...	00:03.130.920	Depth store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.136.050	Store: 1 2 3	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.136.050	Depth store, Stencil store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.136.095	Store: 1 2 3	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.136.096	Depth store, Stencil store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.153.884	Store: 1	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.153.885	Depth load and store, Stencil load and store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.153.981	Load: 1, Store: 1	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.153.982	Depth load and store, Stencil load and store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.154.967	Load: 1, Store: 1	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.155.096	Store: 1	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.334.237	Depth store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.338.868	Store: 1 2 3	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.338.868	Depth store, Stencil store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.338.899	Store: 1 2 3	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.338.900	Depth store, Stencil store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.356.605	Store: 1	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.356.605	Depth load and store, Stencil load and store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.356.680	Load: 1, Store: 1	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.356.680	Depth load and store, Stencil load and store	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.357.255	Load: 1, Store: 1	MetalApp.app (2...
Framebuffer	Has color load/stores	00:03.357.316	Store: 1	MetalApp.app (2...
Framebuffer	Has depth-stencil load/st...	00:03.392.711	Depth store	MetalApp.app (2...

Nothing to display.

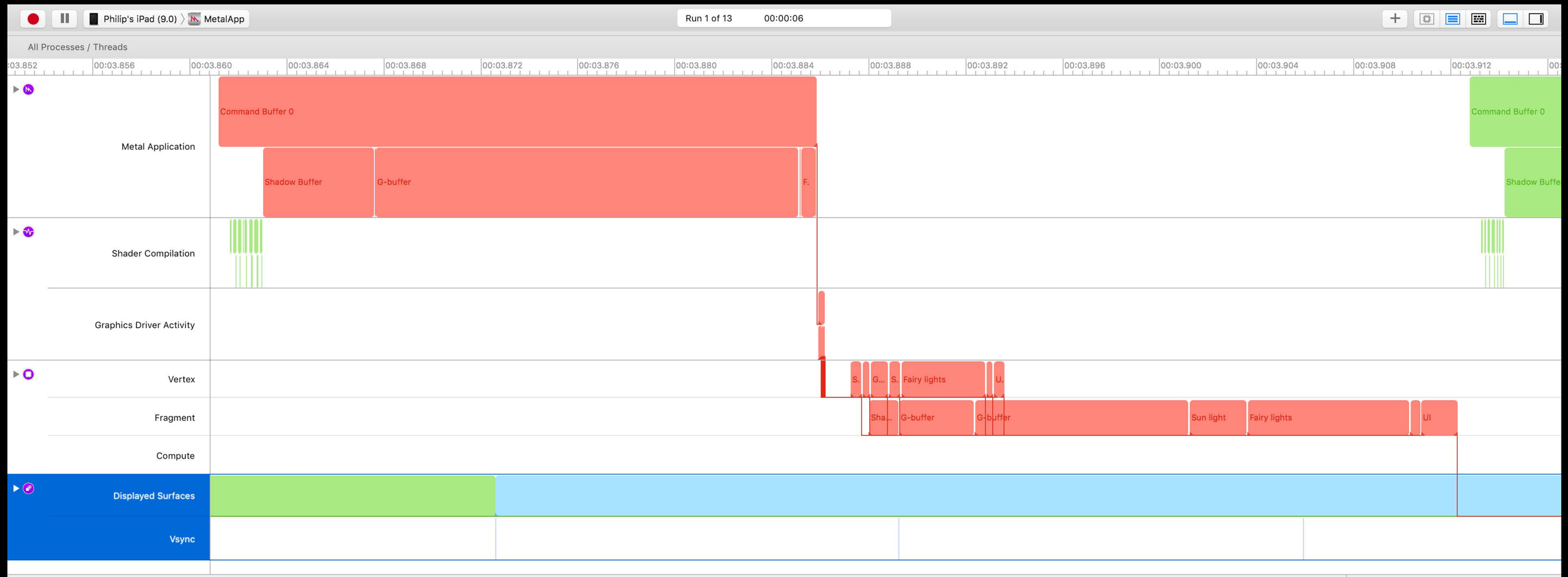


Details 〉 Summary

Instrument Detail

Event / State	Count	Total Duration	Min Duration	Avg Duration	Max Duration	Std Dev Duration
* All *	2,089	7.69 s	46.67 µs	3.68 ms	151.23 ms	10.11 ms
Vertex	1,041	3.84 s	52.17 µs	3.69 ms	151.23 ms	11.96 ms
Fragment	1,048	3.85 s	46.67 µs	3.68 ms	129.73 ms	7.85 ms

Nothing to display.



Details 〉 Displayed Surfaces

Instrument Detail

Start Time	End Time	Duration	Process
00:05.689.596	00:05.722.696	33.10 ms	MetalApp.app (2...
00:05.722.696	00:05.772.699	50.00 ms	MetalApp.app (2...
00:05.772.699	00:05.822.695	50.00 ms	MetalApp.app (2...
00:05.822.695	00:05.872.697	50.00 ms	MetalApp.app (2...
00:05.872.697	00:05.922.705	50.01 ms	MetalApp.app (2...
00:05.922.705	00:05.989.368	66.66 ms	MetalApp.app (2...
00:05.989.368	00:06.039.378	50.01 ms	MetalApp.app (2...
00:06.039.378	00:06.089.375	50.00 ms	MetalApp.app (2...
00:06.089.375	00:06.139.386	50.01 ms	MetalApp.app (2...
00:06.139.386	00:06.189.379	49.99 ms	MetalApp.app (2...
00:06.189.379	00:06.239.387	50.01 ms	MetalApp.app (2...
00:06.239.387	00:06.289.393	50.01 ms	MetalApp.app (2...
00:06.289.393	00:06.306.271	16.88 ms	MetalApp.app (2...
00:06.306.271	00:06.339.397	33.13 ms	
00:06.339.397	00:06.389.393	50.00 ms	MetalApp.app (2...
00:06.389.393	00:06.439.391	50.00 ms	MetalApp.app (2...
00:06.439.391	00:06.489.395	50.00 ms	MetalApp.app (2...
00:06.489.395	00:06.539.395	50.00 ms	MetalApp.app (2...
00:06.539.395	00:06.589.402	50.01 ms	MetalApp.app (2...
00:06.589.402	00:06.656.072	66.67 ms	MetalApp.app (2...
00:06.656.072	00:06.706.079	50.01 ms	MetalApp.app (2...
00:06.706.079	00:06.756.080	50.00 ms	MetalApp.app (2...
00:06.756.080	00:06.806.087	50.01 ms	MetalApp.app (2...

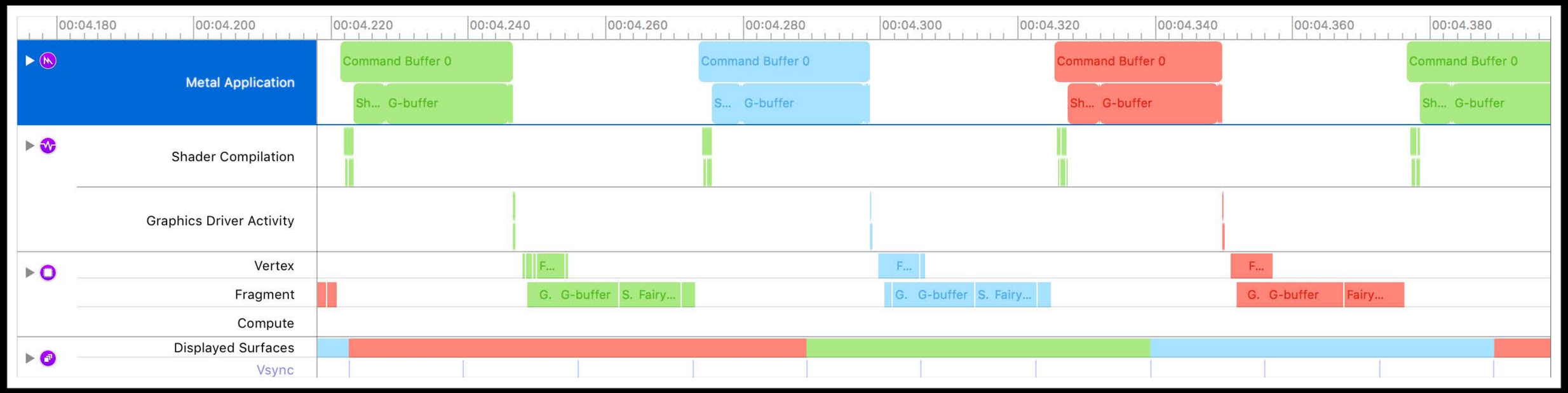
Nothing to display.

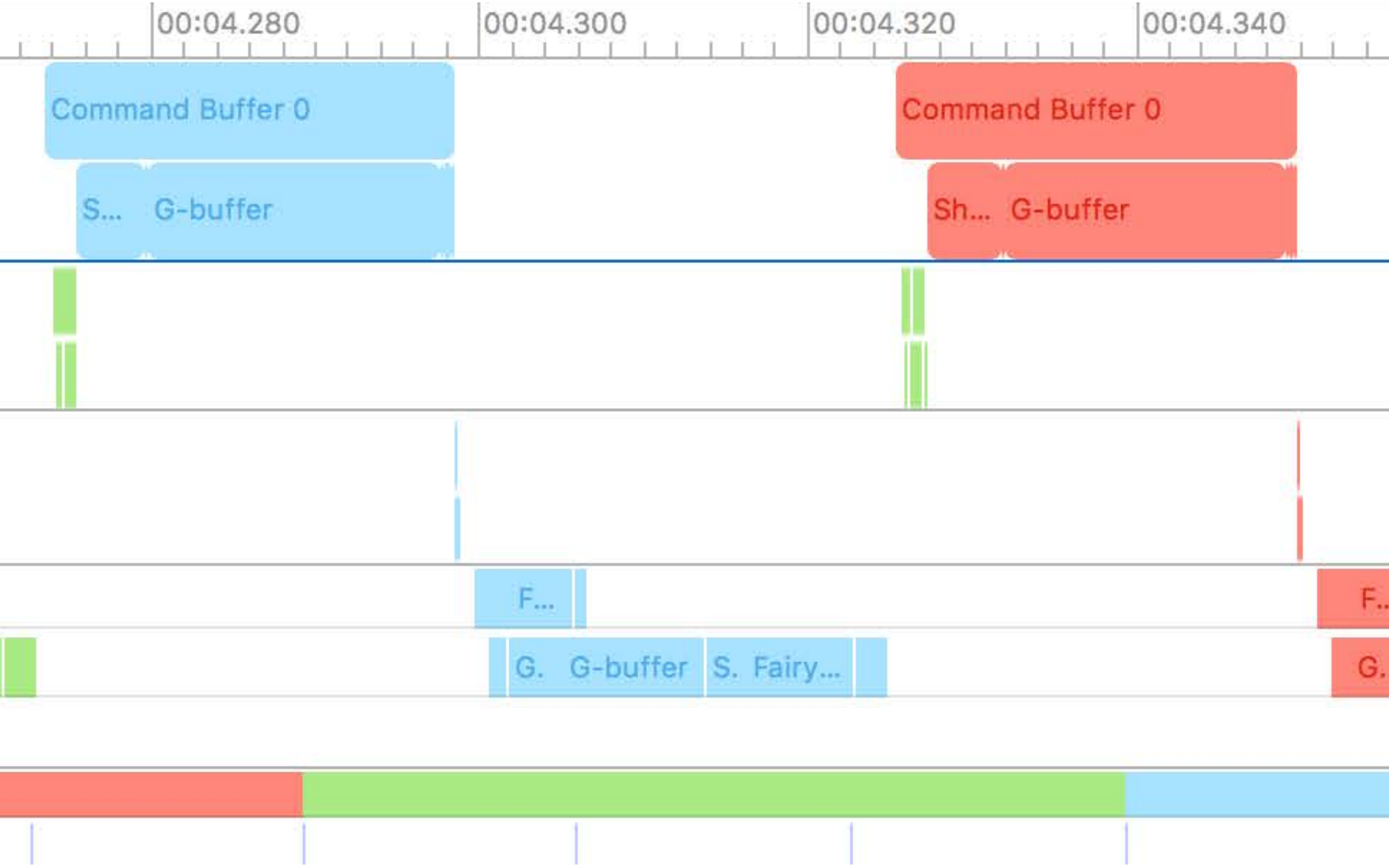
Demo

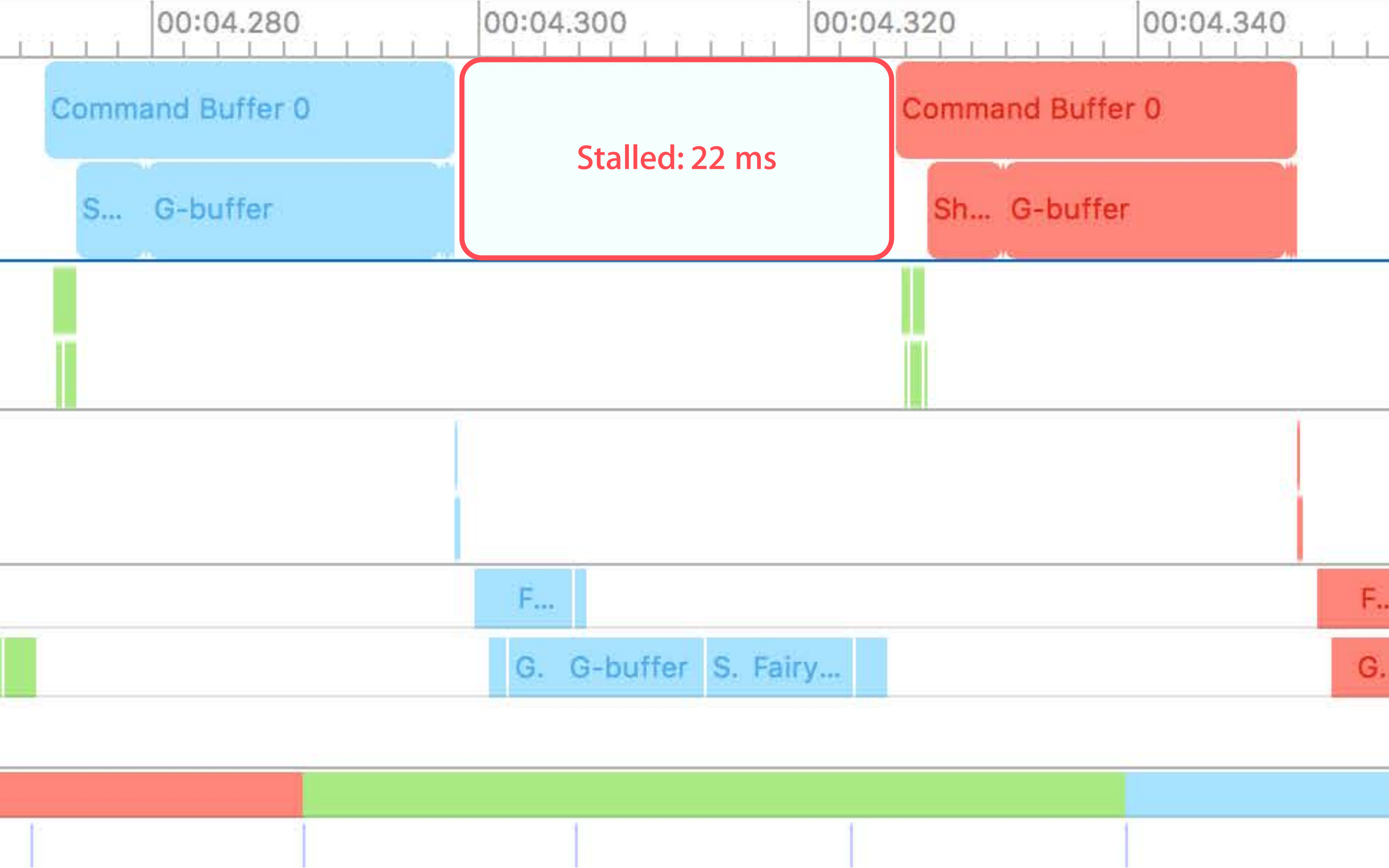
Metal System Trace

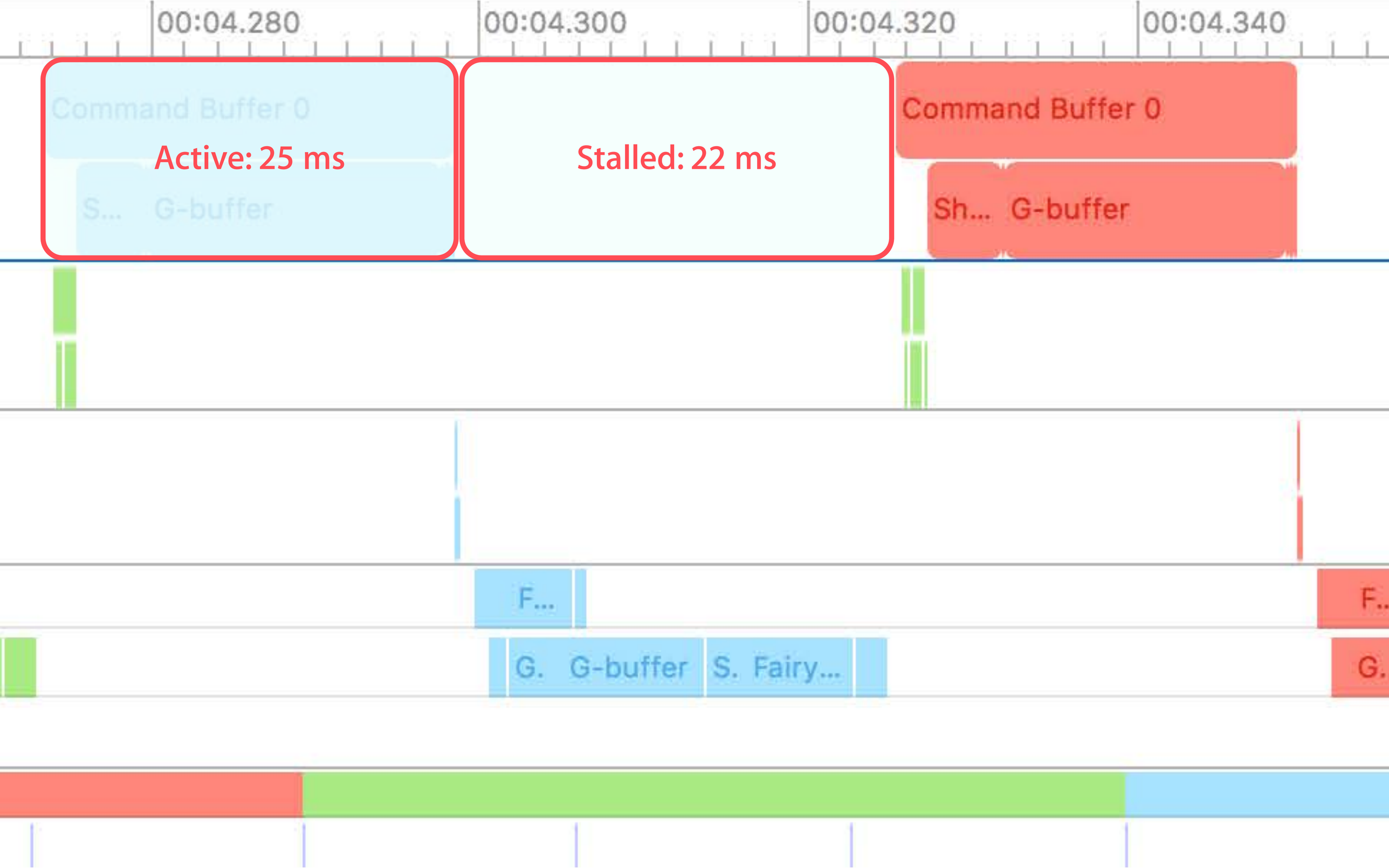
Serhat Tekin
GPU Software Developer Technologies

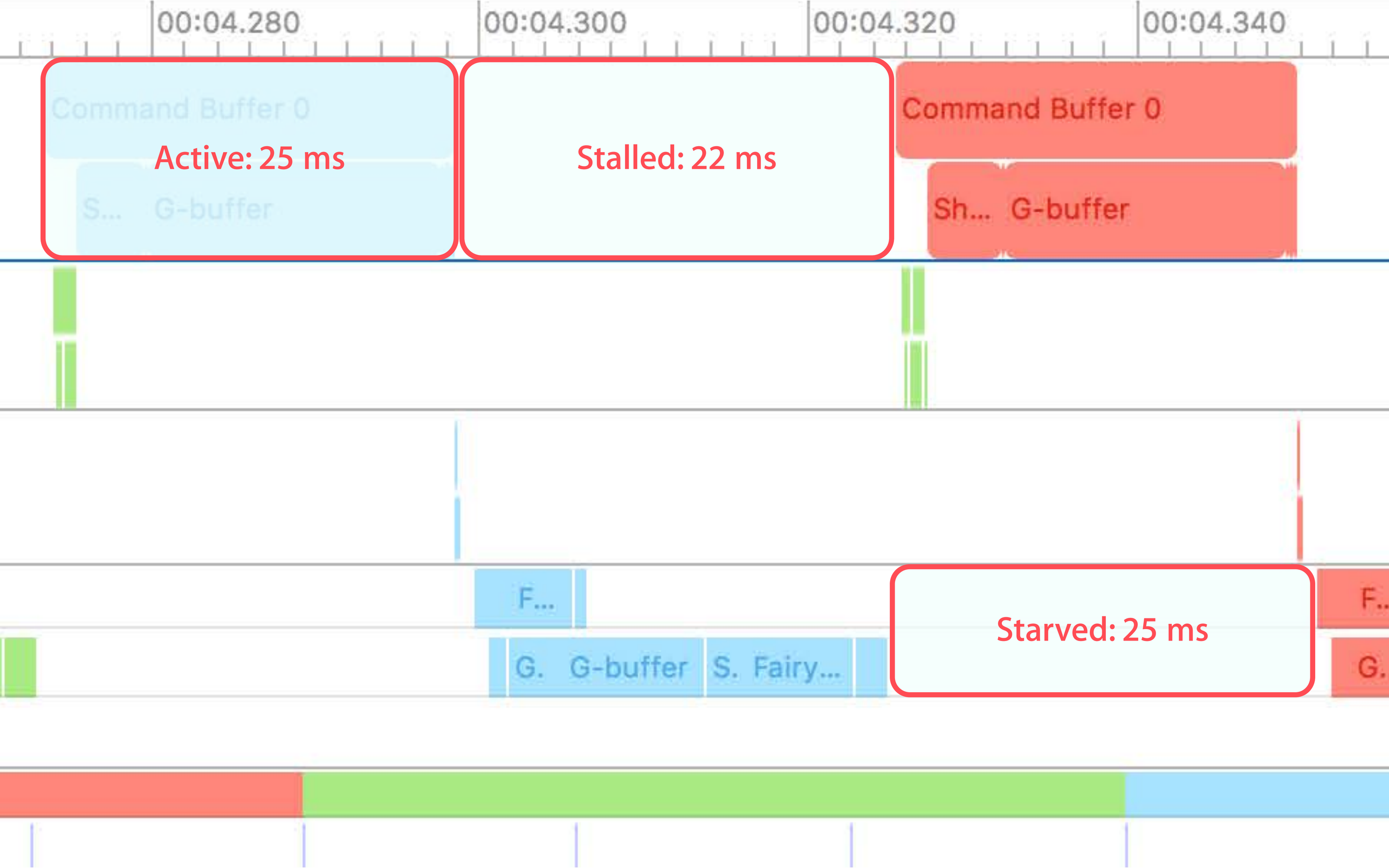
Metal Performance Best Practices

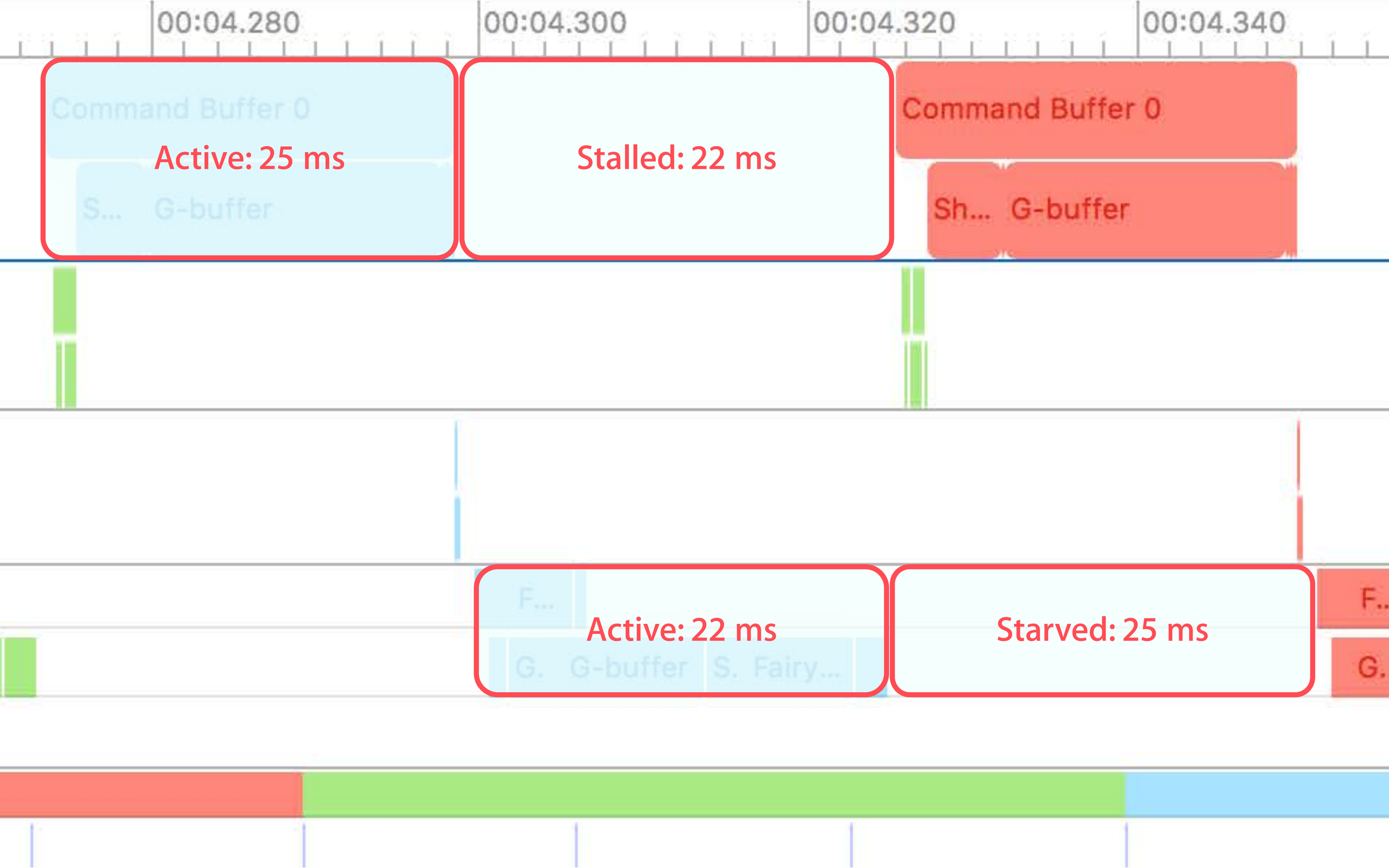

00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
Metal Application
Command Buffer 0
Sh... G-buffer
Command Buffer 0
S... G-buffer
Command Buffer 0
Sh... G-buffer
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F...
F...
F...
Fragment
G. G-buffer
S. Fairy...
G. G-buffer
S. Fairy...
G. G-buffer
Fairy...
Compute
Displayed Surfaces
Vsync

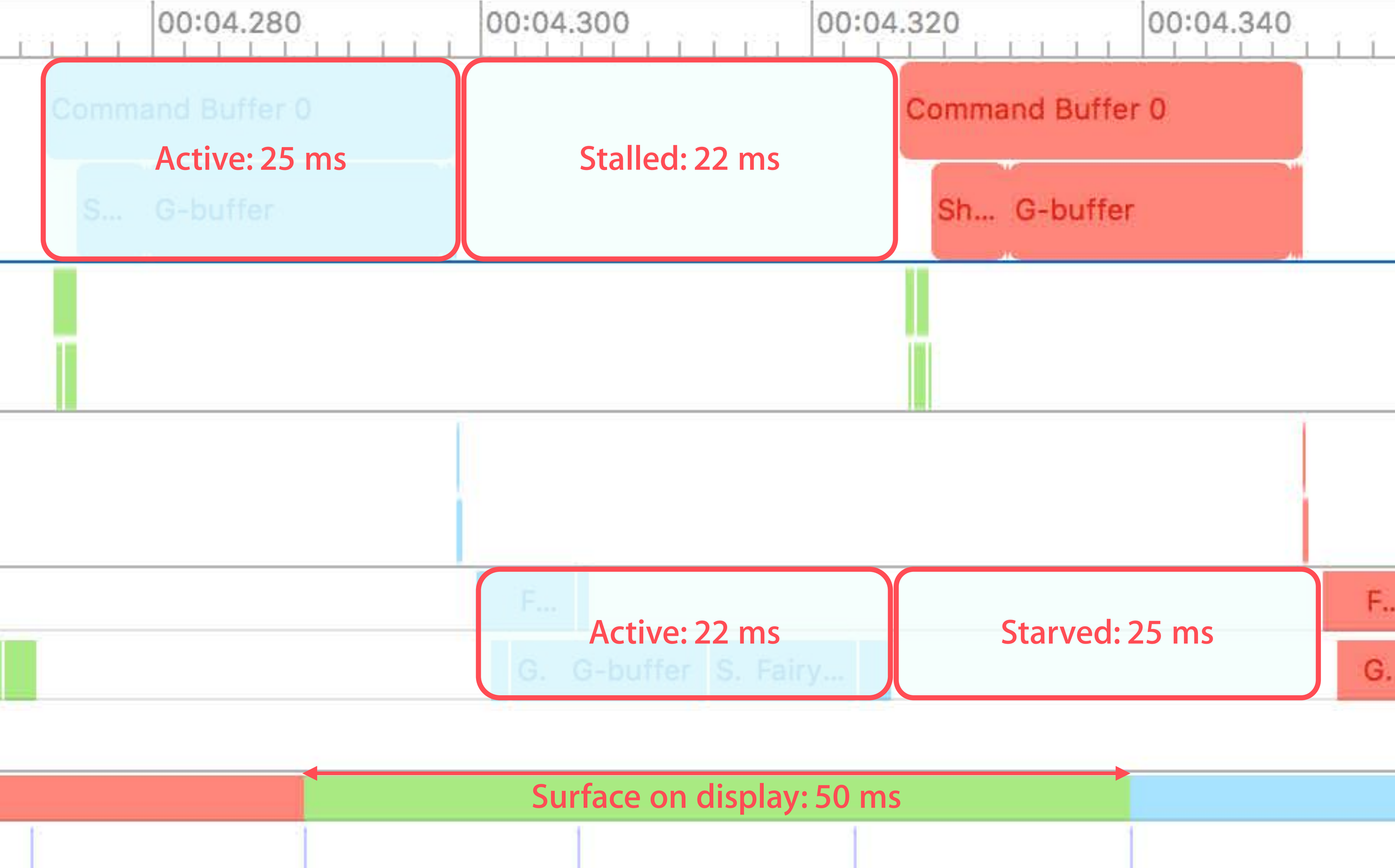

00:04.280
00:04.300
00:04.320
00:04.340
Command Buffer 0
S...
G-buffer
Command Buffer 0
Sh...
G-buffer
F...
G.
G-buffer
S.
Fairy...
F..
G.


00:04.280
00:04.300
00:04.320
00:04.340
Command Buffer 0
S... G-buffer
Stalled: 22 ms
Command Buffer 0
Sh... G-buffer
F...
G. G-buffer
S. Fairy...


00:04.280
00:04.300
00:04.320
00:04.340
Command Buffer 0
Active: 25 ms
S... G-buffer
Stalled: 22 ms
Command Buffer 0
Sh... G-buffer
F...
F...
G. G-buffer
S. Fairy...
G.


00:04.280
00:04.300
00:04.320
00:04.340
Command Buffer 0
Active: 25 ms
S... G-buffer
Stalled: 22 ms
Command Buffer 0
Sh... G-buffer
F...
G. G-buffer
S. Fairy...
Starved: 25 ms
F...
G.


00:04.280
00:04.300
00:04.320
00:04.340
Command Buffer 0
Active: 25 ms
S...
G-buffer
Stalled: 22 ms
Command Buffer 0
Sh...
G-buffer
F...
Active: 22 ms
G.
G-buffer
S.
Fairy...
Starved: 25 ms
F..
G.


00:04.280
00:04.300
00:04.320
00:04.340
Command Buffer 0
Active: 25 ms
S...
G-buffer
Stalled: 22 ms
Command Buffer 0
Sh...
G-buffer
F...
Active: 22 ms
G.
G-buffer
S.
Fairy...
Starved: 25 ms
F..
G.
Surface on display: 50 ms

