#### Graphics and Games

### What's New in Metal Part 1 Session 604

Aaftab Munshi GPU Software Engineer James Ding GPU Software Engineer Jose Enrique D'Arnaude del Castillo GPU Software Engineer Alp Yucebilgin GPU Software Engineer

© 2016 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.

#### #WWDC16

### Metal at WWDC This Year A look at the sessions

Adopting Metal

Part One

- Fundamental Concepts
- Basic Drawing
- Lighting and Texturing

PartTwo

- Dynamic Data Management
- CPU-GPU Synchronization
- Multithreaded Encoding

### Metal at WWDC This Year A look at the sessions

What's New in Metal

Part One

- Tessellation
- Resource Heaps and Memoryless
  Render Targets
- Improved Tools

Part Two

- Function Specialization and Function Resource Read-Writes
- Wide Color and Texture Assets
- Additions to Metal Performance Shaders

### Metal at WWDC This Year A look at the sessions

Advanced Shader Optimization

- Shader Performance Fundamentals
- Tuning Shader Code





NEW





Wide Color and Texture Assets

Additions to Metal Performance Shaders

#### Function Specialization and Function Resource Read-Writes





Wide Color and Texture Assets

Additions to Metal Performance Shaders

#### Function Specialization and Function Resource Read-Writes





Function Specialization and Function Resource Read-Writes

Wide Color and Texture Assets

Additions to Metal Performance Shaders

NEW



## Agenda

#### Tessellation

Resource Heaps and Memoryless Render Targets

Function Specialization and Function Resource Read-Writes

Wide Color and Texture Assets

Additions to Metal Performance Shaders



Improved Tools

## Tessellation

Aaftab Munshi GPU Software Engineer

NEW



Large increase in memory bandwidth required to render high resolution geometry

Large increase in memory bandwidth required to render high resolution geometry Describe input geometry as a coarser or lower resolution model

- Large increase in memory bandwidth required to render high resolution geometry Describe input geometry as a coarser or lower resolution model Generate high resolution model from the coarser model
- High resolution model not stored in graphics memory
- Method to generate high resolution model is programmable
  Tessellation A technique to amplify and refine the geometric details of an object













Tessellation in Metal

### Metal Tessellation A simpler and efficient approach

### Metal Tessellation A simpler and efficient approach

Modern clean-sheet approach Easy to use Efficient and performant

#### Tessellation in Metal Availability

# macOS Sierra

All Configurations

#### Tessellation in Metal Availability

# macOS Sierra

All Configurations



A9 Processor

### Tessellation in Metal Topics we'll cover

Tessellation in Metal Topics we'll cover

Metal Graphics Pipeline Rendering Geometry Adopting Metal Tessellation

Metal Graphics Pipeline With Tessellation

A patch is a parametric surface made from spline curves

Described by a set of control vertices



- A patch is a parametric surface made from spline curves
- Described by a set of control vertices
- Tessellation controls how to render the patch as triangles



- A patch is a parametric surface made from spline curves
- Described by a set of control vertices
- Tessellation controls how to render the patch as triangles



- A patch is a parametric surface made from spline curves
- Described by a set of control vertices
- Tessellation controls how to render the patch as triangles



## Metal Tessellation First stage — Tessellation kernel



**Graphics Memory** 

## Metal Tessellation First stage — Tessellation kernel

**Graphics Memory** 

Buffers, Textures Input Control Point

**Tessellation Kernel** 

A programmable stage

## Metal Tessellation First stage — Tessellation kernel



#### A programmable stage Computations to generate

- Tessellation factors for a patch determines how much to subdivide the patch
- Additional patch data, if required

Graphics Memory


Graphics Memory



#### A fixed-function stage but configurable

**Graphics Memory** 



#### A fixed-function stage but configurable

- Subdivides the patch into triangles using the tessellation factors
- Triangles describe the high resolution geometry to draw