1

Create Expensive Objects Upfront and Reuse

Shaders and State

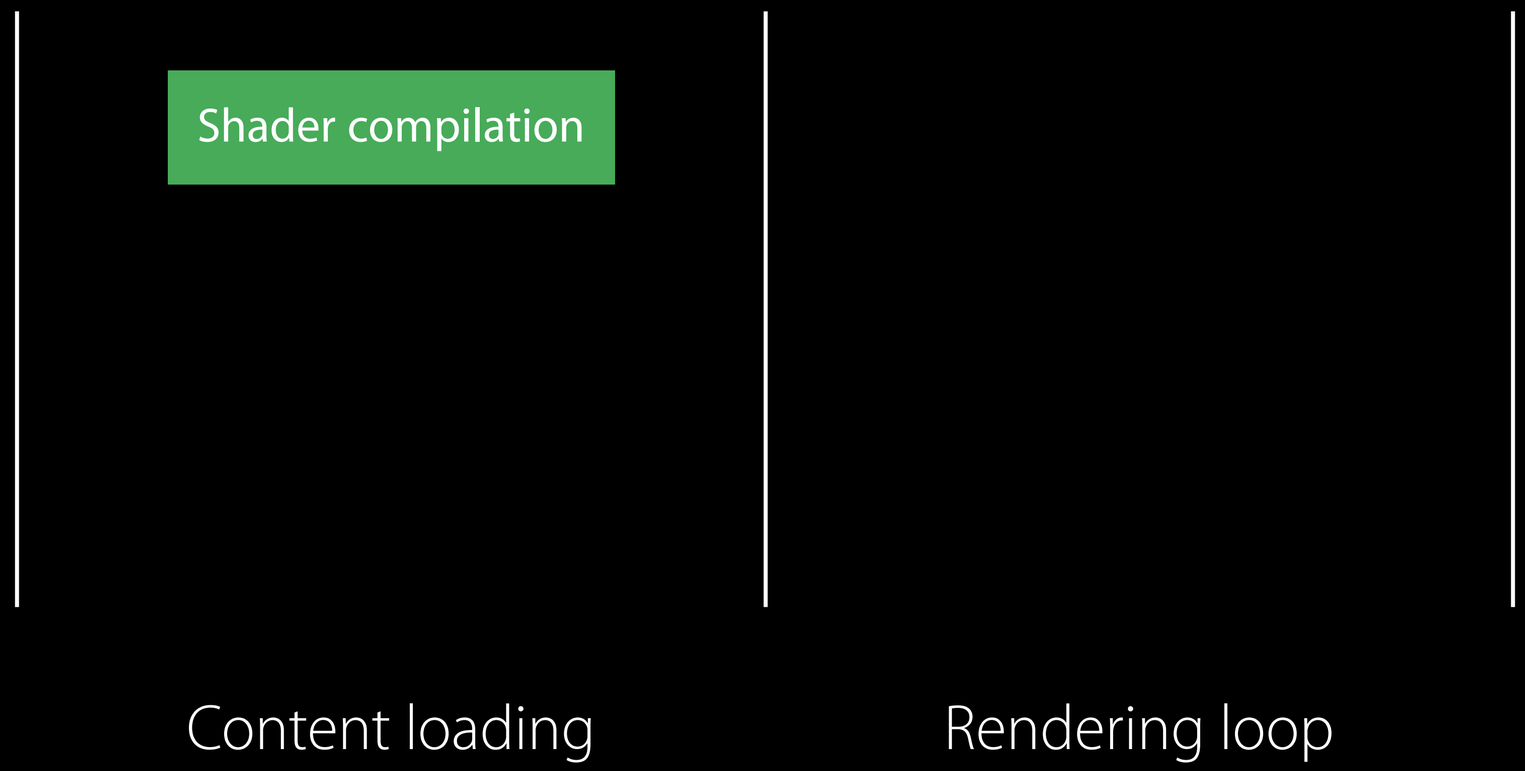
In a legacy app

Content loading

Rendering loop

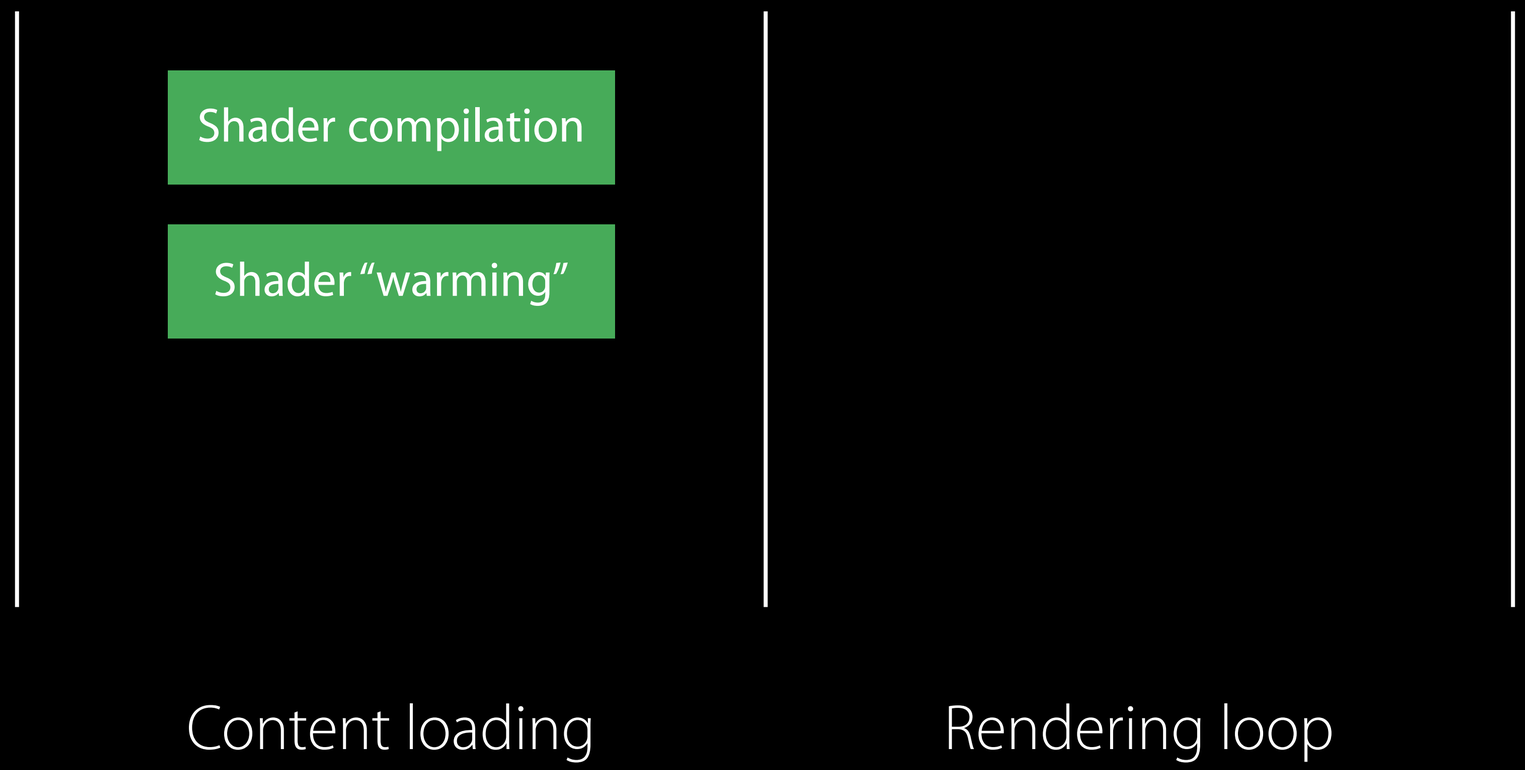
Shaders and State

In a legacy app



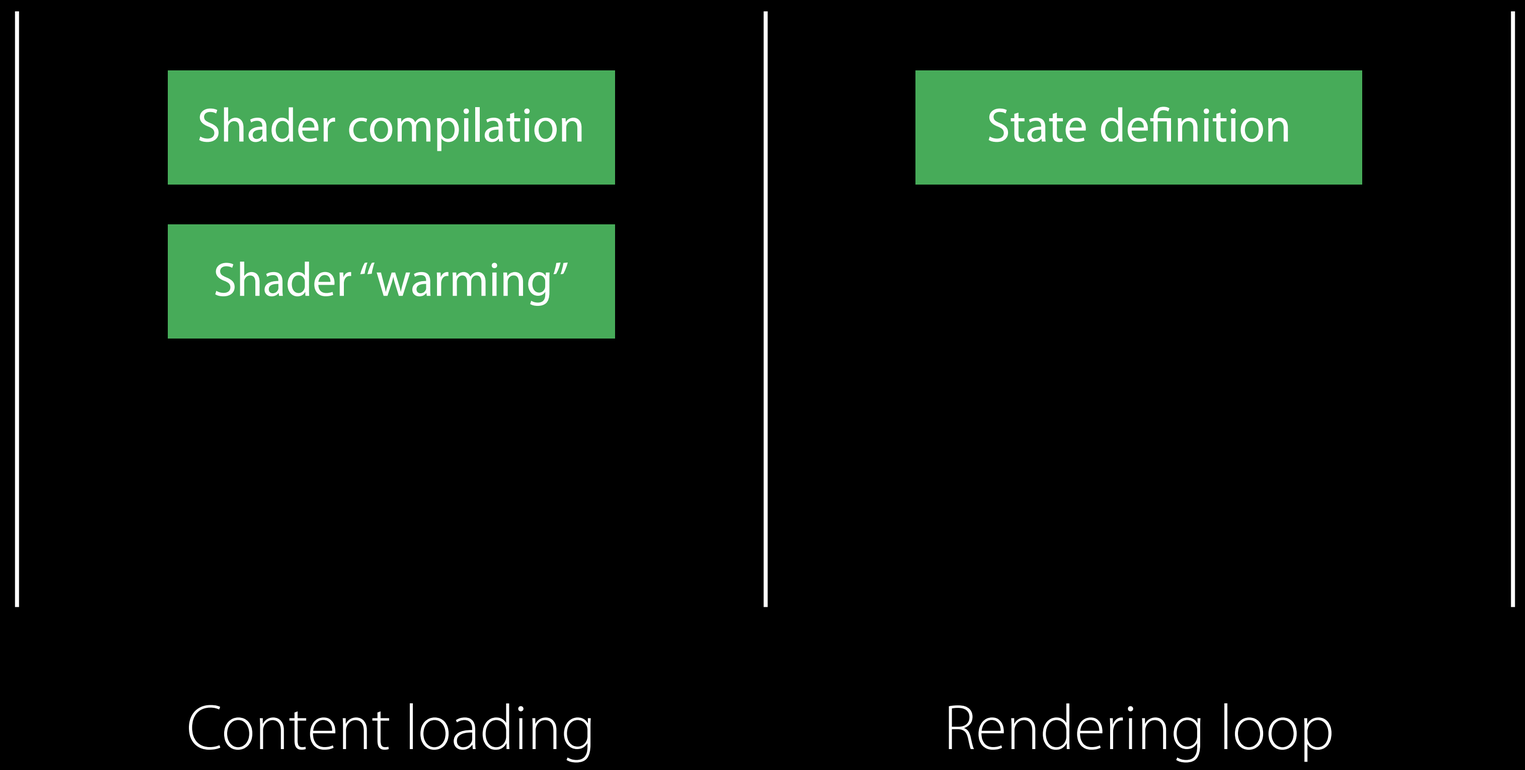
Shaders and State

In a legacy app



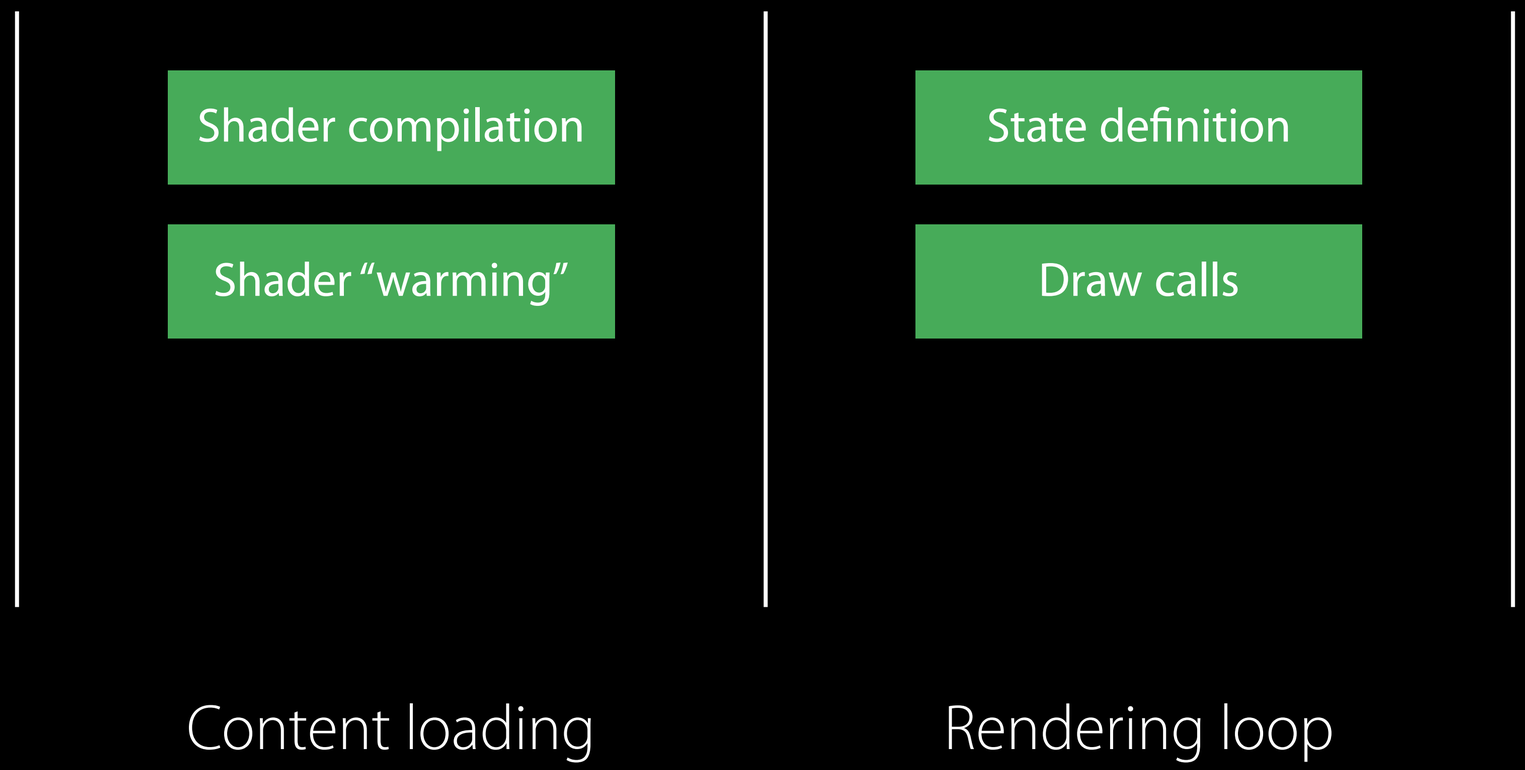
Shaders and State

In a legacy app



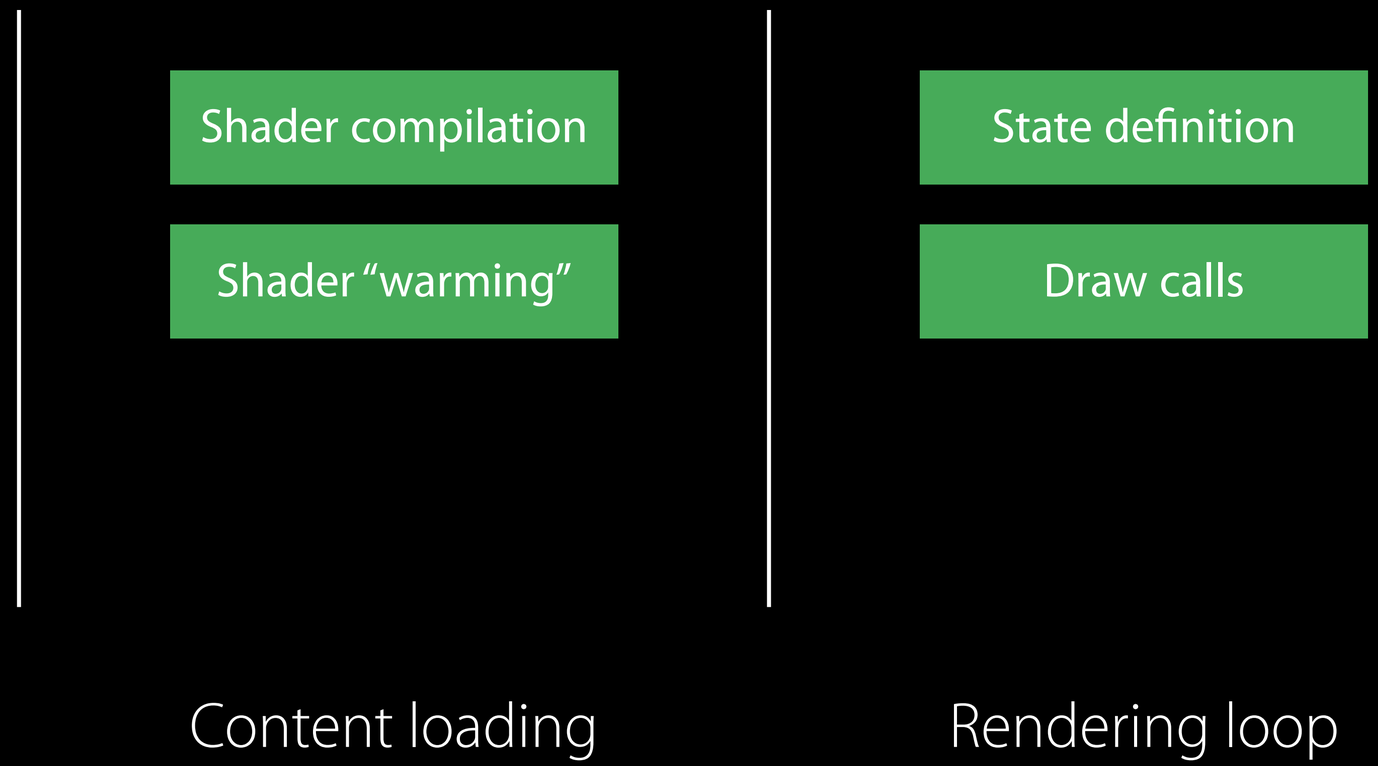
Shaders and State

In a legacy app



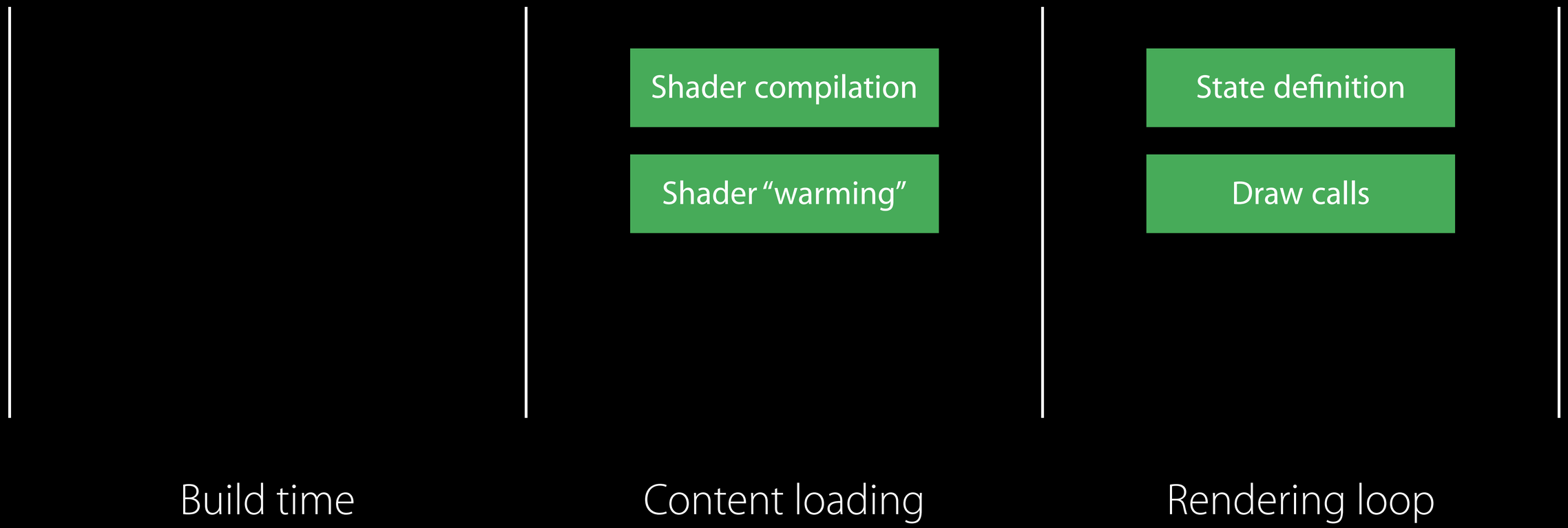
Shaders and State

In a Metal app



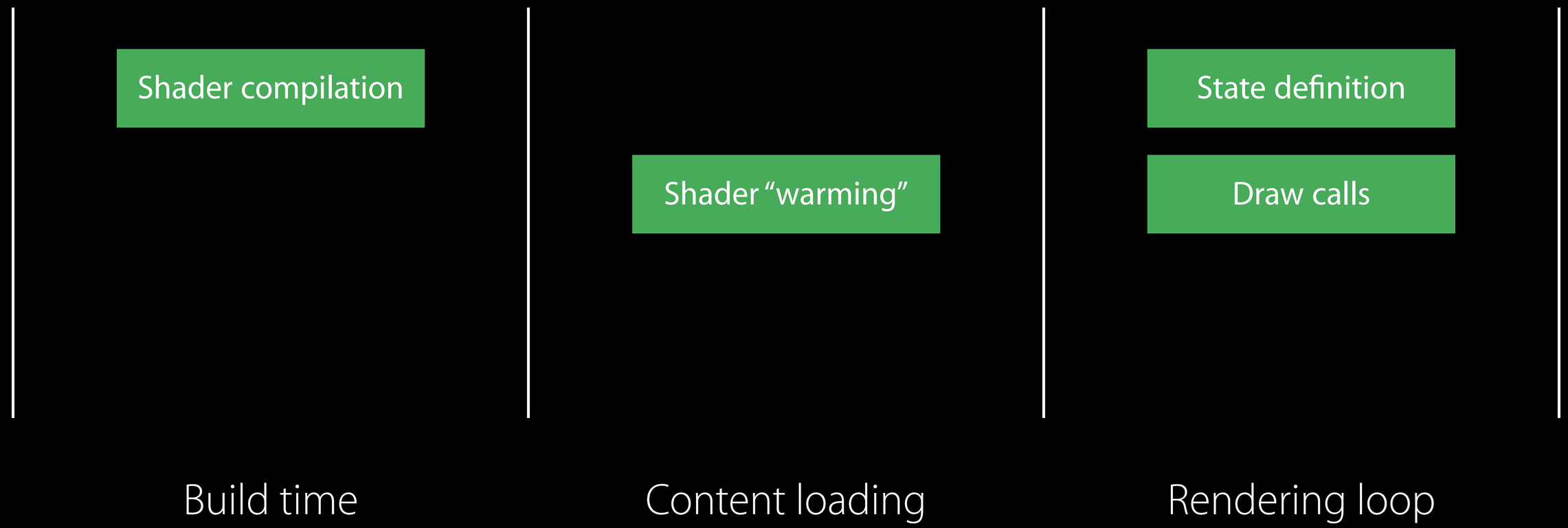
Shaders and State

In a Metal app



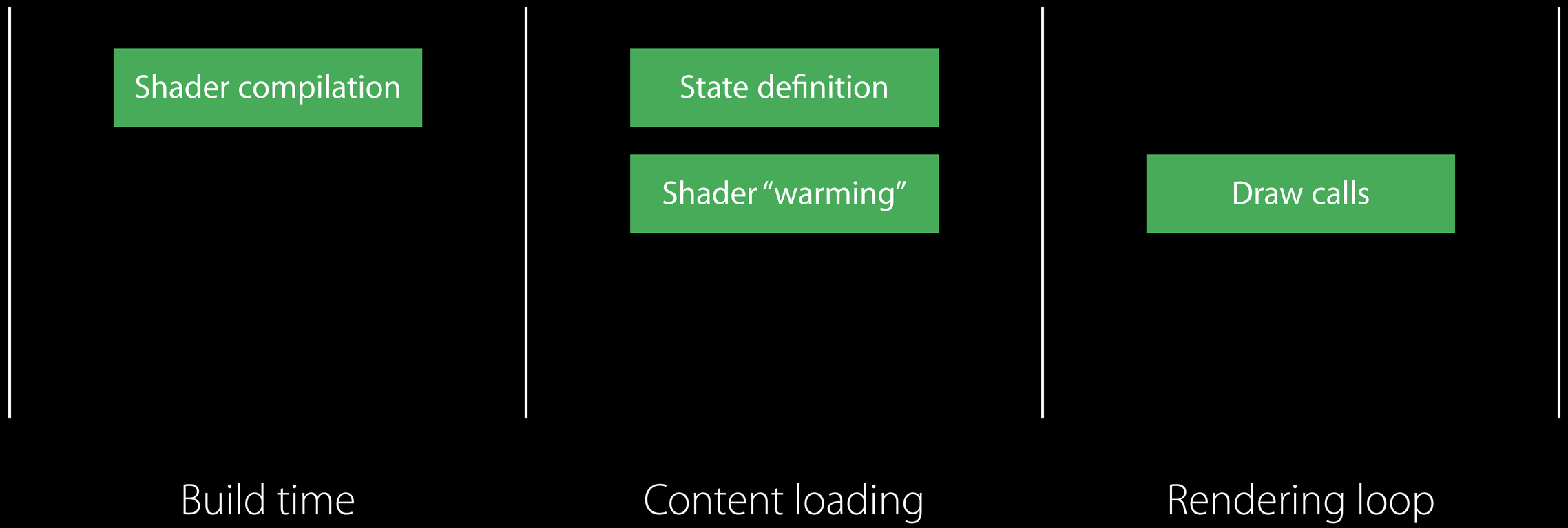
Shaders and State

In a Metal app



Shaders and State

In a Metal app



Shaders and State

In a Metal app

Build time	Content loading	Rendering loop
Shader compilation	State definition	Draw calls

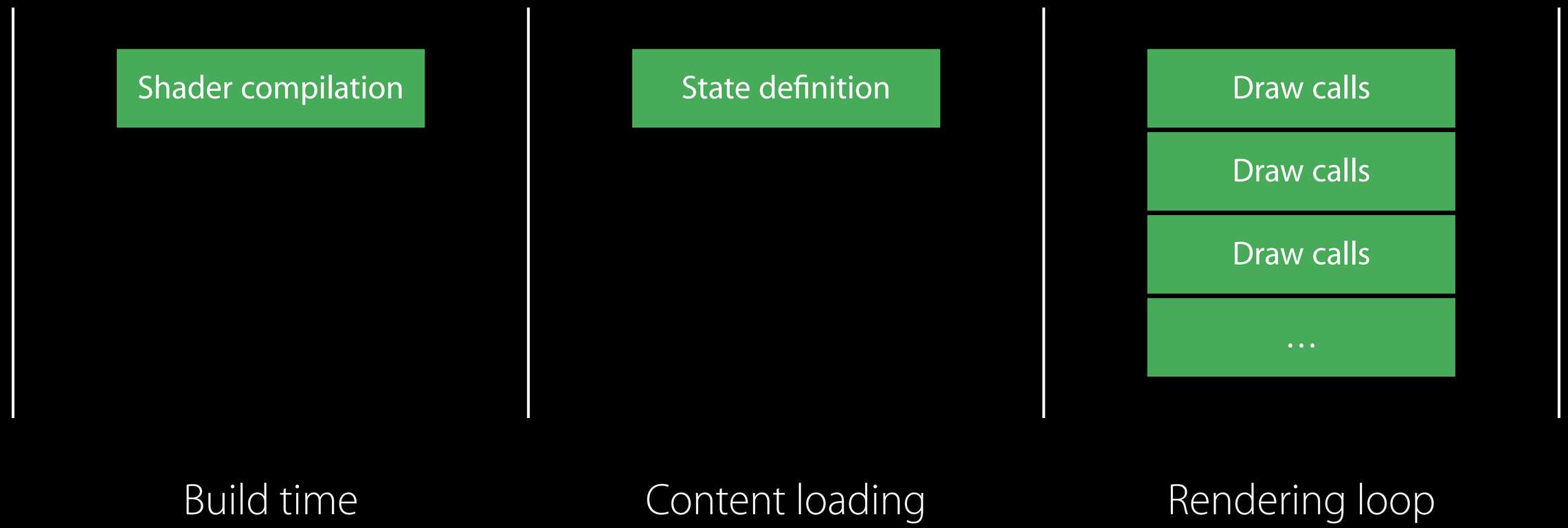
Shaders and State

In a Metal app

Shader compilation	State definition	Draw calls
Build time	Content loading	Rendering loop

Shaders and State

In a Metal app



Metal State Management

Fundamentals

Metal State Management

Fundamentals

Metal facilitates upfront state definition

- Decouples state validation and compilation from draw commands
- Allows expensive operations to be moved out of rendering loop

Metal State Management

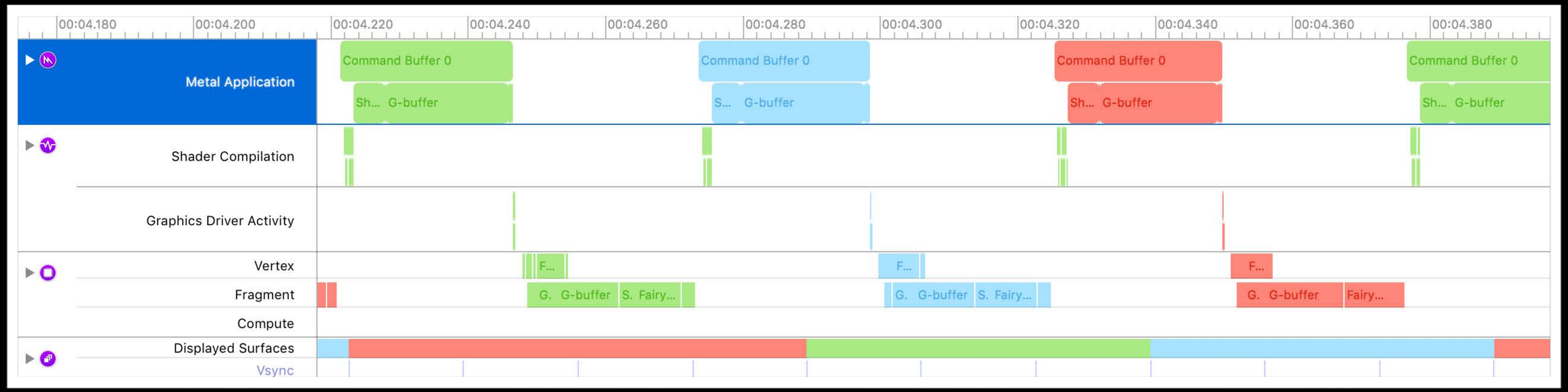
Fundamentals

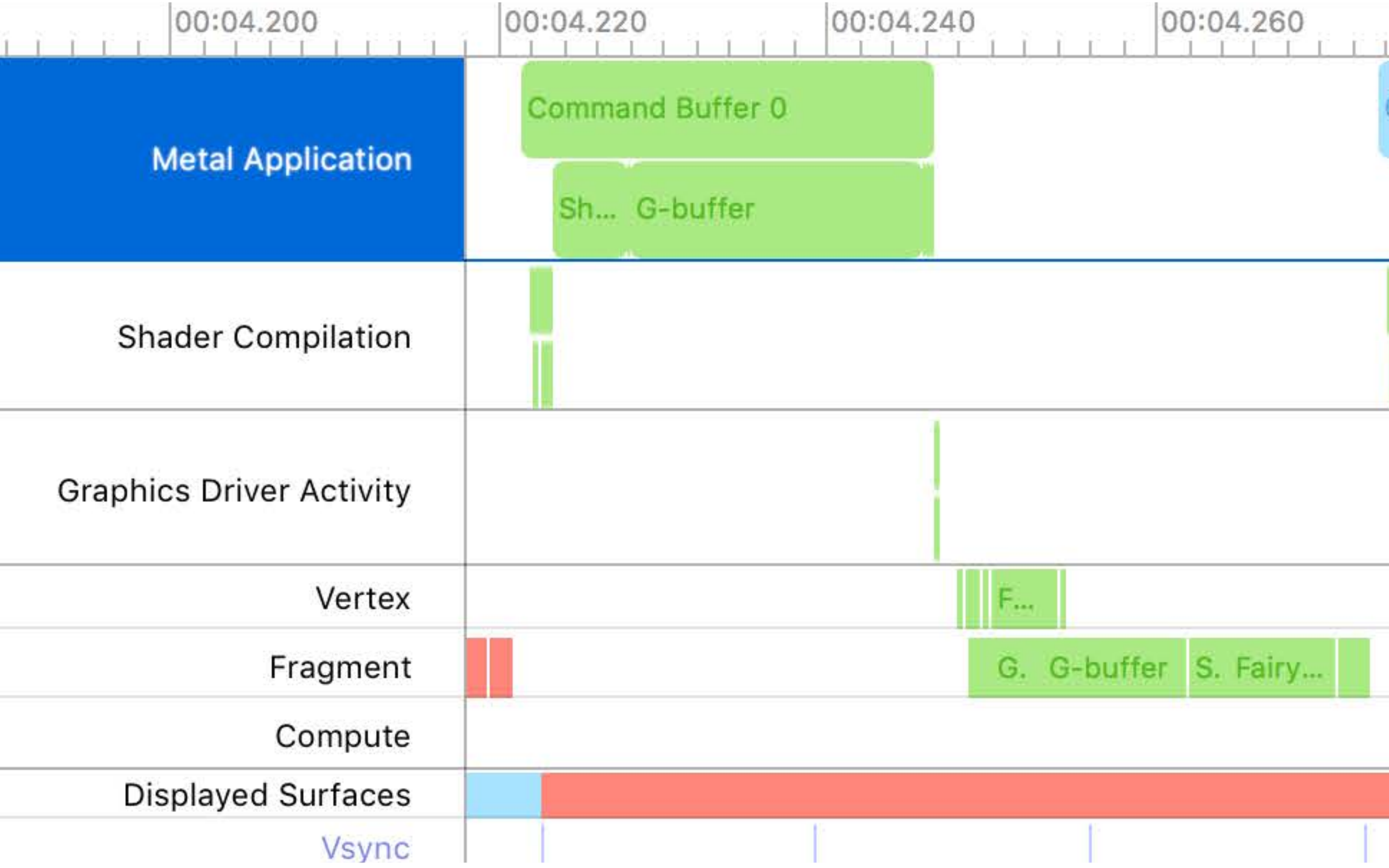
Metal facilitates upfront state definition

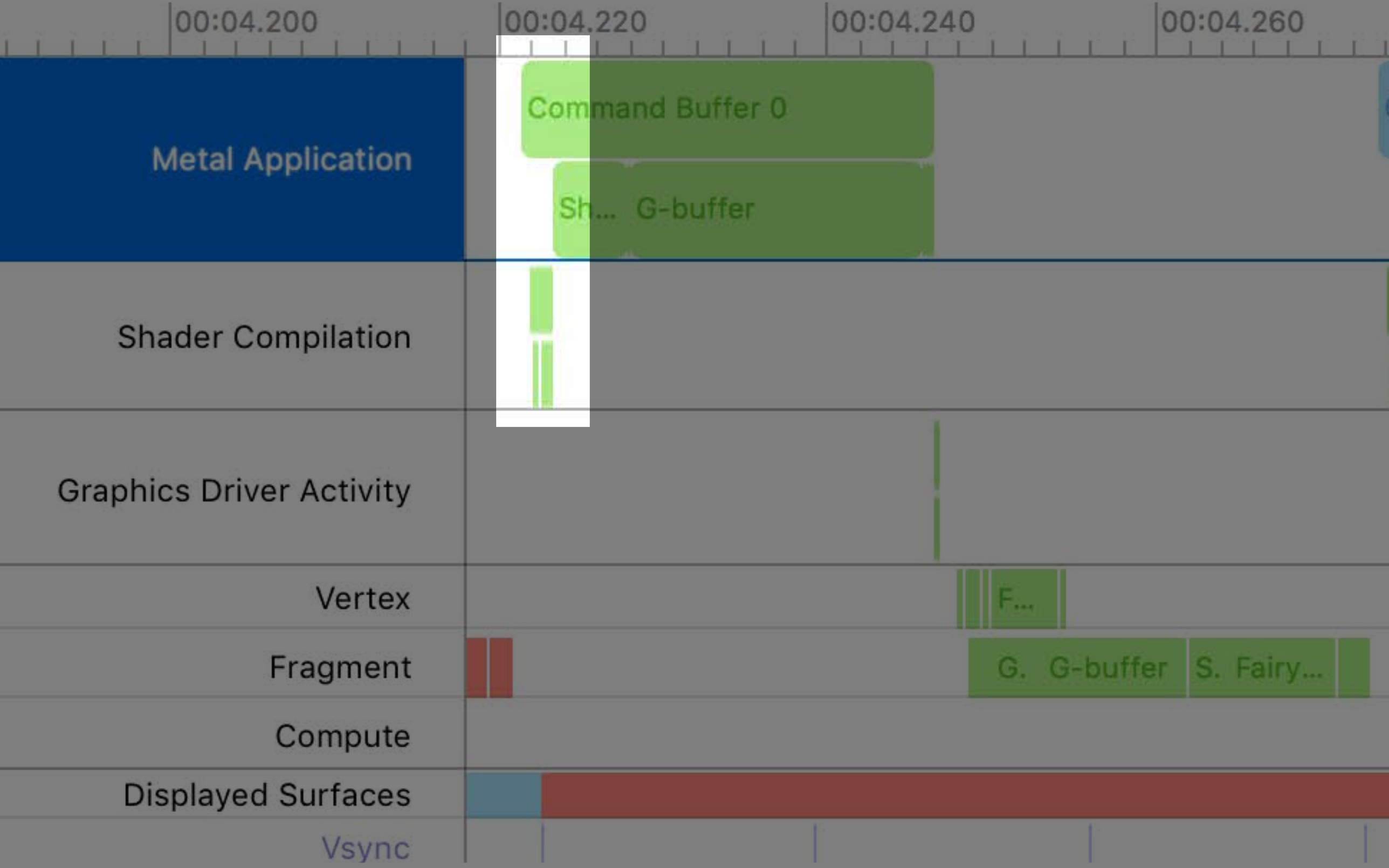
- Decouples state validation and compilation from draw commands
- Allows expensive operations to be moved out of rendering loop

Expensive to create state encapsulated in immutable state objects

- Intended to be created upfront and reused many times


00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
Metal Application
Command Buffer 0
Sh... G-buffer
S... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F...
Fragment
G. G-buffer
S. Fairy...
Fairy...
Compute
Displayed Surfaces
Vsync


00:04.200
00:04.220
00:04.240
00:04.260
Metal Application
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F...
Fragment
G. G-buffer
S. Fairy...
Compute
Displayed Surfaces
Vsync


00:04.200
00:04.220
00:04.240
00:04.260
Metal Application
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F...
Fragment
G. G-buffer
S. Fairy...
Compute
Displayed Surfaces
Vsync

MetalApp

FPS 25 FPS

- 0 zOnly <- [0x1007c1af0 newFunctionWithName:"zOnly"]
- 1 Shadow Render <- [Device newRenderPipelineStateWithDescriptor:<d...
- 2 gBufferVert <- [0x1007c1af0 newFunctionWithName:"gBufferVert"]
- 3 gBufferFrag <- [0x1007c1af0 newFunctionWithName:"gBufferFrag"]
- 4 skyboxVert <- [0x1007c1af0 newFunctionWithName:"skyboxVert"]
- 5 skyboxFragAlt <- [0x1007c1af0 newFunctionWithName:"skyboxFragAl...
- 6 Skybox Render <- [Device newRenderPipelineStateWithDescriptor:<d...
- 7 GBuffer Render <- [Device newRenderPipelineStateWithDescriptor:<d...
- 8 lightVert <- [0x1007c1af0 newFunctionWithName:"lightVert"]
- 9 compositionVertex <- [0x1007c1af0 newFunctionWithName:"composit...
- 10 fairyVertex <- [0x1007c1af0 newFunctionWithName:"fairyVertex"]
- 11 fairyFragmentAlt <- [0x1007c1af0 newFunctionWithName:"fairyFrag...
- 12 lightFragAlt <- [0x1007c1af0 newFunctionWithName:"lightFragAlt"]
- 13 compositionFragAlt <- [0x1007c1af0 newFunctionWithName:"compo...
- 14 UIVert <- [0x1007c1af0 newFunctionWithName:"UIVert"]
- 15 UIFrag <- [0x1007c1af0 newFunctionWithName:"UIFrag"]
- 16 Light Mask Render <- [Device newRenderPipelineStateWithDescripto...
- 17 Light Color Render <- [Device newRenderPipelineStateWithDescripto...
- 18 Composition Render <- [Device newRenderPipelineStateWithDescrip...
- 19 Fairy Sprites <- [Device newRenderPipelineStateWithDescriptor:<dat...
- 20 UI <- [Device newRenderPipelineStateWithDescriptor:<data> error:0...
- 21 0x100358aa0 <- [MTLLayer nextDrawable]
- 22 0x100346d50 <- [MTLDrawable texture]
- 23 0x10121b700 <- [0x1007c0cf0 commandBuffer]
- CommandBuffer 0x10121b700
  - Shadow Buffer 11.31 µs
  - G-buffer 0.00 ns
  - G-buffer 11.10 ms ⚠
  - Sun light 3.08 ms
  - Fairy lights 958.72 µs
  - Fairy sprites 84.84 µs
  - UI 1.05 ms
  - 1030 [addCompletedHandler:0x16fdfd090]
  - 1031 [presentDrawable:0x100358aa0]
  - 1032 [commit]

- 8 lightVert <- [0x1007c1af0 newFunctionWithName:"lightVert"]
- 9 compositionVertex <- [0x1007c1af0 newFunctionWithName:"composit...
- 10 fairyVertex <- [0x1007c1af0 newFunctionWithName:"fairyVertex"]
- 11 fairyFragmentAlt <- [0x1007c1af0 newFunctionWithName:"fairyFrag...
- 12 lightFragAlt <- [0x1007c1af0 newFunctionWithName:"lightFragAlt"]
- 13 compositionFragAlt <- [0x1007c1af0 newFunctionWithName:"compo...
- 14 UIVert <- [0x1007c1af0 newFunctionWithName:"UIVert"]
- 15 UIFrag <- [0x1007c1af0 newFunctionWithName:"UIFrag"]
- 16 Light Mask Render <- [Device newRenderPipelineStateWithDescripto...
- 17 Light Color Render <- [Device newRenderPipelineStateWithDescripto...
- 18 Composition Render <- [Device newRenderPipelineStateWithDescrip...
- 19 Fairy Sprites <- [Device newRenderPipelineStateWithDescriptor:<dat...
- 20 UI <- [Device newRenderPipelineStateWithDescriptor:<data> error:0...
- 21 0x100358aa0 <- [MTLLayer nextDrawable]
- 22 0x100346d50 <- [MTLDrawable texture]
- 23 0x10121b700 <- [0x1007c0cf0 commandBuffer]

8 lightVert <- [0x1007c1af0 newFunctionWithName:"lightVert"]

9 compositionVertex <- [0x1007c1af0 newFunctionWithName:"composit...

10 fairyVertex <- [0x1007c1af0 newFunctionWithName:"fairyVertex"]

11 fairyFragmentAlt <- [0x1007c1af0 newFunctionWithName:"fairyFrag...

12 lightFragAlt <- [0x1007c1af0 newFunctionWithName:"lightFragAlt"]

13 compositionFragAlt <- [0x1007c1af0 newFunctionWithName:"compo...

14 UIVert <- [0x1007c1af0 newFunctionWithName:"UIVert"]

15 UIFrag <- [0x1007c1af0 newFunctionWithName:"UIFrag"]

16 Light Mask Render <- [Device newRenderPipelineStateWithDescripto...

17 Light Color Render <- [Device newRenderPipelineStateWithDescripto...

18 Composition Render <- [Device newRenderPipelineStateWithDescrip...

19 Fairy Sprites <- [Device newRenderPipelineStateWithDescriptor:<dat...

20 UI <- [Device newRenderPipelineStateWithDescriptor:<data> error:0...

21 0x100358aa0 <- [MTLLayer nextDrawable]

22 0x100346d50 <- [MTLDrawable texture]

23 0x10121b700 <- [0x1007c0cf0 commandBuffer]

Shader Library

MTLLibrary

Shader Library

MTLLibrary

A library of compiled shaders

Shader Library

MTLLibrary

A library of compiled shaders

Create offline using Xcode

- Automatically generates default library
- Accessed with `[MTLDevice newDefaultLibrary]`

Shader Library

MTLLibrary

A library of compiled shaders

Create offline using Xcode

- Automatically generates default library
- Accessed with `[MTLDevice newDefaultLibrary]`

Use command-line tools with shader creation pipeline

- Developer/Platforms/iPhoneOS.platform/usr/bin/metal, metal-ar, metallib
- Accessed with `[MTLDevice newLibraryWithFile:error:]`

Shader Library

MTLLibrary

A library of compiled shaders

Create offline using Xcode

- Automatically generates default library
- Accessed with `[MTLDevice newDefaultLibrary]`

Use command-line tools with shader creation pipeline

- Developer/Platforms/iPhoneOS.platform/usr/bin/metal, metal-ar, metallib
- Accessed with `[MTLDevice newLibraryWithFile:error:]`

Create asynchronously if runtime creation unavoidable

- `[MTLDevice newLibraryWithSource:options:completionHandler:]`

Device and Command Queue

MTLDevice and MTLCommandQueue

Device and Command Queue

MTLDevice and MTLCommandQueue

Represent a single GPU and queue of command buffers awaiting execution

Device and Command Queue

MTLDevice and MTLCommandQueue

Represent a single GPU and queue of command buffers awaiting execution

Create during app initialization and reuse throughout app lifetime

Device and Command Queue

MTLDevice and MTLCommandQueue

Represent a single GPU and queue of command buffers awaiting execution

Create during app initialization and reuse throughout app lifetime

Create one per GPU used

Render and Compute Pipeline State

MTLRenderPipelineState and MTLComputePipelineState

Render and Compute Pipeline State

MTLRenderPipelineState and MTLComputePipelineState

Encapsulate programmable GPU pipeline state

Render and Compute Pipeline State

MTLRenderPipelineState and MTLComputePipelineState

Encapsulate programmable GPU pipeline state

Create during content loading and reuse often

Render and Compute Pipeline State

MTLRenderPipelineState and MTLComputePipelineState

Encapsulate programmable GPU pipeline state

Create during content loading and reuse often

Can be created asynchronously

- `[MTLDevice newRenderPipelineStateWithDescriptor:options:completionHandler:]`
- `MTLDevice newComputePipelineStateWithDescriptor:options:completionHandler:]`

Render and Compute Pipeline State

MTLRenderPipelineState and MTLComputePipelineState

Encapsulate programmable GPU pipeline state

Create during content loading and reuse often

Can be created asynchronously

- `[MTLDevice newRenderPipelineStateWithDescriptor:options:completionHandler:]`
- `MTLDevice newComputePipelineStateWithDescriptor:options:completionHandler:]`

Don't obtain reflection data if not required

Depth/Stencil and Sampler State

MTLDepthStencilState and MTLSamplerState

Depth/Stencil and Sampler State

MTLDepthStencilState and MTLSamplerState

Encapsulate fixed-function GPU pipeline state

Depth/Stencil and Sampler State

MTLDepthStencilState and MTLSamplerState

Encapsulate fixed-function GPU pipeline state

Create during content loading along with pipeline state

Depth/Stencil and Sampler State

MTLDepthStencilState and MTLSamplerState

Encapsulate fixed-function GPU pipeline state

Create during content loading along with pipeline state

Some Metal implementations may hash state objects to avoid duplication

Buffer and Texture Resources

MTLBuffer and MTLTexture

Buffer and Texture Resources

MTLBuffer and MTLTexture

Data consumed by GPU

Buffer and Texture Resources

MTLBuffer and MTLTexture

Data consumed by GPU

Create during content loading and reuse often

Buffer and Texture Resources

MTLBuffer and MTLTexture

Data consumed by GPU

Create during content loading and reuse often

Create multiple copies of dynamic resources in order to buffer updates

State	When to Create

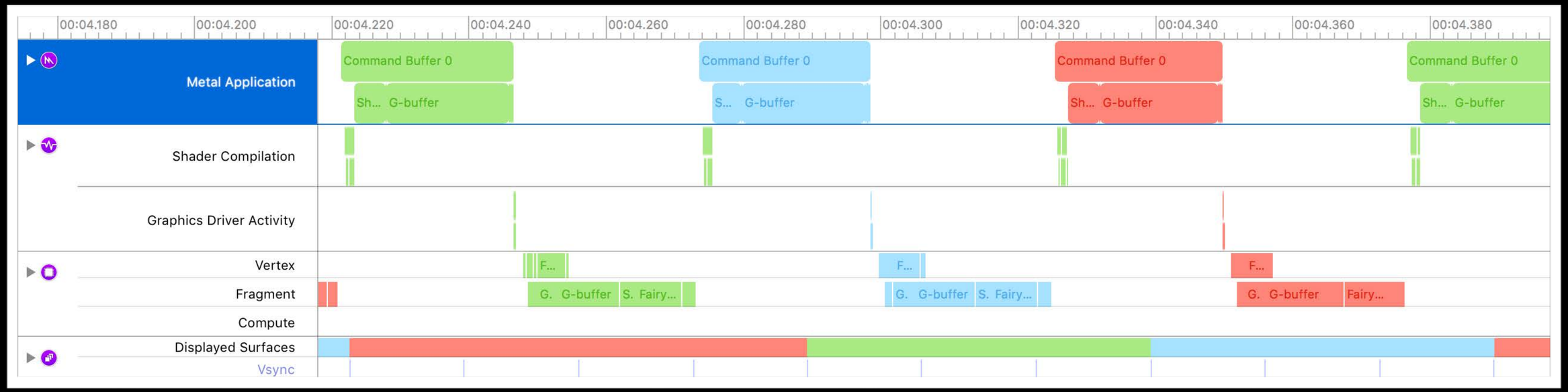
State	When to Create
`MTLLibrary`	App building

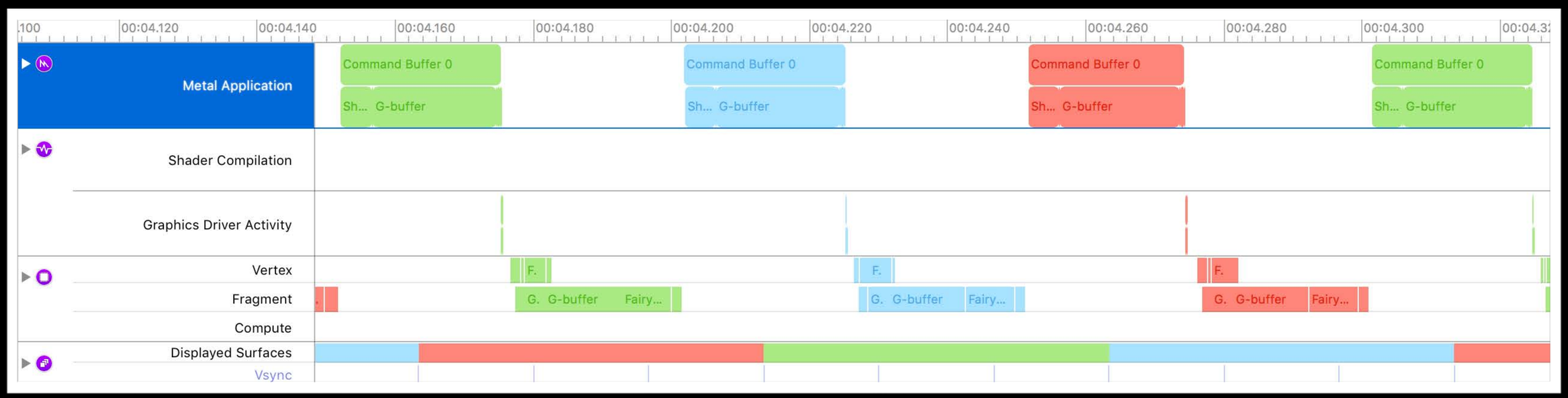
State	When to Create
`MTLLibrary`	App building
`MTLDevice` `MTLCommandQueue`	App initialization

State	When to Create
`MTLLibrary`	App building
`MTLDevice` `MTLCommandQueue`	App initialization
`MTLRenderPipelineState` `MTLComputePipelineState`	Content loading

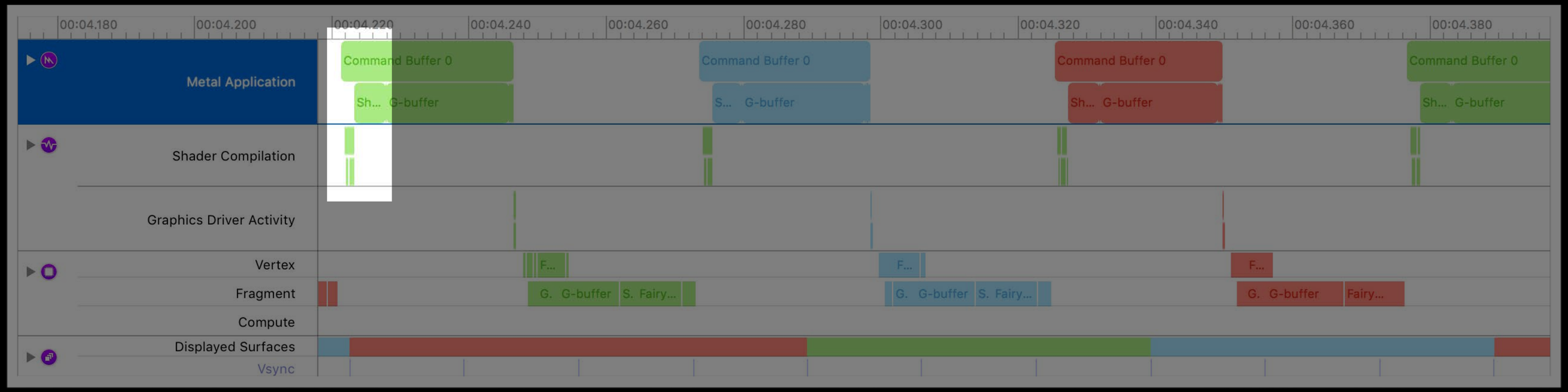
State	When to Create
`MTLLibrary`	App building
`MTLDevice` `MTLCommandQueue`	App initialization
`MTLRenderPipelineState` `MTLComputePipelineState`	Content loading
`MTLDepthStencilState` `MTLSamplerState`	Content loading

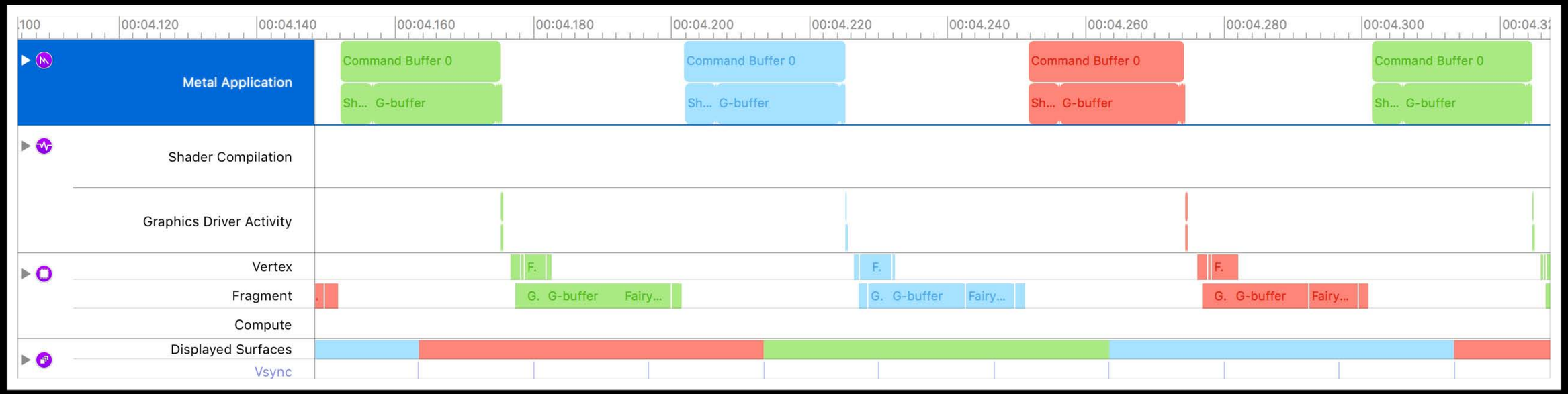
State	When to Create
`MTLLibrary`	App building
`MTLDevice` `MTLCommandQueue`	App initialization
`MTLRenderPipelineState` `MTLComputePipelineState`	Content loading
`MTLDepthStencilState` `MTLSamplerState`	Content loading
`MTLTexture` `MTLBuffer`	Content loading


00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
Metal Application
Command Buffer 0
Sh... G-buffer
S... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F...
Fragment
G. G-buffer
S. Fairy...
Fairy...
Compute
Displayed Surfaces
Vsync

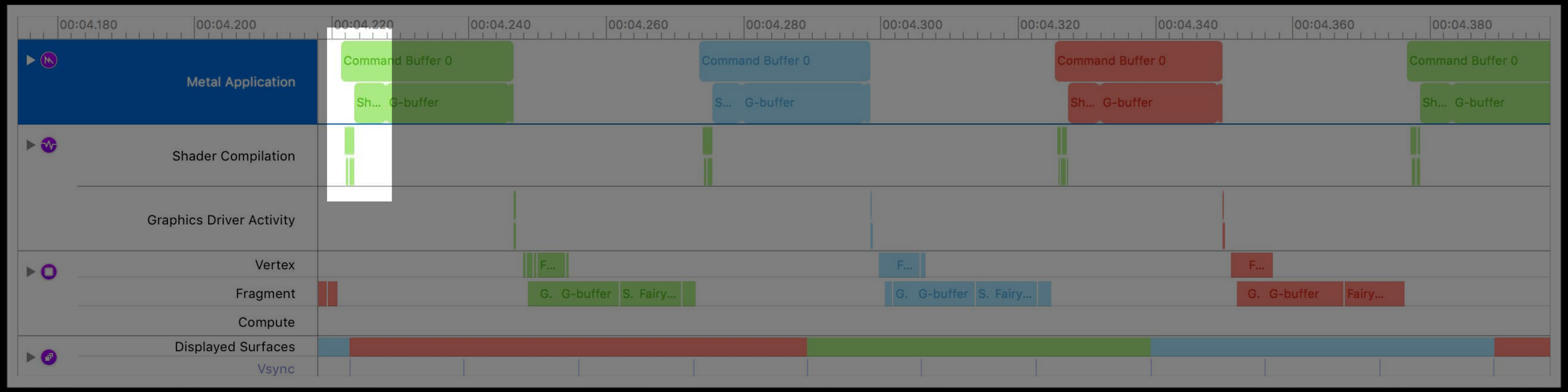

00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
00:04.300
Metal Application
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F.
Fragment
G. G-buffer
Fairy...
Compute
Displayed Surfaces
Vsync

Before

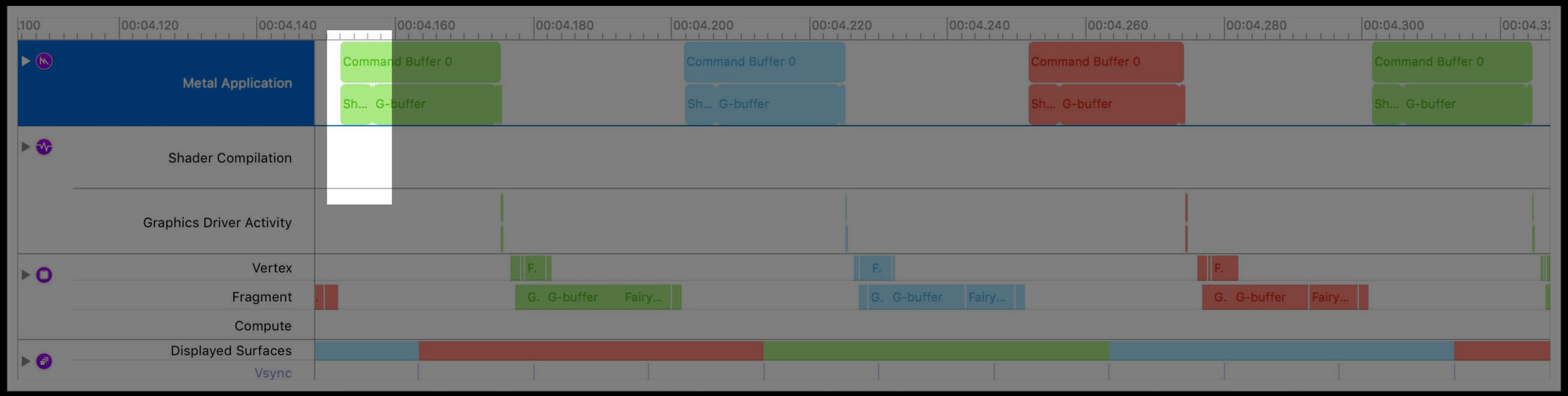

00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
Metal Application
Command Buffer 0
Sh... G-buffer
S... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
G. G-buffer
S. Fairy...
Fairy...


00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
00:04.300
Metal Application
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F.
G. G-buffer
Fairy...