**Graphics Memory** 

the tessellation factors ometry to draw





#### A fixed-function stage but configurable

- Subdivides the patch into triangles using the tessellation factors
- Triangles describe the high resolution geometry to draw
- Triangle list not stored in graphics memory
- Generates parametric (u, v) values for vertices on subdivided patch

#### **Graphics Memory**

the tessellation factors ometry to draw

y ices on subdivided patch



### Metal Tessellation Third stage — Post Tessellation Vertex Shader



**Graphics Memory** 

### Metal Tessellation Third stage — Post Tessellation Vertex Shader



#### A programmable stage

**Graphics Memory** 

Post-Tessellation Vertex Shader

## Metal Tessellation Third stage — Post Tessellation Vertex Shader



#### A programmable stage

- Executes for vertices generated by tessellator
- Evaluates position and other attributes on high resolution surface
- Similar role as the domain shader in DirectX

### Metal Tessellation Complete pipeline



#### Graphics Memory

Post-Tessellation Vertex Shader

## Metal Tessellation Complete pipeline



Tessellation factors and patch data can also be generated by vertex or fragment shader Compute kernel preferred as it can asynchronously execute with draw commands on GPU

## Metal Tessellation Complete pipeline



- Tessellation factors and patch data can also be generated by vertex or fragment shader Compute kernel preferred as it can asynchronously execute with draw commands on GPU Tessellation kernel does not need to execute every frame
- Apply a scale value to the tessellation factors

### Metal Graphics Pipeline Without Tessellation



**Graphics Memory** 

### Metal Graphics Pipeline Without Tessellation

Vertex Data Buffers, Textures

Vertex Shader



#### Metal Graphics Pipeline With Tessellation



**Graphics Memory** 



#### Metal Graphics Pipeline With Tessellation



# Rendering Geometry with Tessellation



## Rendering Geometry with Tessellation

# Rendering Geometry with Tessellation

Writing a Post-Tessellation Vertex Shader Specifying Patch Data Inputs Configuring the Tessellator Drawing Patches

[[ patch(quad, 16) ]] vertex VertexOutput myPostTessellationVertexShader(uint patchID [[ patch\_id ]], float2 patchUV [[ position\_in\_patch ]], MyPatchData patchData [[ stage\_in ]], ...)

A vertex shader that executes for vertices of a tessellated surface

[[ patch(quad, 16) ]] vertex VertexOutput myPostTessellationVertexShader(uint patchID [[ patch\_id ]],

float2 patchUV [[ position\_in\_patch ]], MyPatchData patchData [[ stage\_in ]], ...)

A vertex shader that executes for vertices of a tessellated surface

[[ patch(quad, 16) ]]

vertex VertexOutput

myPostTessellationVertexShader(uint patchID [[ patch\_id ]], float2 patchUV [[ position\_in\_patch ]], MyPatchData patchData [[ stage\_in ]],

...)

A vertex shader that executes for vertices of a tessellated surface Shader Inputs

Patch Data vs. Vertex Data

[[ patch(quad, 16) ]] vertex VertexOutput myPostTessellationVertexShader(uint patchID [[ patch\_id ]], float2 patchUV [[ position\_in\_patch ]], MyPatchData patchData [[ stage\_in ]], ...)

- A vertex shader that executes for vertices of a tessellated surface Shader Inputs
- Patch Data vs. Vertex Data

[[ patch(quad, 16) ]] vertex VertexOutput myPostTessellationVertexShader(uint patchID [[ patch\_id ]], float2 patchUV [[ position\_in\_patch ]], MyPatchData patchData [[ stage\_in ]],

...)

A vertex shader that executes for vertices of a tessellated surface Shader Inputs

Patch Data vs. Vertex Data

[[ patch(quad, 16) ]] vertex VertexOutput myPostTessellationVertexShader(uint patchID [[ patch\_id ]], float2 patchUV [[ position\_in\_patch ]], MyPatchData patchData [[ stage\_in ]],

...)