Before



After



Metal Optimization Best Practices

Create state and objects upfront and reuse

- Compile shader source offline
- Keep the rendering loop free of object creation

2

Buffer Dynamic Shared Resources

Dynamic Resources

Dynamic Resources

GPU resources updated by CPU many times after creation

- Shader constants
- Vertex and index buffers for dynamic geometry
- Dynamic textures

Shared Storage Mode

MTLStorageModeShared

Shared Storage Mode

MTLStorageModeShared

Creates resource in memory shared by CPU and GPU

Shared Storage Mode

MTLStorageModeShared

Creates resource in memory shared by CPU and GPU

Default storage mode on iOS

Shared Storage Mode

MTLStorageModeShared

Creates resource in memory shared by CPU and GPU

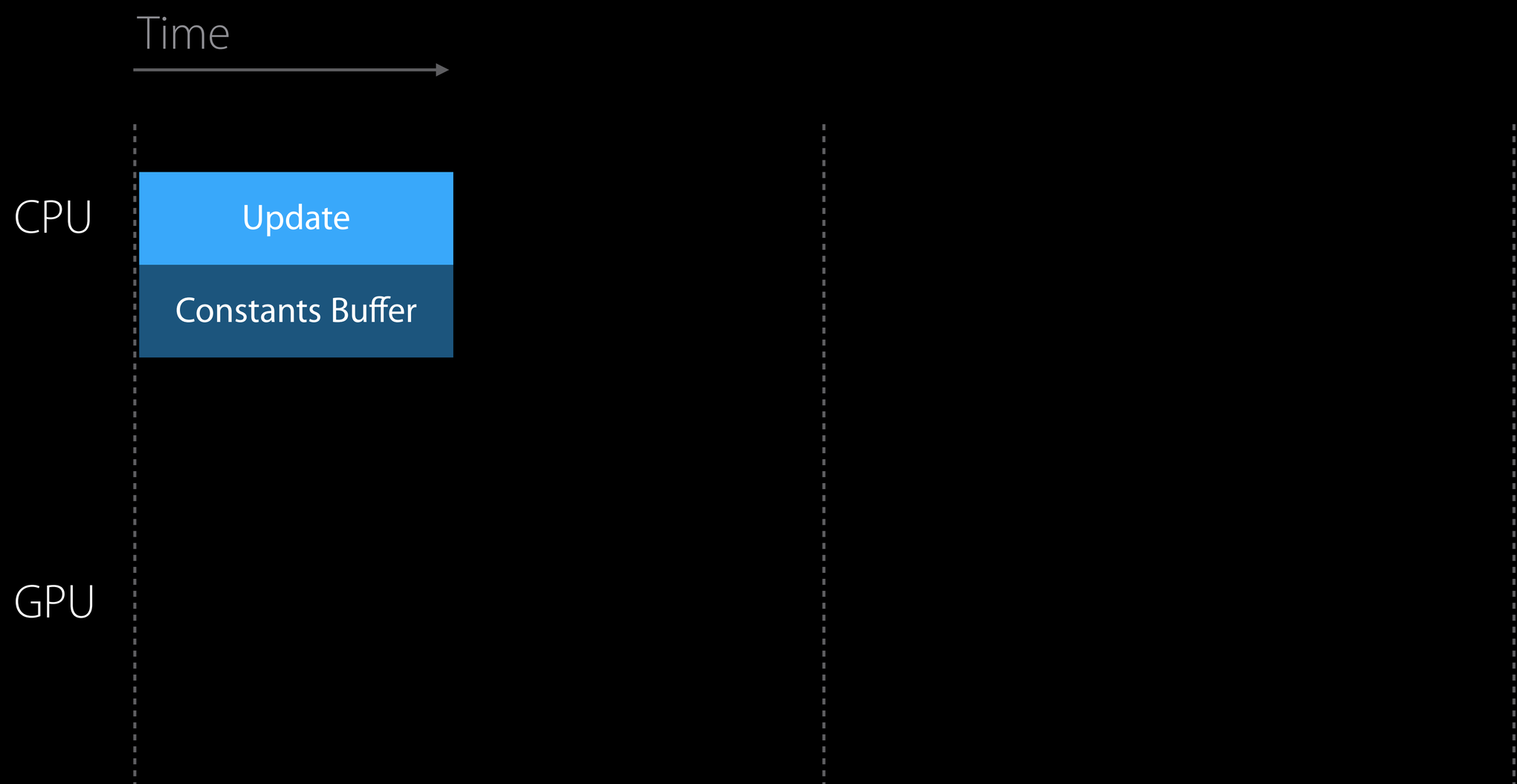
Default storage mode on iOS

Allows unsynchronized CPU access

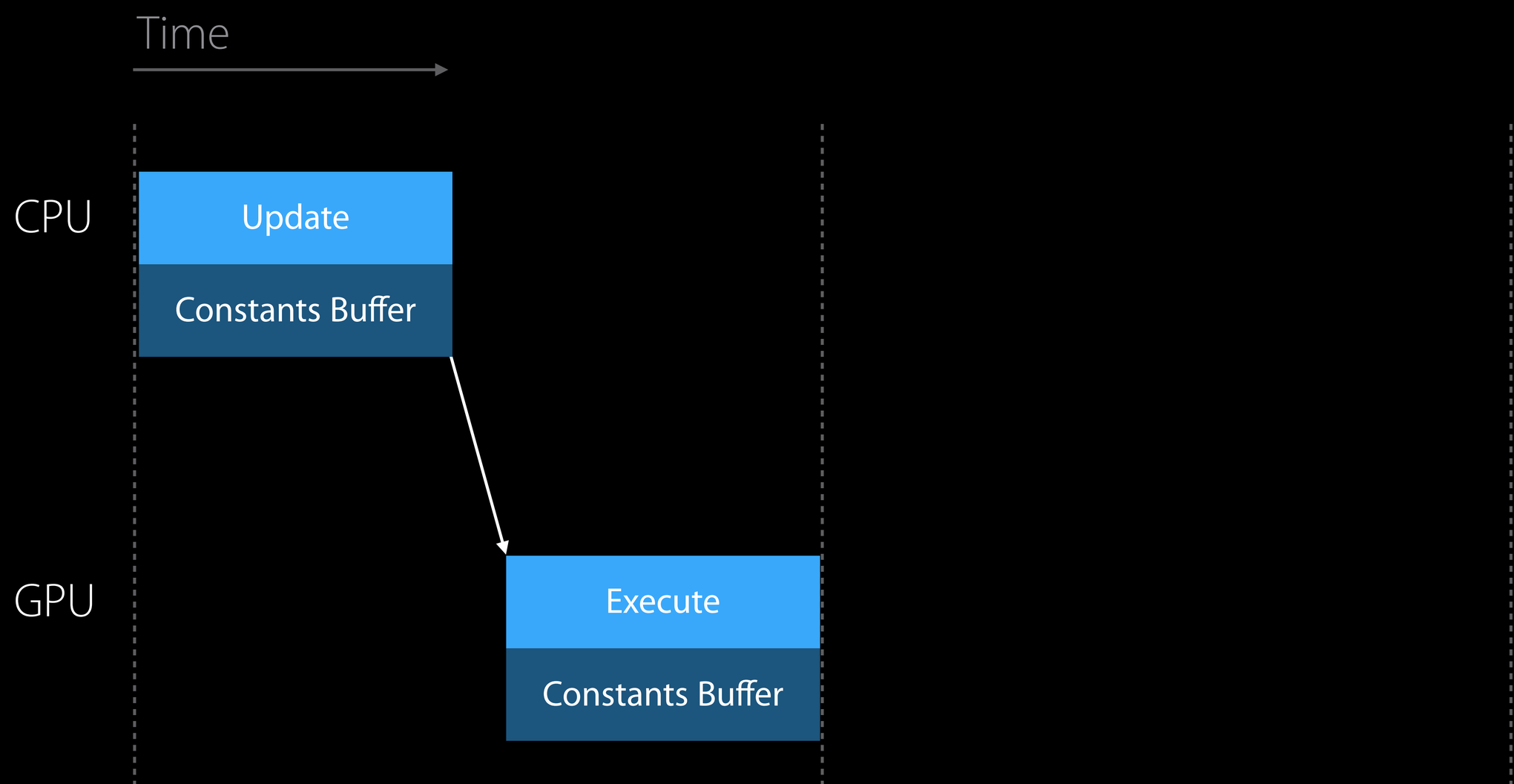
- Data modified through persistent pointer
- App responsible for not overwriting data in use by GPU

Single Buffer

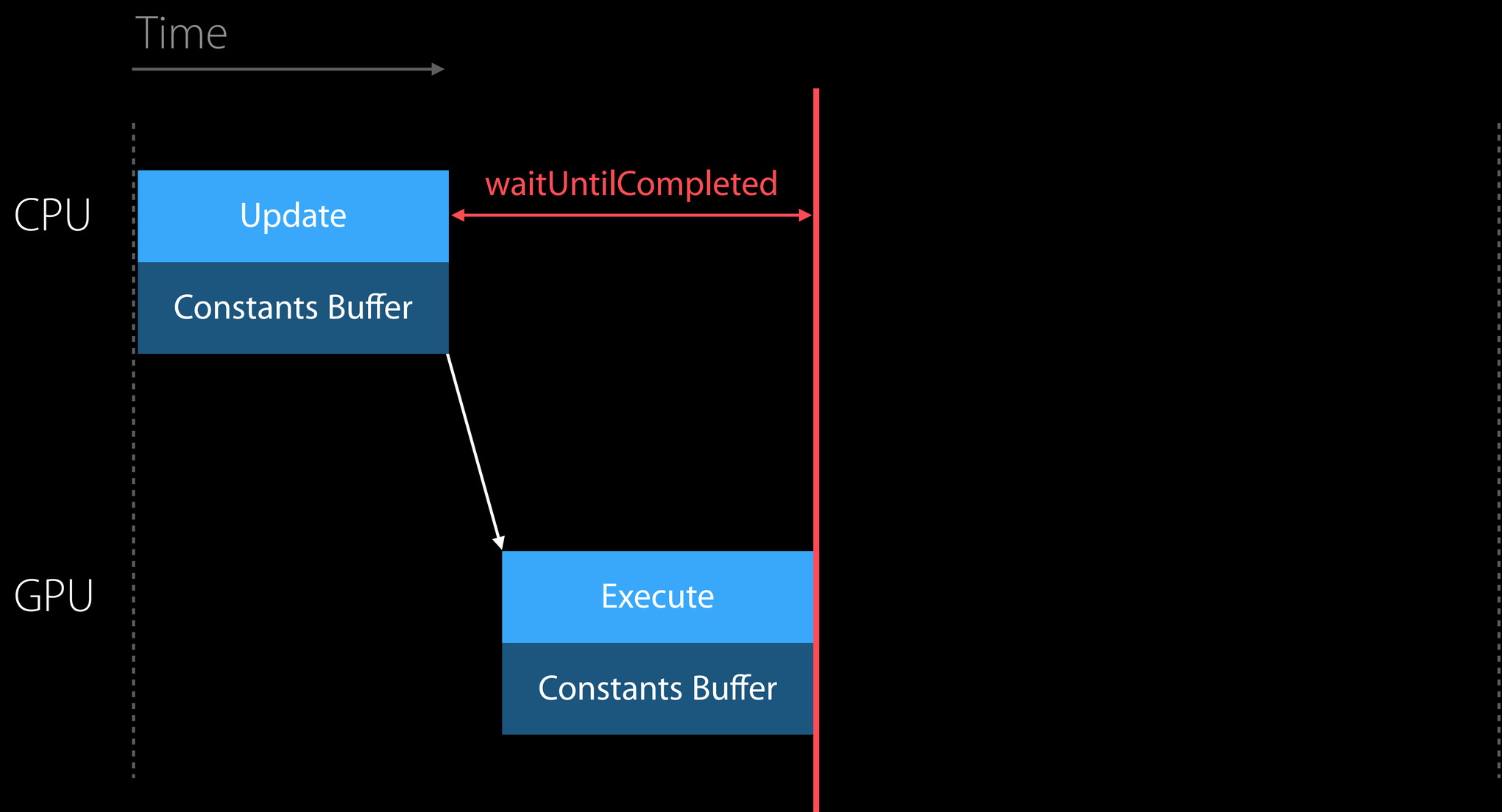
Single Buffer



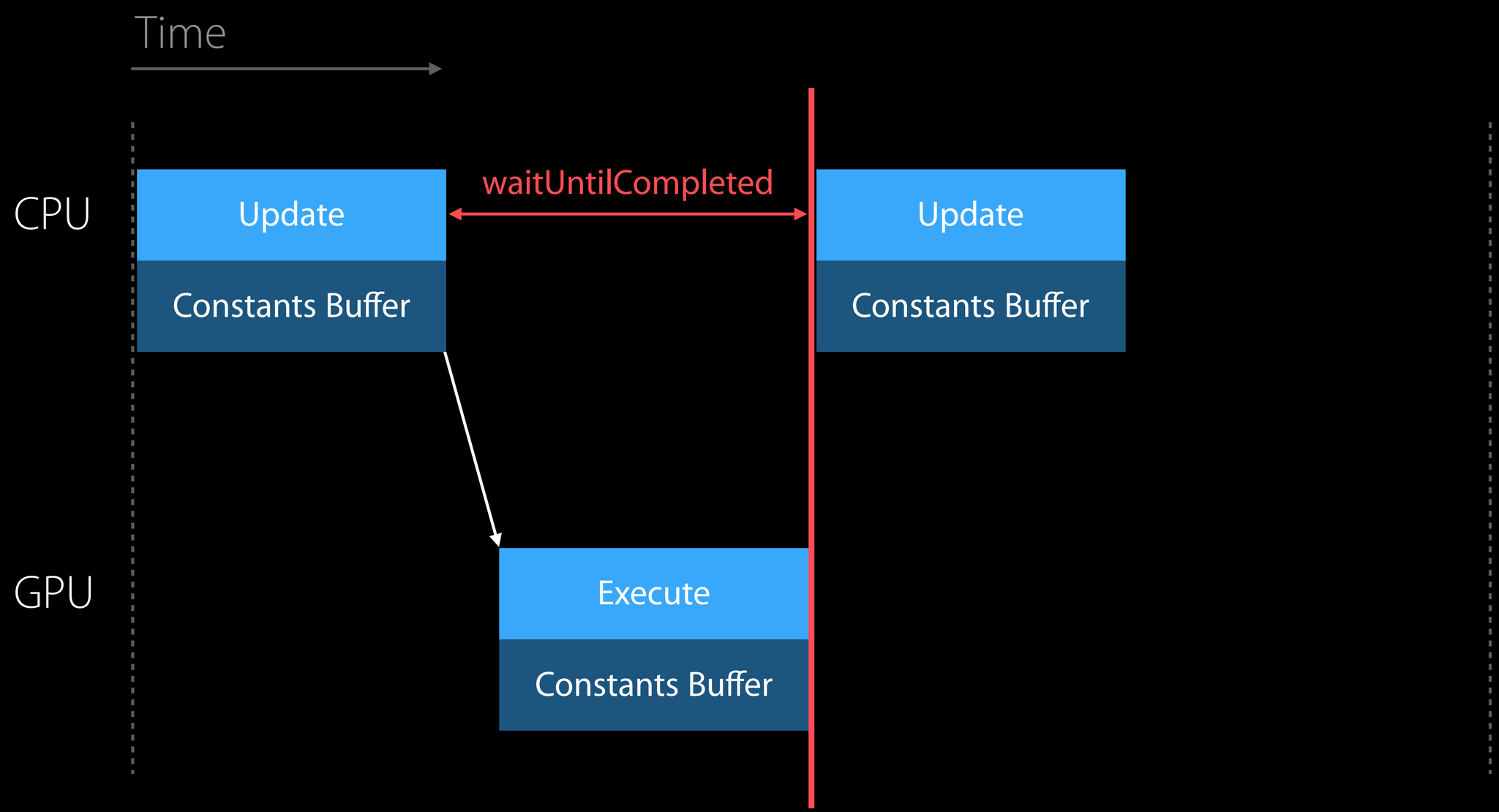
Single Buffer



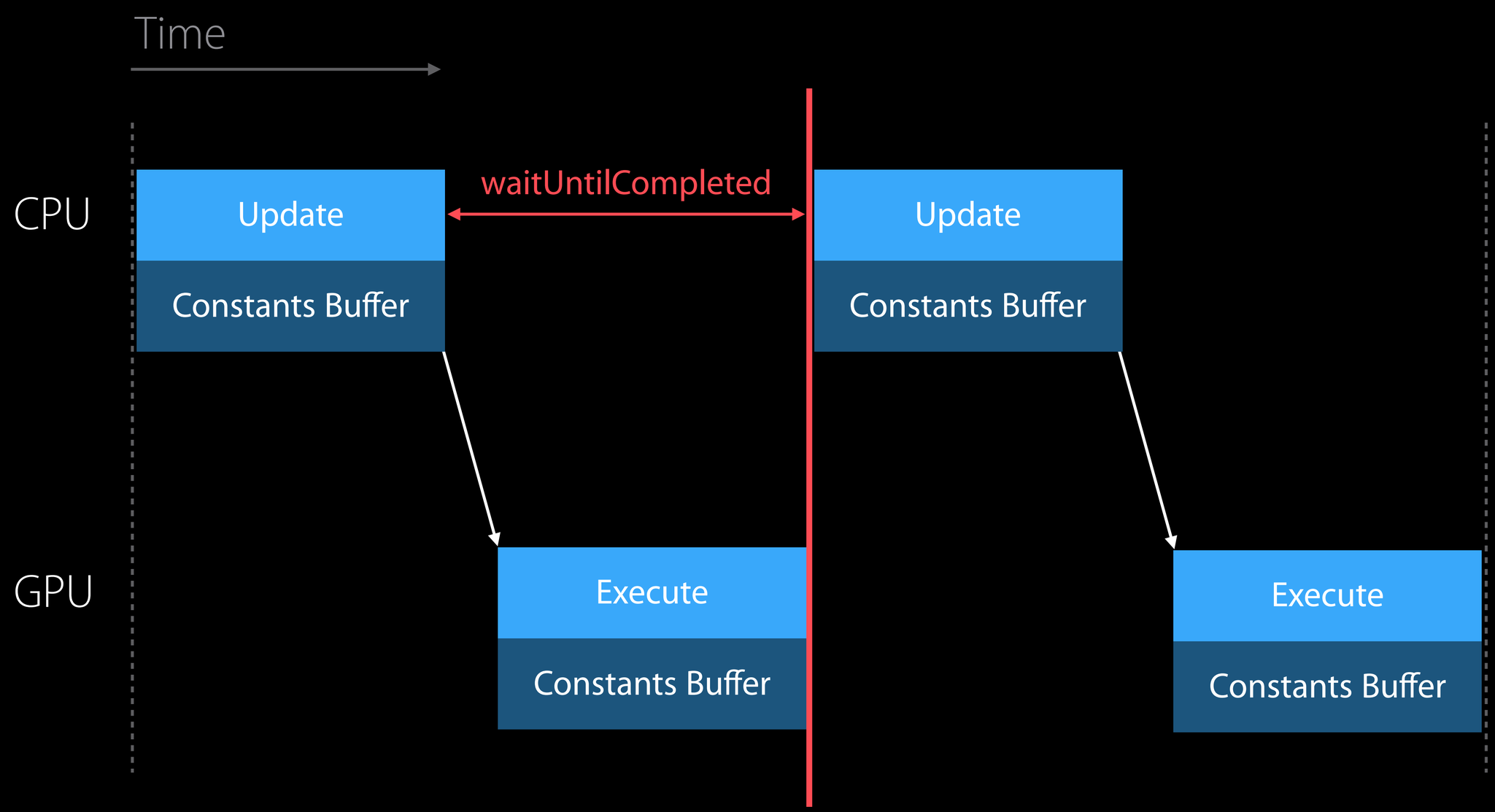
Single Buffer



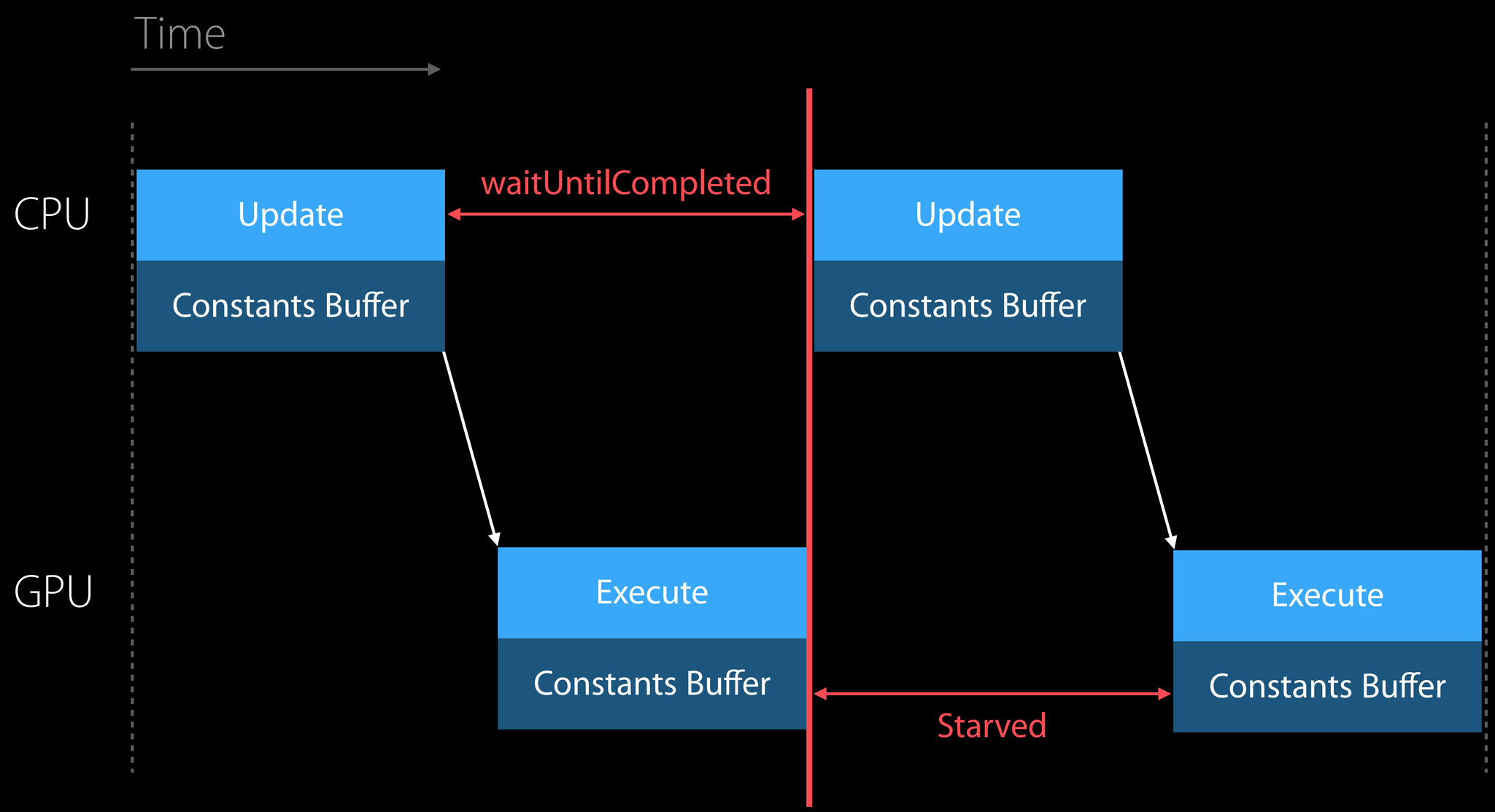
Single Buffer

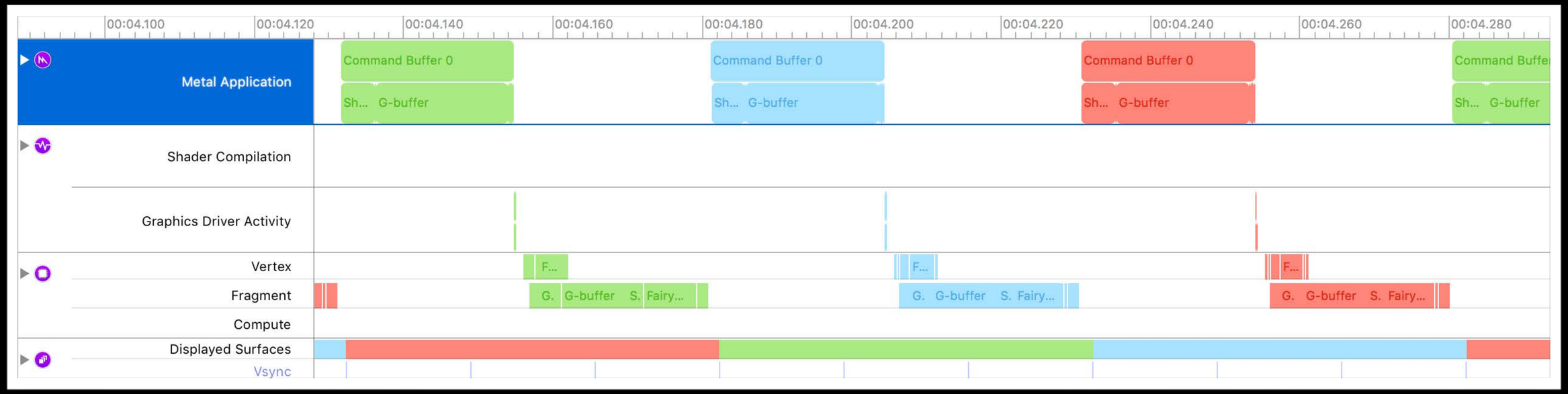


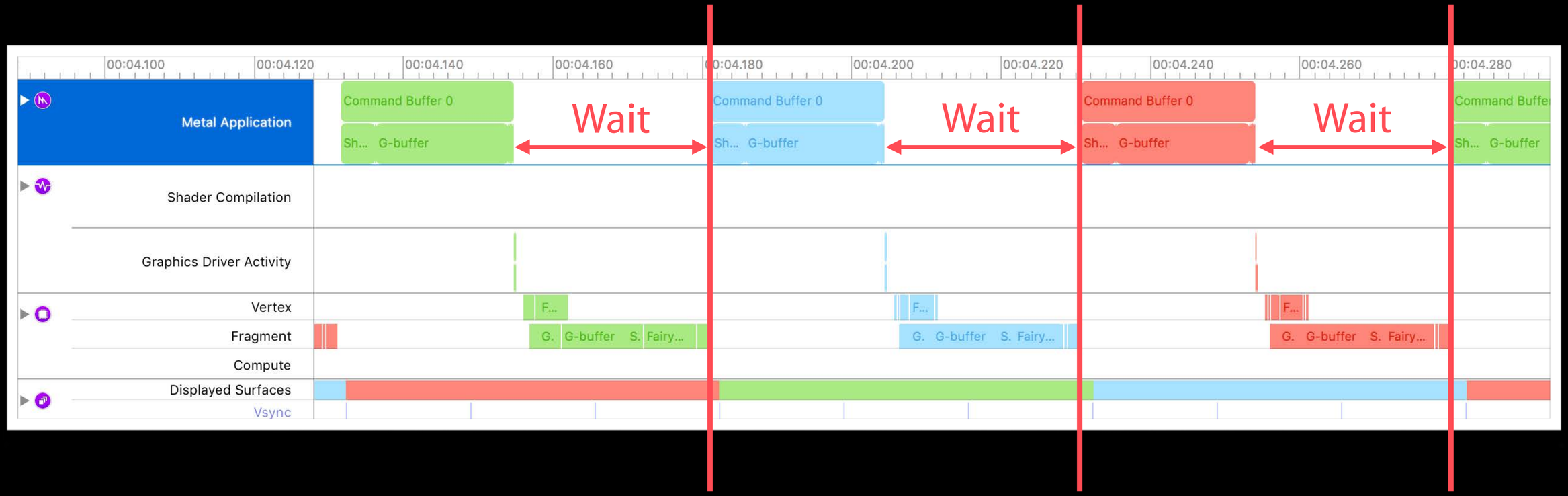
Single Buffer



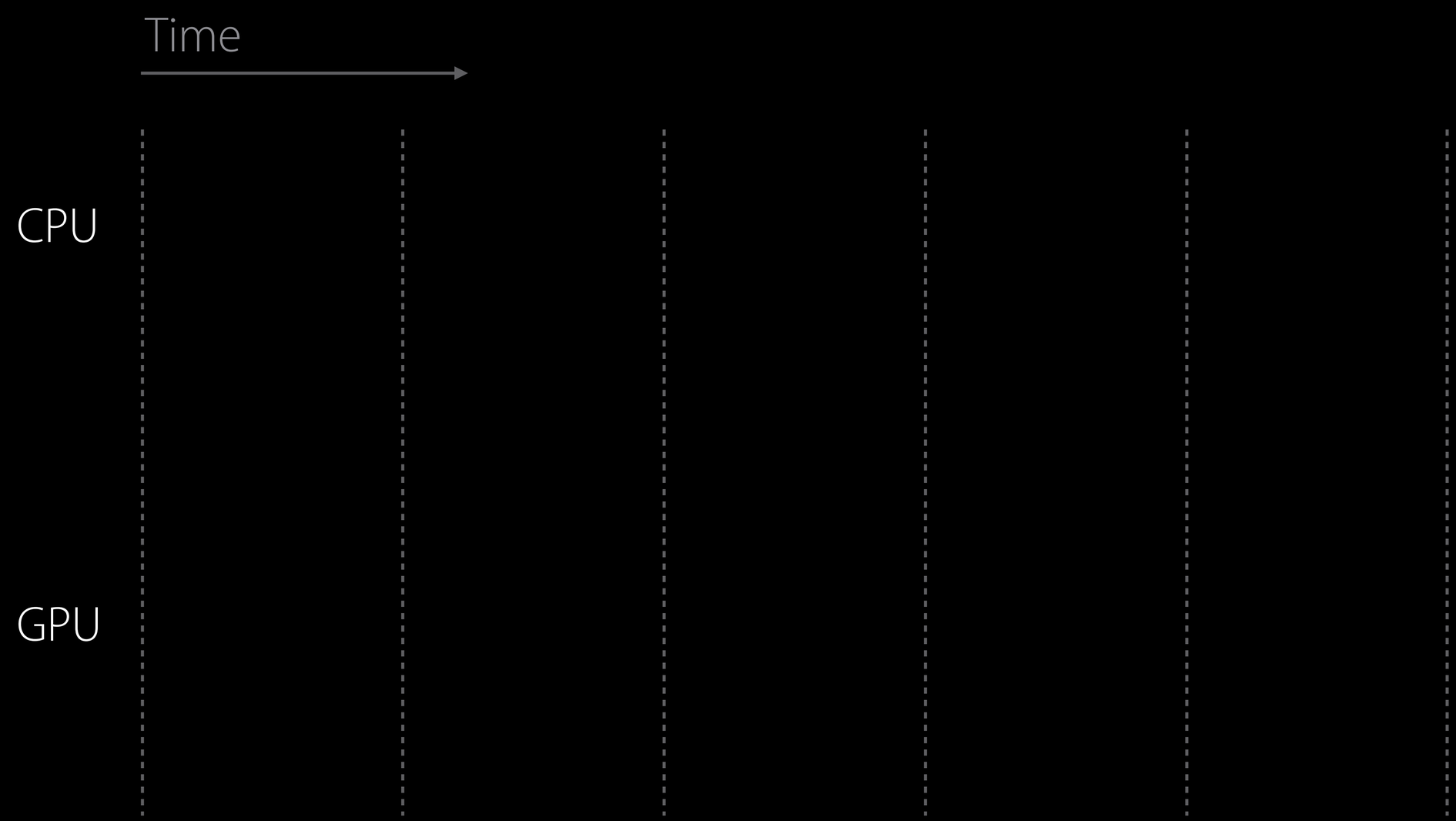
Single Buffer



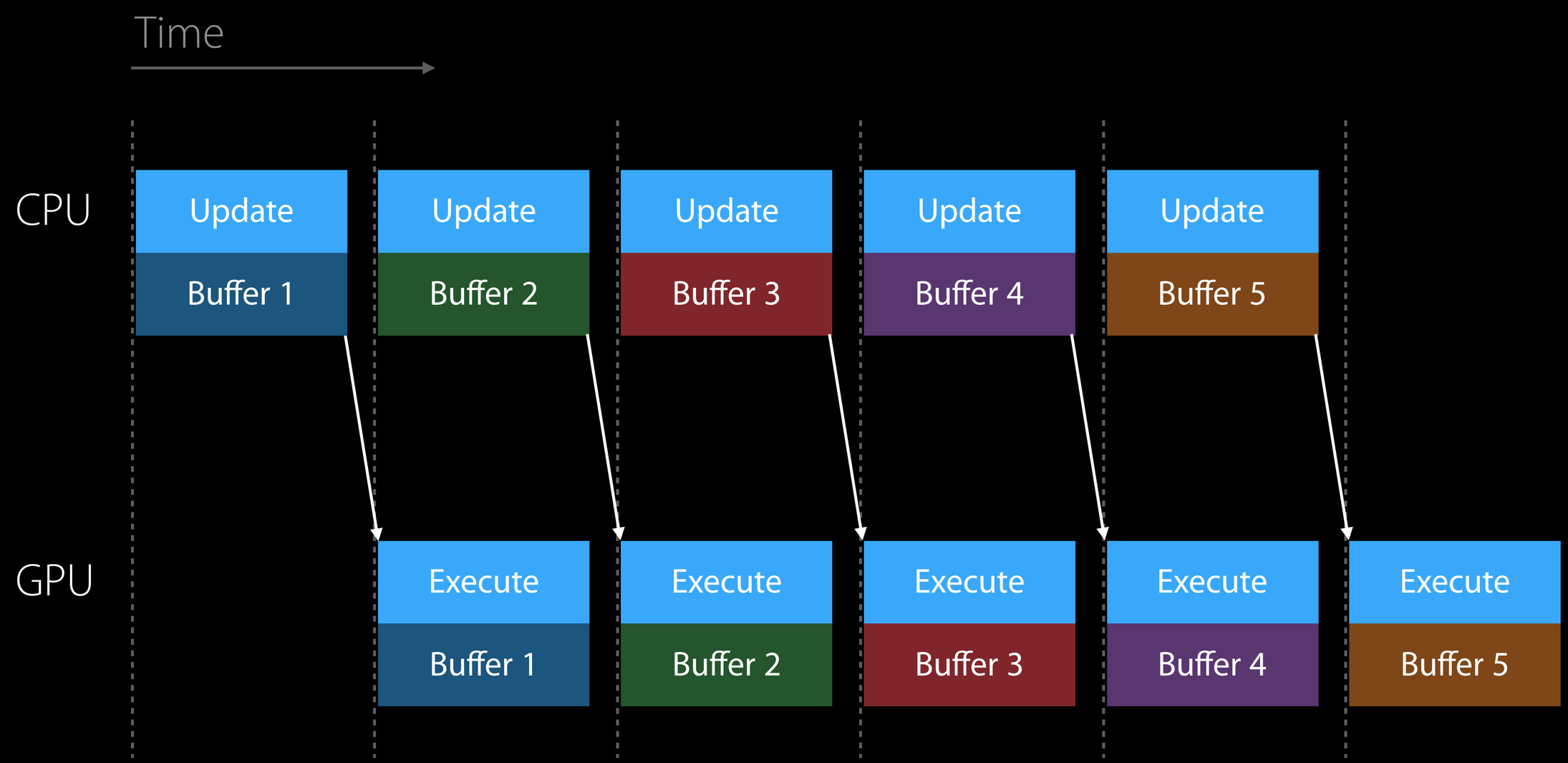

00:04.100
00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
Metal Application
Command Buffer 0
Sh...
G-buffer
Command Buffer 0
Sh...
G-buffer
Command Buffer 0
Sh...
G-buffer
Command Buffe
Sh...
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F...
F...
F...
Fragment
G.
G-buffer
S.
Fairy...
G.
G-buffer
S.
Fairy...
G.
G-buffer
S.
Fairy...
Compute
Displayed Surfaces
Vsync


00:04.100
00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
Metal Application
Command Buffer 0
Sh... G-buffer
Wait
Command Buffer 0
Sh... G-buffer
Wait
Command Buffer 0
Sh... G-buffer
Wait
Command Buffe
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
F...
Fragment
G. G-buffer S. Fairy...
Compute
Displayed Surfaces
Vsync

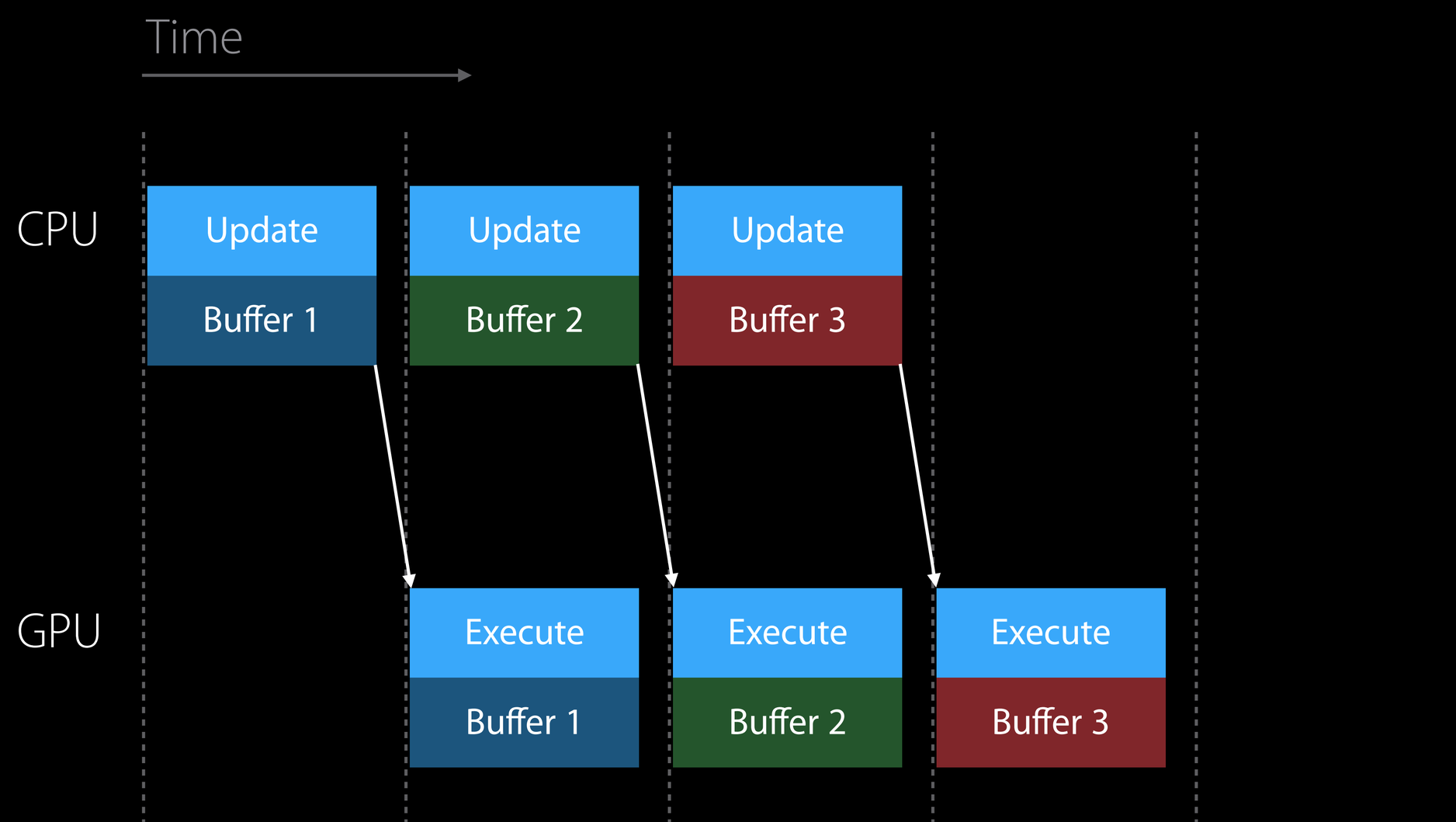
Creating New Buffers per Frame



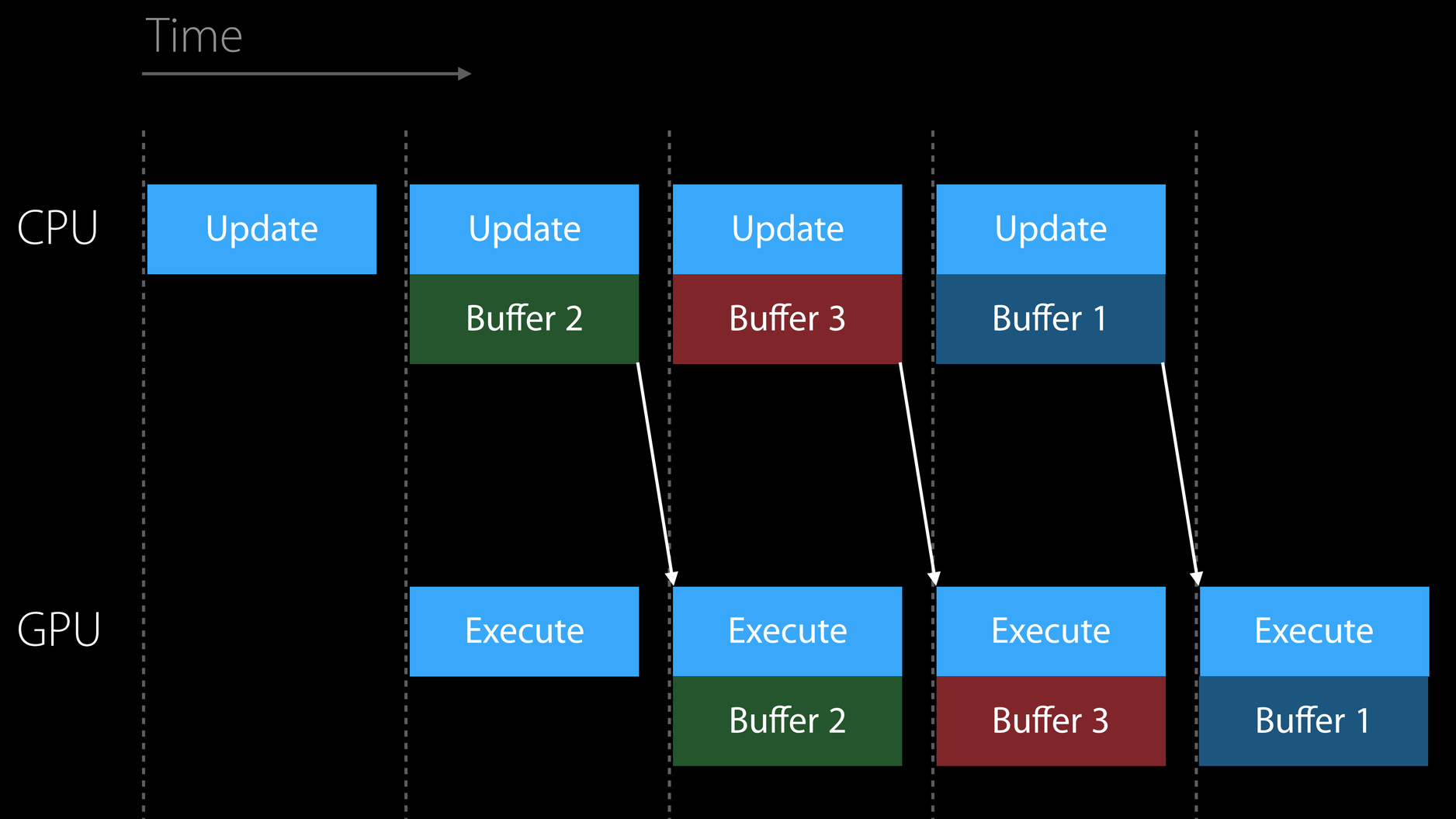
Creating New Buffers per Frame



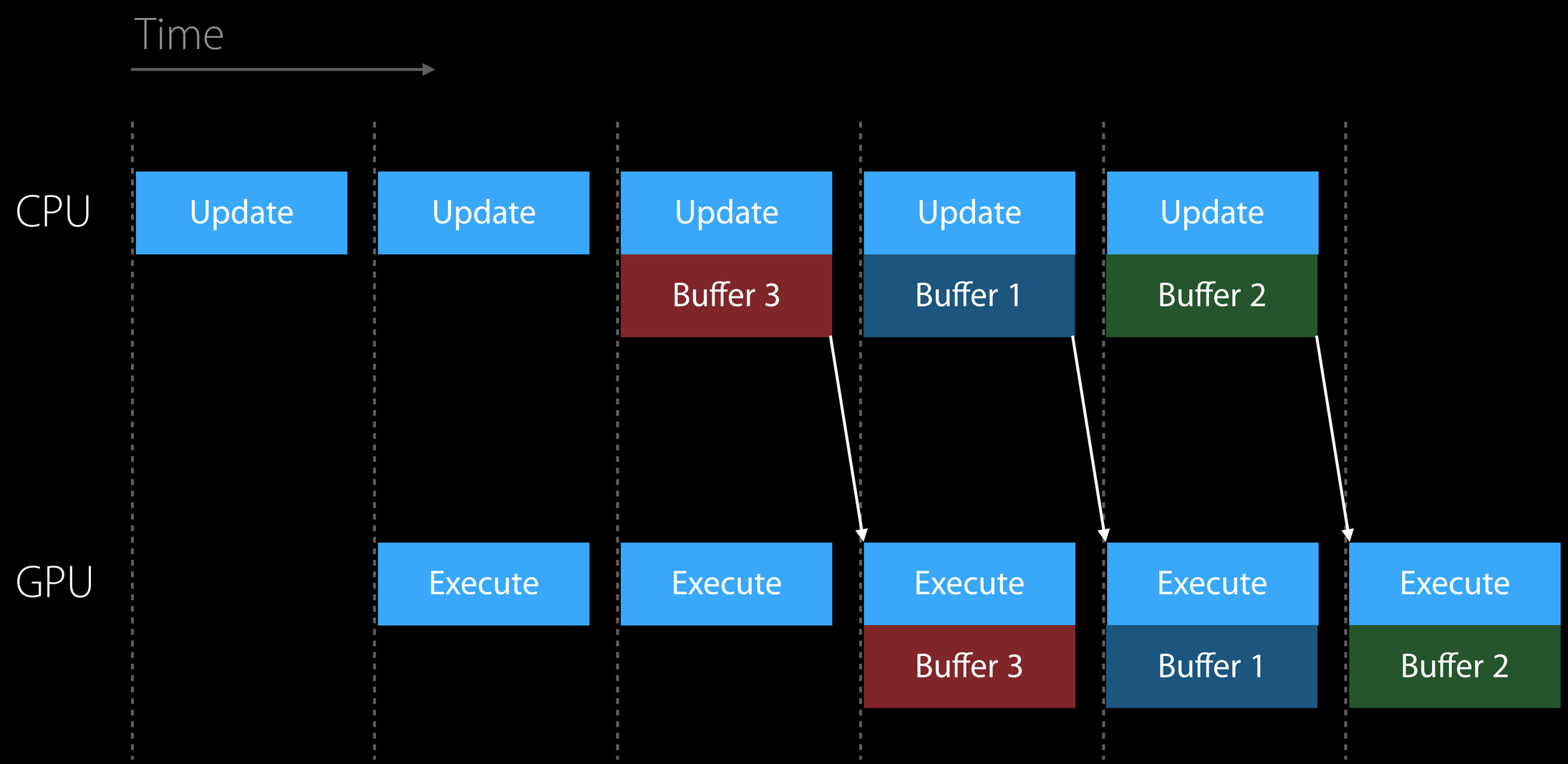
Triple Buffering Scheme



Triple Buffering Scheme



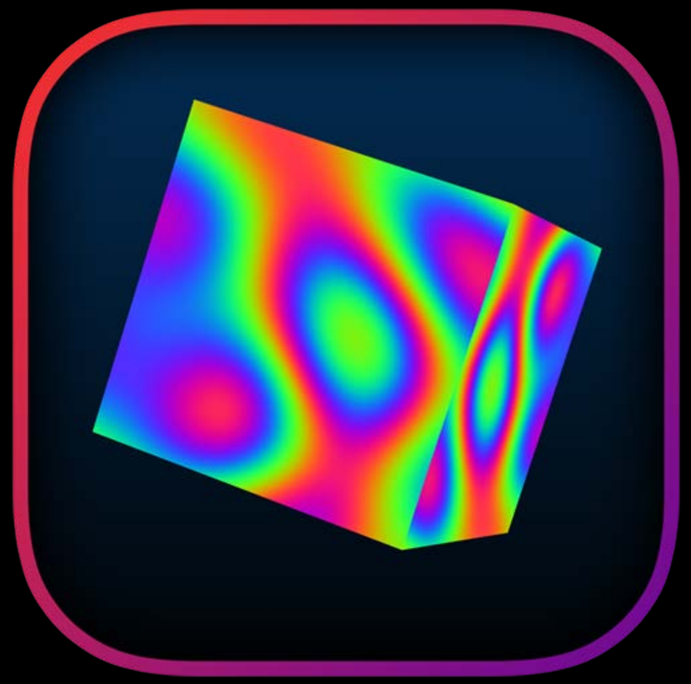
Triple Buffering Scheme

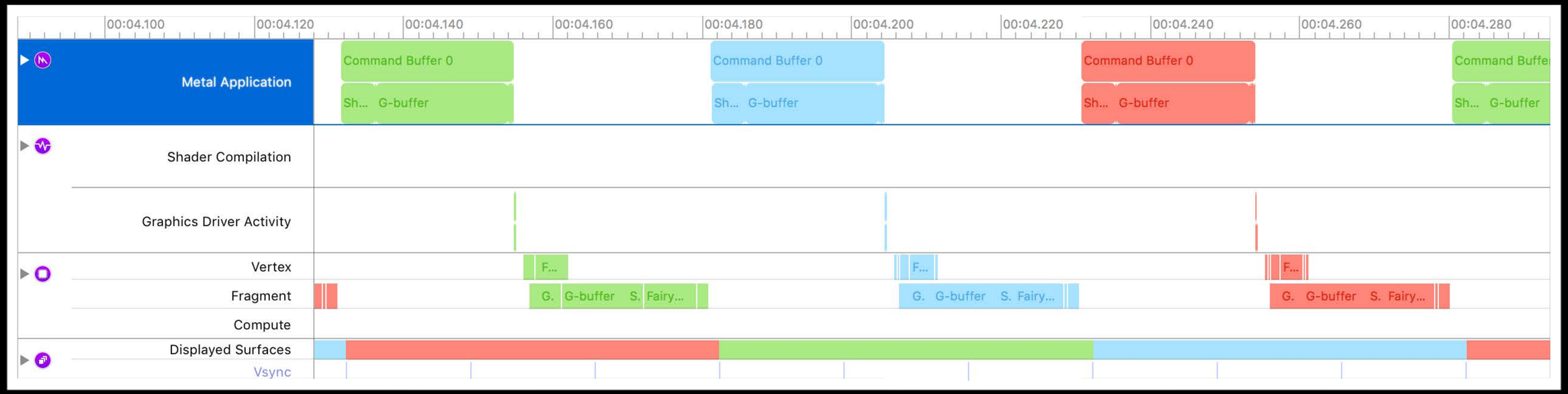


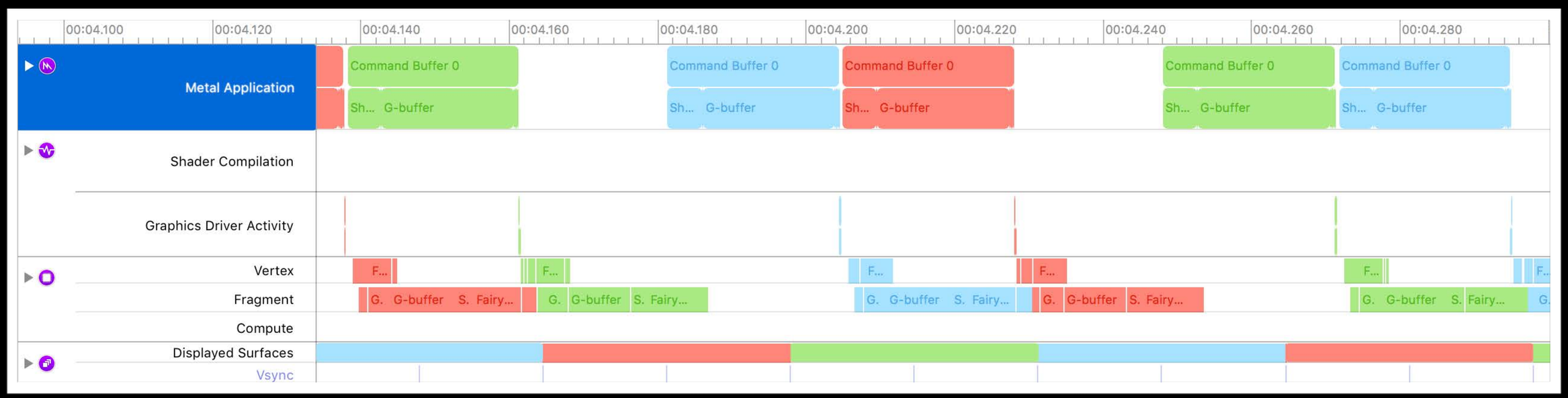
Sample Code

MetalUniformStreaming

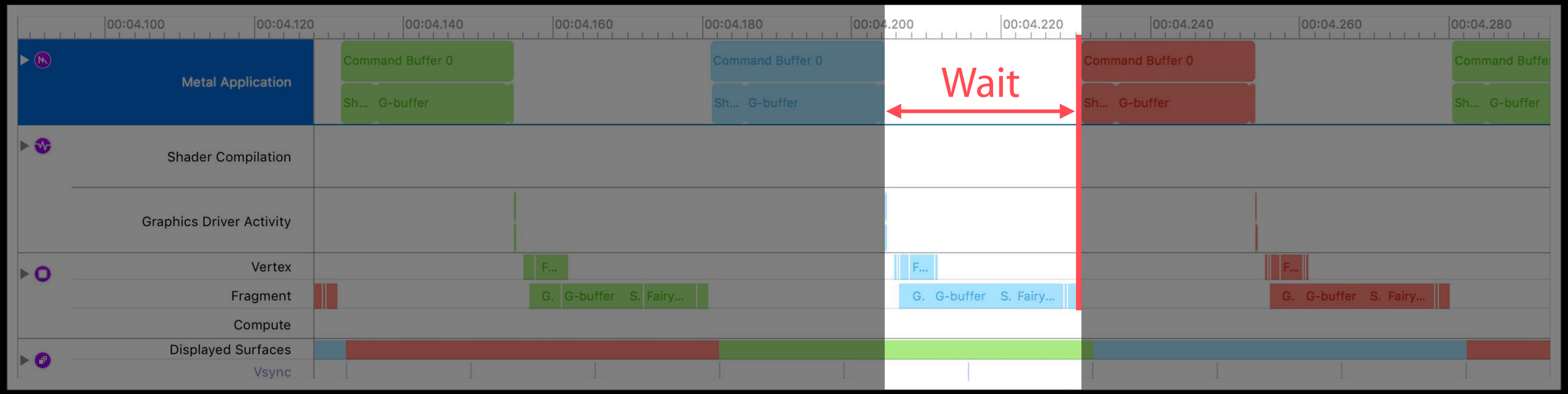
https://developer.apple.com/library/ios/samplecode/MetalUniformStreaming

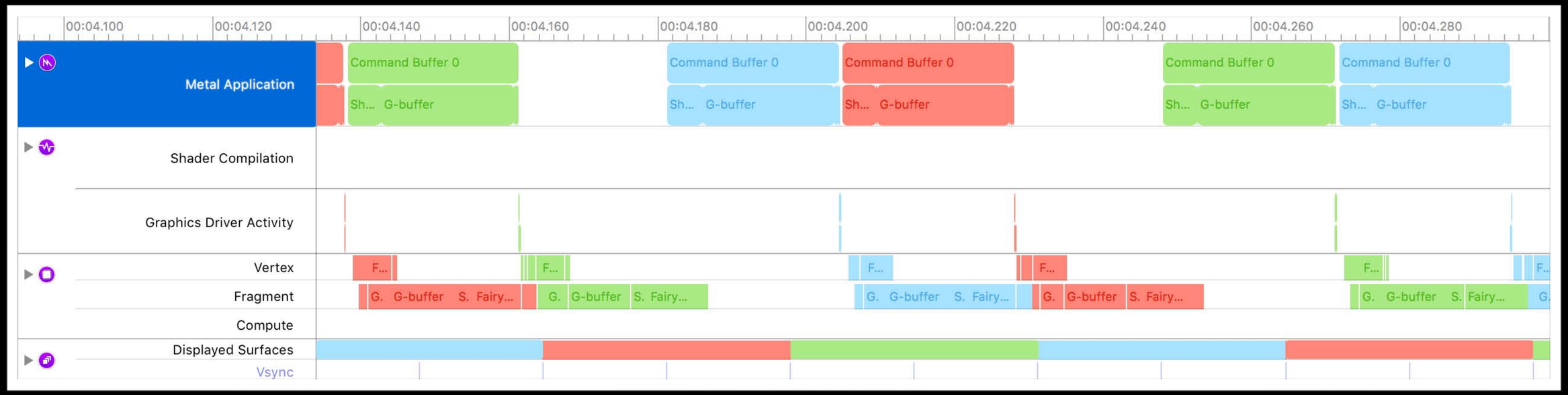



00:04.100
00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
Metal Application
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
G. G-buffer S. Fairy...

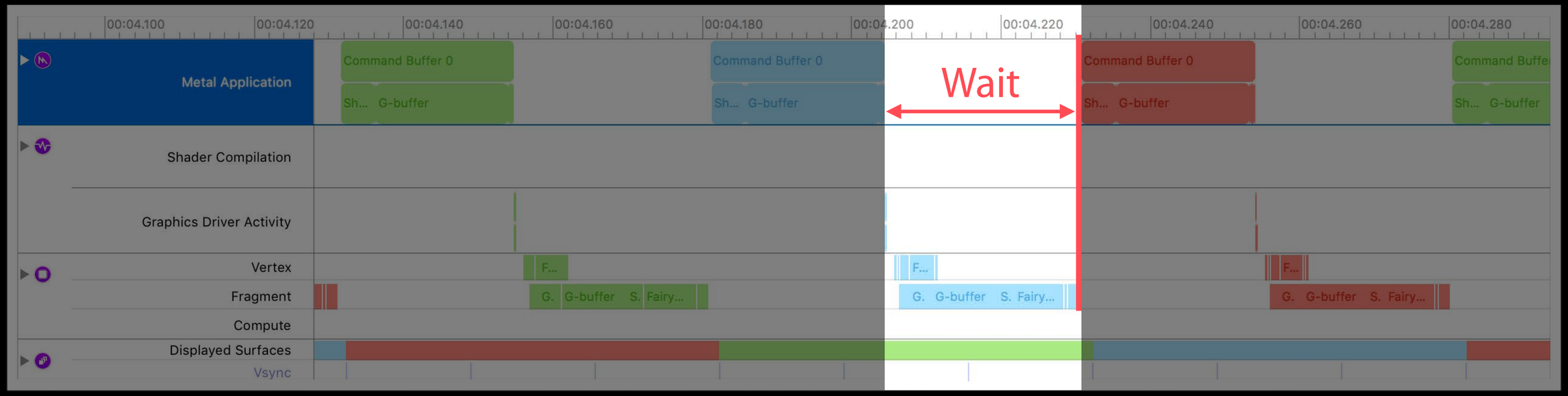

00:04.100
00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
Metal Application
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
G. G-buffer S. Fairy...

Before


00:04.100
00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
Metal Application
Command Buffer 0
Sh... G-buffer
Wait
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
G. G-buffer S. Fairy...


00:04.100
00:04.120
00:04.140
00:04.160
00:04.180
00:04.200
00:04.220
00:04.240
00:04.260
00:04.280
Metal Application
Command Buffer 0
Sh... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
G. G-buffer S. Fairy...

Before



After

Metal Optimization Best Practices

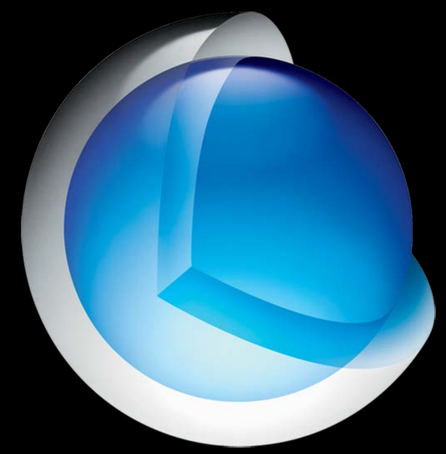
Buffer dynamic shared resources

- Most efficient way of updating dynamic resources between frames
- Enforce safety by using semaphore to stop CPU getting too far ahead of GPU

3

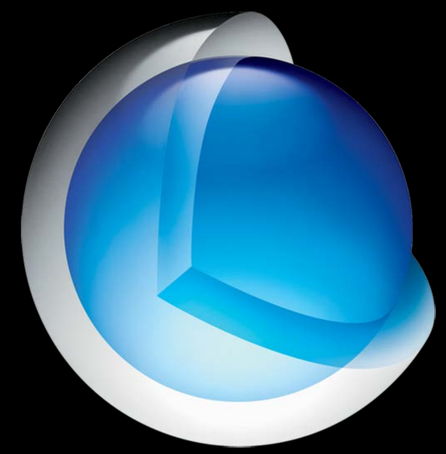
Acquire the Drawable at the Latest Opportunity

Drawable Surfaces



Drawable Surfaces

Surface into which app draws its visible output

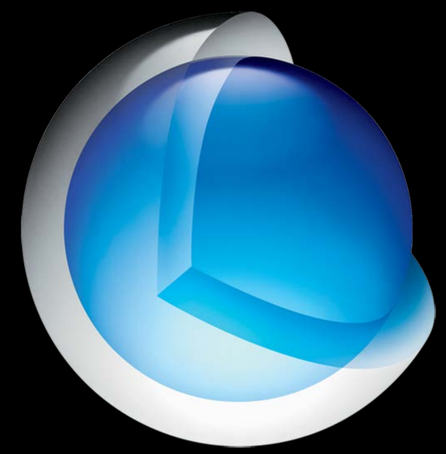


Drawable Surfaces

Surface into which app draws its visible output

CAMetalLayer has a limited pool of CAMetalDrawable objects

- Free drawables returned to pool at display intervals



Drawable Surfaces

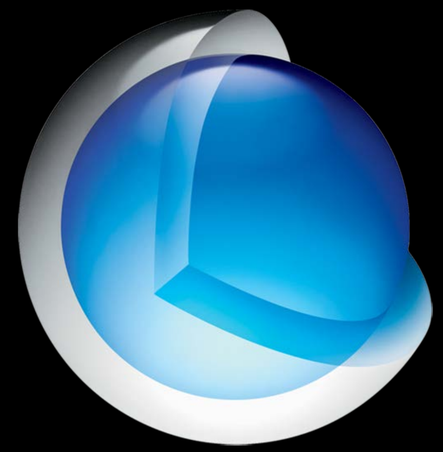
Surface into which app draws its visible output

CAMetalLayer has a limited pool of CAMetalDrawable objects

- Free drawables returned to pool at display intervals

Drawables may be used by multiple display stages at once:

- Your application
- GPU
- Core Animation (if compositing enabled)
- Display hardware



Acquiring a Drawable Surface

Acquiring a Drawable Surface

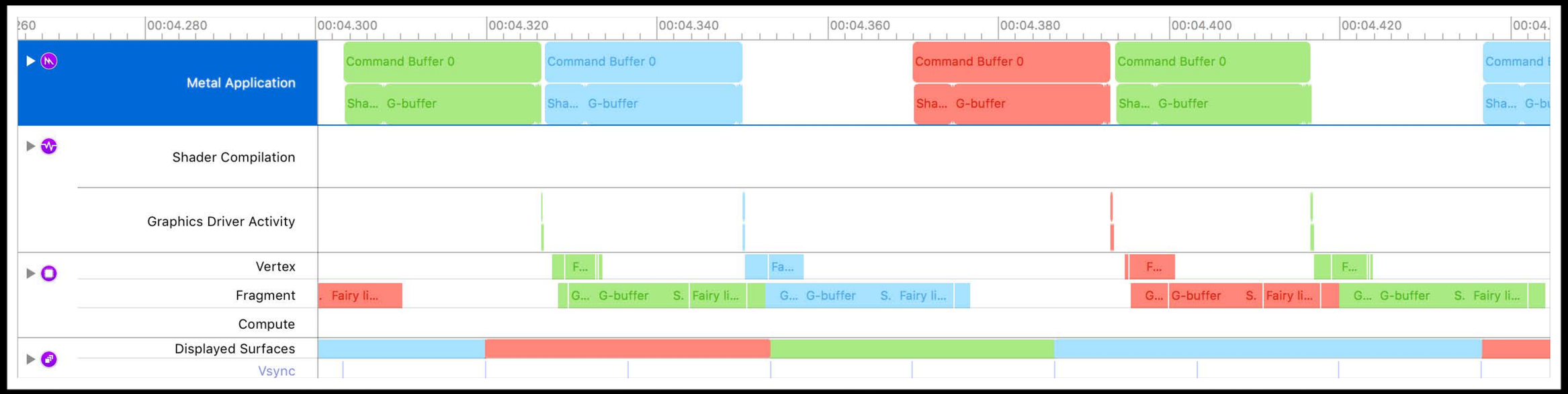
Drawable acquired with `nextDrawable` method

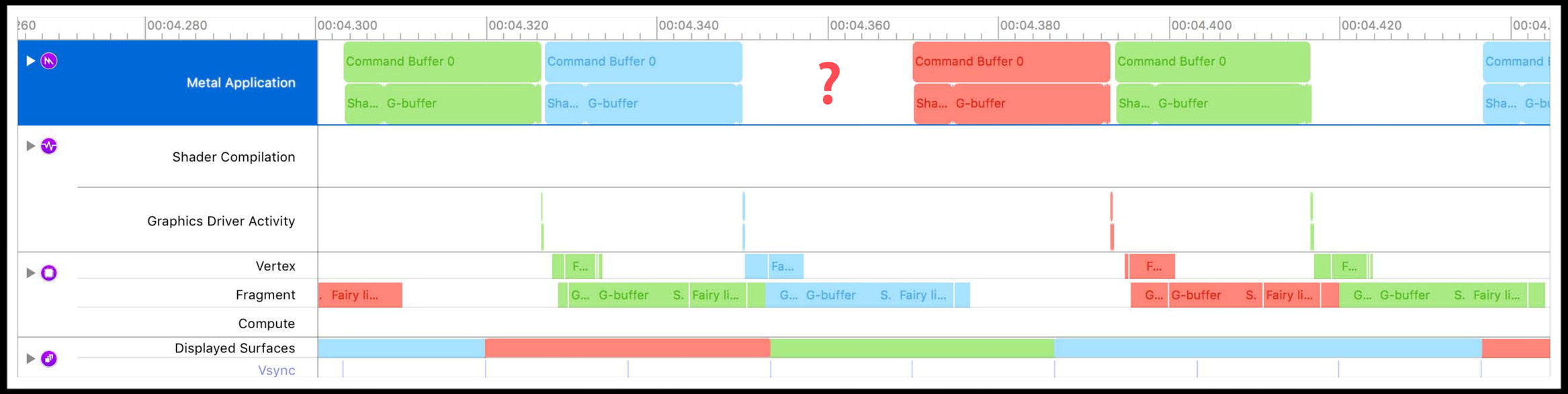
Acquiring a Drawable Surface

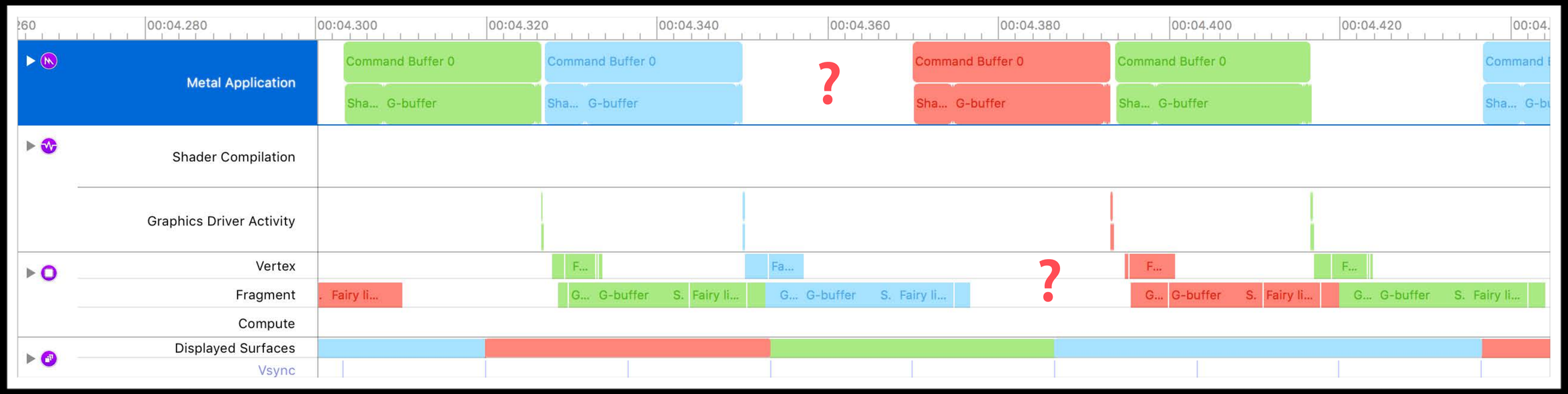
Drawable acquired with `nextDrawable` method

Method returns once a drawable is available

- Calling thread will block until at least next display interval if none available


00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
00:04.400
00:04.420
Metal Application
Command Buffer 0
Sha... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Fairy li...
G... G-buffer S.
Displayed Surfaces
Vsync


00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
00:04.400
00:04.420
Metal Application
Command Buffer 0
Command Buffer 0
Command Buffer 0
Command Buffer 0
Sha... G-buffer
Sha... G-buffer
Sha... G-buffer
Sha... G-buffer
?
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Fairy li...
G... G-buffer S.
Displayed Surfaces
Vsync


00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
00:04.400
00:04.420
Metal Application
Command Buffer 0
Command Buffer 0
Command Buffer 0
Command Buffer 0
Sha... G-buffer
Sha... G-buffer
Sha... G-buffer
Sha... G-buffer
?
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Fa...
Fairy li...
G... G-buffer S.
Displayed Surfaces
Vsync

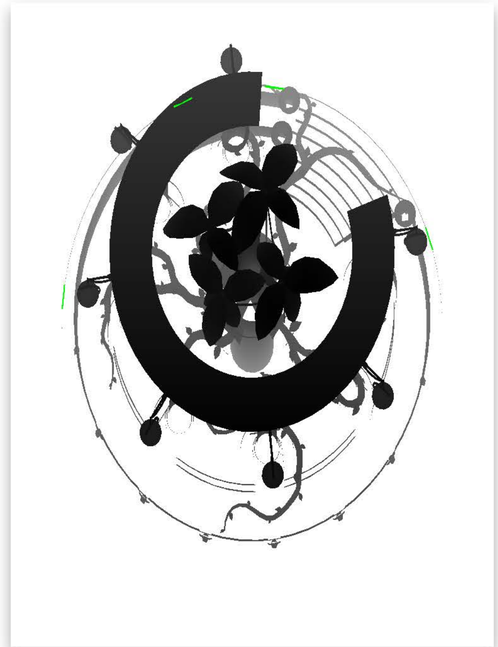
MetalApp 〉 Philip's iPad (9.0)

Running MetalApp on Philip's iPad

MetalApp

FPS	55 FPS
0 0x1003a6680 <- [MTLLayer nextDrawable]	
1 0x10104b7e0 <- [MTLDrawable texture]	
2 0x10039d0c0 <- [0x10103b540 commandBuffer]	
CommandBuffer 0x10039d0c0	
Shadow Buffer	10.24 µs
G-buffer	0.00 ns
G-buffer	11.97 ms
Sun light	3.22 ms
Fairy lights	951.21 µs
Fairy sprites	79.52 µs
UI	1.06 ms
1009 [addCompletedHandler:0x16fdfd090]	
1010 [presentDrawable:0x1003a6680]	
1011 [commit]	

MetalApp.gputrace 〉 CommandB...10039d0c0 〉 Shadow Buffer 〉 14 [endEncoding]



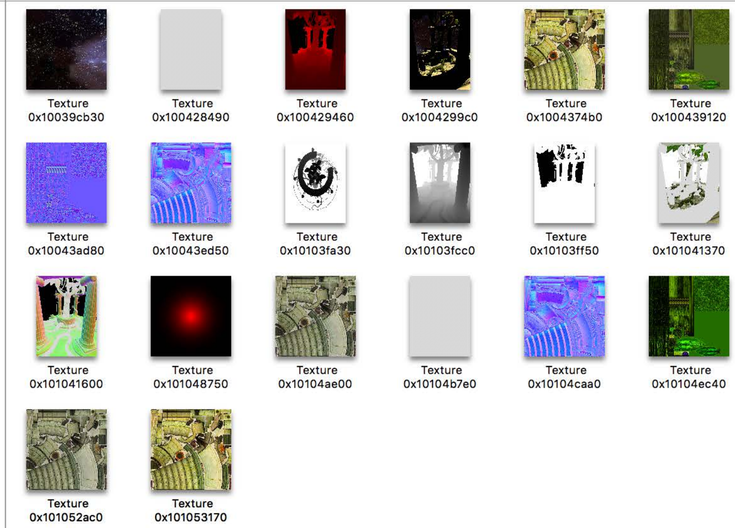
Depth Attachment

MetalApp 〉 CommandBuffer 0x10039d0c0 〉 Shadow Buffer 〉 14 [endEncoding]

- "Shadow Buffer" (0x10038f590)
- Encoder Statistics 3 draw calls, 12MB load/store bandwidth
- Frame Statistics 271 draw calls, 246MB load/store bandwidth

Auto

All GPU Objects 〉 Textures 〉 No Selection



"Shadow Buffer" (0x10038f590)

MTL Command Encoder

MetalApp 〉 Philip's iPad (9.0)

MetalApp

FPS	55 FPS
0 0x1003a6680 <- [MTLLayer nextDrawable]	
1 0x10104b7e0 <- [MTLDrawable texture]	
2 0x10039d0c0 <- [0x10103b540 commandBuffer]	
CommandBuffer 0x10039d0c0	
Shadow Buffer	10.24 µs ⚠
G-buffer	0.00 ns
G-buffer	11.97 ms
Sun light	3.22 ms
Fairy lights	951.21 µs
Fairy sprites	79.52 µs
UI	1.06 ms
1009 [addCompletedHandler:0x16fdfd090]	
1010 [presentDrawable:0x1003a6680]	
1011 [commit]	

MetalApp 〉 Philip's iPad (9.0)

MetalApp

FPS	55 FPS
0 0x1003a6680 <- [MTLLayer nextDrawable]	
1 0x10104b7e0 <- [MTLDrawable texture]	
2 0x10039d0c0 <- [0x10103b540 commandBuffer]	
CommandBuffer 0x10039d0c0	
Shadow Buffer	10.24 µs ⚠
G-buffer	0.00 ns
G-buffer	11.97 ms
Sun light	3.22 ms
Fairy lights	951.21 µs
Fairy sprites	79.52 µs
UI	1.06 ms
1009 [addCompletedHandler:0x16fdfd090]	
1010 [presentDrawable:0x1003a6680]	
1011 [commit]	

MetalApp 〉 Philip's iPad (9.0)

By File | By Type

- MetalApp 1 issue
  - CAMetalLayer nextDrawable Called Early
    - Your application called CAMetalLayer nextDrawable earlier than needed. This may block execution until a CAMetalDrawable is available. Delay the call to nextDrawable until the drawable is needed for encoding.
      : function index of NextDrawable = 20, function index of violating function = 21
      AAPLRenderer.mm

MetalApp 〉 Philip's iPad (9.0)

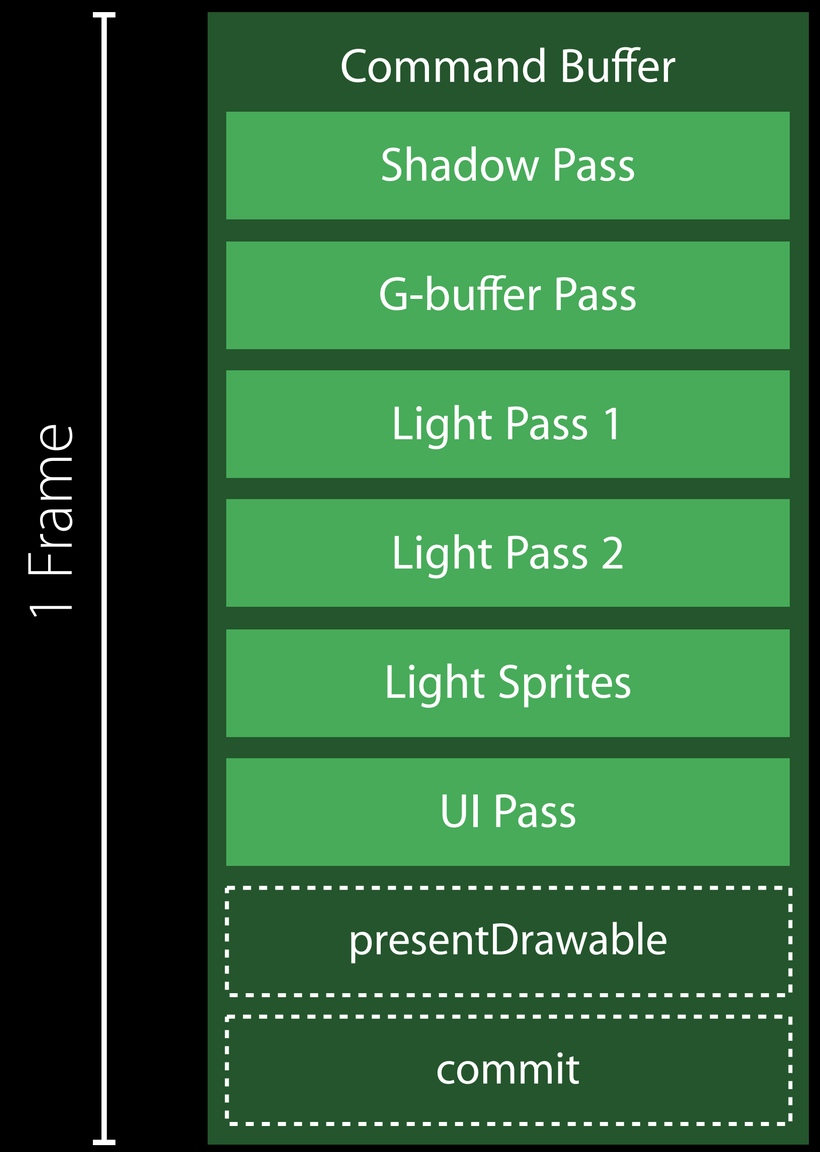
By File | By Type

MetalApp 1 issue

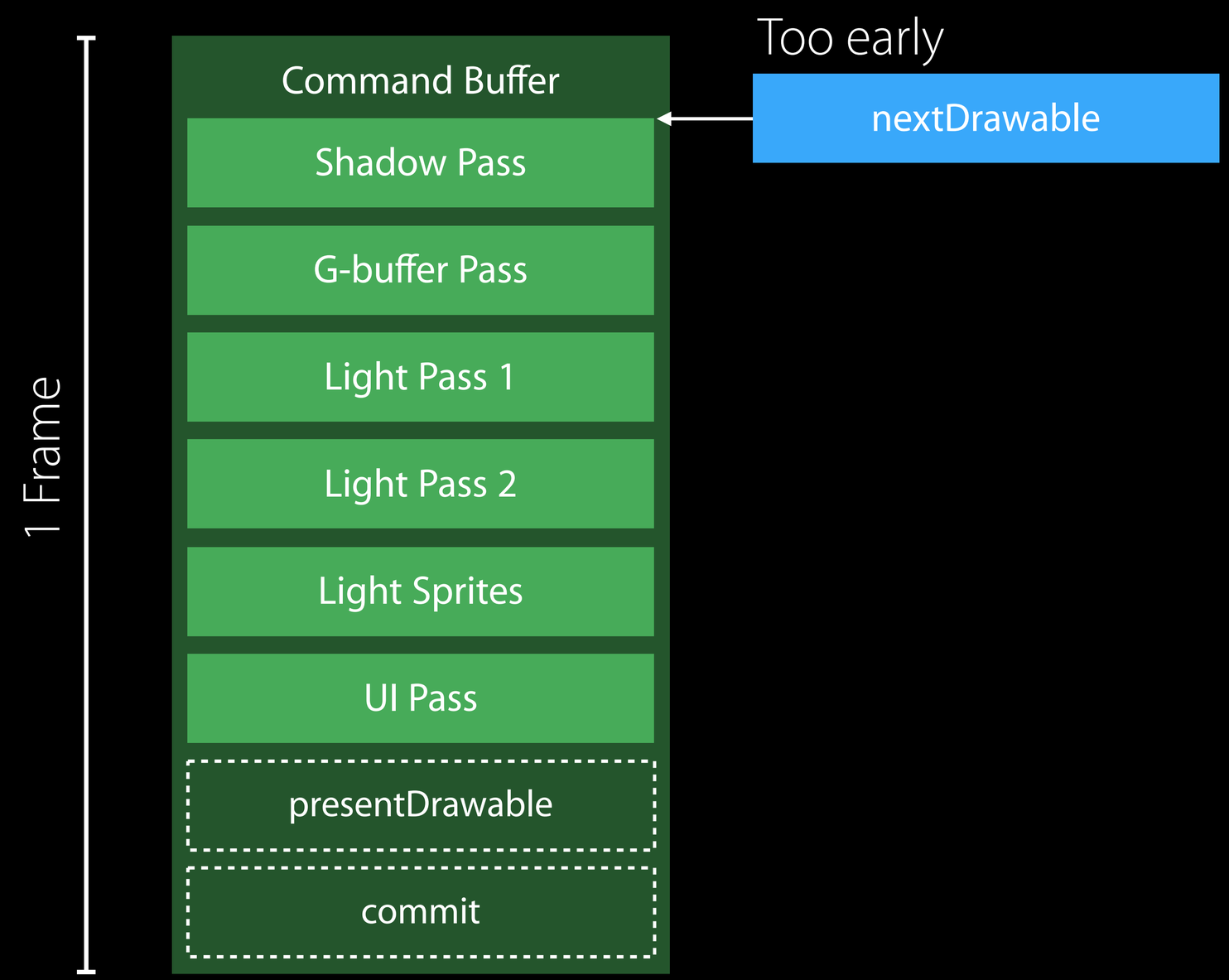
CAMetalLayer nextDrawable Called Early

Your application called CAMetalLayer nextDrawable earlier than needed. This may block execution until a CAMetalDrawable is available. Delay the call to nextDrawable until the drawable is needed for encoding.
: function index of NextDrawable = 20, function index of violating function = 21
AAPLRenderer.mm

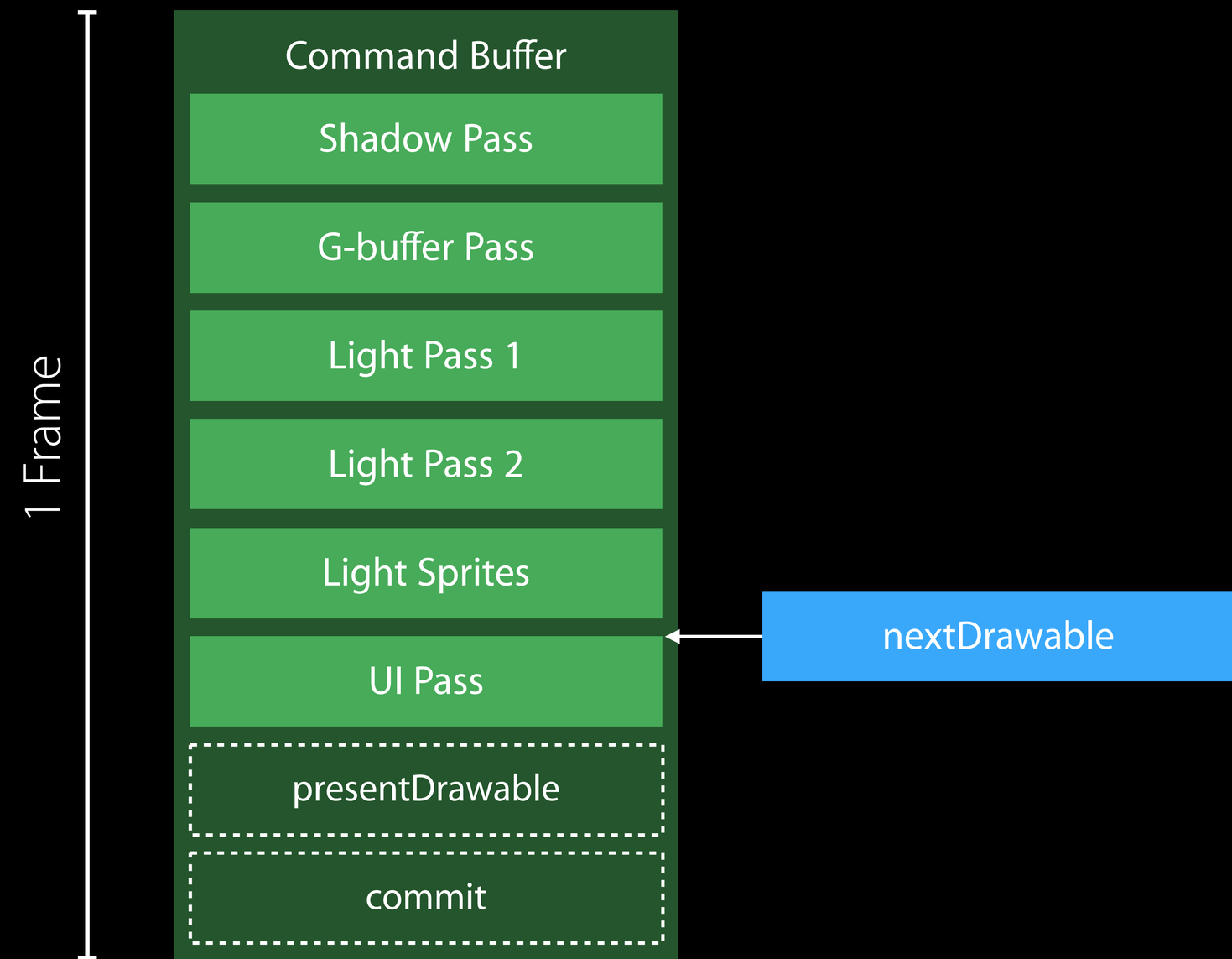
When to Acquire Your Drawable

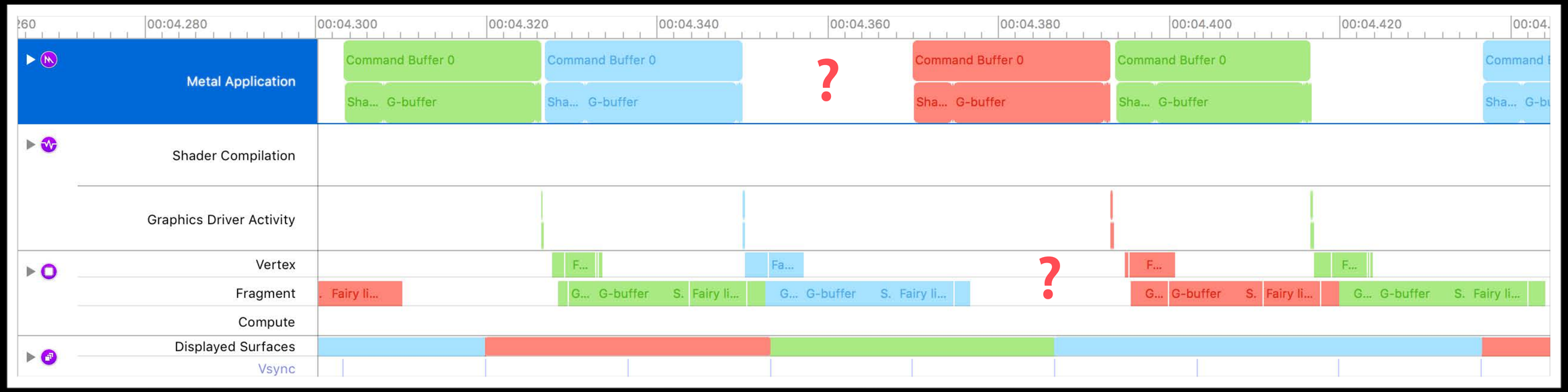


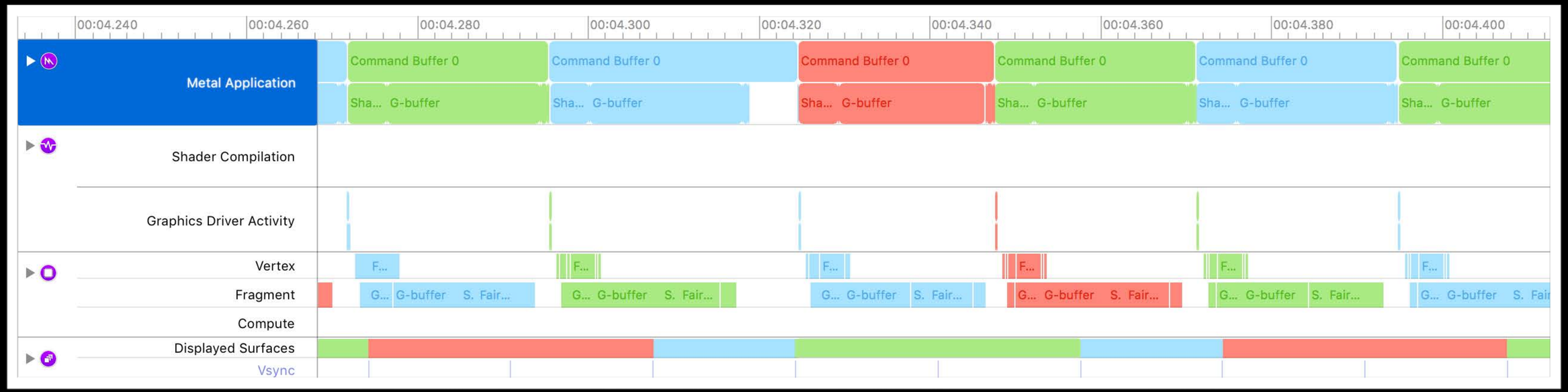
When to Acquire Your Drawable



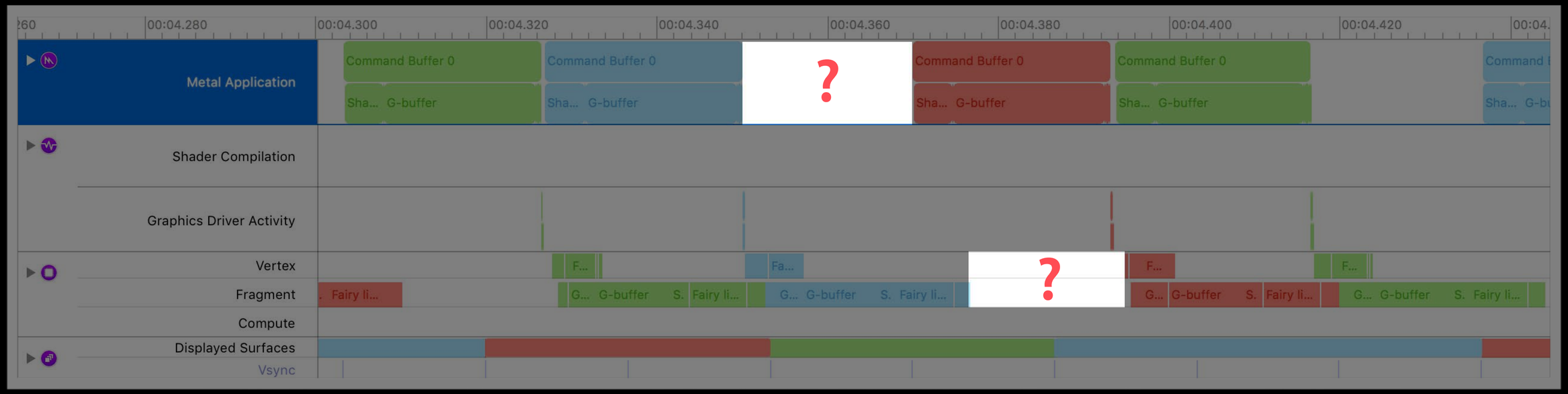
When to Acquire Your Drawable

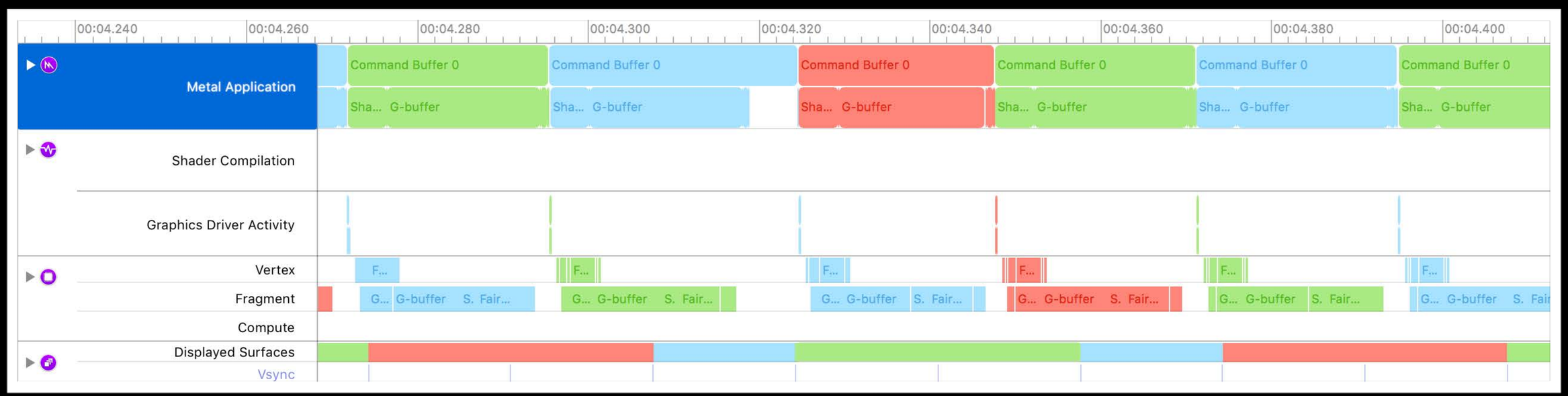



00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
00:04.400
00:04.420
Metal Application
Command Buffer 0
Sha... G-buffer
?
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
Fa...
Fairy li...
G... G-buffer S. Fairy li...

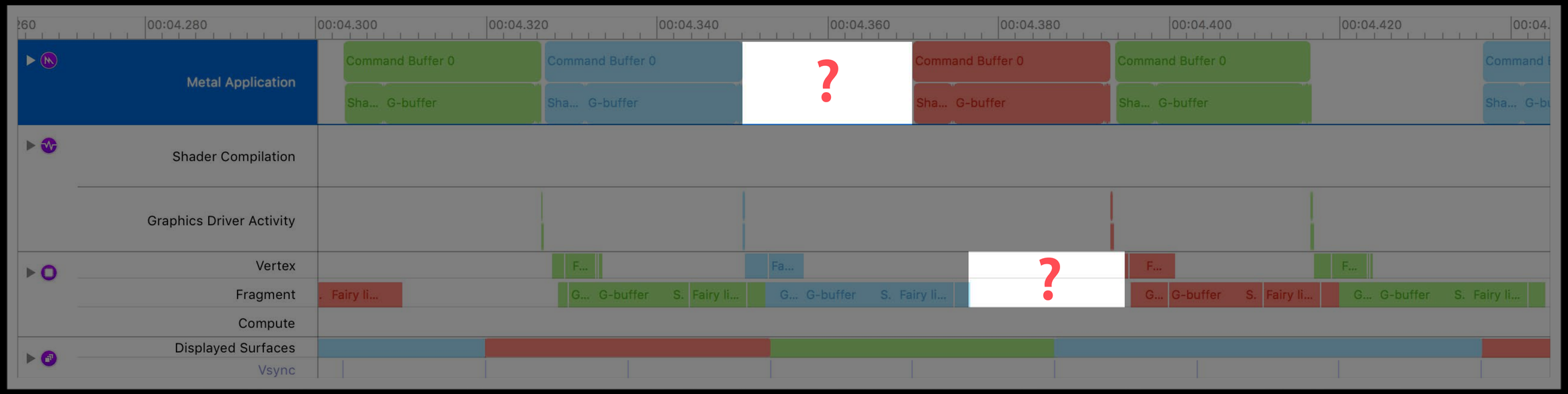

00:04.240
00:04.260
00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
00:04.400
Metal Application
Command Buffer 0
Sha... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
G... G-buffer S. Fair...

Before

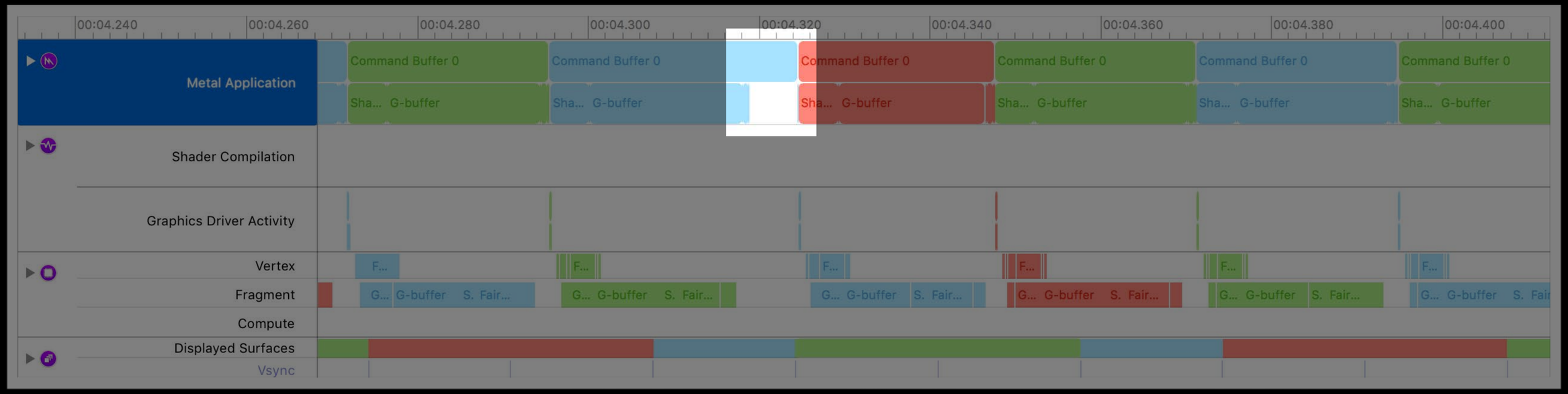

00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
00:04.400
00:04.420
Metal Application
Command Buffer 0
Sha... G-buffer
?
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
Fa...
Fairy li...
G... G-buffer S. Fairy li...


00:04.240
00:04.260
00:04.280
00:04.300
00:04.320
00:04.340
00:04.360
00:04.380
00:04.400
Metal Application
Command Buffer 0
Sha... G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
F...
G... G-buffer S. Fair...