A vertex shader that executes for vertices of a tessellated surface Shader Inputs

Patch Data vs. Vertex Data

Shader Outputs

• Similar to outputs from a vertex shader

```
[[ patch(quad, 16) ]]
vertex VertexOutput
myPostTessellationVertexShader(uint patchID [[ patch_id ]],
                                ...)
```

float2 patchUV [[ position\_in\_patch ]], MyPatchData patchData [[ stage\_in ]],

Per-patch and control-point data

- Multiple control-points per patch
  - Example: 16 control-points in a Bezier patch

Per-patch and control-point data

- Multiple control-points per patch

- Example: 16 control-points in a Bezier patch Use MTLVertexDescriptor to specify the patch data layout in memory

Per-patch and control-point data

- Multiple control-points per patch
  - Example: 16 control-points in a Bezier patch

Use MTLVertexDescriptor to specify the patch data layout in memory

- Attribute index used to identify and match patch input data with MTLVertexDescriptor • Control-point data specified as a templated type

- Declared as a struct with [[ stage\_in ]] qualifier in post-tessellation vertex shader

// struct that stores the per-patch and control-point data
struct MyPatchData {
 PerPatchData patchData;
 patch\_control\_point<ControlPointData> controlPointData;
};

// post-tessellation vertex shader
[[ patch(quad, 16) ]]
vertex VertexOutput
myPostTessellationVertexShader(MyPatchData patchData [[ stage\_in ]], ...)

// Per-Patch Data
struct PerPatchData {
 float b [[ attribute(2) ]];
 uint c [[ attribute(3) ]];
};

// struct that stores the per-patch and control-point data
struct MyPatchData {
 PerPatchData patchData;
 patch\_control\_point<ControlPointData> controlPointData;

};

// post-tessellation vertex shader
[[ patch(quad, 16) ]]
vertex VertexOutput
myPostTessellationVertexShader(MyPatchData patchData [[ stage\_in ]], ...)

// Per-Patch Data
struct PerPatchData {
 float b [[ attribute(2) ]];
 uint c [[ attribute(3) ]];
};

// struct that stores the per-patch and control-point data
struct MyPatchData {
 PerPatchData patchData;
 patch\_control\_point<ControlPointData> controlPointData;
};

// post-tessellation vertex shader
[[ patch(quad, 16) ]]
vertex VertexOutput
myPostTessellationVertexShader(MyPatchData patchData [[ stage\_in ]], ...)

```
// Per-Patch Data
struct PerPatchData {
  float b [[ attribute(2) ]];
  uint c [[ attribute(3) ]];
};
```

// struct that stores the per-patch and control-point data
struct MyPatchData {
 PerPatchData patchData;
 patch\_control\_point<ControlPointData> controlPointData;
};

// post-tessellation vertex shader
[[ patch(quad, 16) ]]
vertex VertexOutput
myPostTessellationVertexShader(MyPatchData patchData [[ stage\_in ]], ...)

```
// Per-Patch Data
struct PerPatchData {
  float b [[ attribute(2) ]];
  uint c [[ attribute(3) ]];
};
```

```
Control-Point Data
struct ControlPointData {
  float3 position [[ attribute(0) ]];
                 [[ attribute(1) ]];
 uint
       a
};
```

```
// struct that stores the per-patch and control-point data
struct MyPatchData {
  PerPatchData patchData;
```

};

patch\_control\_point<ControlPointData> controlPointData;

```
// post-tessellation vertex shader
[[ patch(quad, 16) ]]
vertex VertexOutput
myPostTessellationVertexShader(MyPatchData patchData [[ stage_in ]], ...)
```

// Per-Patch Data struct PerPatchData { float b [[ attribute(2) ]]; uint c [[ attribute(3) ]]; };

```
Control-Point Data
struct ControlPointData {
 float3 position [[ attribute(0) ]];
               [[ attribute(1) ]];
 uint a
};
```

```
// struct that stores the per-patch and control-point data
struct MyPatchData {
  PerPatchData patchData;
```

```
patch_control_point<ControlPointData> controlPointData;
```

```
};
```

```
// post-tessellation vertex shader
```

```
[[ patch(quad, 16) ]]
```

vertex VertexOutput

myPostTessellationVertexShader(MyPatchData patchData [[ stage\_in ]], ...)