Before



After



Metal Optimization Best Practices

Acquire the drawable at the latest opportunity

- Immediately before the render pass in which it is used
- Hides long latency if no drawables available

4

Don't Waste Render Command Encoders

Render Command Encoder

Render Command Encoder

Generates hardware commands for single rendering pass

- Specifies states for vertex and fragment stages of 3D pipeline
- Interleaves resources, state changes, and draw calls
- All rendering to a single set of render attachments

Render Encoders and iOS Device GPUs

Render Encoders and iOS Device GPUs

Tile-Based Deferred Renderer

Render Encoders and iOS Device GPUs

Tile-Based Deferred Renderer

Each render command encoder results in two GPU phases

Render Encoders and iOS Device GPUs

Tile-Based Deferred Renderer

Each render command encoder results in two GPU phases

- Vertex phase transforms and bins encoded draws into tiles

Render Encoders and iOS Device GPUs

Tile-Based Deferred Renderer

Each render command encoder results in two GPU phases

- Vertex phase transforms and bins encoded draws into tiles
- Fragment phase processes visible pixels in each tile

Render Encoders and iOS Device GPUs

Tile-Based Deferred Renderer

Each render command encoder results in two GPU phases

- Vertex phase transforms and bins encoded draws into tiles
- Fragment phase processes visible pixels in each tile
  - Processing performed in fast on-chip tile buffers

MetalApp 〉 Philip's iPad (9.0)

Running MetalApp on Philip's iPad

MetalApp

FPS	55 FPS
0 0x10046fd60 <- [MTLLayer nextDrawable]	
1 0x100457070 <- [MTLDrawable texture]	
2 0x100447ca0 <- [0x100430bc0 commandBuffer]	
CommandBuffer 0x100447ca0	
Shadow Buffer	11.99 µs
G-buffer	0.00 ns
G-buffer	11.77 ms
Sun light	3.30 ms
Fairy lights	1.04 ms
Fairy sprites	81.37 µs
UI	1.14 ms
1009 [addCompletedHandler:0x16fdfd090]	
1010 [presentDrawable:0x10046fd60]	
1011 [commit]	

MetalApp.gputrace 〉 CommandBuffer 0x100447ca0 〉 G-buffer 〉 18 [endEncoding]

Color Attachment 0

Color Attachment 1

Color Attachment 2

Depth Attachment

Stencil Attachment

... 〉 All 〉 No Selection 〉 4

FS Color0 0x100437870

FS Color1 0x100437df0

FS Color2 0x100438350

FS Depth 0x100437350

FS Stencil 0x1004375e0

MetalApp 〉 CommandB...100447ca0 〉 G-buffer 〉 18 [endEncoding]

"G-buffer" (0x100452130)

Encoder Statistics 0 draw calls, 51MB load/store bandwidth

- **Draw Calls** = 0
- **Load Bandwidth** = 0B
- **Store Bandwidth** = 51MB

Frame Statistics 271 draw calls, 246MB load/store bandwidth

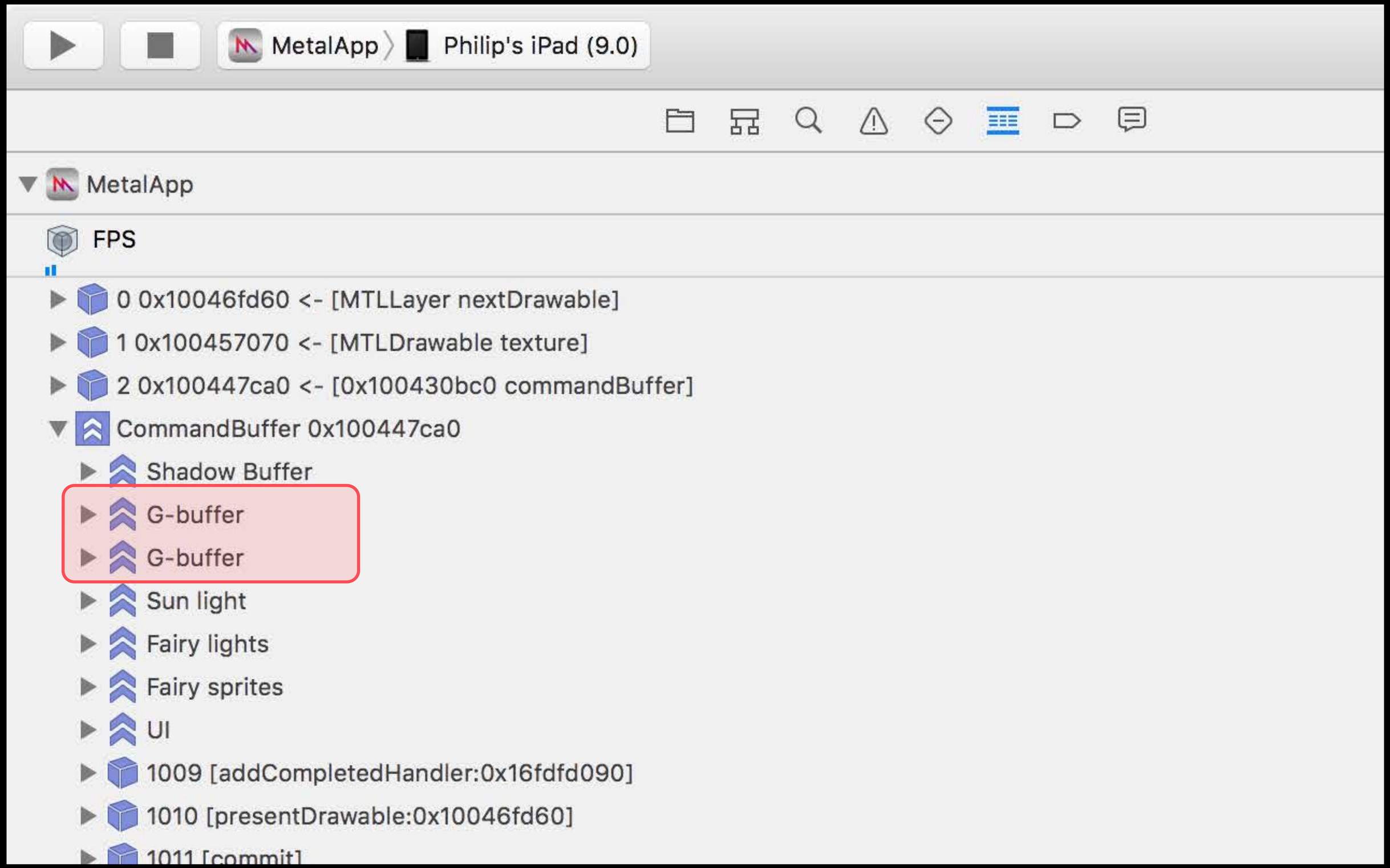
- **Frame Rate** = 56.5fps
- **Draw Calls** = 271
- **Command Buffers** = 1
- **Render Encoders** = 7
- **Load Bandwidth** = 54MB
- **Store Bandwidth** = 192MB

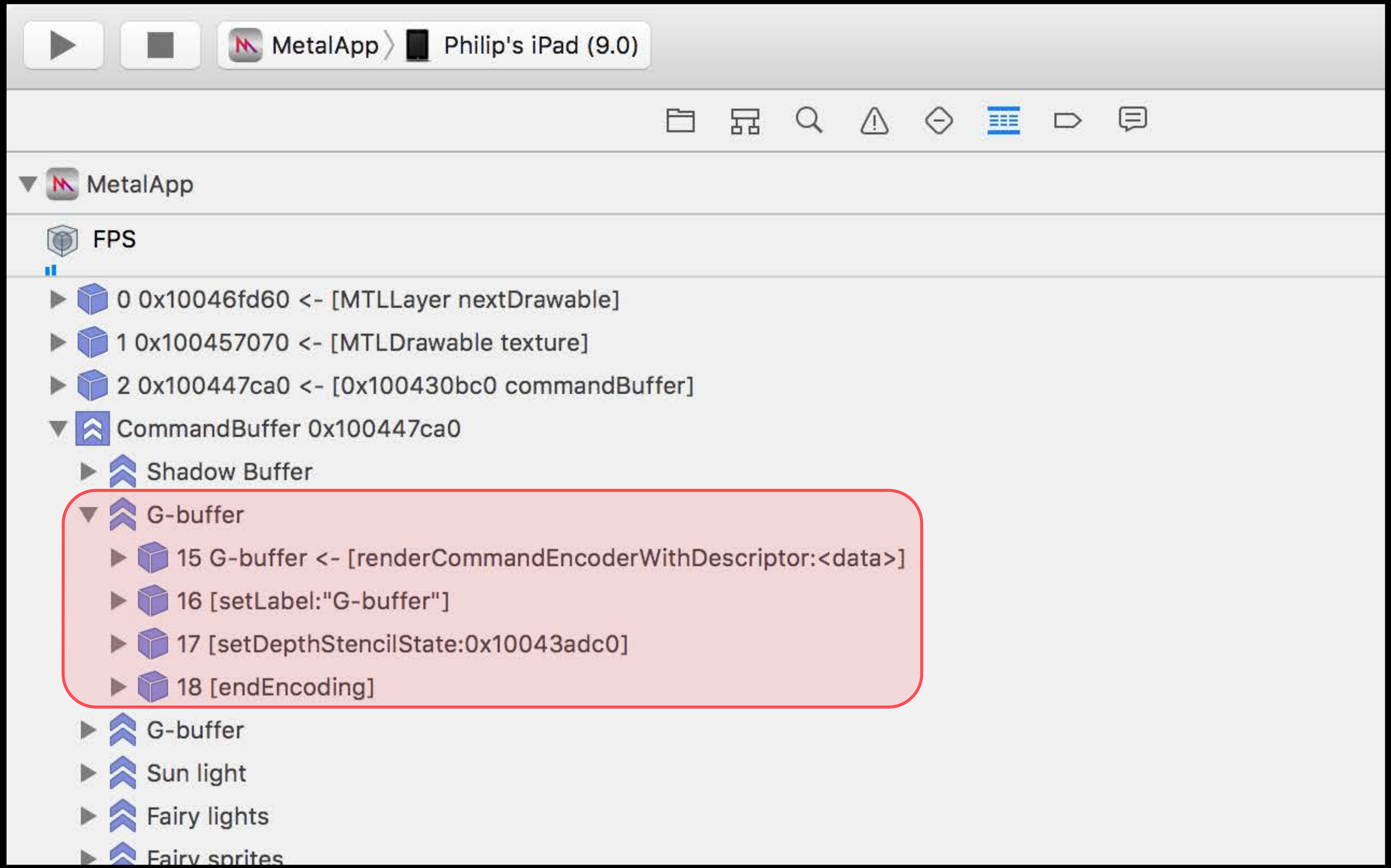
Auto

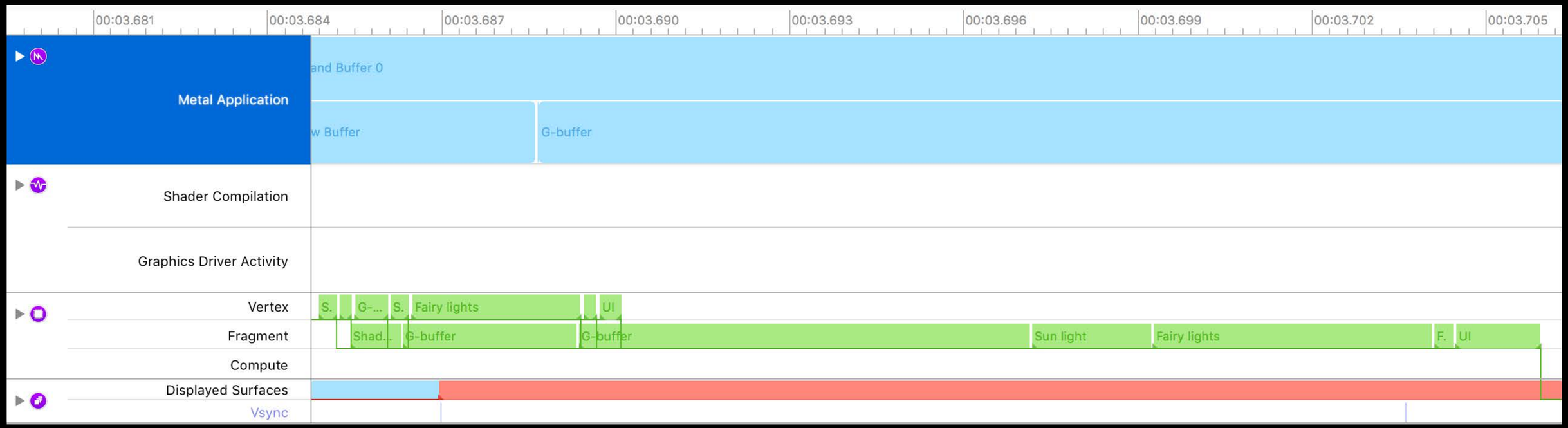
"G-buffer" (0x100452130)

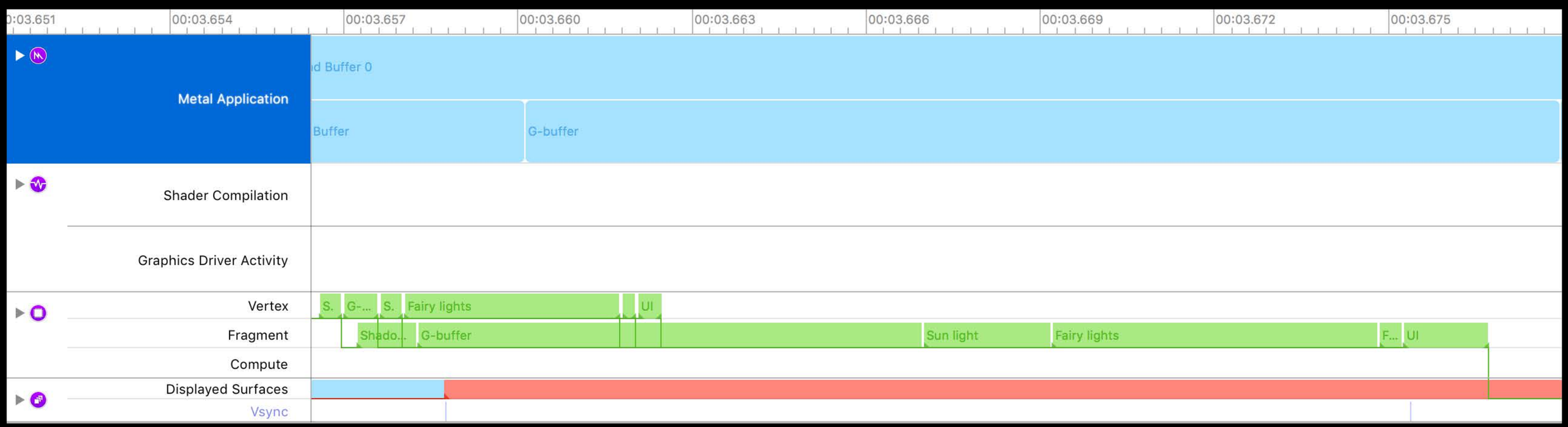
- **IsEncodingEnded** = YES
- **RenderPipelineState 0x0** Does not exist
- **TriangleFillMode** = Fill
- **Viewport** = 0, 0 - 1536, 2048 / 0 - 1
- **ScissorRect** = 0,0 - 1536 * 2048
- **BlendColor** = 0, 0, 0, 1
- **CullMode** = None
- **FrontFacingWinding** = Clockwise
- **DepthBias** = 0.00
- **DepthSlopeScale** = 0.00
- **DepthBiasClamp** = 0.00
- **DepthClipMode** = Clip
- **DepthStencilState 0x10043adc0** Depth: {LessEqual, Write Yes} Stencil: {Always (Keep, Keep, Keep)}
- **StencilFrontReferenceValue** = 0
- **StencilBackReferenceValue** = 0
- **VisibilityResultMode** = Disabled
- **RenderPassDescriptor** *1536x2048 Color(x3)/Depth/Stencil*

MTL Command Encoder

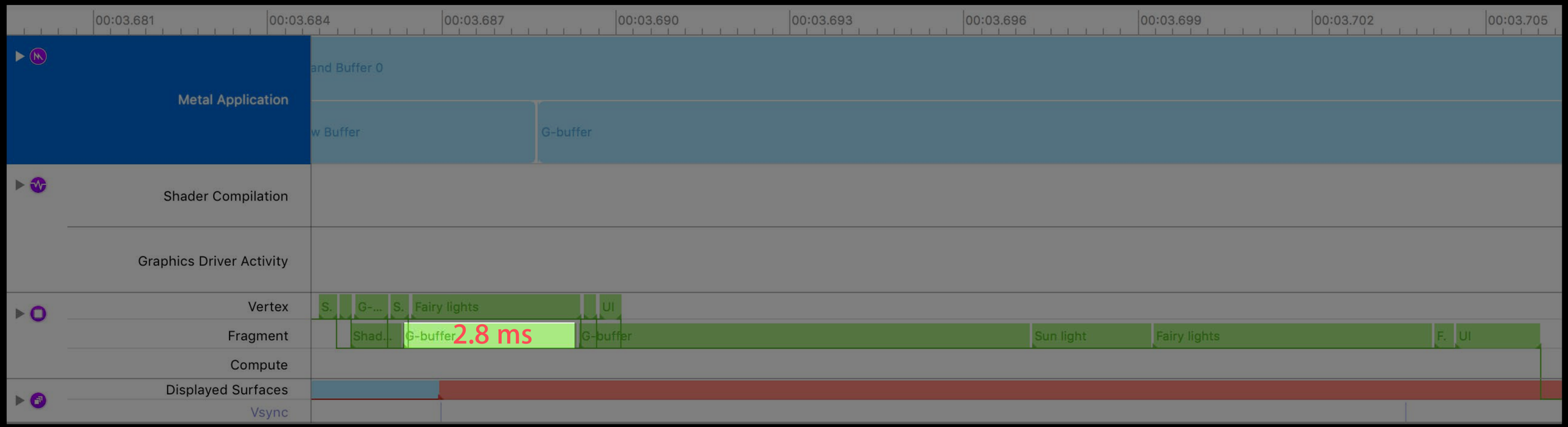

MetalApp
Philip's iPad (9.0)
MetalApp
FPS
0 0x10046fd60 <- [MTLLayer nextDrawable]
1 0x100457070 <- [MTLDrawable texture]
2 0x100447ca0 <- [0x100430bc0 commandBuffer]
CommandBuffer 0x100447ca0
Shadow Buffer
G-buffer
G-buffer
Sun light
Fairy lights
Fairy sprites
UI
1009 [addCompletedHandler:0x16fdfd090]
1010 [presentDrawable:0x10046fd60]
1011 [commit]

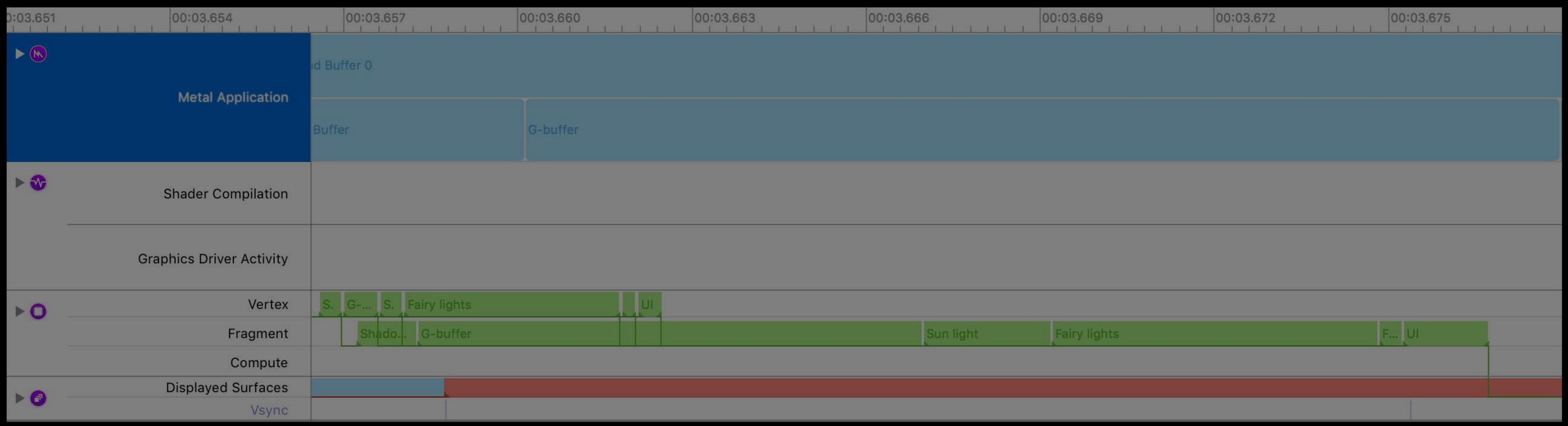

MetalApp
Philip's iPad (9.0)
MetalApp
FPS
0 0x10046fd60 <- [MTLLayer nextDrawable]
1 0x100457070 <- [MTLDrawable texture]
2 0x100447ca0 <- [0x100430bc0 commandBuffer]
CommandBuffer 0x100447ca0
Shadow Buffer
G-buffer
15 G-buffer <- [renderCommandEncoderWithDescriptor:<data>]
16 [setLabel:"G-buffer"]
17 [setDepthStencilState:0x10043adc0]
18 [endEncoding]
G-buffer
Sun light
Fairy lights
Fairy sprites


00:03.681
00:03.684
00:03.687
00:03.690
00:03.693
00:03.696
00:03.699
00:03.702
00:03.705
Metal Application
and Buffer 0
w Buffer
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
S.
G-...
S.
Fairy lights
UI
Shad...
G-buffer
G-buffer
Sun light
Fairy lights
F.
UI
Displayed Surfaces
Vsync


00:03.651
00:03.654
00:03.657
00:03.660
00:03.663
00:03.666
00:03.669
00:03.672
00:03.675
Metal Application
d Buffer 0
Buffer
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
S.
G-...
S.
Fairy lights
UI
Shado...
G-buffer
Sun light
Fairy lights
F...
UI
Displayed Surfaces
Vsync

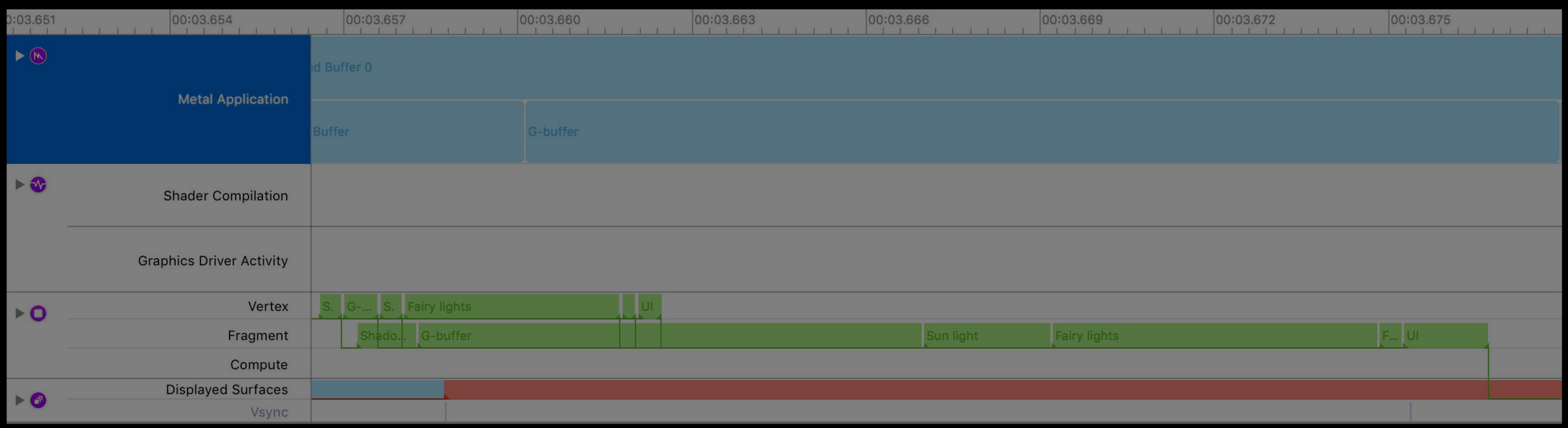
Before


00:03.681
00:03.684
00:03.687
00:03.690
00:03.693
00:03.696
00:03.699
00:03.702
00:03.705
Metal Application
and Buffer 0
w Buffer
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
S.
G-...
S.
Fairy lights
UI
Fragment
Shad...
G-buffer
2.8 ms
G-buffer
Sun light
Fairy lights
F.
UI
Compute
Displayed Surfaces
Vsync


00:03.651
00:03.654
00:03.657
00:03.660
00:03.663
00:03.666
00:03.669
00:03.672
00:03.675
Metal Application
d Buffer 0
Buffer
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
S.
G-...
S.
Fairy lights
UI
Fragment
Shado...
G-buffer
Sun light
Fairy lights
F...
UI
Compute
Displayed Surfaces
Vsync

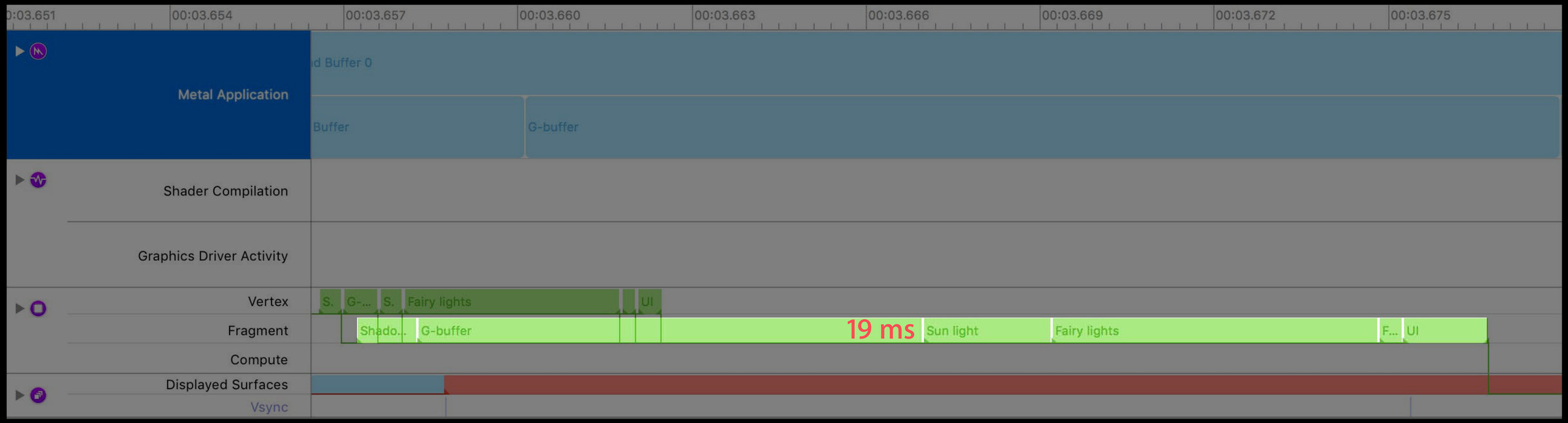
Before


00:03.681
00:03.684
00:03.687
00:03.690
00:03.693
00:03.696
00:03.699
00:03.702
00:03.705
Metal Application
and Buffer 0
w Buffer
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
S.
G-...
S.
Fairy lights
UI
Shad...
G-buffer
G-buffer
22 ms
Sun light
Fairy lights
F.
UI


00:03.651
00:03.654
00:03.657
00:03.660
00:03.663
00:03.666
00:03.669
00:03.672
00:03.675
Metal Application
d Buffer 0
Buffer
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
S.
G-...
S.
Fairy lights
UI
Shado...
G-buffer
Sun light
Fairy lights
F...
UI

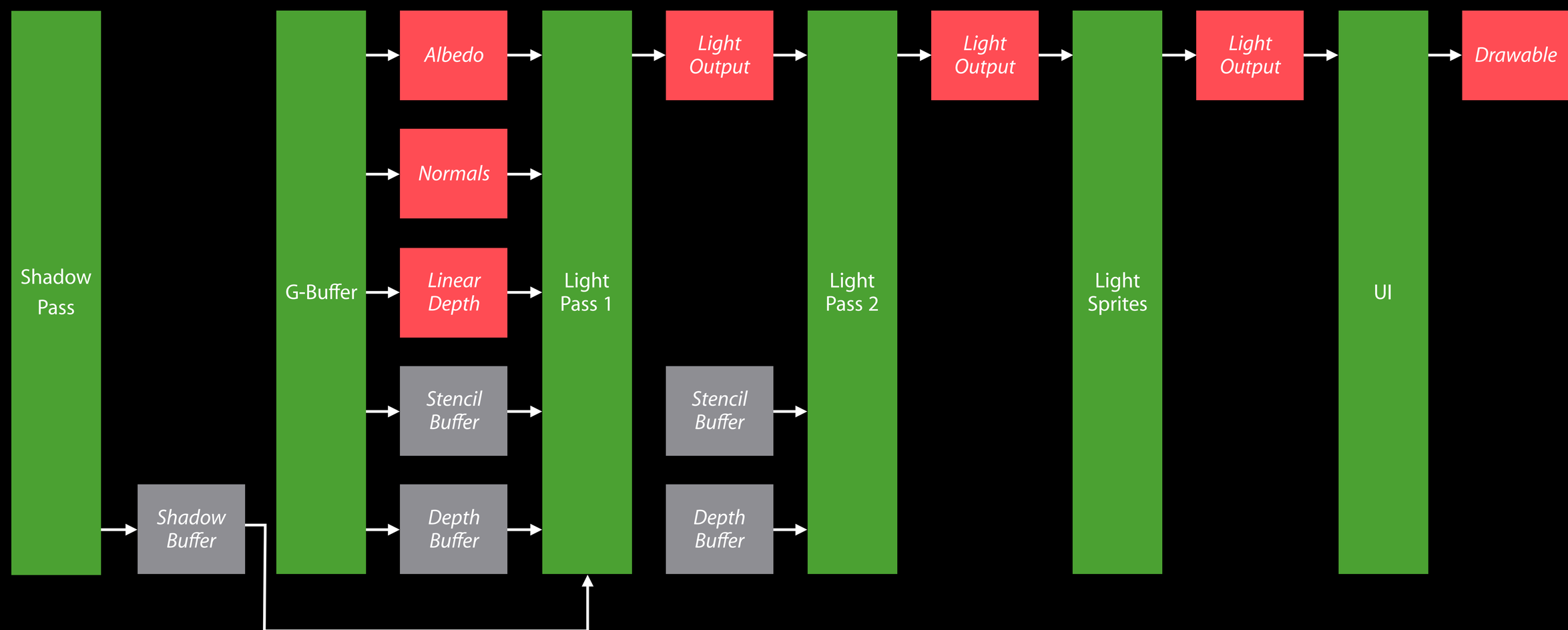
Before

After



Render Passes in Example App

Render Passes in Example App



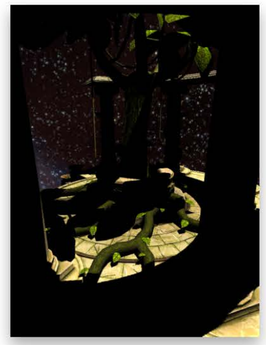
MetalApp 〉 Philip's iPad

Running MetalApp on Philip's iPad

68 [drawPrimitives:3 vertexStart:0 vertexCount:6 instanceCount:1]

All GPU Objects 〉 Textures 〉 No Selection

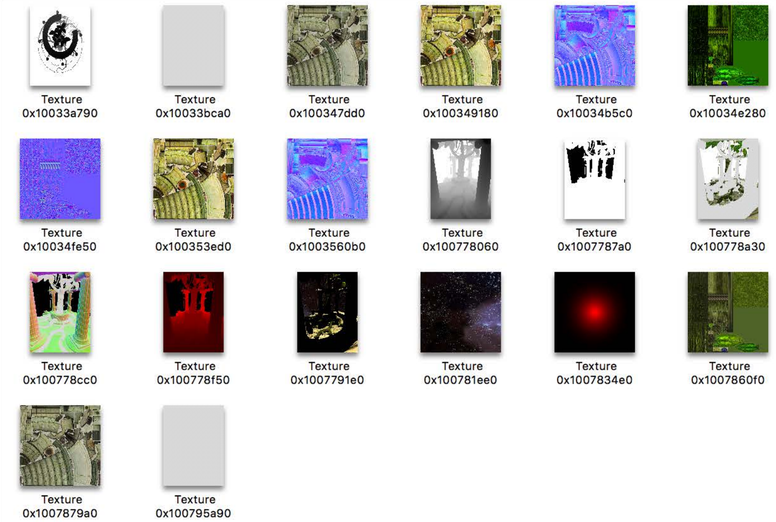
- MetalApp
  - FPS — 60 FPS
  - 0 0x10079d9a0 <- [MTLLayer nextDrawable]
  - 1 0x100795a90 <- [MTLDrawable texture]
  - 2 0x100795ff0 <- [0x100335da0 commandBuffer]
  - CommandBuffer 0x100795ff0
    - Shadow Buffer — 9.37 µs
    - G-buffer — 10.18 ms
    - Sun light — 2.84 ms
    - Fairy lights — 825.65 µs
    - Fairy sprites — 69.31 µs
    - UI — 980.66 µs
    - 1005 [addCompletedHandler:0x16fdfd090]
    - 1006 [presentDrawable:0x10079d9a0]
    - 1007 [commit]



Color Attachment 0

Depth Attachment

Stencil Attachment



Sun 〉 68 [drawPrimitives:3 vertexStart:0 vertexCount:6 instanceCount:1]

- **"Composition Render" (0x100780050)** compositionVertex/compositionFragAlt
- **RenderPipeline Performance** 2.16 ms (14.5%)
- **Fragment Texture 0** 0x100778a30 - 1536 x 2048, BGRA8Unorm
- **Fragment Texture 1** 0x100778cc0 - 1536 x 2048, BGRA8Unorm
- **Fragment Texture 2** 0x1007791e0 - 1536 x 2048, BGRA8Unorm
- **Fragment Buffer 0** 0x10124c9c0
- **Vertex Buffer 0** 0x10033e1a0
- **Vertex Buffer 1** 0x10124bcc0
- **"Sun light" (0x1007a5530)**
- **Encoder Statistics** 7 draw calls, 42MB load/store bandwidth
- **Frame Statistics** 271 draw calls, 195MB load/store bandwidth

- **"Sun light" (0x1007a5530)**
  - **IsEncodingEnded** = NO
  - **"Composition Render" (0x100780050)** compositionVertex/compositionFragAlt
  - **TriangleFillMode** = Fill
  - **Viewport** = 0, 0 - 1536, 2048 / 0 - 1
  - **ScissorRect** = 0,0 - 1536 * 2048
  - **BlendColor** = 0, 0, 0, 1
  - **CullMode** = None
  - **FrontFacingWinding** = Clockwise
  - **DepthBias** = 0.00
  - **DepthSlopeScale** = 0.00
  - **DepthBiasClamp** = 0.00
  - **DepthClipMode** = Clip
  - **DepthStencilState 0x10033d9e0** Depth: {Always, Write No} Stencil: {Equal (Keep, Keep, Keep), WriteMask(0x00)}
  - **StencilFrontReferenceValue** = 128
  - **StencilBackReferenceValue** = 128
  - **VisibilityResultMode** = Disabled
  - **Fragment Texture 0** 0x100778a30 - 1536 x 2048, BGRA8Unorm
  - **Fragment Texture 1** 0x100778cc0 - 1536 x 2048, BGRA8Unorm

Auto

MTL Command Encoder

MetalApp

FPS 60 FPS

- 0 0x10079d9a0 <- [MTLLayer nextDrawable]
- 1 0x100795a90 <- [MTLDrawable texture]
- 2 0x100795ff0 <- [0x100335da0 commandBuffer]
- CommandBuffer 0x100795ff0
  - Shadow Buffer 9.37 µs
  - G-buffer 10.18 ms
  - Sun light 2.84 ms ⚠️
  - Fairy lights 825.65 µs
  - Fairy sprites 69.31 µs
  - UI 980.66 µs
  - 1005 [addCompletedHandler:0x16fdfd090]
  - 1006 [presentDrawable:0x10079d9a0]
  - 1007 [commit]

MetalApp

FPS 60 FPS

- 0 0x10079d9a0 <- [MTLLayer nextDrawable]
- 1 0x100795a90 <- [MTLDrawable texture]
- 2 0x100795ff0 <- [0x100335da0 commandBuffer]
- CommandBuffer 0x100795ff0
  - Shadow Buffer 9.37 µs
  - G-buffer 10.18 ms
  - Sun light 2.84 ms ⚠
  - Fairy lights 825.65 µs
  - Fairy sprites 69.31 µs
  - UI 980.66 µs
  - 1005 [addCompletedHandler:0x16fdfd090]
  - 1006 [presentDrawable:0x10079d9a0]
  - 1007 [commit]

By File By Type

MetalApp 1 issue

Inefficient Use of Command Encoders

Your application created separate command encoders which can be combined into a single encoder for better performance.

AAPLRenderer.mm

RenderCommandEncoder Sun light (0x1007a5530)

RenderCommandEncoder Fairy lights (0x1007a5740)

By File By Type

MetalApp 1 issue

Inefficient Use of Command Encoders

Your application created separate command encoders which can be combined into a single encoder for better performance.

AAPLRenderer.mm

RenderCommandEncoder Sun light (0x1007a5530)

RenderCommandEncoder Fairy lights (0x1007a5740)

By File By Type

MetalApp 1 issue

Inefficient Use of Command Encoders

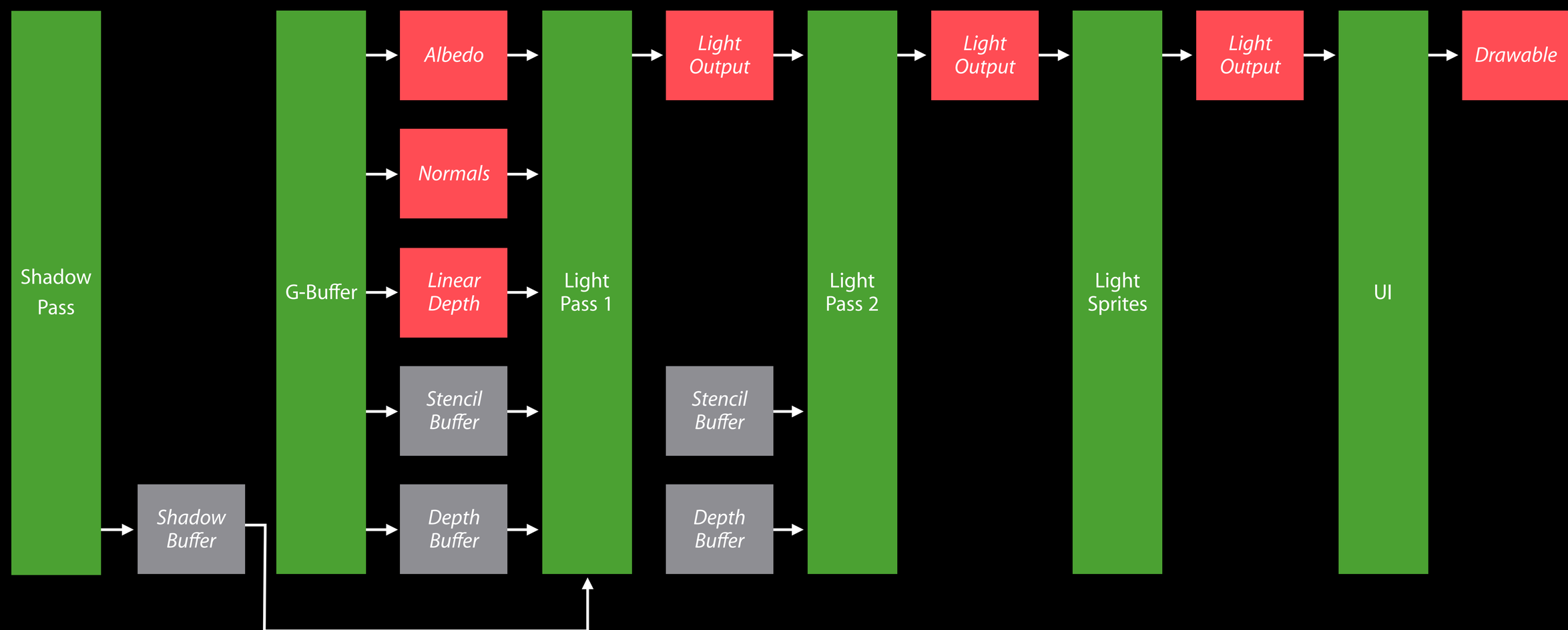
Your application created separate command encoders which can be combined into a single encoder for better performance.

AAPLRenderer.mm

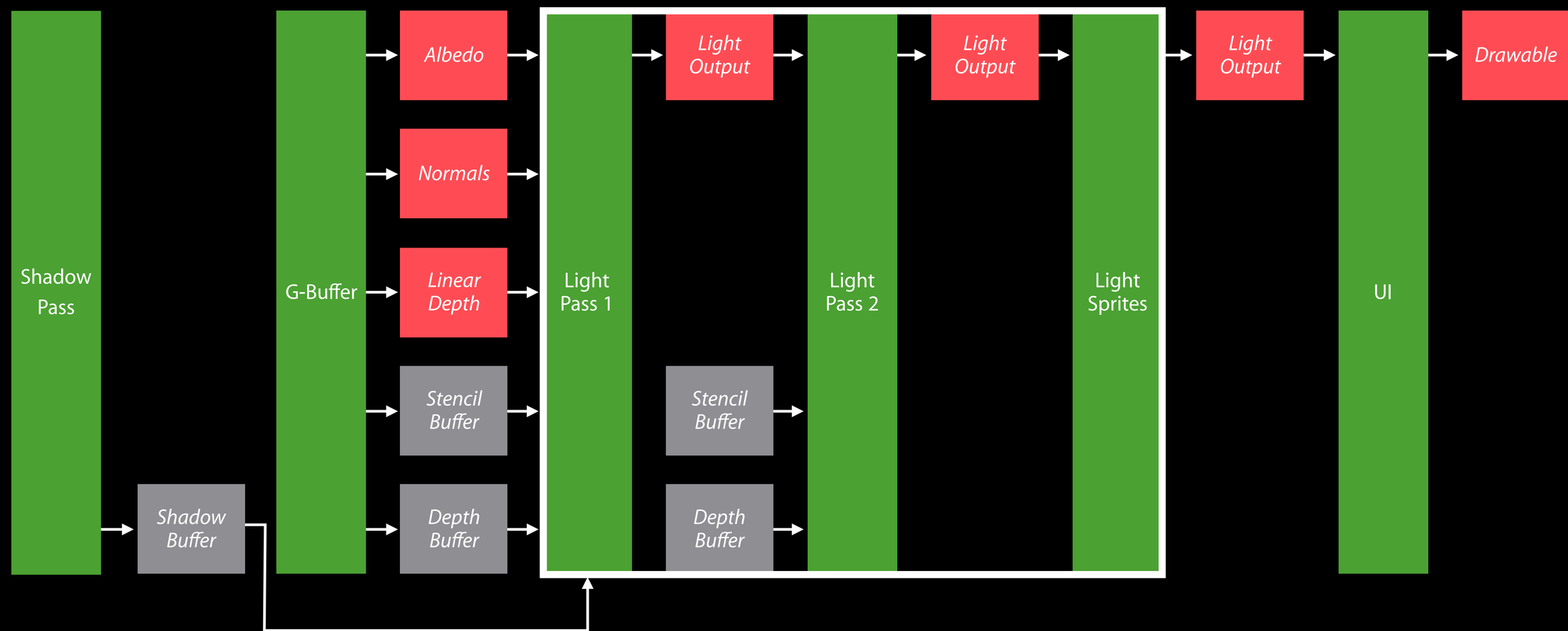
RenderCommandEncoder Sun light (0x1007a5530)

RenderCommandEncoder Fairy lights (0x1007a5740)

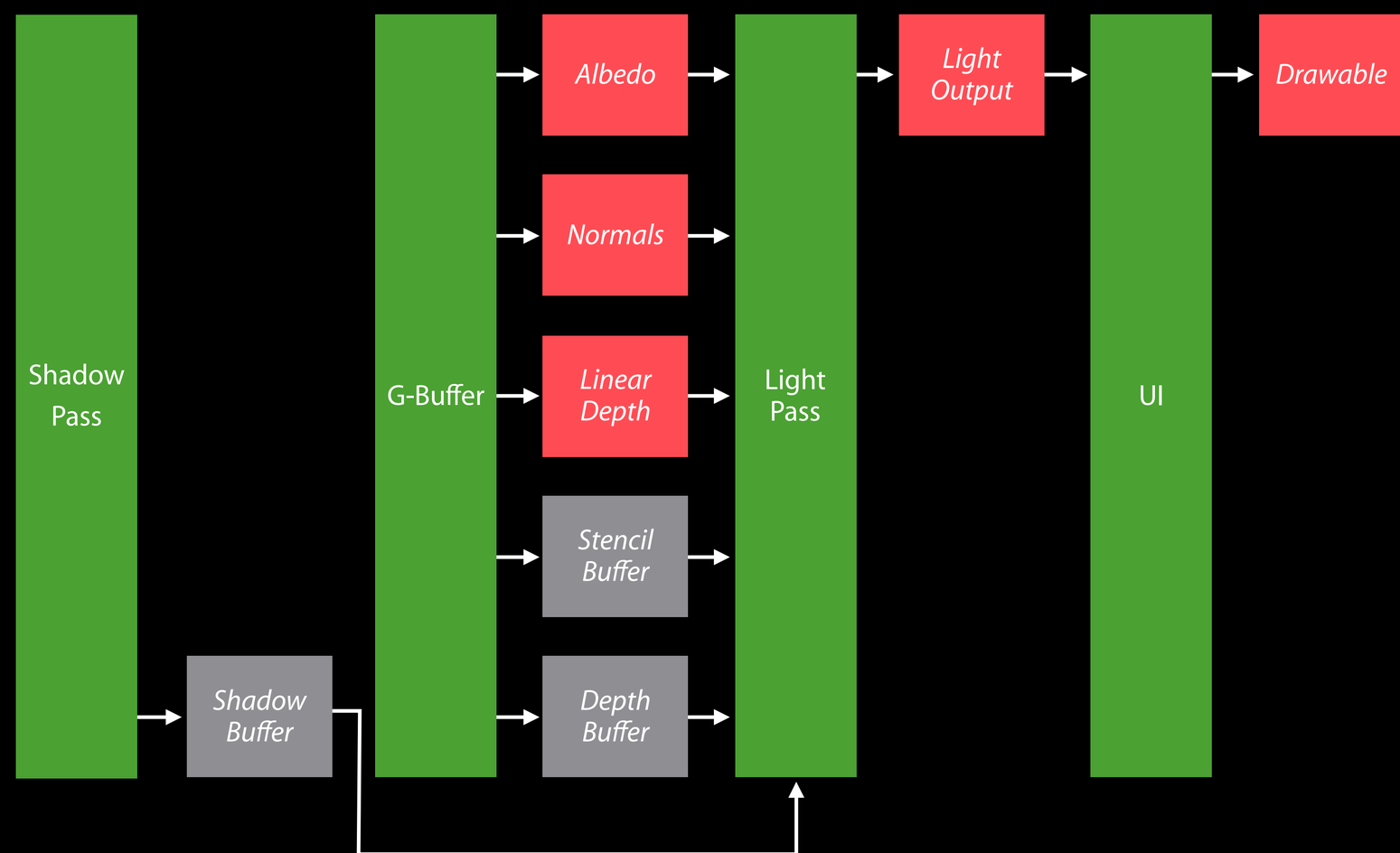
Merging Passes



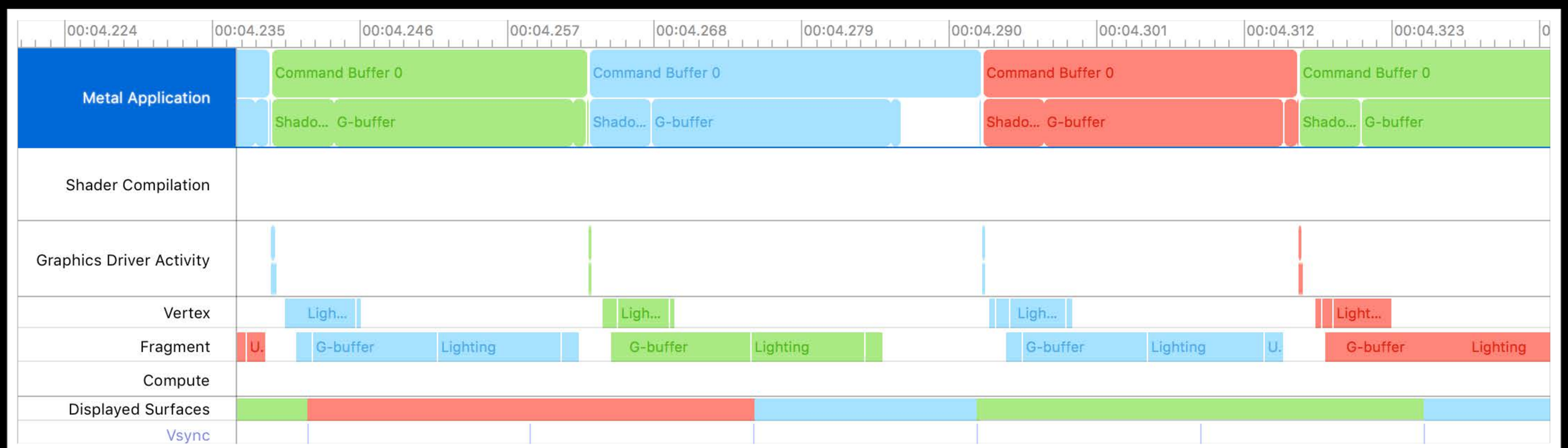
Merging Passes

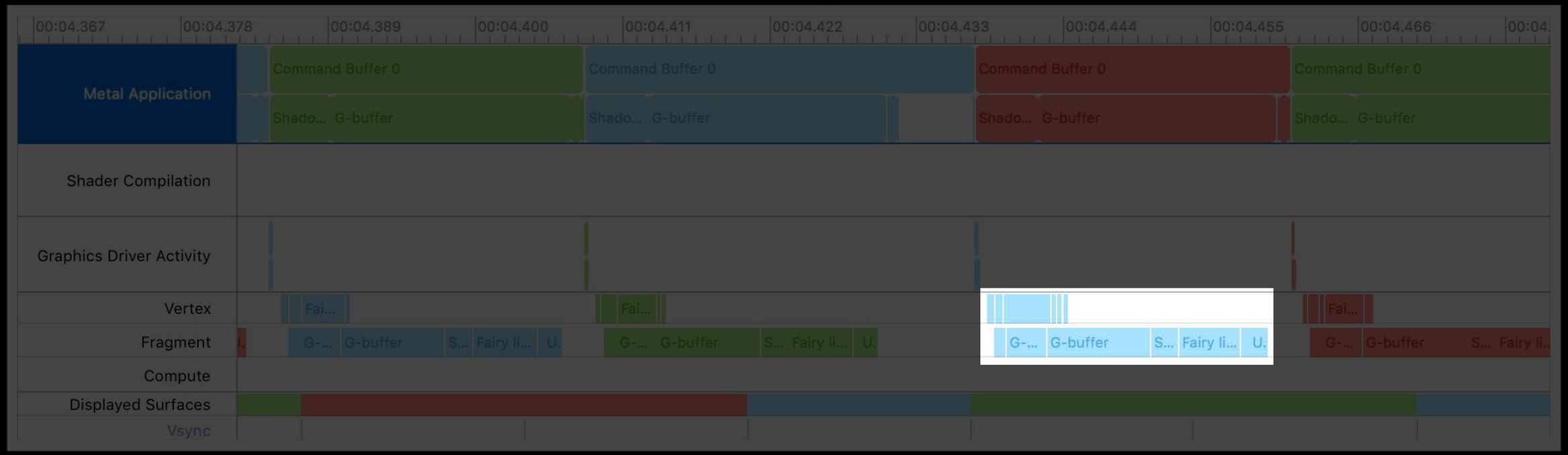


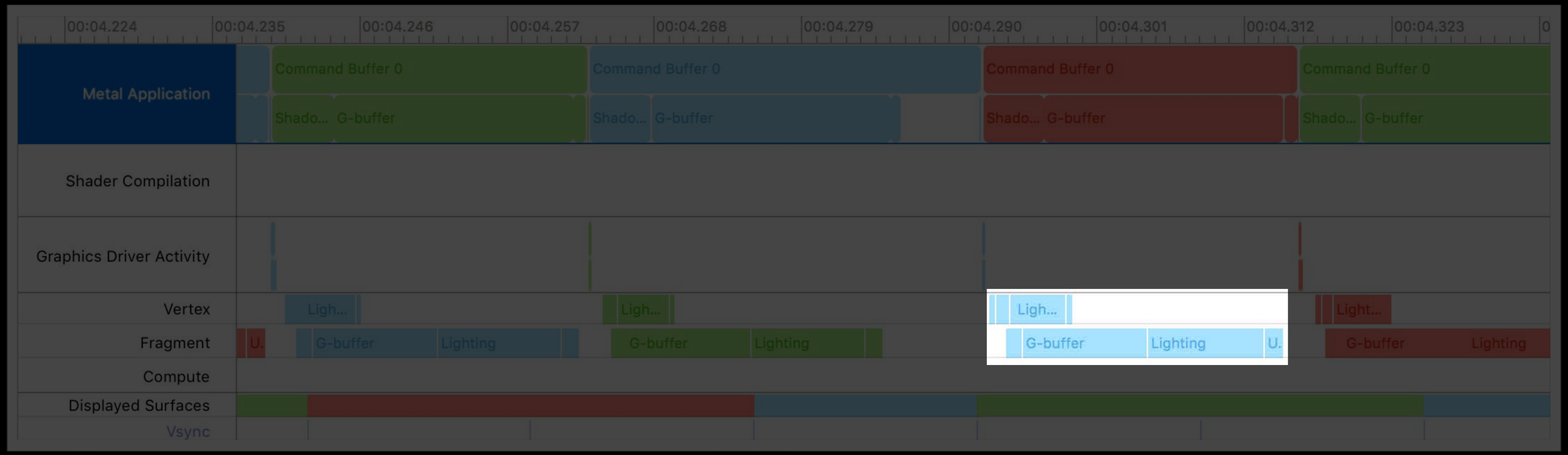
From Six Passes to Four Passes

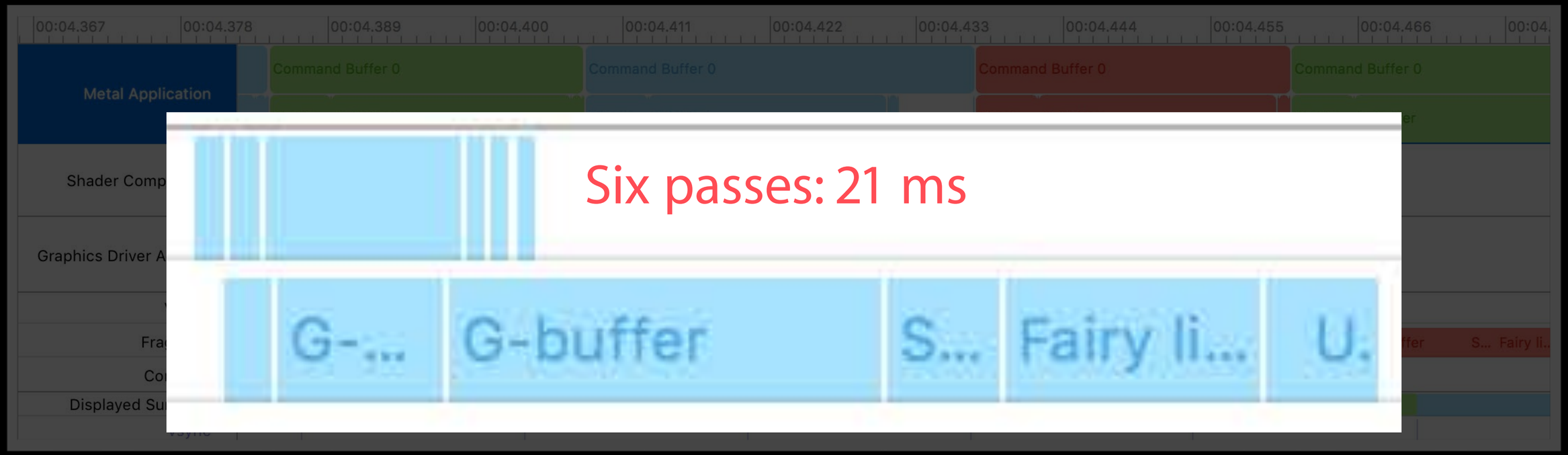


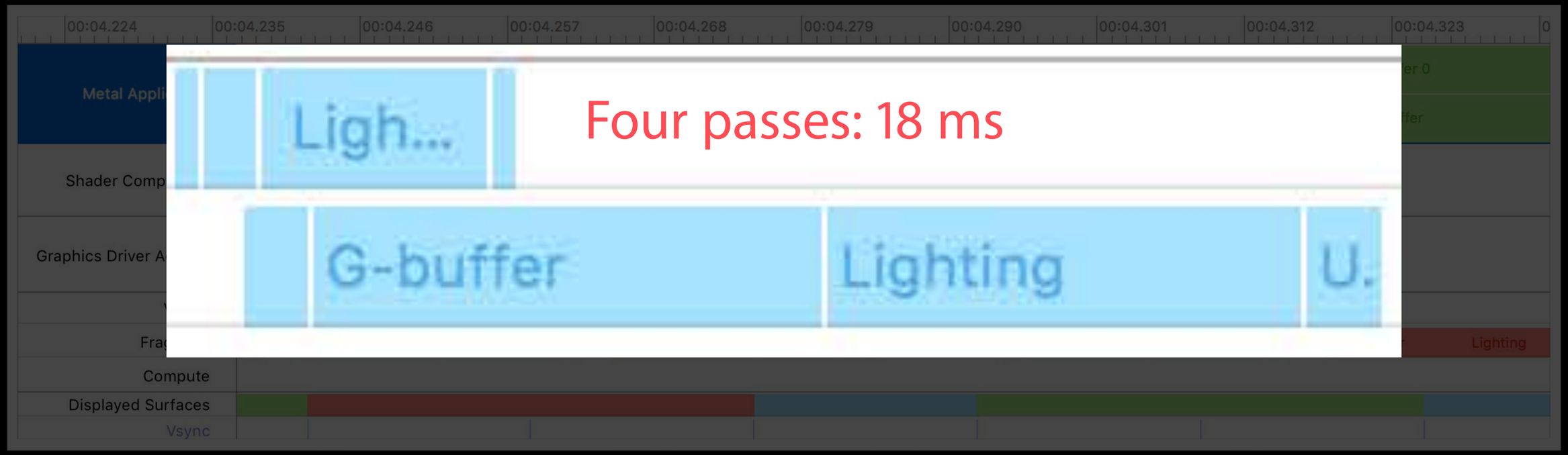

00:04.367
00:04.378
00:04.389
00:04.400
00:04.411
00:04.422
00:04.433
00:04.444
00:04.455
00:04.466
Metal Application
Command Buffer 0
Shado...
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fai...
Fragment
G-...
S...
Fairy li...
U.
Compute
Displayed Surfaces
Vsync


00:04.224
00:04.235
00:04.246
00:04.257
00:04.268
00:04.279
00:04.290
00:04.301
00:04.312
00:04.323
Metal Application
Command Buffer 0
Shado...
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Ligh...
Light...
Fragment
U.
Lighting
Compute
Displayed Surfaces
Vsync

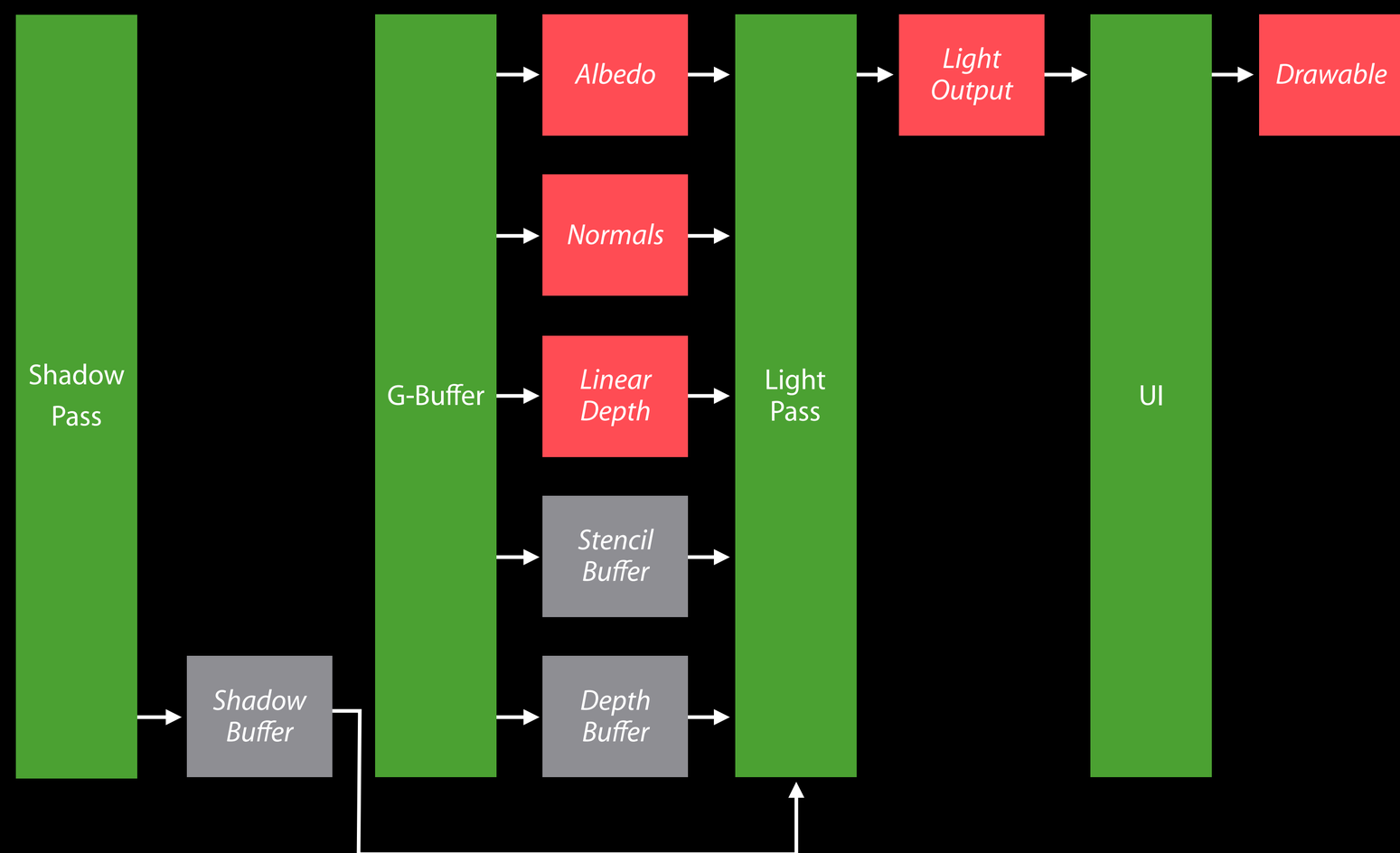

00:04.367
00:04.378
00:04.389
00:04.400
00:04.411
00:04.422
00:04.433
00:04.444
00:04.455
00:04.466
Metal Application
Command Buffer 0
Shado...
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Fai...
Fragment
G-...
S...
Fairy li...
U.
Compute
Displayed Surfaces
Vsync


00:04.224
00:04.235
00:04.246
00:04.257
00:04.268
00:04.279
00:04.290
00:04.301
00:04.312
00:04.323
Metal Application
Command Buffer 0
Shado...
G-buffer
Shader Compilation
Graphics Driver Activity
Vertex
Ligh...
Light...
Fragment
U.
Lighting
Compute
Displayed Surfaces
Vsync

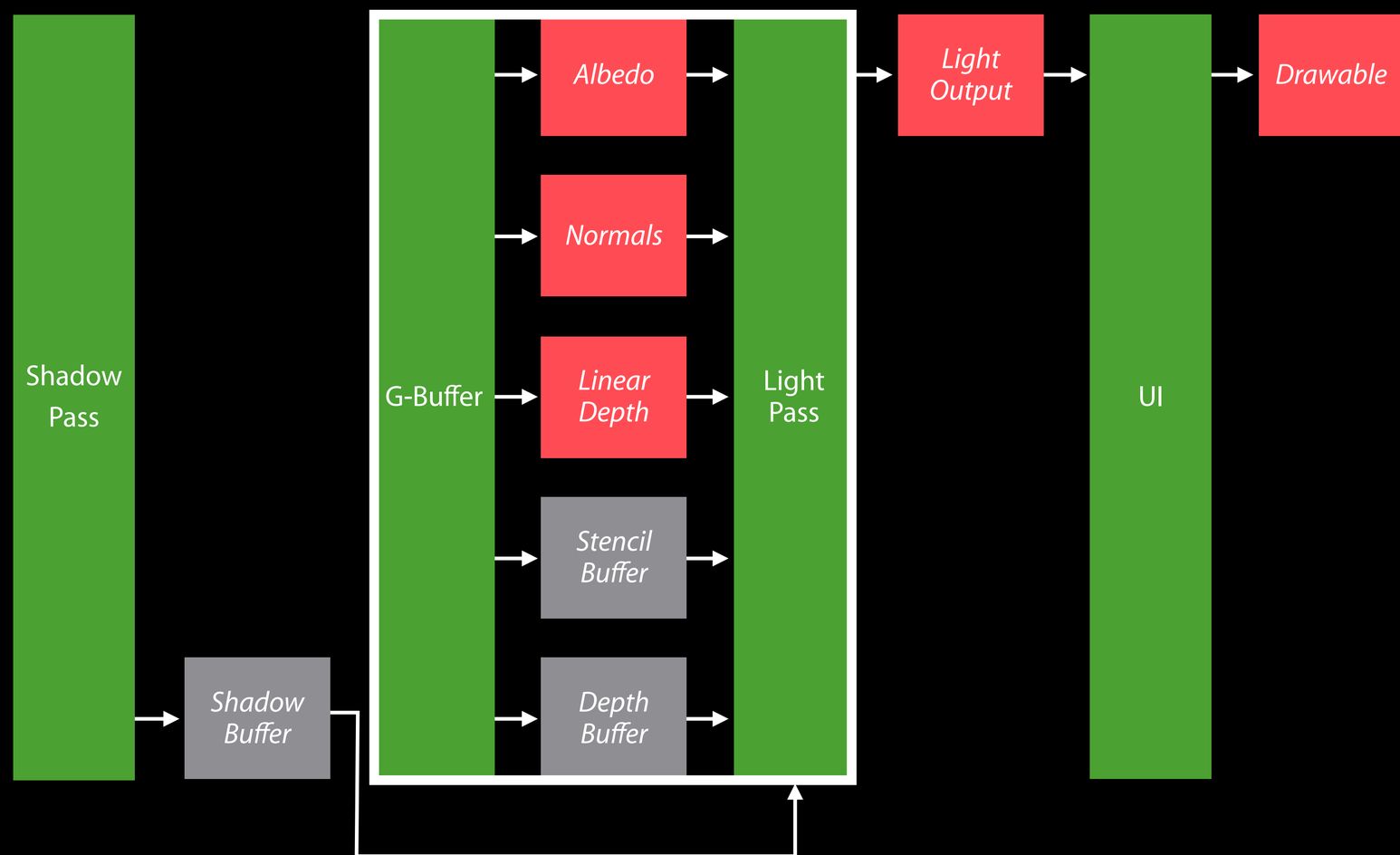

Six passes: 21 ms
G-...
G-buffer
S...
Fairy li...
U.


Four passes: 18 ms
Ligh...
G-buffer
Lighting
U.

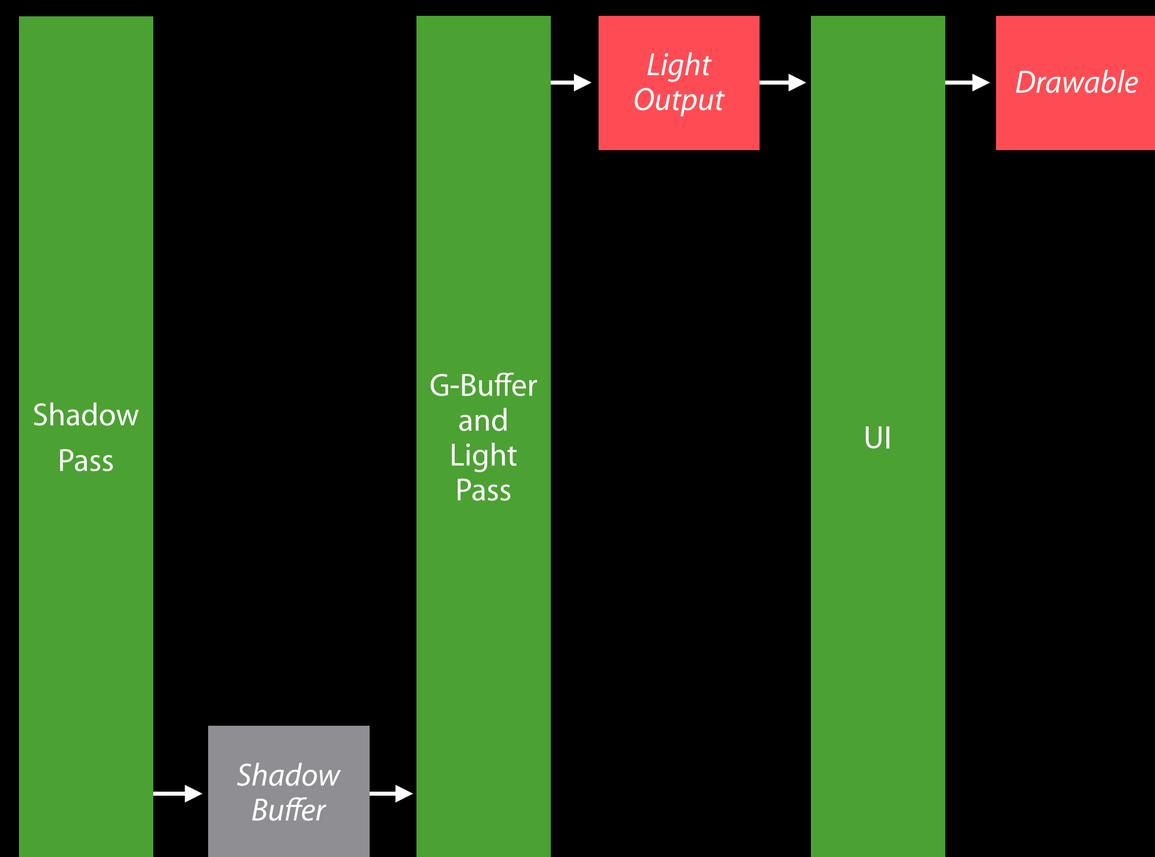
More Merging



More Merging

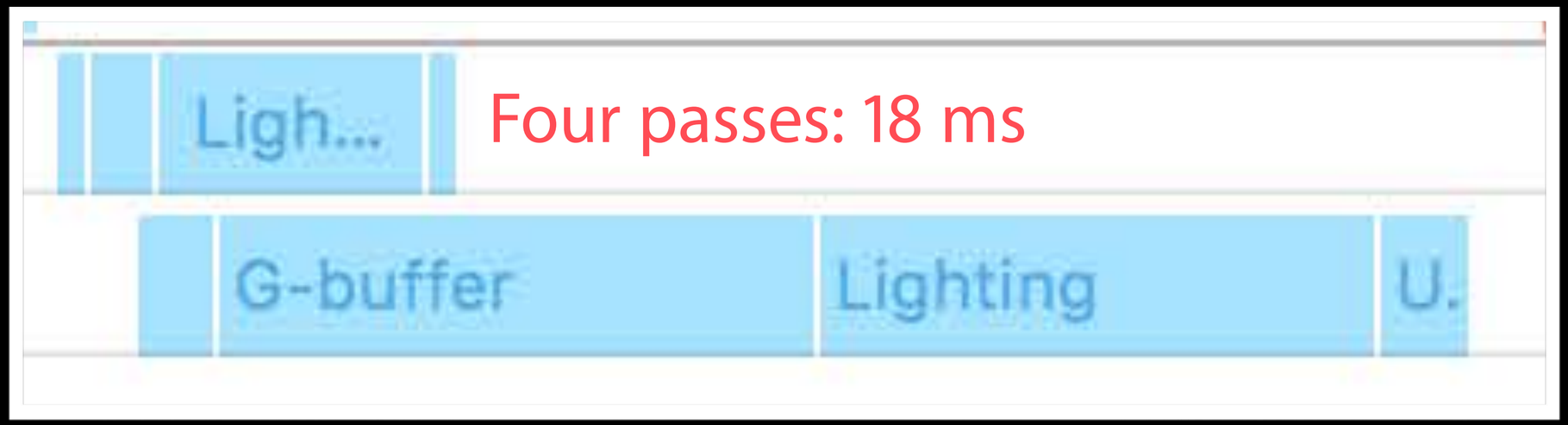


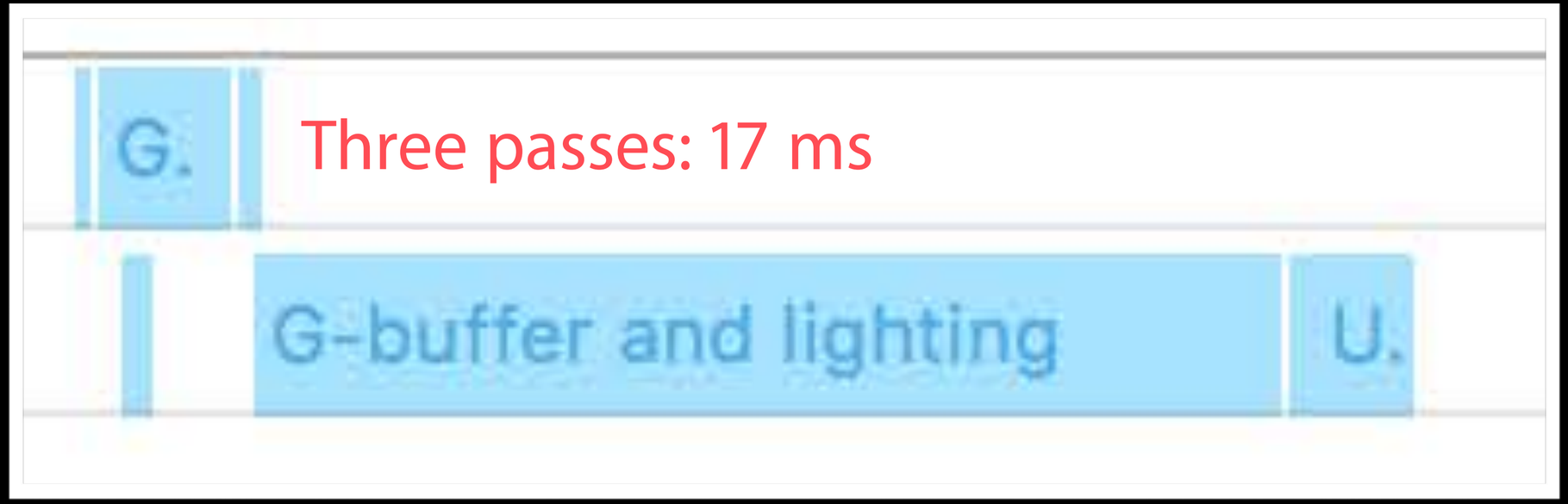
Down to Three Passes




Ligh...
G-buffer
Lighting
U.


G.
G-buffer and lighting
U.

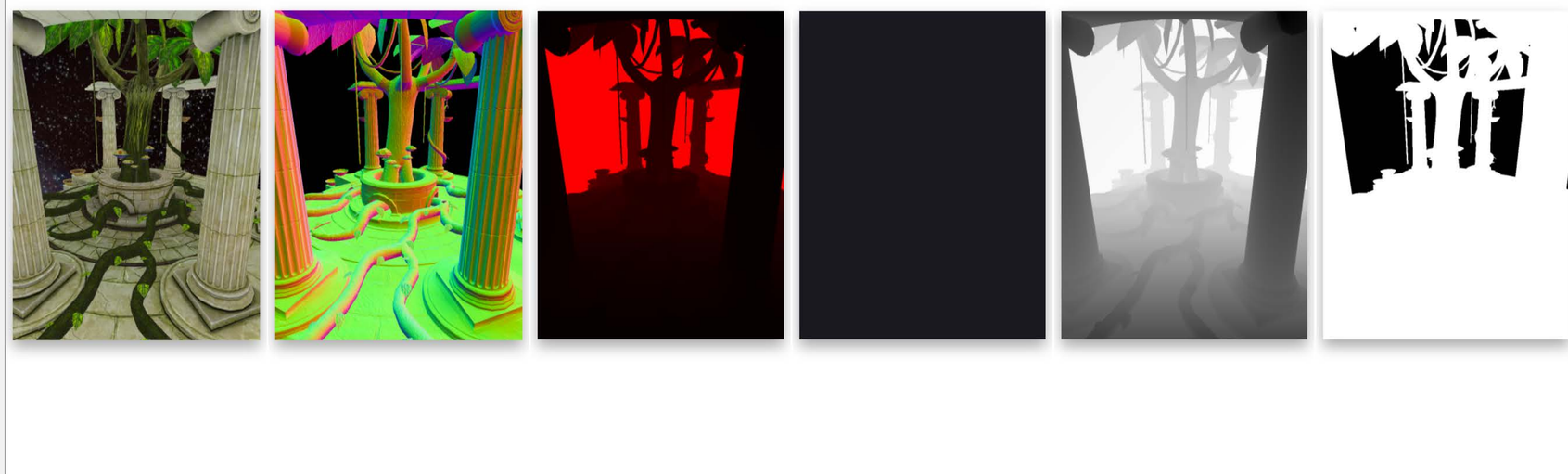

Ligh...
Four passes: 18 ms
G-buffer
Lighting
U.


G.
Three passes: 17 ms
G-buffer and lighting
U.

MetalApp 〉 Philip's iPad (9.0) | Running MetalApp on Philip's iPad

MetalApp.gputrace 〉 Comma...050f4a0 〉 G-buffe...lighting 〉 G-buffer pass 〉 structure 〉 54 [drawIndexedPrimitives:3 indexCount:48 indexType:0 indexBuffer:0x100505580 indexBufferOffset:164544 instanceCount:1]

- MetalApp
  - FPS — 60 FPS
  - 0 0x1003e3be0 <- [MTLLayer nextDrawable]
  - 1 0x1003ab230 <- [MTLDrawable texture]
  - 2 0x10050f4a0 <- [0x1003ba180 commandBuffer]
  - CommandBuffer 0x10050f4a0
    - Shadow Buffer — 7.31 µs
    - G-buffer and lighting — 14.60 ms
      - 15 G-buffer and lighting <- [renderCommandEncoderWithDescriptor:<data>]
      - G-buffer pass — 14.60 ms
        - 16 [pushDebugGroup:"G-buffer pass"]
        - 17 [setLabel:"G-buffer and lighting"]
        - 18 [setDepthStencilState:0x100431a00]
        - skybox — 2.14 ms
        - structure — 5.87 ms
          - 32 [pushDebugGroup:"structure"]
          - 33 [setRenderPipelineState:GBuffer Render]
          - 34 [setCullMode:2]
          - 35 [setDepthStencilState:0x10042e5c0]
          - 36 [setStencilReferenceValue:128]
          - 37 [setVertexBuffer:0x100504c10 offset:0 atIndex:0]
          - 38 [setVertexBuffer:0x100513c40 offset:0 atIndex:1]
          - 39 [setFragmentBuffer:0x10050d780 offset:0 atIndex:0]
          - 40 [setFragmentTexture:0x10045d930 atIndex:0]
          - 41 [setFragmentTexture:0x1003c8ca0 atIndex:1]
          - 42 [setFragmentTexture:0x1003cb510 atIndex:2]
          - 43 [setFragmentTexture:0x10042aaa0 atIndex:3]
          - 44 [drawIndexedPrimitives:3 indexCount:61266 indexType:0 indexBuffer:0... — 4.52 ms
          - 45 [setFragmentTexture:0x100516600 atIndex:0]
          - 46 [setFragmentTexture:0x10045de40 atIndex:1]
          - 47 [setFragmentTexture:0x1003cf350 atIndex:2]
          - 48 [setFragmentTexture:0x10042aaa0 atIndex:3]
          - 49 [drawIndexedPrimitives:3 indexCount:21006 indexType:0 indexBuffer:0x... — 1.35 ms
          - 50 [setFragmentTexture:0x100461100 atIndex:0]
          - 51 [setFragmentTexture:0x1003d0f80 atIndex:1]
          - 52 [setFragmentTexture:0x10045fd70 atIndex:2]
          - 53 [setFragmentTexture:0x10042aaa0 atIndex:3]
          - 54 [drawIndexedPrimitives:3 indexCount:48 indexType:0 indexBuffer:0x100... — 0.00 ns
          - 55 [popDebugGroup]
        - light accumulation — 4.76 ms
        - sun — 1.76 ms
        - fairy sprites — 67.21 µs
        - 2509 [popDebugGroup]
      - 2510 [endEncoding]
    - UI — 578.01 µs
  - 2519 [addCompletedHandler:0x16fdfd470]
  - 2520 [presentDrawable:0x1003e3be0]
  - 2521 [commit]



Color Attachment 0 | Color Attachment 1 | Color Attachment 2 | Color Attachment 3 | Depth Attachment | Stencil Attachment

54 [drawIndexedPrimitives:3 indexCou...xBufferOffset:164544 instanceCount:1]

- **"GBuffer Render" (0x10042ee90)** gBufferVert/gBufferFrag
- **RenderPipeline Performance** 5.87 ms (38.7%)
- **Fragment Texture 0** 0x100461100 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 1** 0x1003d0f80 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 2** 0x10045fd70 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **Index Buffer** Index Buffer 0x100505580, Offset=0x000282c0
- **"G-buffer and lighting" (0x1003e0660)**
- **Encoder Statistics** 267 draw calls, 111MB load/store bandwidth
  - **Draw Calls** = 267
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 63MB
- **Frame Statistics** 271 draw calls, 127MB load/store bandwidth
  - **Frame Rate** = 59.9fps
  - **Draw Calls** = 271
  - **Command Buffers** = 1
  - **Render Encoders** = 3
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 79MB

- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **RenderPassDescriptor** *1536x2048 Color(x4)/Depth/Stencil*
  - **Visibility Result Buffer 0x0** Does not exist
  - **Color Attachment 0** *Load/Store Texture 0x1003bfe40*
    - **Texture 0x1003bfe40** 1536 x 2048, BGRA8Unorm
    - **TextureLevel** = 0
    - **TextureSlice** = 0
    - **TextureDepthPlane** = 0
    - **Resolve Texture 0x0** Does not exist
    - **ResolveTextureLevel** = 0
    - **ResolveTextureSlice** = 0
    - **ResolveTextureDepthPlane** = 0
    - **LoadAction** = Load
    - **StoreAction** = Store
    - **ClearColorRed** = 0.82
    - **ClearColorGreen** = 0.73
    - **ClearColorBlue** = 0.64
    - **ClearColorAlpha** = 1.00
  - **Color Attachment 1** *Load/Store Texture 0x1003c9b10*
  - **Color Attachment 2** *Load/Store Texture 0x1003a2640*
  - **Color Attachment 3** *Load/Store Texture 0x1003d1bd0*
  - **Depth Attachment** *Clear/Store Texture 0x1003cd0a0*
  - **Stencil Attachment** *Clear/Store Texture 0x1003c68b0*

Auto | MTL Command Encoder

uffer:0... 4.52 ms

uffer:0x... 1.35 ms

r:0x100... 0.00 ns

4.76 ms

1.76 ms

67.21 µs

578.01 µs

Color Attachment 0 | Color Attachment 1 | Color Attachment 2 | Color Attachment 3

54 [drawIndexedPrimitives:3 indexCou...xBufferOffset:164544 instanceCount:1]

- **"GBuffer Render" (0x10042ee90)** gBufferVert/gBufferFrag
- **RenderPipeline Performance** 5.87 ms (38.7%)
- **Fragment Texture 0** 0x100461100 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 1** 0x1003d0f80 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 2** 0x10045fd70 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **Index Buffer** Index Buffer 0x100505580, Offset=0x000282c0
- **"G-buffer and lighting" (0x1003e0660)**
- **Encoder Statistics** 267 draw calls, 111MB load/store bandwidth
  - **Draw Calls** = 267
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 63MB
- **Frame Statistics** 271 draw calls, 127MB load/store bandwidth
  - **Frame Rate** = 59.9fps
  - **Draw Calls** = 271
  - **Command Buffers** = 1
  - **Render Encoders** = 3
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 79MB

Auto

- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **RenderPassDescriptor** *1536x2048 Color(x4)/Depth/Stencil*
  - **Visibility Result Buffer 0x0** Does not exist
  - **Color Attachment 0** *Load/Store Texture 0x1003bfe40*
    - **Texture 0x1003bfe40** 1536 x 2048, BGRA8Unorm
    - **TextureLevel** = 0
    - **TextureSlice** = 0
    - **TextureDepthPlane** = 0
    - **Resolve Texture 0x0** Does not exist
    - **ResolveTextureLevel** = 0
    - **ResolveTextureSlice** = 0
    - **ResolveTextureDepthPlane** = 0
    - **LoadAction** = Load
    - **StoreAction** = Store
    - **ClearColorRed** = 0.82
    - **ClearColorGreen** = 0.73
    - **ClearColorBlue** = 0.64
    - **ClearColorAlpha** = 1.00
  - **Color Attachment 1** *Load/Store Texture 0x1003c9b10*
  - **Color Attachment 2** *Load/Store Texture 0x1003a2640*
  - **Color Attachment 3** *Load/Store Texture 0x1003d1bd0*
  - **Depth Attachment** *Clear/Store Texture 0x1003cd0a0*
  - **Stencil Attachment** *Clear/Store Texture 0x1003c68b0*

MTL Command Encoder

54 [drawIndexedPrimitives:3 indexCou...xBufferOffset:164544 instanceCount:1]

- **"GBuffer Render" (0x10042ee90)** gBufferVert/gBufferFrag
- **RenderPipeline Performance** 5.87 ms (38.7%)
- **Fragment Texture 0** 0x100461100 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 1** 0x1003d0f80 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 2** 0x10045fd70 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **Index Buffer** Index Buffer 0x100505580, Offset=0x000282c0
- **"G-buffer and lighting" (0x1003e0660)**
- **Encoder Statistics** 267 draw calls, 111MB load/store bandwidth
  - **Draw Calls** = 267
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 63MB
- **Frame Statistics** 271 draw calls, 127MB load/store bandwidth
  - **Frame Rate** = 59.9fps
  - **Draw Calls** = 271
  - **Command Buffers** = 1
  - **Render Encoders** = 3
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 79MB

Auto

- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **RenderPassDescriptor** *1536x2048 Color(x4)/Depth/Stencil*
  - **Visibility Result Buffer 0x0** Does not exist
  - **Color Attachment 0** *Load/Store Texture 0x1003bfe40*
    - **Texture 0x1003bfe40** 1536 x 2048, BGRA8Unorm
    - **TextureLevel** = 0
    - **TextureSlice** = 0
    - **TextureDepthPlane** = 0
    - **Resolve Texture 0x0** Does not exist
    - **ResolveTextureLevel** = 0
    - **ResolveTextureSlice** = 0
    - **ResolveTextureDepthPlane** = 0
    - **LoadAction** = Load
    - **StoreAction** = Store
    - **ClearColorRed** = 0.82
    - **ClearColorGreen** = 0.73
    - **ClearColorBlue** = 0.64
    - **ClearColorAlpha** = 1.00
  - **Color Attachment 1** *Load/Store Texture 0x1003c9b10*
  - **Color Attachment 2** *Load/Store Texture 0x1003a2640*
  - **Color Attachment 3** *Load/Store Texture 0x1003d1bd0*
  - **Depth Attachment** *Clear/Store Texture 0x1003cd0a0*
  - **Stencil Attachment** *Clear/Store Texture 0x1003c68b0*

MTL Command Encoder

54 [drawIndexedPrimitives:3 indexCou...xBufferOffset:164544 instanceCount:1]

- **"GBuffer Render" (0x10042ee90)** gBufferVert/gBufferFrag
- **RenderPipeline Performance** 5.87 ms (38.7%)
- **Fragment Texture 0** 0x100461100 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 1** 0x1003d0f80 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 2** 0x10045fd70 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **Index Buffer** Index Buffer 0x100505580, Offset=0x000282c0
- **"G-buffer and lighting" (0x1003e0660)**
- **Encoder Statistics** 267 draw calls, 111MB load/store bandwidth
  - **Draw Calls** = 267
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 63MB
- **Frame Statistics** 271 draw calls, 127MB load/store bandwidth
  - **Frame Rate** = 59.9fps
  - **Draw Calls** = 271
  - **Command Buffers** = 1
  - **Render Encoders** = 3
  - **Load Bandwidth** = 48MB
  - **Store Bandwidth** = 79MB

Auto

- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **RenderPassDescriptor** *1536x2048 Color(x4)/Depth/Stencil*
  - **Visibility Result Buffer 0x0** Does not exist
  - **Color Attachment 0** *Load/Store Texture 0x1003bfe40*
    - **Texture 0x1003bfe40** 1536 x 2048, BGRA8Unorm
    - **TextureLevel** = 0
    - **TextureSlice** = 0
    - **TextureDepthPlane** = 0
    - **Resolve Texture 0x0** Does not exist
    - **ResolveTextureLevel** = 0
    - **ResolveTextureSlice** = 0
    - **ResolveTextureDepthPlane** = 0
    - **LoadAction** = Load
    - **StoreAction** = Store
    - **ClearColorRed** = 0.82
    - **ClearColorGreen** = 0.73
    - **ClearColorBlue** = 0.64
    - **ClearColorAlpha** = 1.00
  - **Color Attachment 1** *Load/Store Texture 0x1003c9b10*
  - **Color Attachment 2** *Load/Store Texture 0x1003a2640*
  - **Color Attachment 3** *Load/Store Texture 0x1003d1bd0*
  - **Depth Attachment** *Clear/Store Texture 0x1003cd0a0*
  - **Stencil Attachment** *Clear/Store Texture 0x1003c68b0*

MTL Command Encoder

54 [drawIndexedPrimitives:3 indexCou...xBufferOffset:164544 instanceCount:1]

- **"GBuffer Render" (0x10042ee90)** gBufferVert/gBufferFrag
- **RenderPipeline Performance** 5.87 ms (38.7%)
- **Fragment Texture 0** 0x100461100 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 1** 0x1003d0f80 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 2** 0x10045fd70 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **Index Buffer** Index Buffer 0x100505580, Offset=0x000282c0
- **"G-buffer and lighting" (0x1003e0660)**
- **Encoder Statistics** 267 draw calls, 111MB load/store bandwidth

Load Bandwidth = 48MB

Store Bandwidth = 63MB

Frame Rate = 59.9fps
Draw Calls = 271
Command Buffers = 1
Render Encoders = 3
Load Bandwidth = 48MB
Store Bandwidth = 79MB

Auto

- **Fragment Texture 3** 0x10042aaa0 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x10050d780
- **Vertex Buffer 0** 0x100504c10
- **Vertex Buffer 1** 0x100513c40
- **RenderPassDescriptor** *1536x2048 Color(x4)/Depth/Stencil*
  - **Visibility Result Buffer 0x0** Does not exist
  - **Color Attachment 0** *Load/Store Texture 0x1003bfe40*
    - **Texture 0x1003bfe40** 1536 x 2048, BGRA8Unorm
    - **TextureLevel** = 0
    - **TextureSlice** = 0
    - **TextureDepthPlane** = 0
    - **Resolve Texture 0x0** Does not exist

LoadAction = Load

StoreAction = Store

StoreAction = Store
ClearColorRed = 0.82
ClearColorGreen = 0.73
ClearColorBlue = 0.78
ClearColorAlpha = 1

1 *Load/Store Textu*
2 *Load/Store Text*
3 *Load/Store Text*
Clear/Store Textu
t *Clear/Store Text*

- **Color Attachm** *x1003c9b10*
- **Color Attachm** *x1003a2640*
- **Color Attachm** *x1003d1bd0*
- **Depth Attachn** *x1003cd0a0*
- **Stencil Attach** *x1003c68b0*

MTL Command Encoder

54 [drawIndexedPrimitives:3 indexCou...xBufferOffset:164544 instanceCount:1]

- **"GBuffer Render" (0x100438eb0)** gBufferVert/gBufferFrag
- **RenderPipeline Performance** 5.69 ms (38.8%)
- **Fragment Texture 0** 0x1003662c0 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 1** 0x1004461a0 - 2048 x 2048, RGBA8Unorm
- **Fragment Texture 2** 0x100446880 - 2048 x 2048, RGBA8Uno...
- **Fragment Texture 3** 0x100434580 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x100443600
- **Vertex Buffer 0** 0x1018738c0
- **Vertex Buffer 1** 0x101879de0
- **Index Buffer** Index Buffer 0x1018766b0, Offset=0x000282c0
- **"G-buffer and lighting" (0x10035c9e0)**
- **Encoder Statistics** 267 draw calls, 12MB load/store bandwidth

Load Bandwidth = 0B

Store Bandwidth = 12MB

Frame Rate = 60.0fps
Draw Calls = 271
Command Buffers = 1
Render Encoders = 3
Load Bandwidth = 0B
Store Bandwidth = 28MB

Auto

- **Fragment Texture 3** 0x100434580 - 1024 x 1024, Depth32Float
- **Fragment Buffer 0** 0x100443600
- **Vertex Buffer 0** 0x1018738c0
- **Vertex Buffer 1** 0x101879de0
- **RenderPassDescriptor** *1536x2048 Color(x4)/Depth/Stencil*
  - **Visibility Result Buffer 0x0** Does not exist
  - **Color Attachment 0** *Clear/Store Texture 0x1004342f0*
    - **Texture 0x1004342f0** 1536 x 2048, BGRA8Unorm
    - **TextureLevel** = 0
    - **TextureSlice** = 0
    - **TextureDepthPlane** = 0
    - **Resolve Texture 0x0** Does not exist

LoadAction = Clear

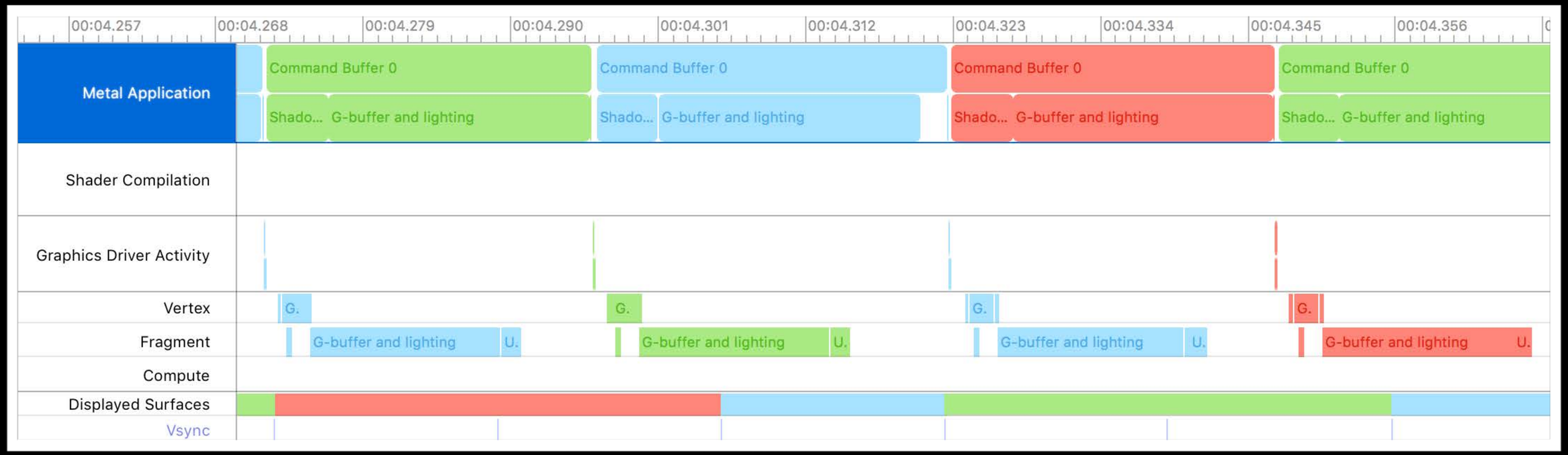
StoreAction = Store

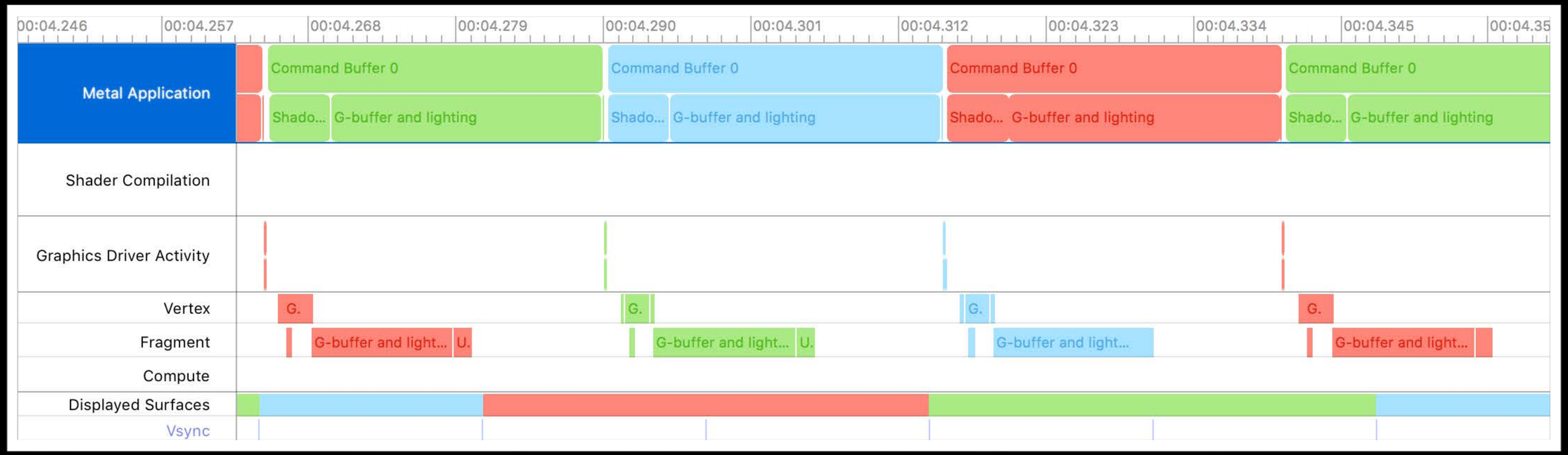
StoreAction = Store
ClearColorRed = 0.82
ClearColor
ClearColor
ClearColor

1 *Clear/DontCare*
2 *Clear/DontCare*
3 *Clear/DontCare*
Clear/DontCare T
t *Clear/DontCare*

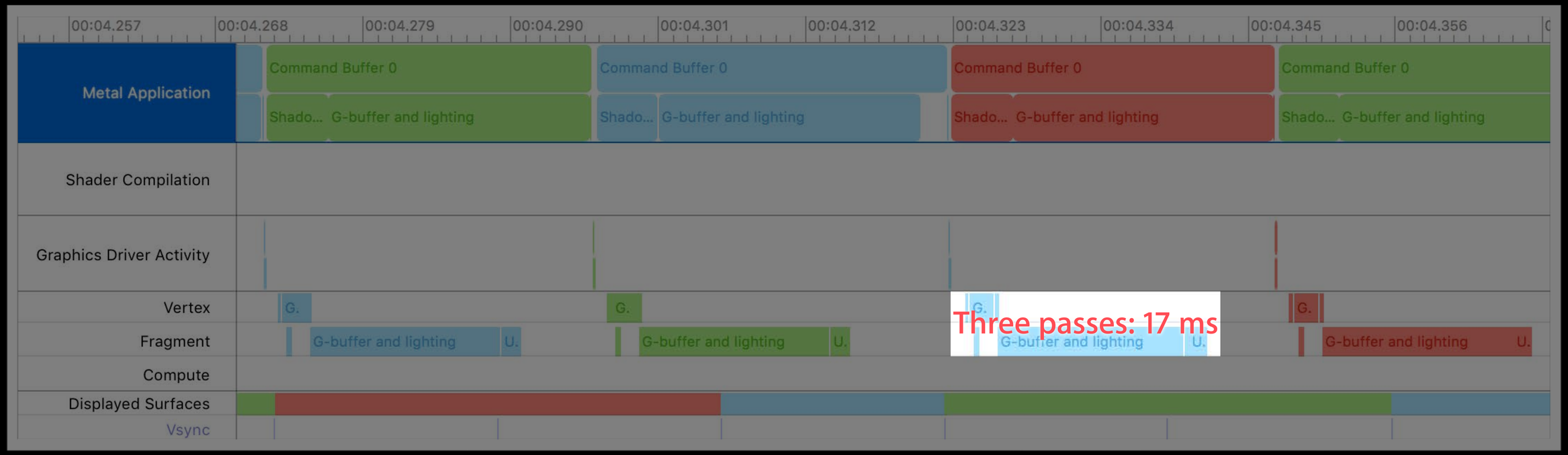
- **Color Attachm** *ure 0x10044aa90*
- **Color Attachm** *ure 0x10044ad20*
- **Color Attachm** *ure 0x10034b990*
- **Depth Attachn** *re 0x100336d30*
- **Stencil Attach** *ure 0x10035b040*

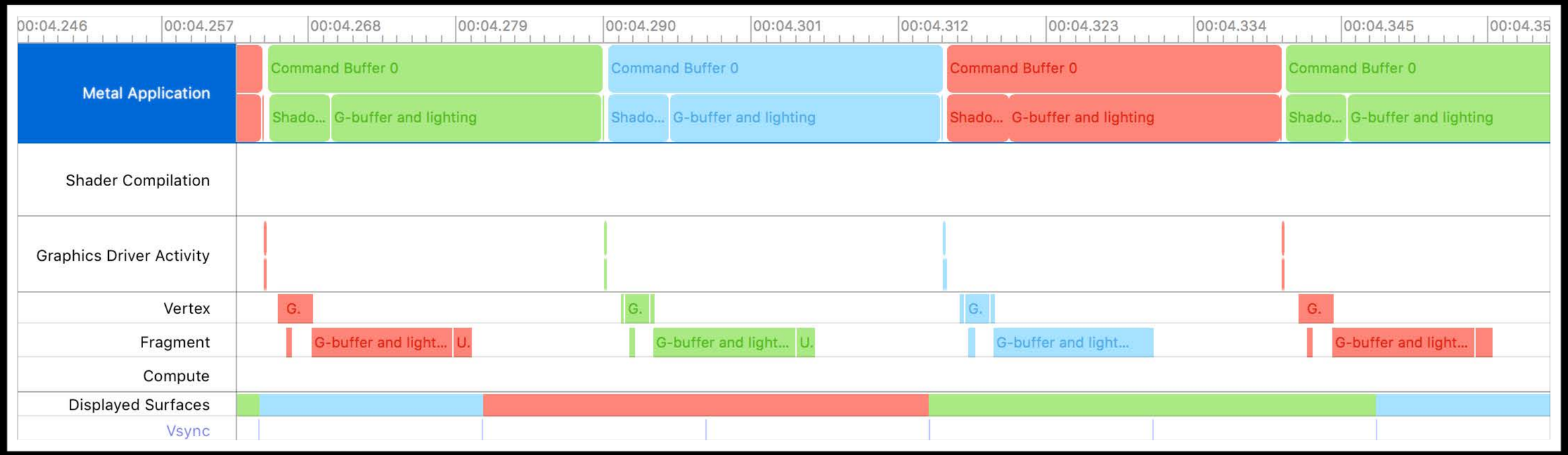
MTL Command Encoder


00:04.257
00:04.268
00:04.279
00:04.290
00:04.301
00:04.312
00:04.323
00:04.334
00:04.345
00:04.356
Metal Application
Command Buffer 0
Command Buffer 0
Command Buffer 0
Command Buffer 0
Shado...
G-buffer and lighting
Shado...
G-buffer and lighting
Shado...
G-buffer and lighting
Shado...
G-buffer and lighting
Shader Compilation
Graphics Driver Activity
Vertex
G.
G.
G.
G.
Fragment
G-buffer and lighting
U.
G-buffer and lighting
U.
G-buffer and lighting
U.
G-buffer and lighting
U.
Compute
Displayed Surfaces
Vsync


00:04.246
00:04.257
00:04.268
00:04.279
00:04.290
00:04.301
00:04.312
00:04.323
00:04.334
00:04.345
00:04.35
Metal Application
Command Buffer 0
Command Buffer 0
Command Buffer 0
Command Buffer 0
Shado...
G-buffer and lighting
Shado...
G-buffer and lighting
Shado...
G-buffer and lighting
Shado...
G-buffer and lighting
Shader Compilation
Graphics Driver Activity
Vertex
G.
G.
G.
G.
Fragment
G-buffer and light...
U.
G-buffer and light...
U.
G-buffer and light...
G-buffer and light...
Compute
Displayed Surfaces
Vsync

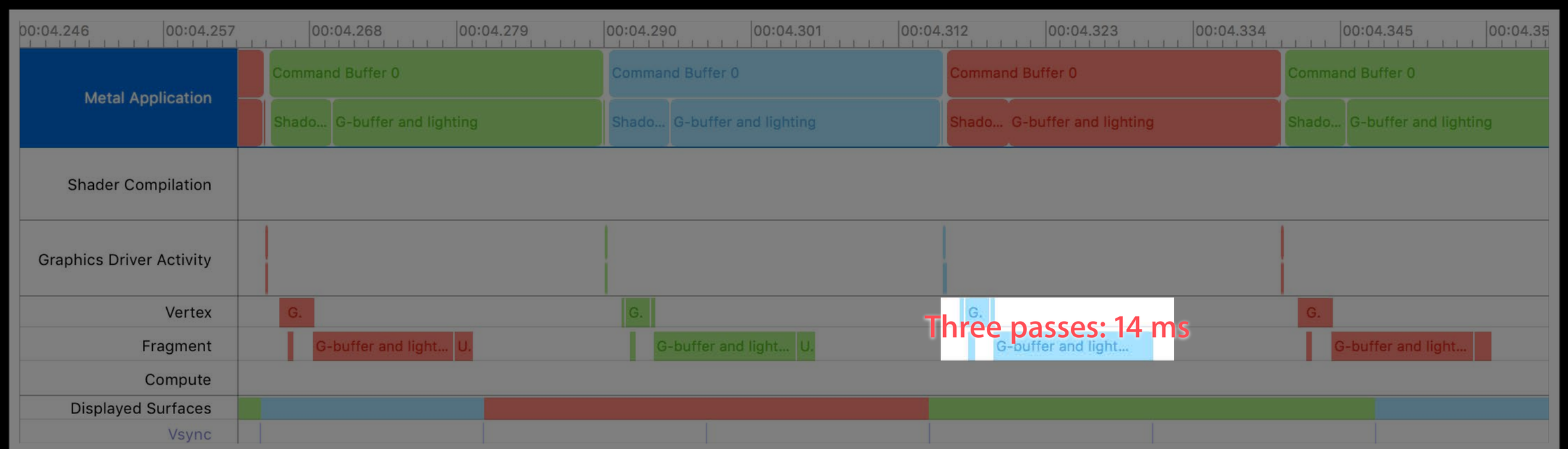
Before


00:04.257
00:04.268
00:04.279
00:04.290
00:04.301
00:04.312
00:04.323
00:04.334
00:04.345
00:04.356
Metal Application
Command Buffer 0
Shado...
G-buffer and lighting
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
G.
U.
Three passes: 17 ms


00:04.246
00:04.257
00:04.268
00:04.279
00:04.290
00:04.301
00:04.312
00:04.323
00:04.334
00:04.345
00:04.35
Metal Application
Command Buffer 0
Shado...
G-buffer and lighting
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
G.
G-buffer and light...
U.