// Per-Patch Data struct PerPatchData { float b [[ attribute(2) ]]; uint c [[ attribute(3) ]]; };

// struct that stores the per-patch and control-point data
struct MyPatchData {

PerPatchData patchData;

patch\_control\_point<ControlPointData> controlPointData;
};

// post-tessellation vertex shader
[[ patch(quad, 16) ]]
vertex VertexOutput
myPostTessellationVertexShader(MyPatchData patchData [[ stage\_in ]], ...)

// Per-Patch Data
struct PerPatchData {
 float b [[ attribute(2) ]];
 uint c [[ attribute(3) ]];
};

# Configuring the Tessellator
# Configuring the Tessellator

Tessellation Properties

- Specified in MTLRenderPipelineDescriptor

# Configuring the Tessellator

Tessellation Properties

- Specified in MTLRenderPipelineDescriptor
   Tessellation Factors
- Edge and Inside factors
- 16-bit floating-point values

// specify the tessellation factor buffer in the MTLRenderCommandEncoder
renderEncoder.setTessellationFactorBuffer(buffer, offset: offset, instanceStride: stride)

#### Drawing Patches New APIs in MTLRenderCommandEncoder

## Drawing Patches New APIs in MTLRenderCommandEncoder

- Non-indexed and indexed draw patch APIs
- Specify patch start, number of patches to draw, control-point index buffer ...
- renderEncoder.drawPatches(numberOfControlPoints,
- renderEncoder.drawIndexedPatches(numberOfControlPoints,

```
patchStart: patchStart, numberOfPatches: n, ...)
```

```
patchStart: patchStart, numberOfPatches: n, ...,
```

controlPointIndexBuffer: buffer,

```
controlPointIndexBufferOffset: offset, ...)
```

## Drawing Patches New APIs in MTLRenderCommandEncoder

- Non-indexed and indexed draw patch APIs
- Specify patch start, number of patches to draw, control-point index buffer ...
- renderEncoder.drawPatches(numberOfControlPoints,
- renderEncoder.drawIndexedPatches(numberOfControlPoints,

#### DrawIndirect variants

- Allows previous draw or dispatch commands to generate draw parameters

```
patchStart: patchStart, numberOfPatches: n, ...)
       patchStart: patchStart, numberOfPatches: n, ...,
       controlPointIndexBuffer: buffer,
       controlPointIndexBufferOffset: offset, ...)
```

Draw parameters such as patch start, patch count specified in a buffer filled out on GPU

Adopting Metal Tessellation





"Unity Technologies is proud to be working with Apple to make our Metal renderer the best of its kind, enabling Unity developers to harness the power of Metal. Later this year, we will be shipping support for Metal Tessellation, Metal Compute, and the ability to write native Metal shaders in Unity."

Unity Technology





#### UNREAL ENGINE

"Metal provides great performance and efficiency improvements alongside a wonderfully clean programming model. We're very enthusiastic about the ongoing revolution in high-end graphics on iOS and Mac enabled by Metal."

Epic Games

### Demo Tessellation in Action

Adopting Metal Tessellation Digital content creation tools



OpenSubdiv

"OpenSubdiv is a key technology at Pixar that helps our artists create expressive performances and beautiful worlds. Pixar is thrilled to see the high performance and full fidelity of OpenSubdiv realized with Metal Tessellation on macOS and iOS devices and with Apple's contribution of a native Metal implementation to the OpenSubdiv open source project."