Before

After



Metal Optimization Best Practices

Don't waste render encoders

- Don't start encoding if there's nothing to draw
- Merge encoders that render to the same attachments

5

Consider Multithreading if Still CPU-Bound

Metal Multithreading

Metal Multithreading

Metal designed to facilitate multithreading

Metal Multithreading

Metal designed to facilitate multithreading

Thread-safe, efficient, and scalable implementation

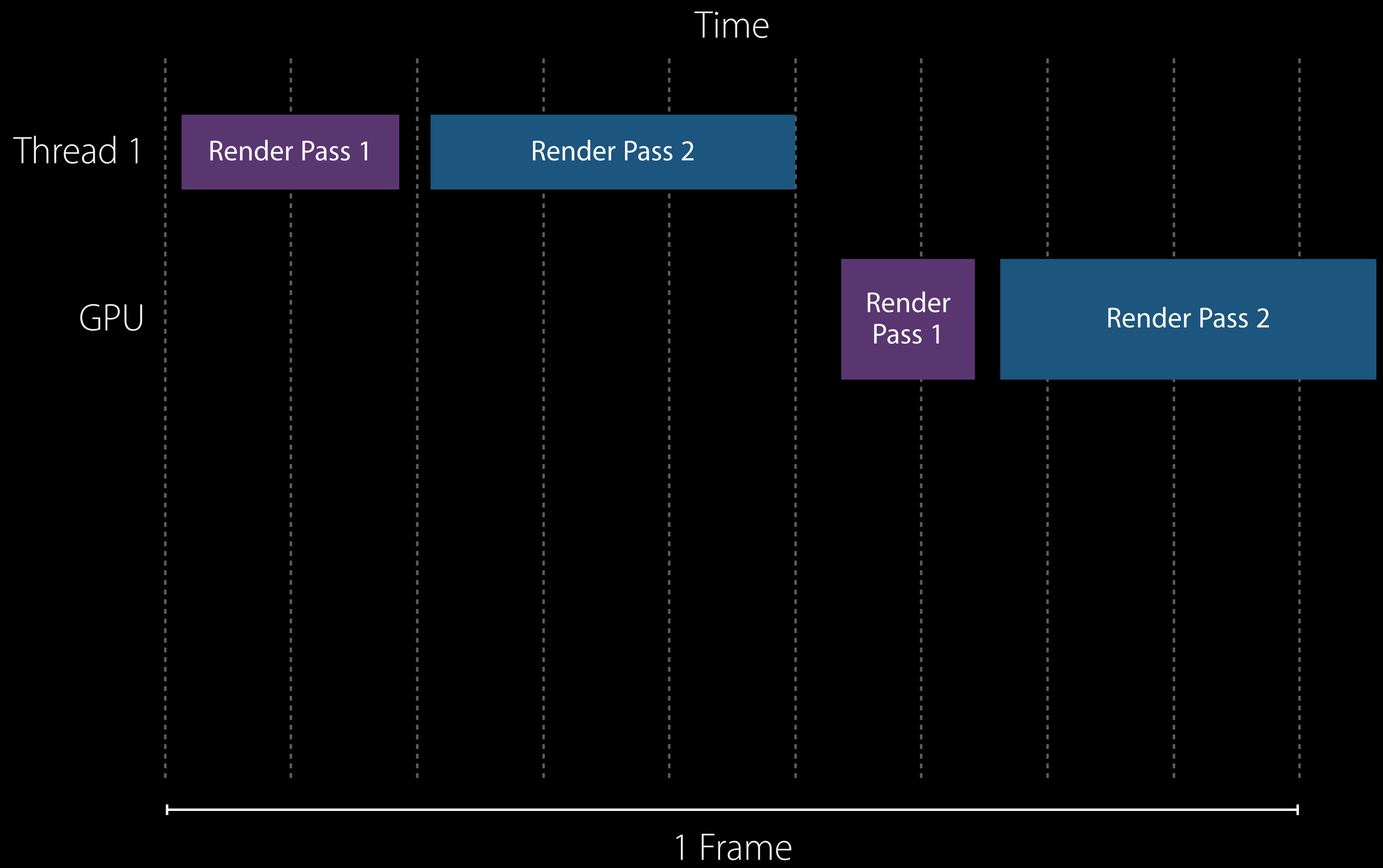
Metal Multithreading

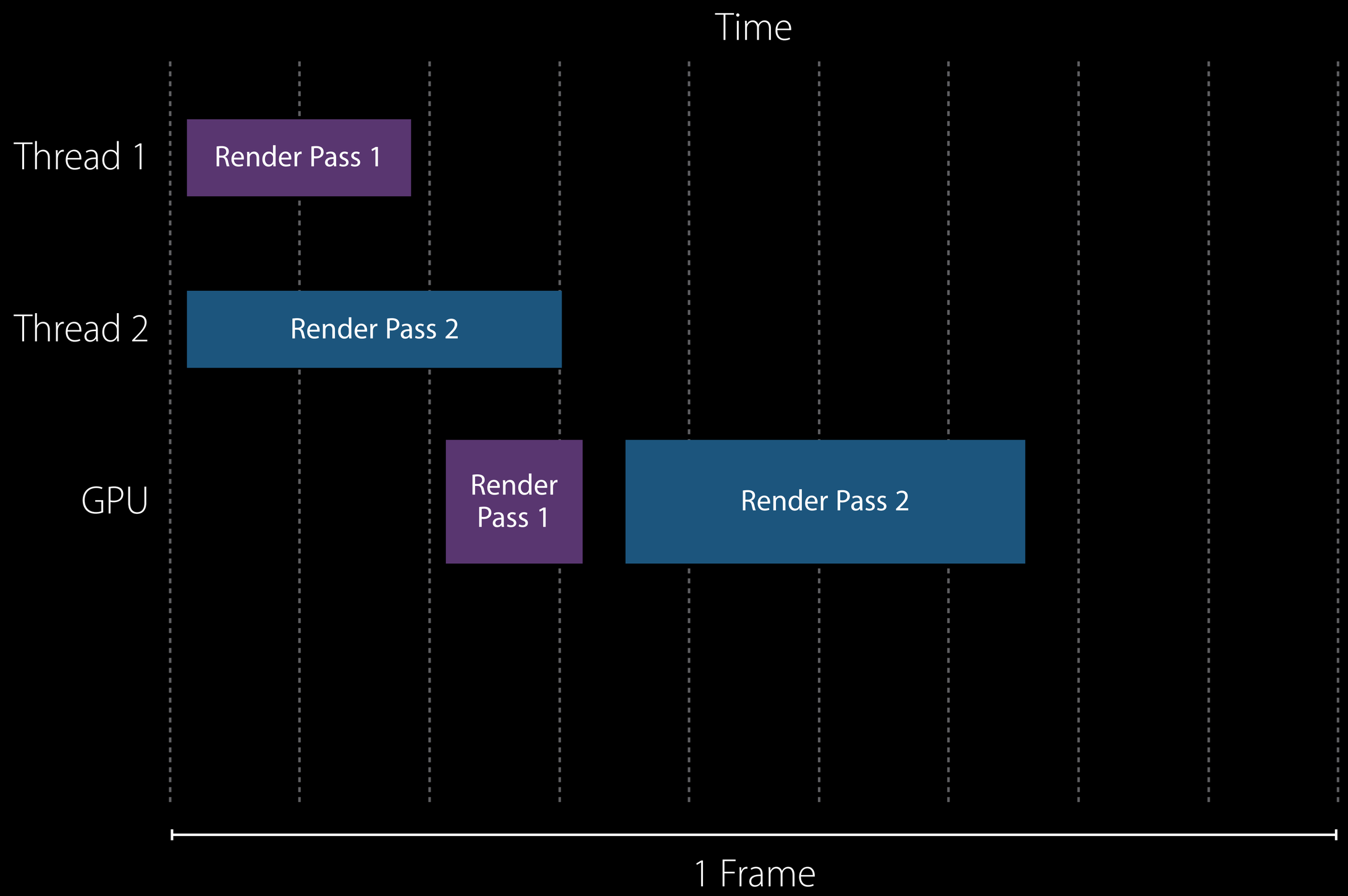
Metal designed to facilitate multithreading

Thread-safe, efficient, and scalable implementation

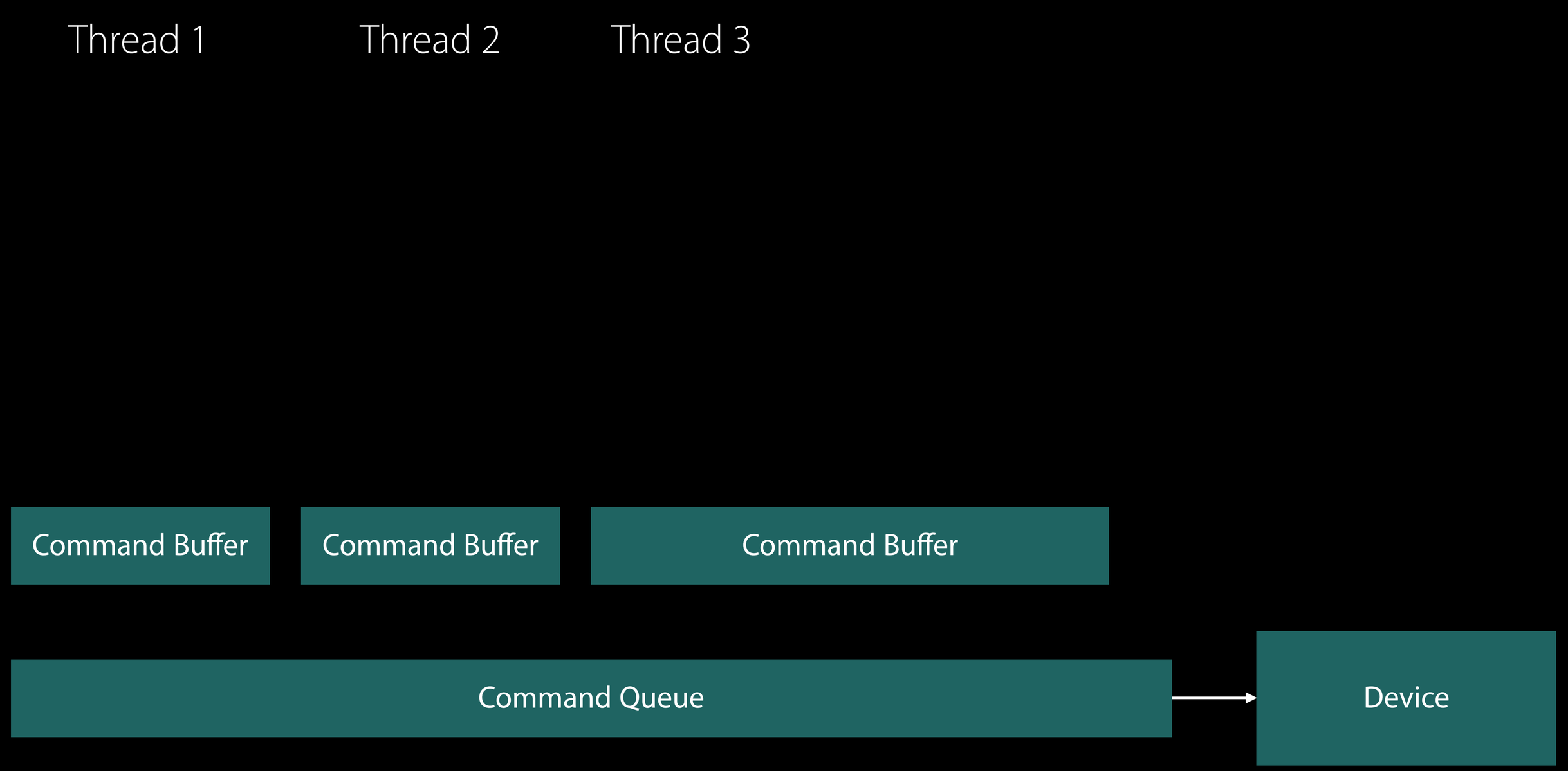
Multiple command buffers can be encoded in parallel

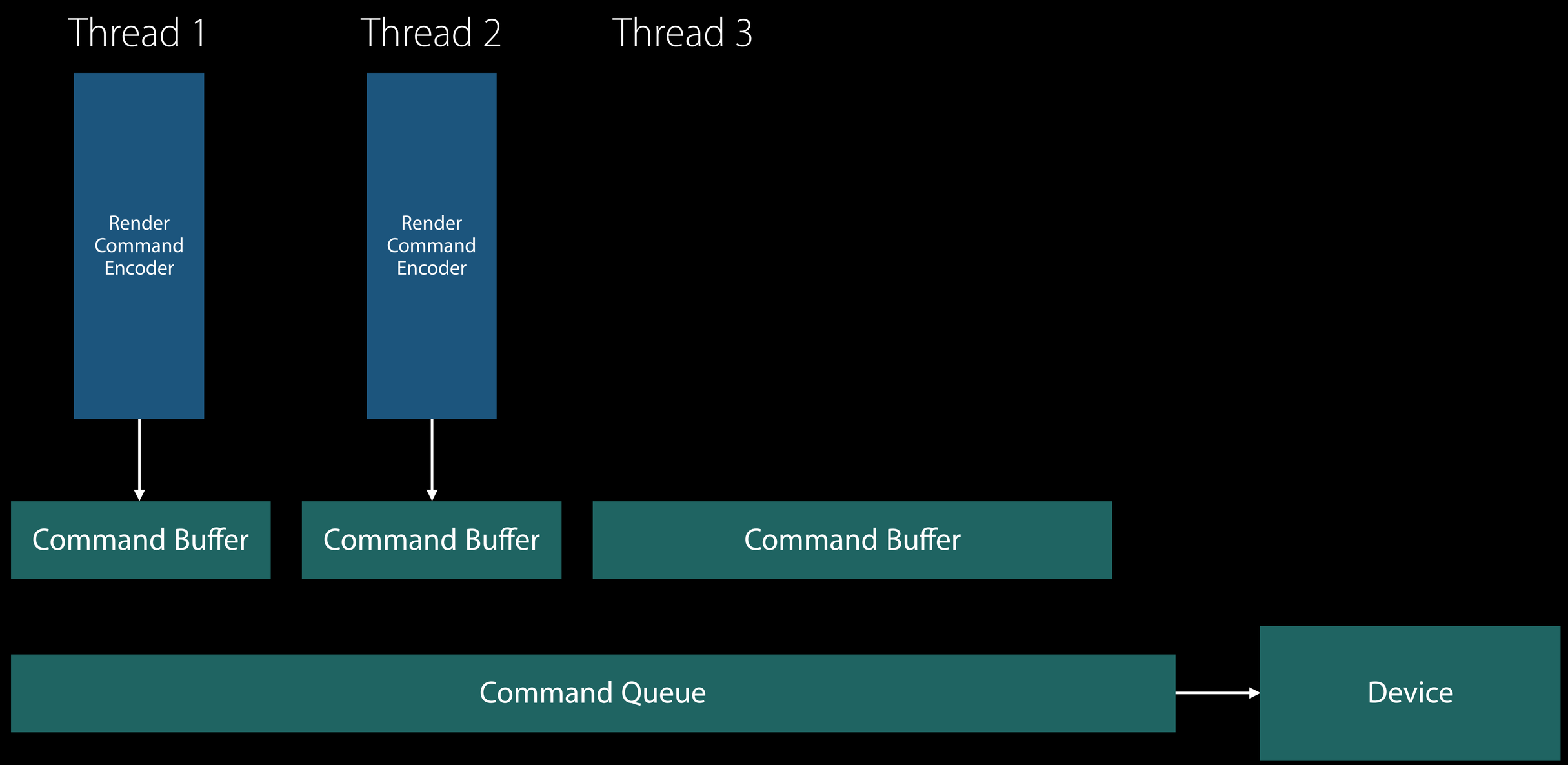
- App controls execution order

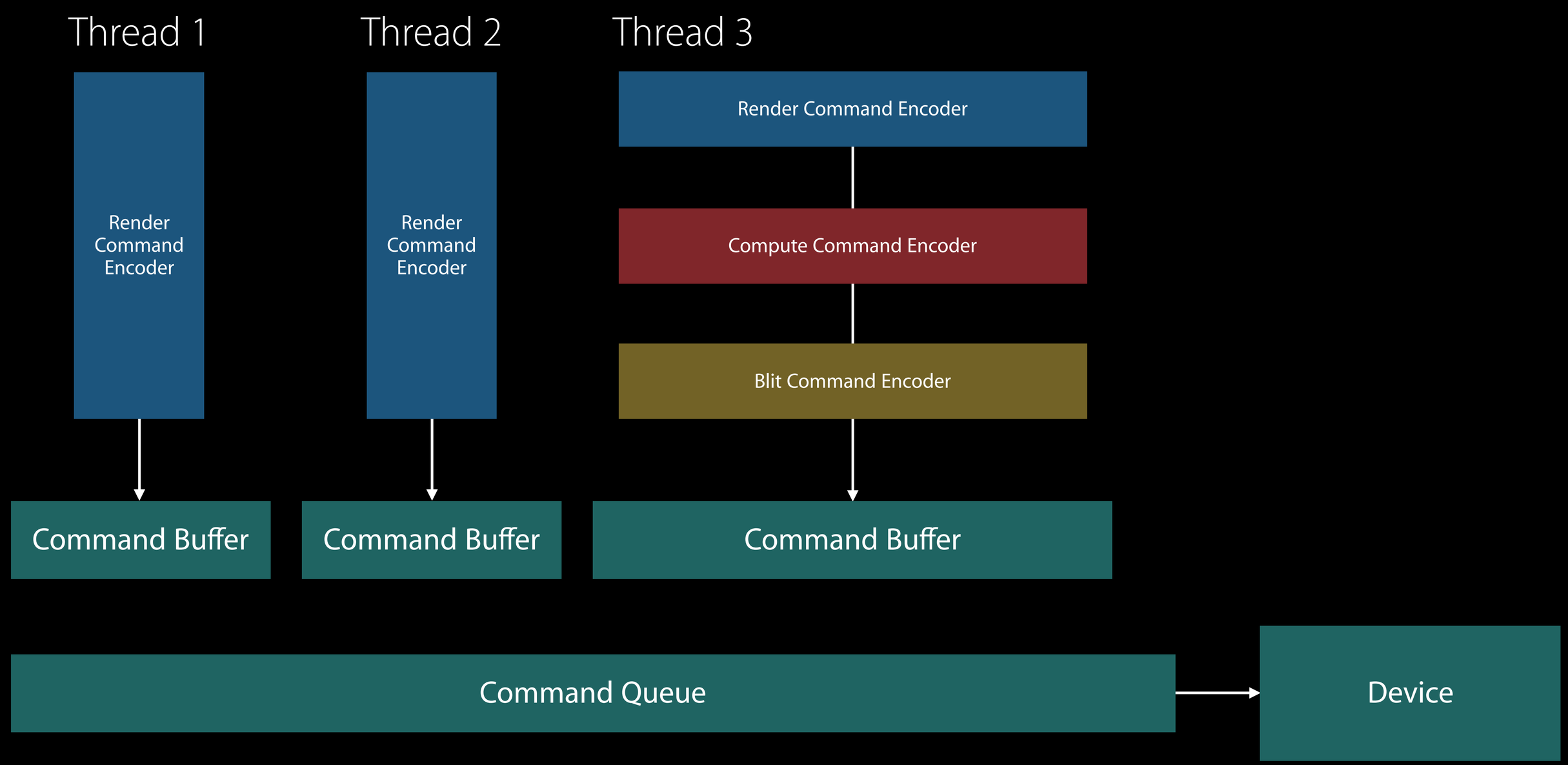

Time
Thread 1
Render Pass 1
Render Pass 2
GPU
Render Pass 1
Render Pass 2
1 Frame

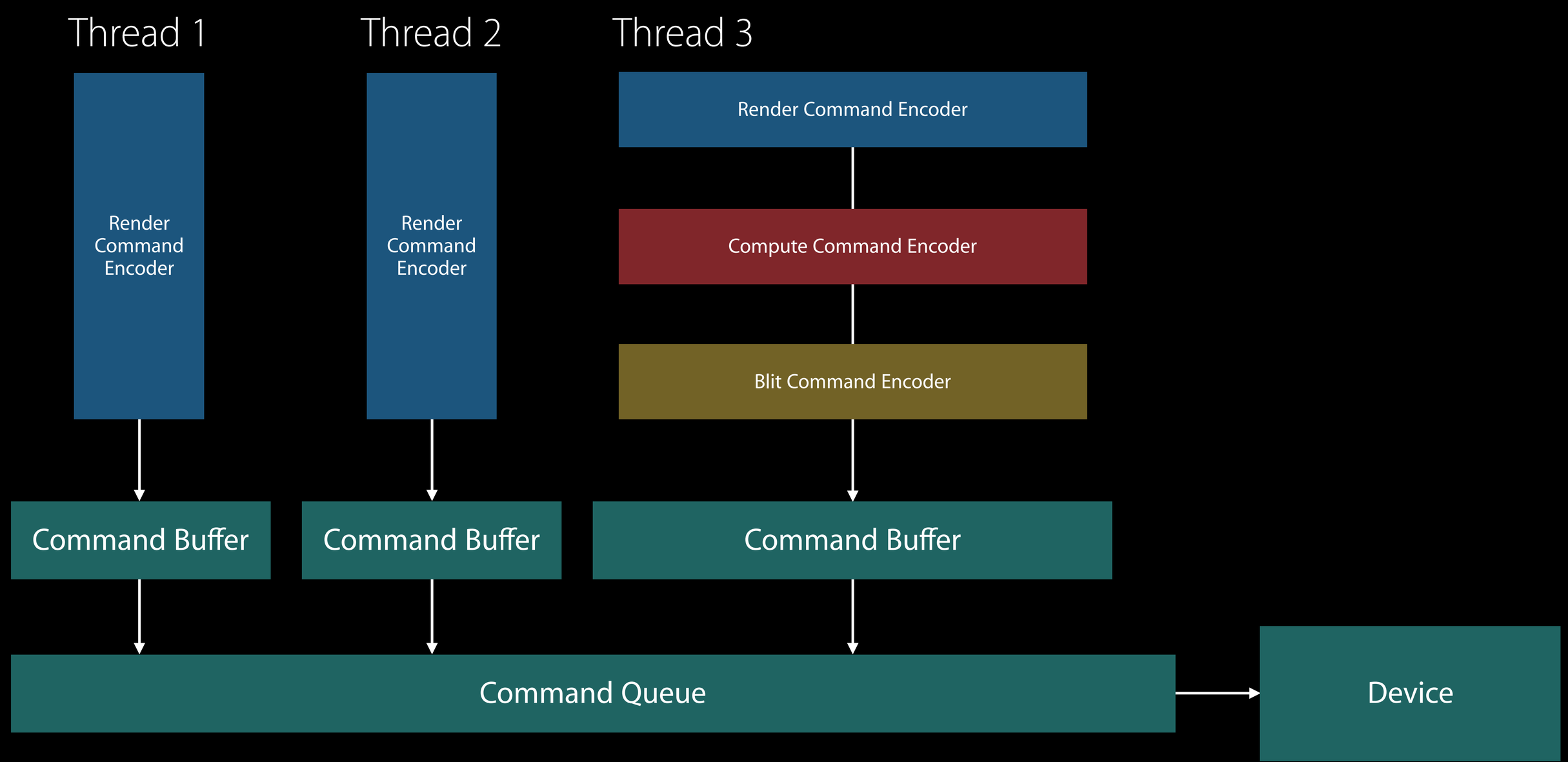

Time
Thread 1
Render Pass 1
Thread 2
Render Pass 2
GPU
Render Pass 1
Render Pass 2
1 Frame


Command Queue
Device


Thread 1
Thread 2
Thread 3
Command Buffer
Command Buffer
Command Buffer
Command Queue
Device


Thread 1
Thread 2
Thread 3
Render Command Encoder
Render Command Encoder
Command Buffer
Command Buffer
Command Buffer
Command Queue
Device


Thread 1
Thread 2
Thread 3
Render Command Encoder
Render Command Encoder
Render Command Encoder
Compute Command Encoder
Blit Command Encoder
Command Buffer
Command Buffer
Command Buffer
Command Queue
Device


Thread 1
Thread 2
Thread 3
Render Command Encoder
Render Command Encoder
Render Command Encoder
Compute Command Encoder
Blit Command Encoder
Command Buffer
Command Buffer
Command Buffer
Command Queue
Device

Setting Up Threaded Passes

```
// create one command buffer per thread
id <MTLCommandBuffer> commandBuffer1 = [commandQueue commandBuffer];
id <MTLCommandBuffer> commandBuffer2 = [commandQueue commandBuffer];

// create and initialize render pass descriptors
// reserve each buffer a place in command queue to enforce GPU submission order
[commandBuffer1 enqueue];
[commandBuffer2 enqueue];

// create per-thread command encoders
id <MTLRenderCommandEncoder> pass1RCE =
   [commandBuffer1 renderCommandEncoderWithDescriptor:renderPass1Desc];
id <MTLRenderCommandEncoder> pass2RCE =
   [commandBuffer2 renderCommandEncoderWithDescriptor:renderPass2Desc];

// encode commands and commit buffer for each thread
[pass1RCE draw...];          [pass2RCE draw...];
[pass1RCE endEncoding];      [pass2RCE endEncoding];
[commandBuffer1 commit];     [commandBuffer2 commit];
```

Setting Up Threaded Passes

```
// create one command buffer per thread
id <MTLCommandBuffer> commandBuffer1 = [commandQueue commandBuffer];
id <MTLCommandBuffer> commandBuffer2 = [commandQueue commandBuffer];
```

```
// create and initialize render pass descriptors
// reserve each buffer a place in command queue to enforce GPU submission order
[commandBuffer1 enqueue];
[commandBuffer2 enqueue];

// create per-thread command encoders
id <MTLRenderCommandEncoder> pass1RCE =
   [commandBuffer1 renderCommandEncoderWithDescriptor:renderPass1Desc];
id <MTLRenderCommandEncoder> pass2RCE =
   [commandBuffer2 renderCommandEncoderWithDescriptor:renderPass2Desc];

// encode commands and commit buffer for each thread
[pass1RCE draw...];          [pass2RCE draw...];
[pass1RCE endEncoding];      [pass2RCE endEncoding];
[commandBuffer1 commit];     [commandBuffer2 commit];
```

Setting Up Threaded Passes

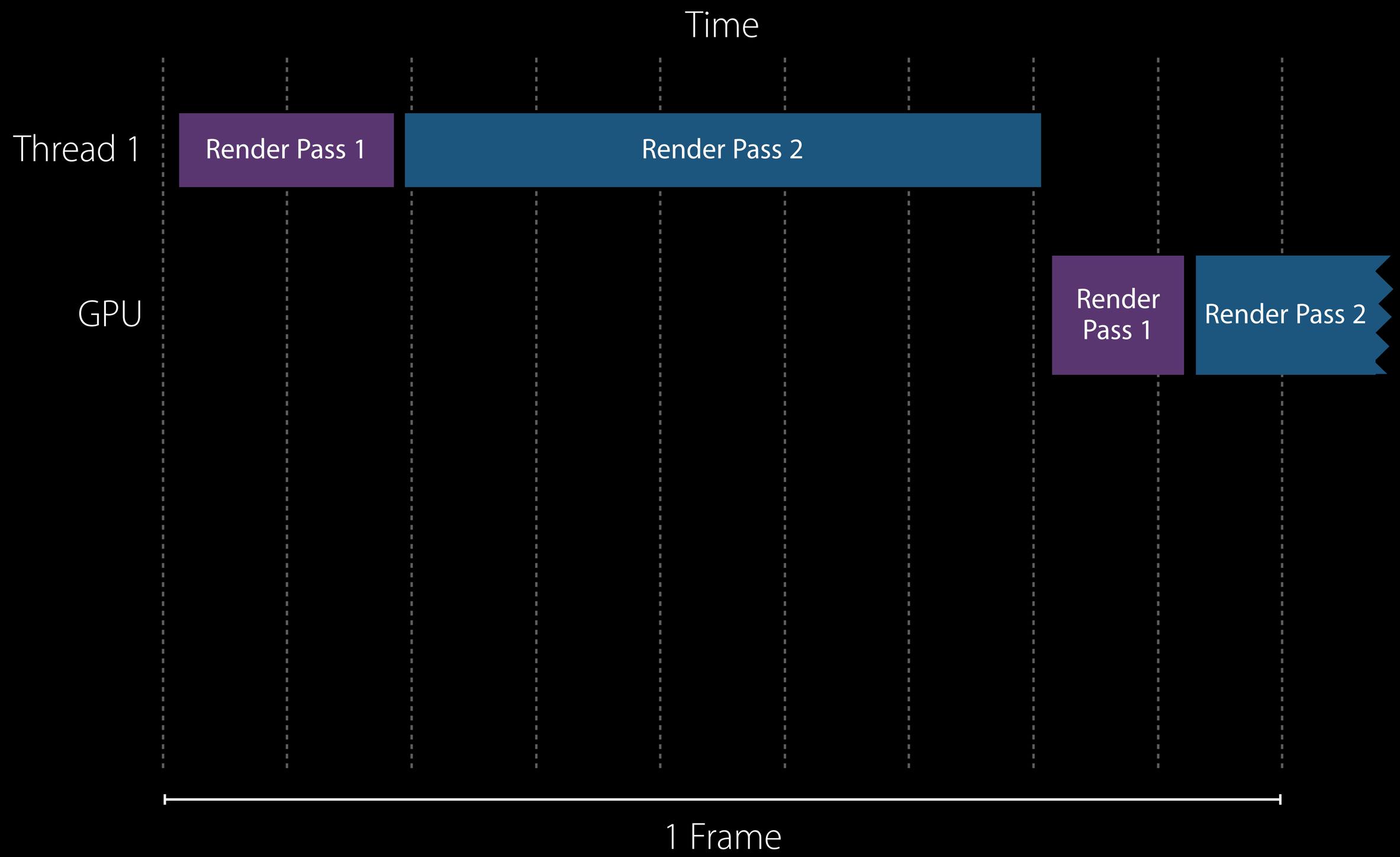
```
// create one command buffer per thread
id <MTLCommandBuffer> commandBuffer1 = [commandQueue commandBuffer];
id <MTLCommandBuffer> commandBuffer2 = [commandQueue commandBuffer];

// create and initialize render pass descriptors
// reserve each buffer a place in command queue to enforce GPU submission order
[commandBuffer1 enqueue];
[commandBuffer2 enqueue];

// create per-thread command encoders
id <MTLRenderCommandEncoder> pass1RCE =
   [commandBuffer1 renderCommandEncoderWithDescriptor:renderPass1Desc];
id <MTLRenderCommandEncoder> pass2RCE =
   [commandBuffer2 renderCommandEncoderWithDescriptor:renderPass2Desc];

// encode commands and commit buffer for each thread
[pass1RCE draw...];          [pass2RCE draw...];
[pass1RCE endEncoding];      [pass2RCE endEncoding];
[commandBuffer1 commit];     [commandBuffer2 commit];
```

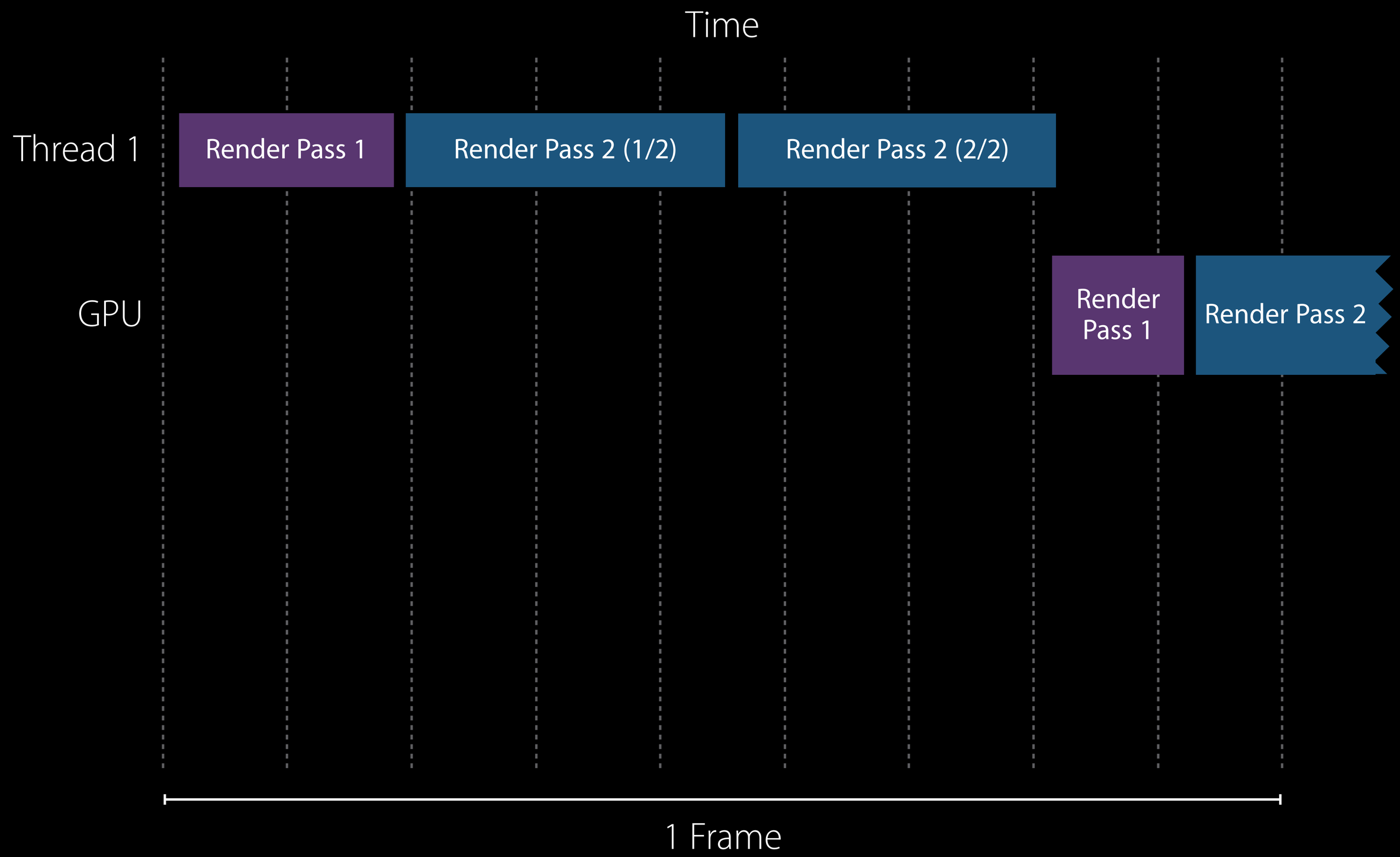
Setting Up Threaded Passes

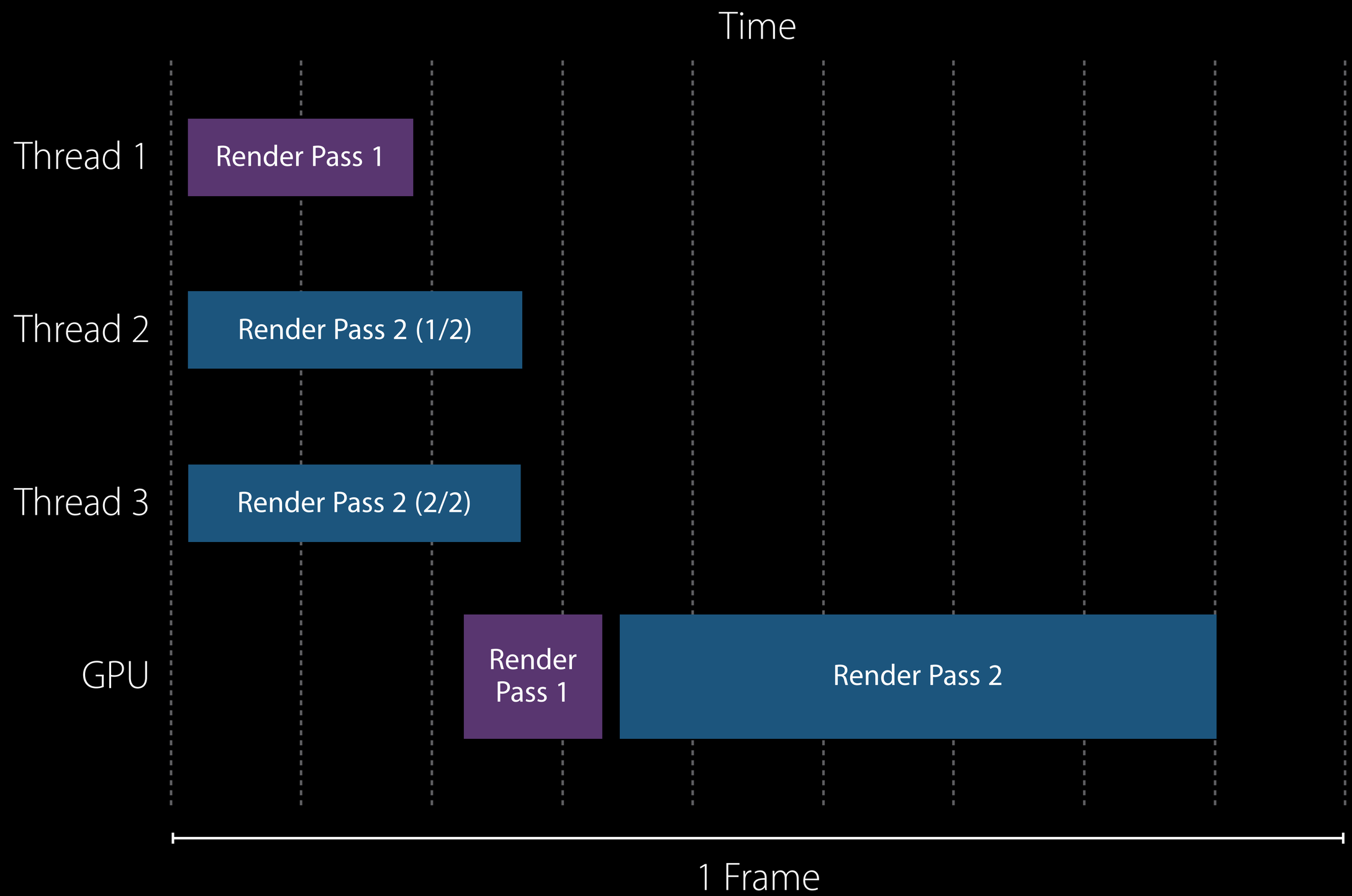
```
// create one command buffer per thread
id <MTLCommandBuffer> commandBuffer1 = [commandQueue commandBuffer];
id <MTLCommandBuffer> commandBuffer2 = [commandQueue commandBuffer];

// create and initialize render pass descriptors
// reserve each buffer a place in command queue to enforce GPU submission order
[commandBuffer1 enqueue];
[commandBuffer2 enqueue];

// create per-thread command encoders
id <MTLRenderCommandEncoder> pass1RCE =
   [commandBuffer1 renderCommandEncoderWithDescriptor:renderPass1Desc];
id <MTLRenderCommandEncoder> pass2RCE =
   [commandBuffer2 renderCommandEncoderWithDescriptor:renderPass2Desc];

// encode commands and commit buffer for each thread
[pass1RCE draw...];          [pass2RCE draw...];
[pass1RCE endEncoding];      [pass2RCE endEncoding];
[commandBuffer1 commit];     [commandBuffer2 commit];
```

Setting Up Threaded Passes

```
// create one command buffer per thread
id <MTLCommandBuffer> commandBuffer1 = [commandQueue commandBuffer];
id <MTLCommandBuffer> commandBuffer2 = [commandQueue commandBuffer];

// create and initialize render pass descriptors
// reserve each buffer a place in command queue to enforce GPU submission order
[commandBuffer1 enqueue];
[commandBuffer2 enqueue];

// create per-thread command encoders
id <MTLRenderCommandEncoder> pass1RCE =
   [commandBuffer1 renderCommandEncoderWithDescriptor:renderPass1Desc];
id <MTLRenderCommandEncoder> pass2RCE =
   [commandBuffer2 renderCommandEncoderWithDescriptor:renderPass2Desc];

// encode commands and commit buffer for each thread
[pass1RCE draw...];
[pass1RCE endEncoding];
[commandBuffer1 commit];

[pass2RCE draw...];
[pass2RCE endEncoding];
[commandBuffer2 commit];
```

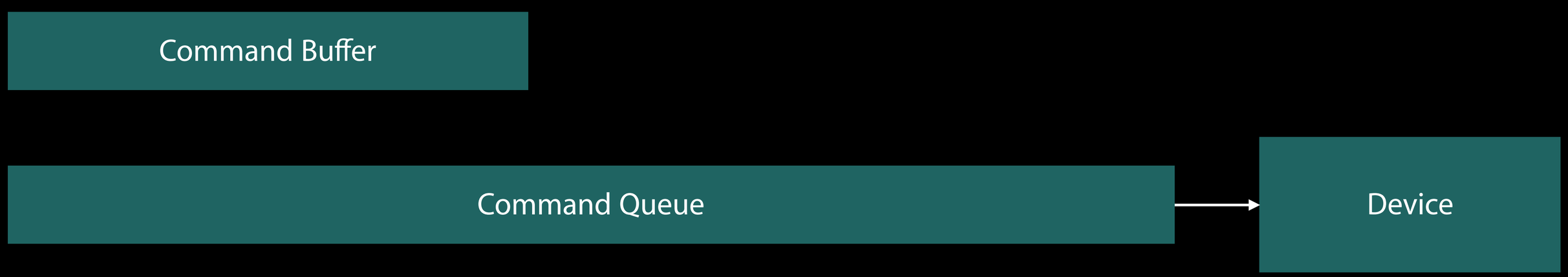

Time
Thread 1
Render Pass 1
Render Pass 2
GPU
Render Pass 1
Render Pass 2
1 Frame


Time
Thread 1
Render Pass 1
Render Pass 2 (1/2)
Render Pass 2 (2/2)
GPU
Render Pass 1
Render Pass 2
1 Frame


Time
Thread 1
Render Pass 1
Thread 2
Render Pass 2 (1/2)
Thread 3
Render Pass 2 (2/2)
GPU
Render Pass 1
Render Pass 2
1 Frame

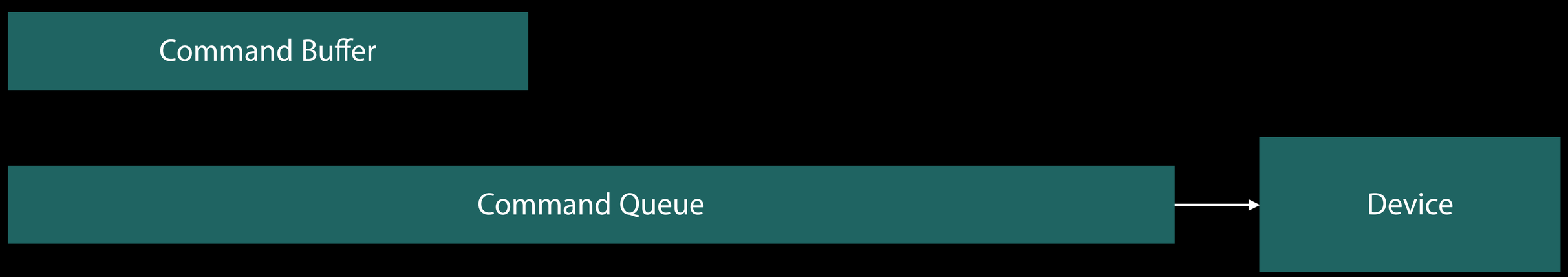

Command Queue
Device

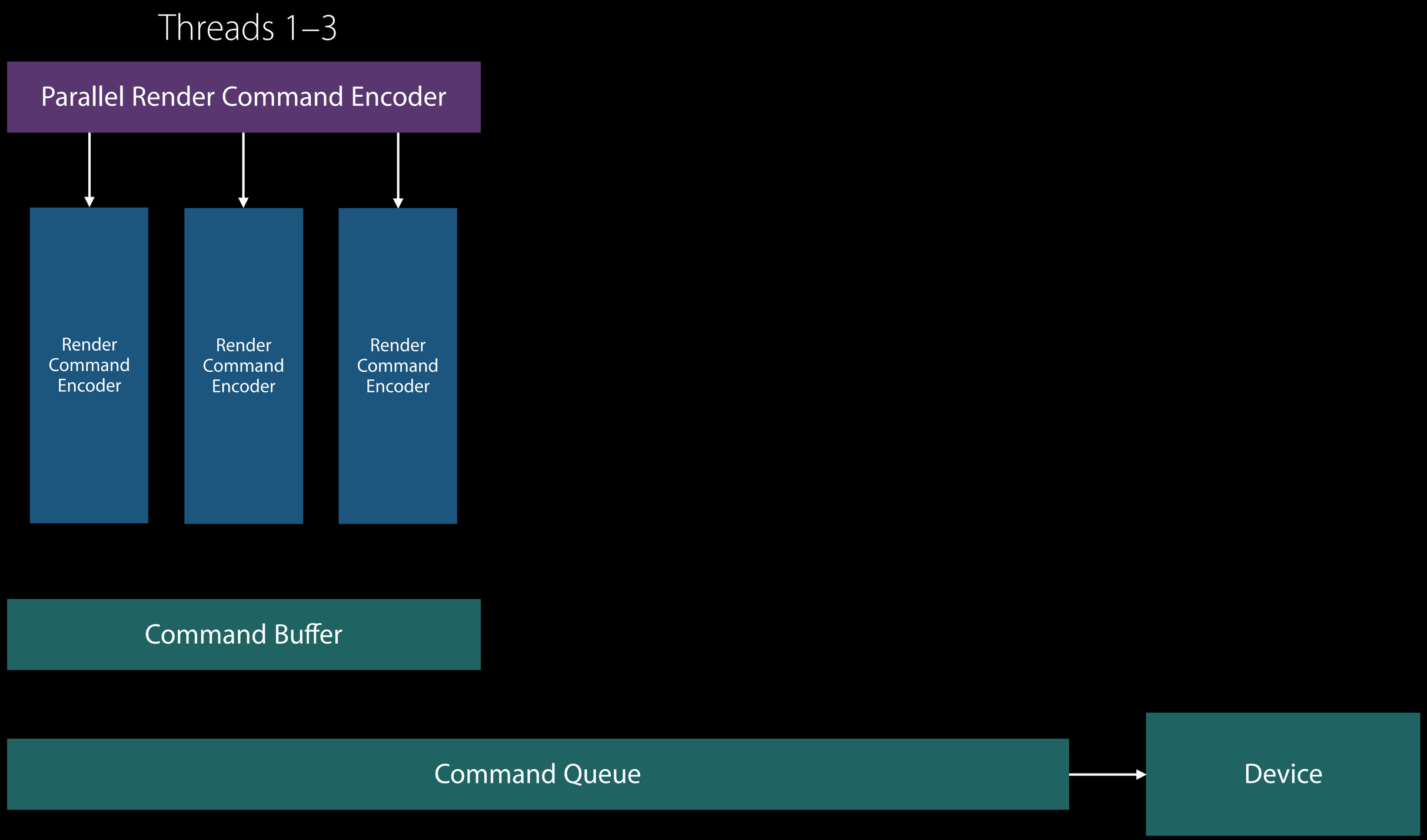
Threads 1–3

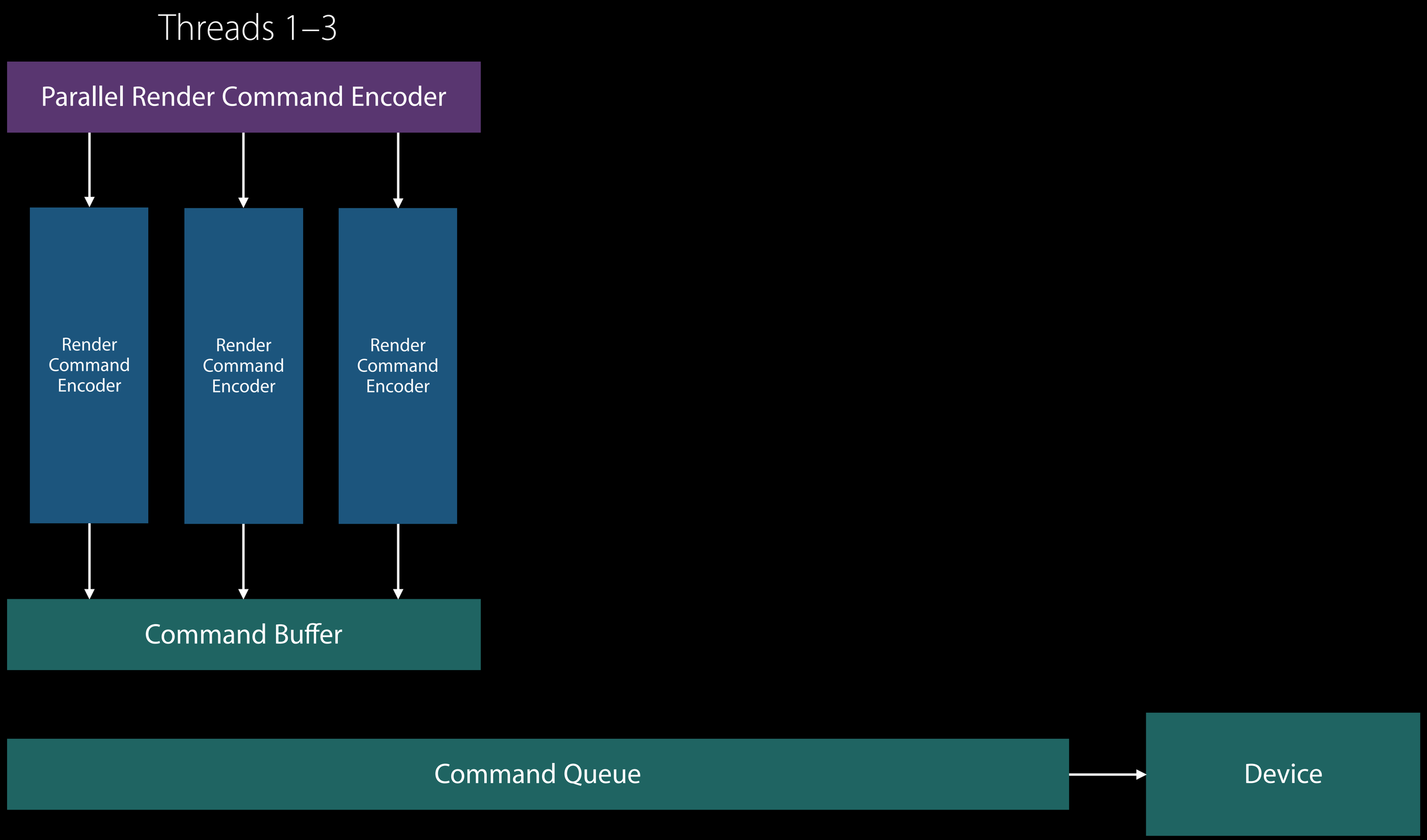


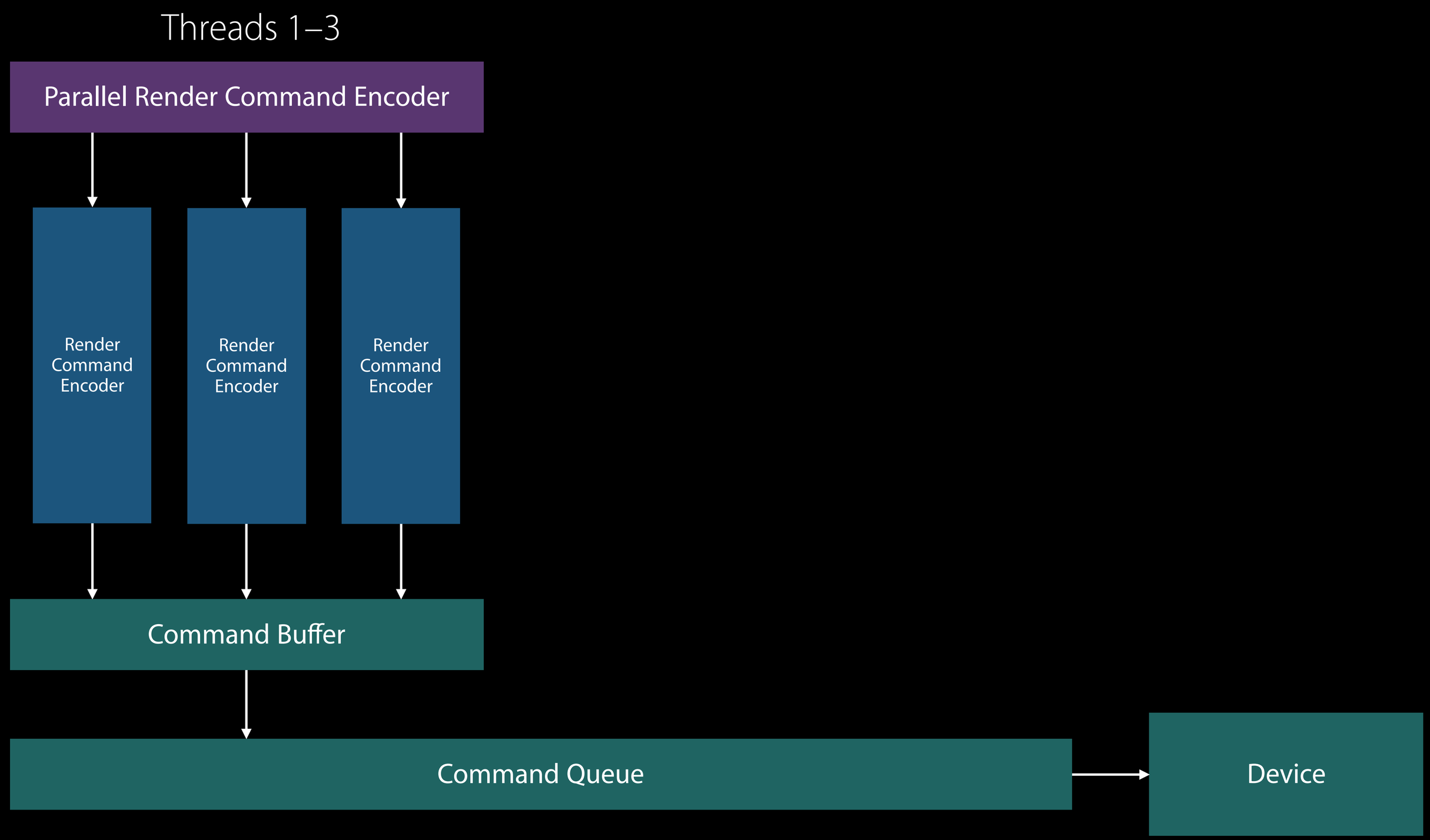
Threads 1–3

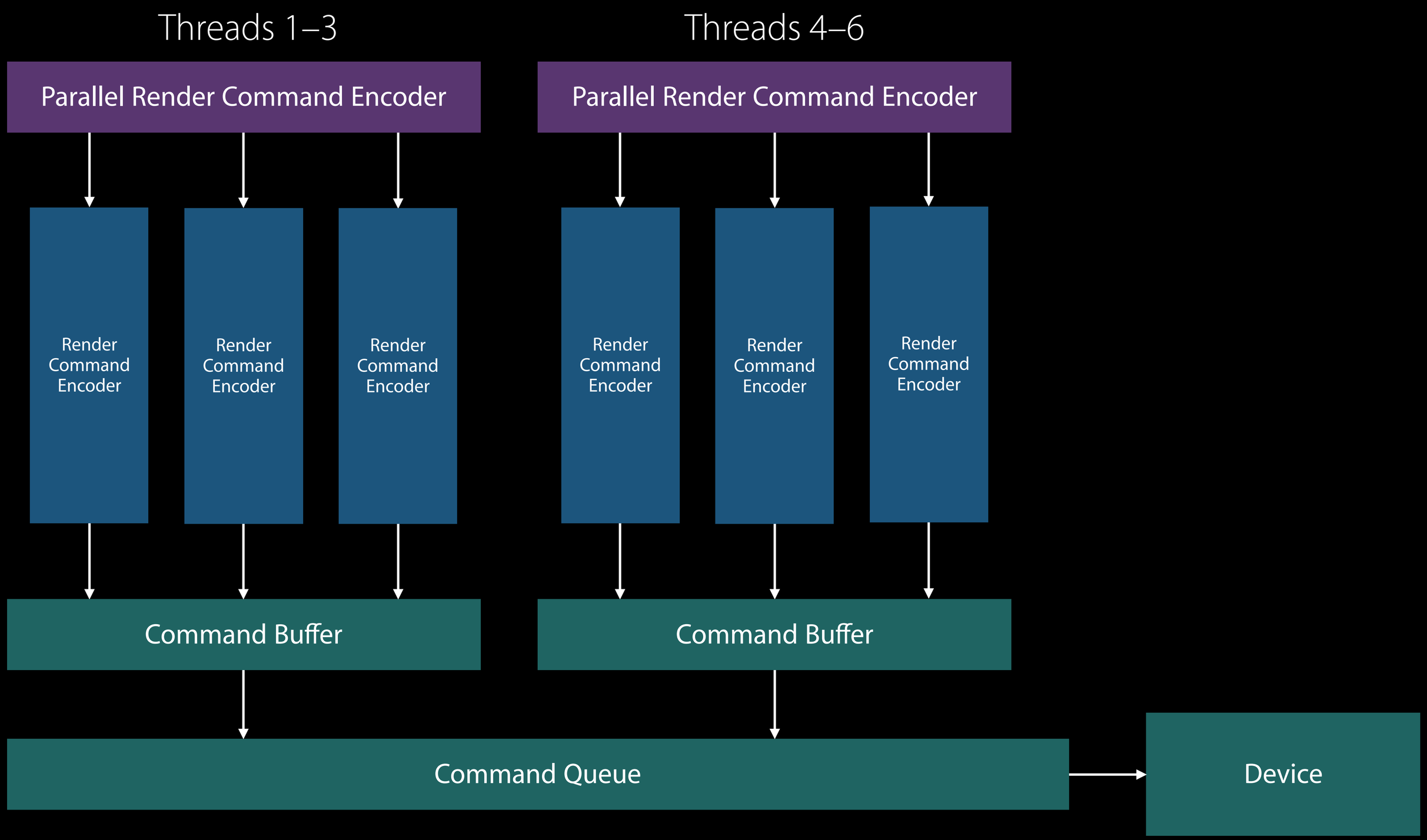
Parallel Render Command Encoder

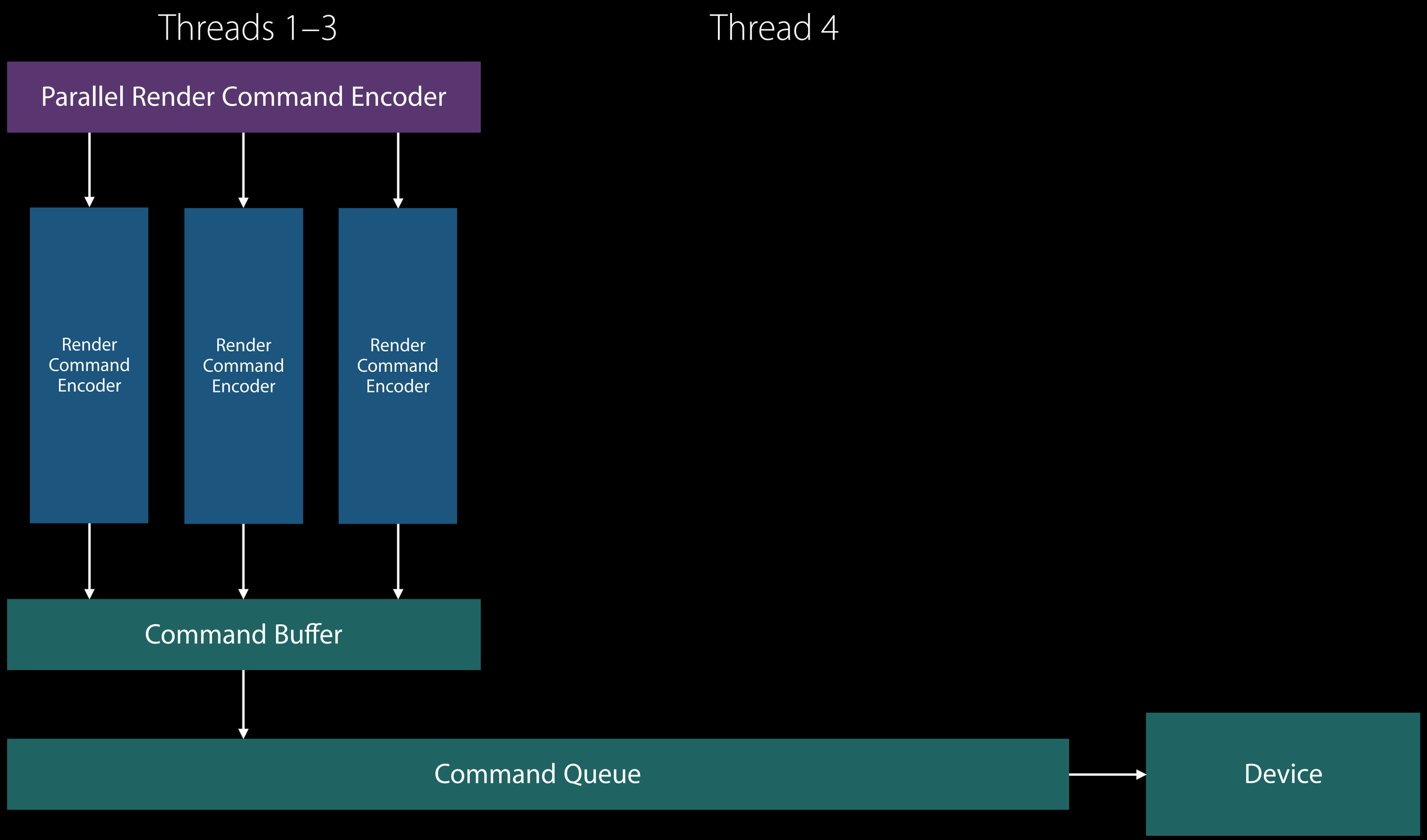


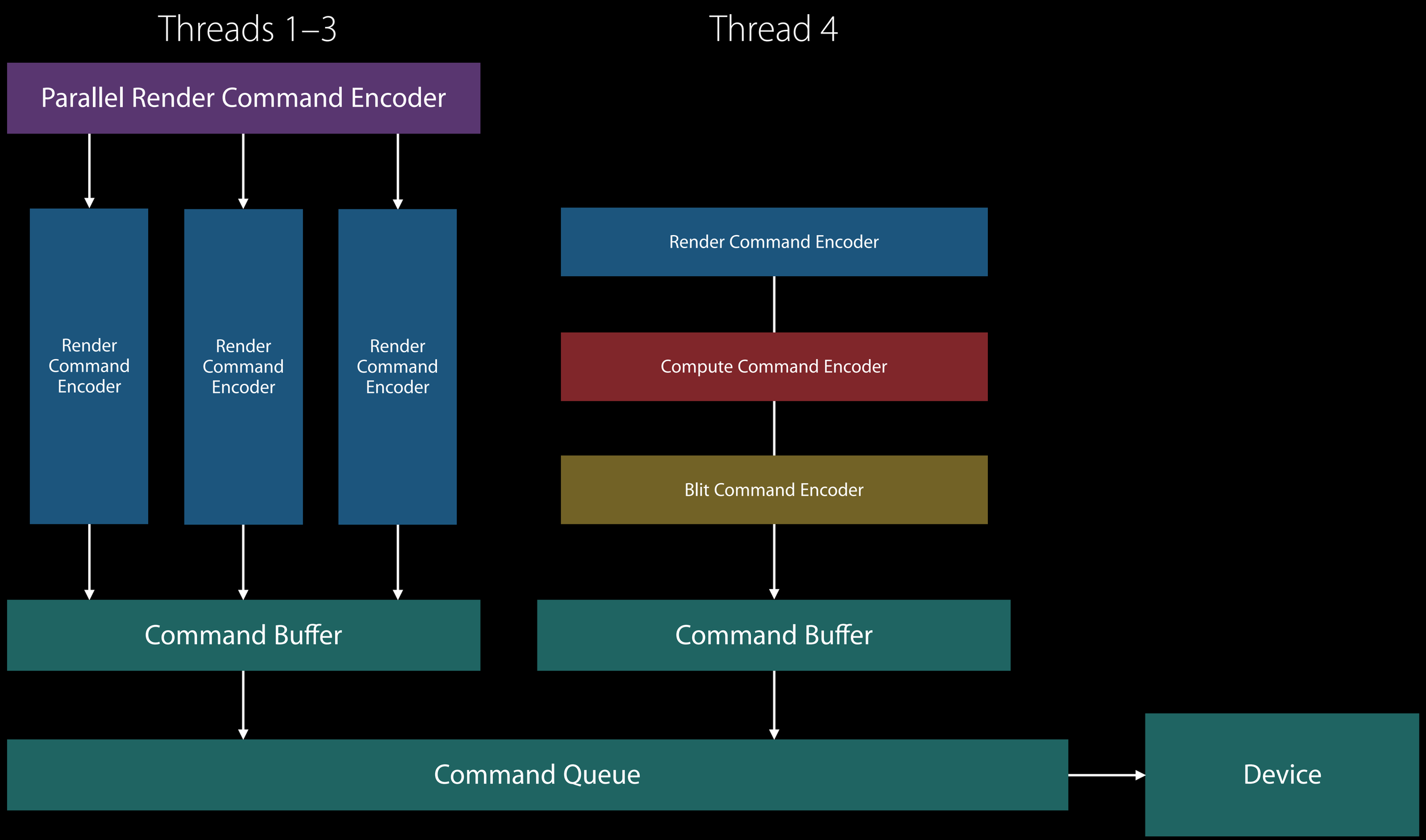

Threads 1–3
Parallel Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Command Buffer
Command Queue
Device


Threads 1–3
Parallel Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Command Buffer
Command Queue
Device


Threads 1–3
Parallel Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Command Buffer
Command Queue
Device


Threads 1–3
Threads 4–6
Parallel Render Command Encoder
Parallel Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Command Buffer
Command Buffer
Command Queue
Device


Threads 1–3
Thread 4
Parallel Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Command Buffer
Command Queue
Device


Threads 1–3
Thread 4
Parallel Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Render Command Encoder
Compute Command Encoder
Blit Command Encoder
Command Buffer
Command Buffer
Command Queue
Device

Setting Up Parallel Encoding

```
// create one command buffer per parallel render command encoder
id <MTLCommandBuffer> commandBuffer = [commandQueue commandBuffer];

// create and initialize render pass descriptor
// create a parallel render command encoder
id <MTLParallelRenderCommandEncoder> parallelRCE =
   [commandBuffer parallelRenderCommandEncoderWithDescriptor:renderPassDesc];

// create subordinate encoders in GPU submission order
id <MTLRenderCommandEncoder> rCE1 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE2 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE3 = [parallelRCE renderCommandEncoder];

// encode commands on each thread
[rCE1 draw...];        [rCE2 draw...];        [rCE3 draw...];
[rCE1 endEncoding];    [rCE2 endEncoding];    [rCE3 endEncoding];

// all subordinate encoders must end encoding before MTLParallelRenderCommandEncoder
[parallelRCE endEncoding];
[commandBuffer commit];
```

Setting Up Parallel Encoding

```
// create one command buffer per parallel render command encoder
id <MTLCommandBuffer> commandBuffer = [commandQueue commandBuffer];
```

```
// create and initialize render pass descriptor
// create a parallel render command encoder
id <MTLParallelRenderCommandEncoder> parallelRCE =
   [commandBuffer parallelRenderCommandEncoderWithDescriptor:renderPassDesc];

// create subordinate encoders in GPU submission order
id <MTLRenderCommandEncoder> rCE1 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE2 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE3 = [parallelRCE renderCommandEncoder];

// encode commands on each thread
[rCE1 draw...];        [rCE2 draw...];        [rCE3 draw...];
[rCE1 endEncoding];    [rCE2 endEncoding];    [rCE3 endEncoding];

// all subordinate encoders must end encoding before MTLParallelRenderCommandEncoder
[parallelRCE endEncoding];
[commandBuffer commit];
```

Setting Up Parallel Encoding

```
// create one command buffer per parallel render command encoder
id <MTLCommandBuffer> commandBuffer = [commandQueue commandBuffer];

// create and initialize render pass descriptor
// create a parallel render command encoder
id <MTLParallelRenderCommandEncoder> parallelRCE =
   [commandBuffer parallelRenderCommandEncoderWithDescriptor:renderPassDesc];

// create subordinate encoders in GPU submission order
id <MTLRenderCommandEncoder> rCE1 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE2 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE3 = [parallelRCE renderCommandEncoder];

// encode commands on each thread
[rCE1 draw...];        [rCE2 draw...];        [rCE3 draw...];
[rCE1 endEncoding];    [rCE2 endEncoding];    [rCE3 endEncoding];

// all subordinate encoders must end encoding before MTLParallelRenderCommandEncoder
[parallelRCE endEncoding];
[commandBuffer commit];
```

Setting Up Parallel Encoding

```
// create one command buffer per parallel render command encoder
id <MTLCommandBuffer> commandBuffer = [commandQueue commandBuffer];

// create and initialize render pass descriptor
// create a parallel render command encoder
id <MTLParallelRenderCommandEncoder> parallelRCE =
   [commandBuffer parallelRenderCommandEncoderWithDescriptor:renderPassDesc];

// create subordinate encoders in GPU submission order
id <MTLRenderCommandEncoder> rCE1 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE2 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE3 = [parallelRCE renderCommandEncoder];

// encode commands on each thread
[rCE1 draw...];        [rCE2 draw...];        [rCE3 draw...];
[rCE1 endEncoding];    [rCE2 endEncoding];    [rCE3 endEncoding];

// all subordinate encoders must end encoding before MTLParallelRenderCommandEncoder
[parallelRCE endEncoding];
[commandBuffer commit];
```

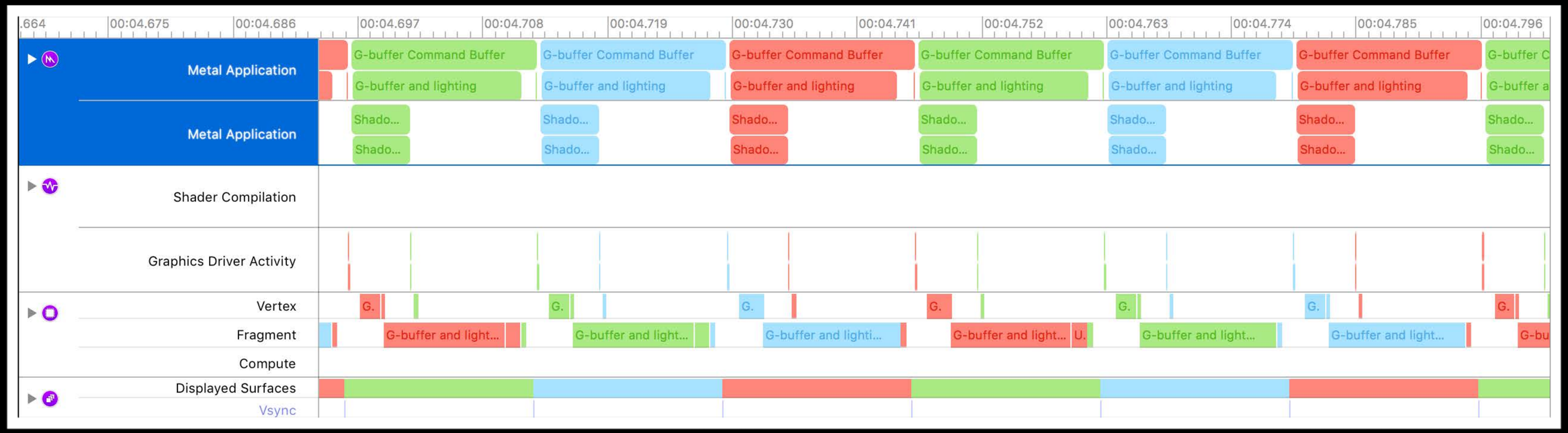
Setting Up Parallel Encoding

```
// create one command buffer per parallel render command encoder
id <MTLCommandBuffer> commandBuffer = [commandQueue commandBuffer];

// create and initialize render pass descriptor
// create a parallel render command encoder
id <MTLParallelRenderCommandEncoder> parallelRCE =
   [commandBuffer parallelRenderCommandEncoderWithDescriptor:renderPassDesc];

// create subordinate encoders in GPU submission order
id <MTLRenderCommandEncoder> rCE1 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE2 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE3 = [parallelRCE renderCommandEncoder];

// encode commands on each thread
[rCE1 draw...];
[rCE1 endEncoding];

[rCE2 draw...];
[rCE2 endEncoding];

[rCE3 draw...];
[rCE3 endEncoding];

// all subordinate encoders must end encoding before MTLParallelRenderCommandEncoder
[parallelRCE endEncoding];
[commandBuffer commit];
```

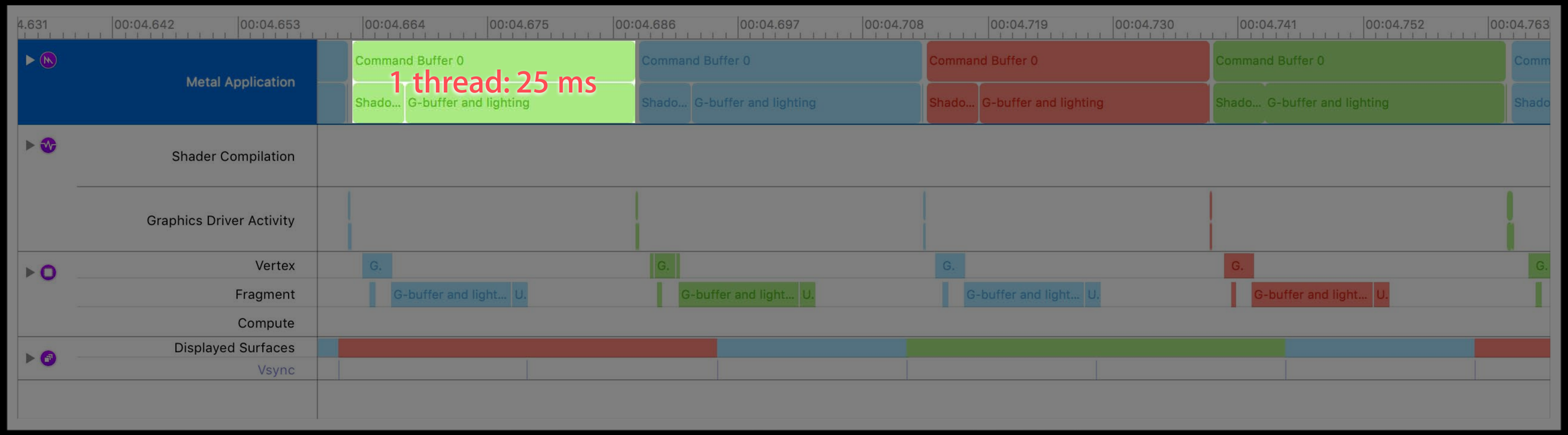
Setting Up Parallel Encoding

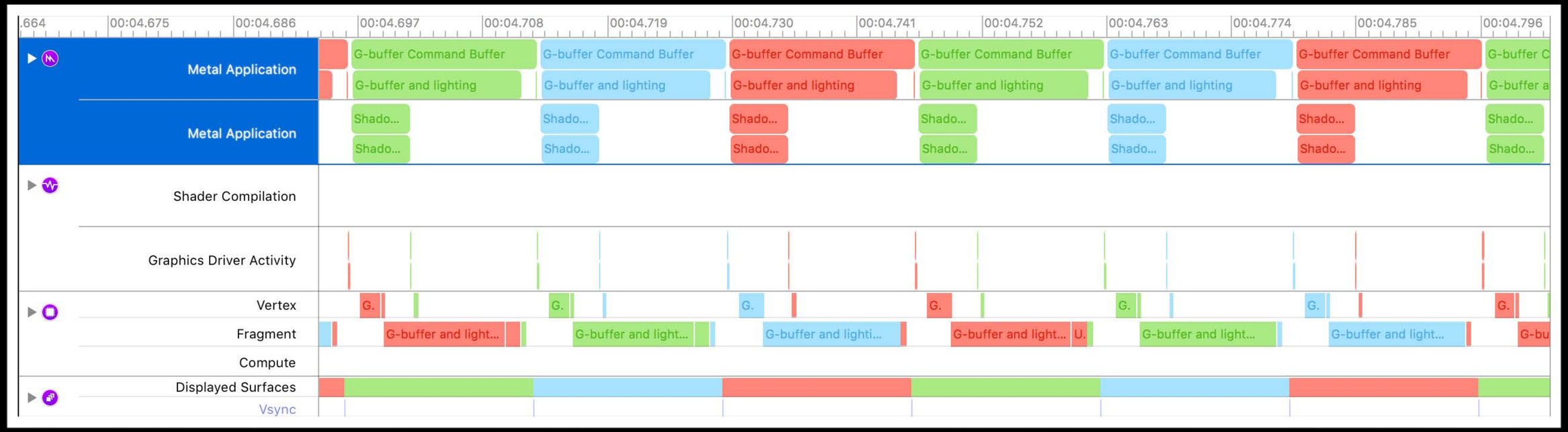
```
// create one command buffer per parallel render command encoder
id <MTLCommandBuffer> commandBuffer = [commandQueue commandBuffer];

// create and initialize render pass descriptor
// create a parallel render command encoder
id <MTLParallelRenderCommandEncoder> parallelRCE =
   [commandBuffer parallelRenderCommandEncoderWithDescriptor:renderPassDesc];

// create subordinate encoders in GPU submission order
id <MTLRenderCommandEncoder> rCE1 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE2 = [parallelRCE renderCommandEncoder];
id <MTLRenderCommandEncoder> rCE3 = [parallelRCE renderCommandEncoder];

// encode commands on each thread
[rCE1 draw...];        [rCE2 draw...];        [rCE3 draw...];
[rCE1 endEncoding];    [rCE2 endEncoding];    [rCE3 endEncoding];

// all subordinate encoders must end encoding before MTLParallelRenderCommandEncoder
[parallelRCE endEncoding];
[commandBuffer commit];
```


4.631
00:04.642
00:04.653
00:04.664
00:04.675
00:04.686
00:04.697
00:04.708
00:04.719
00:04.730
00:04.741
00:04.752
00:04.763
Metal Application
Command Buffer 0
Shado...
G-buffer and lighting
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
G.
G-buffer and light...
U.
Displayed Surfaces
Vsync

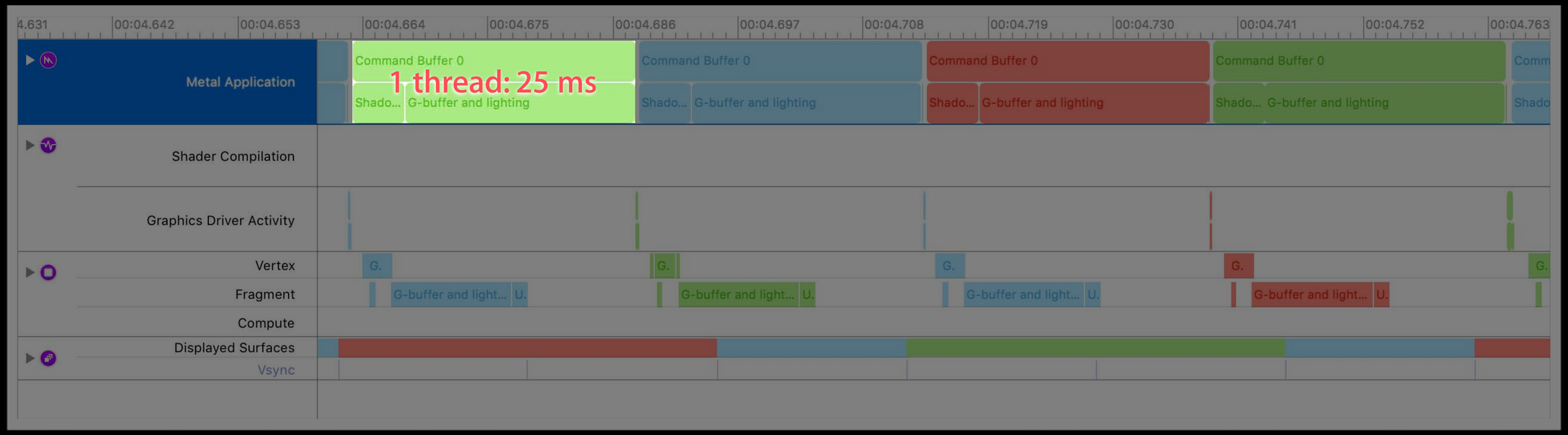

.664
00:04.675
00:04.686
00:04.697
00:04.708
00:04.719
00:04.730
00:04.741
00:04.752
00:04.763
00:04.774
00:04.785
00:04.796
Metal Application
G-buffer Command Buffer
G-buffer and lighting
Shado...
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
G.
G-buffer and light...
G-buffer and lighti...
U.
Displayed Surfaces
Vsync

Before


4.631
00:04.642
00:04.653
00:04.664
00:04.675
00:04.686
00:04.697
00:04.708
00:04.719
00:04.730
00:04.741
00:04.752
00:04.763
Metal Application
Command Buffer 0
1 thread: 25 ms
Shado...
G-buffer and lighting
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
G.
G-buffer and light...
U.


.664
00:04.675
00:04.686
00:04.697
00:04.708
00:04.719
00:04.730
00:04.741
00:04.752
00:04.763
00:04.774
00:04.785
00:04.796
Metal Application
G-buffer Command Buffer
G-buffer and lighting
Shado...
Shader Compilation
Graphics Driver Activity
Vertex
Fragment
Compute
Displayed Surfaces
Vsync
G.
G-buffer and light...
G-buffer and lighti...
U.

Before



After

Metal Optimization Best Practices

Consider multithreading if still CPU-bound

- Render passes can be simultaneously encoded on separate threads
- Use MTLParallelRenderCommandEncoder to split long passes

Summary

Summary

Metal System Trace offers a new insight into Metal app performance

Summary

Metal System Trace offers a new insight into Metal app performance

- Use in conjunction with Xcode to profile early and often

Summary

Metal System Trace offers a new insight into Metal app performance

- Use in conjunction with Xcode to profile early and often

Follow Metal best practices to maximize app performance

Summary

Metal System Trace offers a new insight into Metal app performance

- Use in conjunction with Xcode to profile early and often

Follow Metal best practices to maximize app performance

- Create expensive objects upfront and reuse

Summary

Metal System Trace offers a new insight into Metal app performance

- Use in conjunction with Xcode to profile early and often

Follow Metal best practices to maximize app performance

- Create expensive objects upfront and reuse
- Buffer dynamic shared resources

Summary

Metal System Trace offers a new insight into Metal app performance

- Use in conjunction with Xcode to profile early and often

Follow Metal best practices to maximize app performance

- Create expensive objects upfront and reuse
- Buffer dynamic shared resources
- Acquire the drawable at the latest opportunity

Summary

Metal System Trace offers a new insight into Metal app performance

- Use in conjunction with Xcode to profile early and often

Follow Metal best practices to maximize app performance

- Create expensive objects upfront and reuse
- Buffer dynamic shared resources
- Acquire the drawable at the latest opportunity
- Don't waste render command encoders

Summary

Metal System Trace offers a new insight into Metal app performance

- Use in conjunction with Xcode to profile early and often

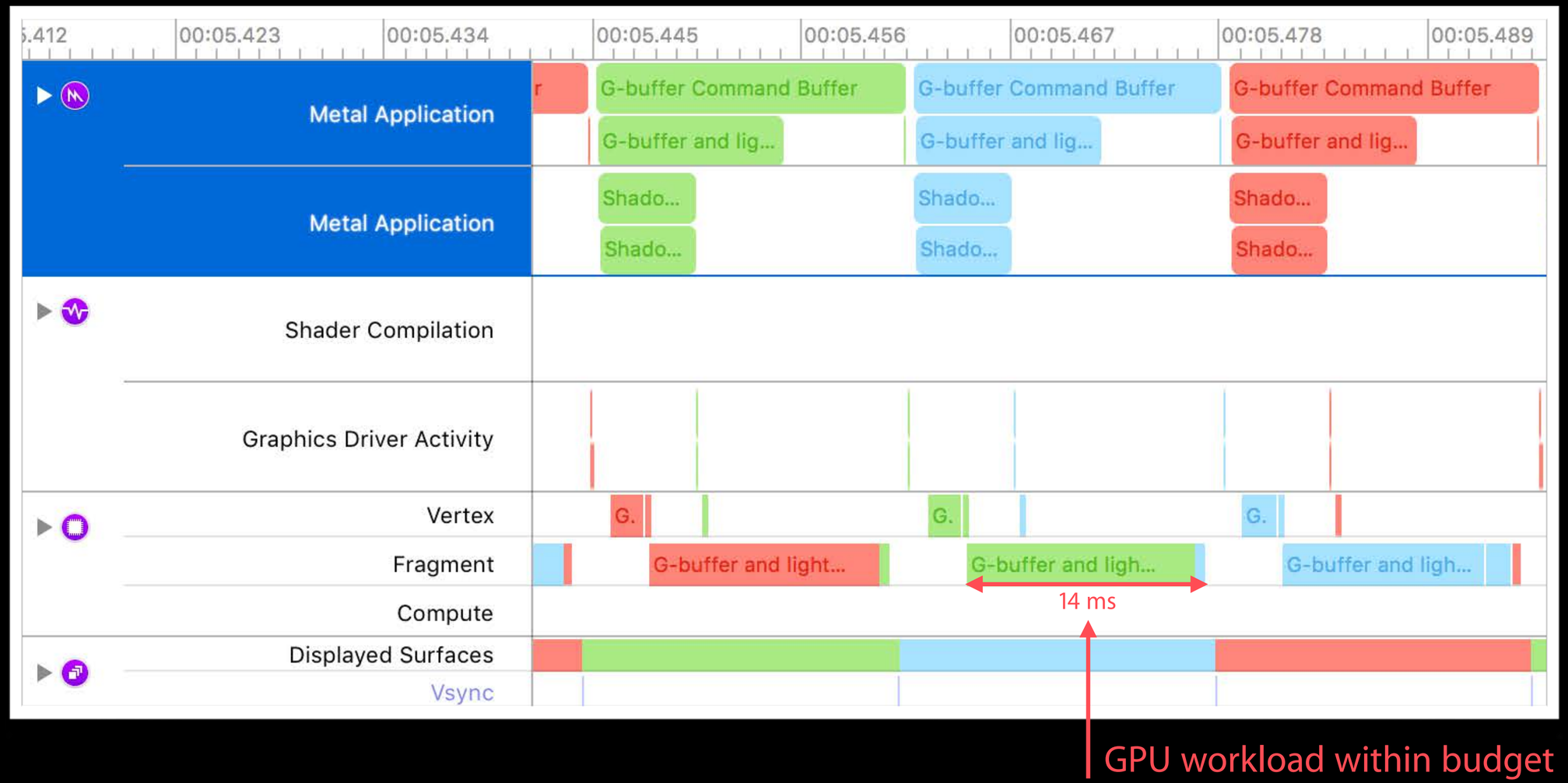
Follow Metal best practices to maximize app performance

- Create expensive objects upfront and reuse
- Buffer dynamic shared resources
- Acquire the drawable at the latest opportunity
- Don't waste render command encoders
- Consider multithreading if still CPU-bound

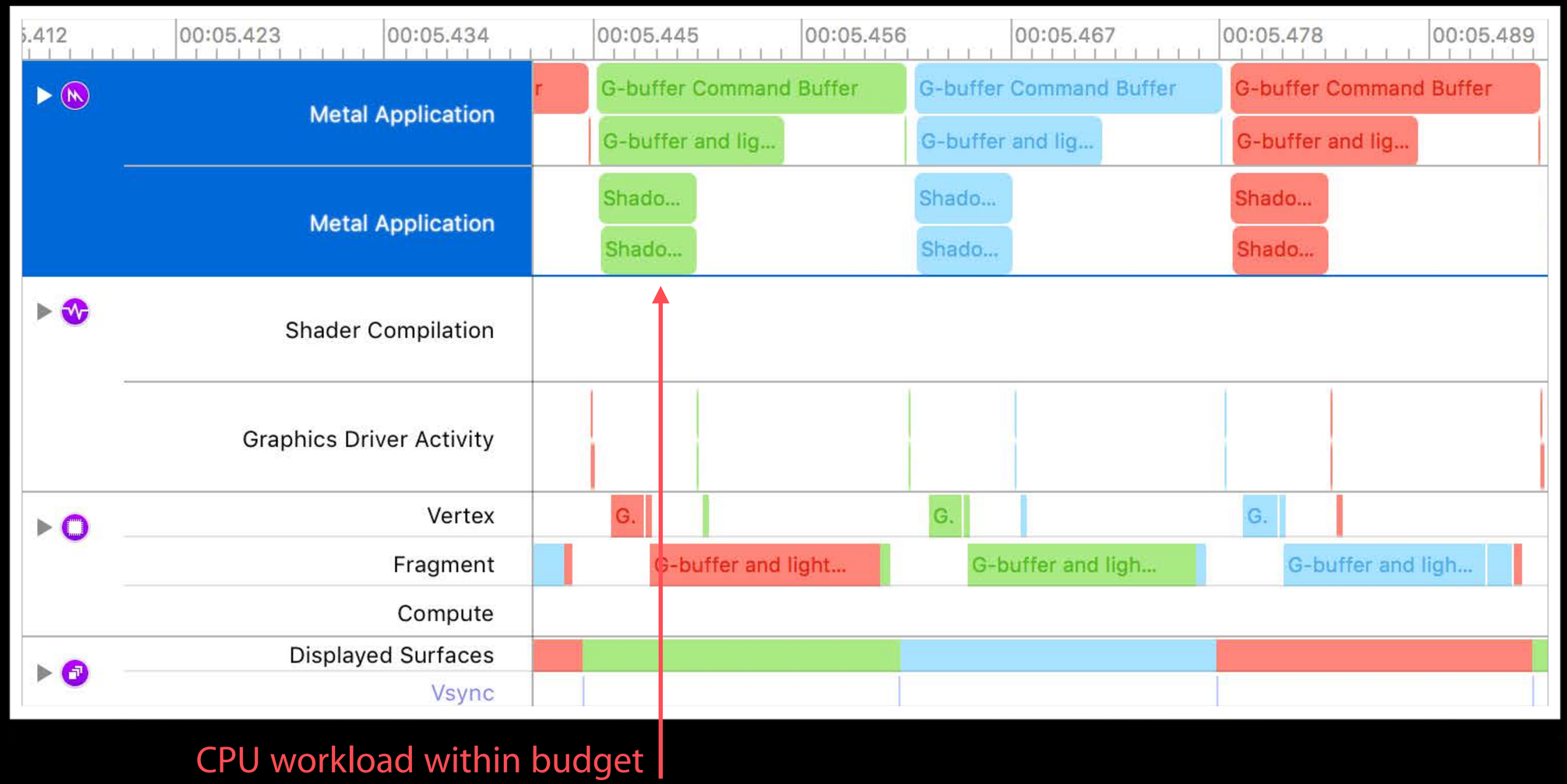
Overall Progress

Overall Progress

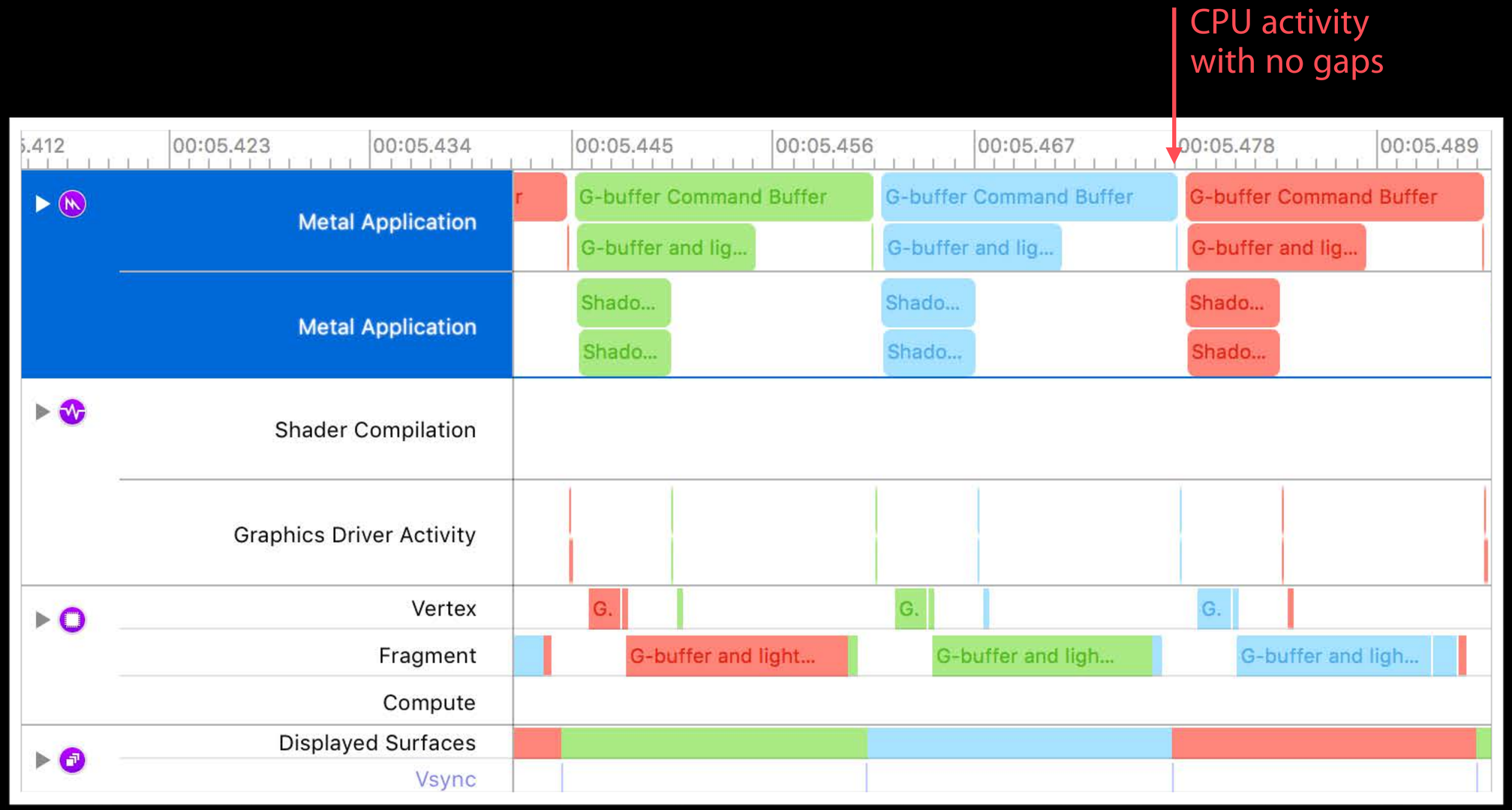
Overall Progress



Overall Progress

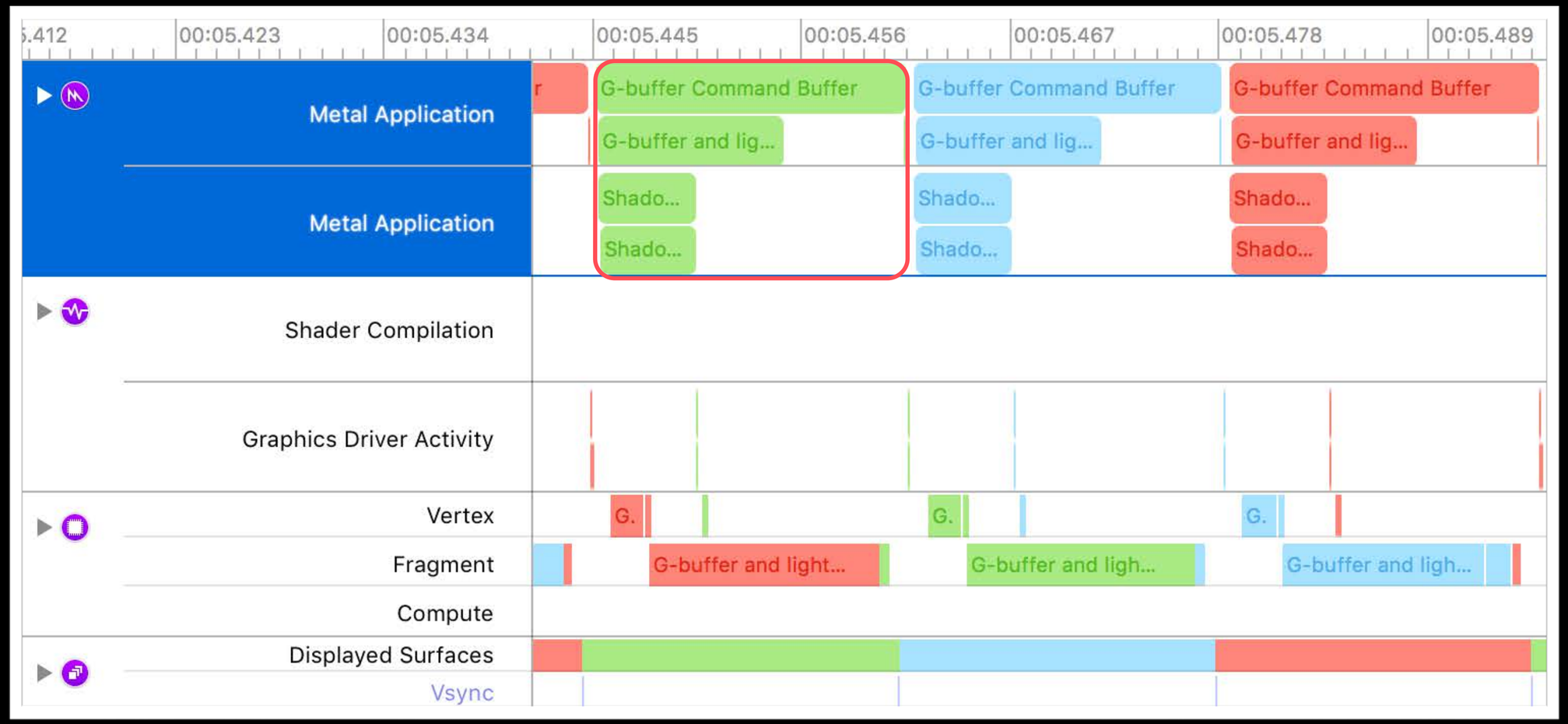


Overall Progress



Overall Progress

CPU encodes passes on multiple threads



Overall Progress

60 FPS !

More Information

Documentation and Videos

Metal Documentation

http://developer.apple.com/metal

Apple Developer Forums

http://developer.apple.com/forums

General Inquiries

Allan Schaffer, Graphics and Games Technologies Evangelist

aschaffer@apple.com

Labs

Metal	Graphics Lab D	Friday 12:00PM

WWDC15