David Yu Senior Software Engineer of Pixar's GPU Team



**Graphics Memory** 







#### One-to-one mapping between domain and post-tessellation vertex shader Tessellator remains the same





#### One-to-one mapping between domain and post-tessellation vertex shader Tessellator remains the same





#### One-to-one mapping between domain and post-tessellation vertex shader Tessellator remains the same Vertex and hull shader to be translated to a Metal kernel



Vertex descriptor used to describe vertex data layout

- [[ stage\_in ]] in a Metal kernel
- Input data layout described in a MTLStageInputOutputDescriptor

Vertex descriptor used to describe vertex data layout

- [[ stage\_in ]] in a Metal kernel
- Input data layout described in a MTLStageInputOutputDescriptor

Vertex and hull shader observations

- Vertex shader executes for each control-point
- Hull shader described by control-point and per-patch functions

Vertex descriptor used to describe vertex data layout

- [[ stage\_in ]] in a Metal kernel
- Input data layout described in a MTLStageInputOutputDescriptor

Vertex and hull shader observations

- Vertex shader executes for each control-point
- Hull shader described by control-point and per-patch functions

Translate vertex and hull shaders to Metal functions



Threadgroup Memory



Threadgroup Memory











**Graphics** Memory



**Graphics Memory** 





#### Tessellation Summary

#### Tessellation Summary

Simple to use and performant Easy to adapt your existing tessellation code to Metal Available on iOS and macOS

### Tessellation Summary

Simple to use and performant Easy to adapt your existing tessellation code to Metal Available on iOS and macOS Call to Action

• Use tessellation to improve the visual content rendered by your application

# Resource Heaps and Memoryless Render Targets

James Ding GPU Software Engineer


Resource Sub-Allocation



Resource Sub-Allocation Resource Aliasing



Resource Sub-Allocation Resource Aliasing Explicit Command Synchronization



Resource Sub-Allocation

Resource creation with MTLDevice

Heavy CPU operation

Resource creation with MTLDevice

Heavy CPU operation

Resource creation with MTLDevice

- Heavy CPU operation
- Resource binding
- Tracking costs add up for complex scenes

Resource creation with MTLDevice

- Heavy CPU operation
- Resource binding
- Tracking costs add up for complex scenes

Resource creation with MTLDevice

Heavy CPU operation

Resource binding

- Tracking costs add up for complex scenes Resource Heaps address these performance issues
- Perform expensive memory operations ahead of time
- Only heaps are tracked, not their resources



Creating Resources with MTLDevice

Memory



Creating Resources with MTLDevice

Memory

Memory



Creating Resources with MTLDevice



Memory



Creating Resources with MTLDevice



Memory

Resource

Resource

Resource



Creating Resources with MTLDevice

Creating Resources with MTLHeap



Memory



Creating Resources with MTLDevice

Creating Resources with MTLHeap







Creating Resources with MTLDevice

Creating Resources with MTLHeap







Creating Resources with MTLDevice

Creating Resources with MTLHeap







Creating Resources with MTLDevice

Creating Resources with MTLHeap



Memory



Creating Resources with MTLDevice

Creating Resources with MTLHeap

Memory

Resource



**NEW** 



Creating Resources with MTLDevice

Creating Resources with MTLHeap





Resource

### NEW



Creating Resources with MTLDevice

Creating Resources with MTLHeap







// Create heap descriptor, including the heap size. let heapDesc = MTLHeapDescriptor() heapDesc.size = heapSize

// Create the heap object ahead of time. let heap = device.newHeap(with:heapDesc)

// Create heap descriptor, including the heap size. let heapDesc = MTLHeapDescriptor() heapDesc.size = heapSize

// Create the heap object ahead of time. let heap = device.newHeap(with:heapDesc)



// Create heap descriptor, including the heap size. let heapDesc = MTLHeapDescriptor() heapDesc.size = heapSize

// Create the heap object ahead of time. let heap = device.newHeap(with:heapDesc)



// Create heap descriptor, including the heap size. let heapDesc = MTLHeapDescriptor() heapDesc.size = heapSize

// Create the heap object ahead of time. let heap = device.newHeap(with:heapDesc)

Use Resource Heaps to create resources on a performance-critical path

Use Resource Heaps to create resources on a performance-critical path

Use Resource Heaps to create resources on a performance-critical path Create multiple, size-bucketed heaps to reduce fragmentation

Use Resource Heaps to create resources on a performance-critical path Create multiple, size-bucketed heaps to reduce fragmentation

Use Resource Heaps to create resources on a performance-critical path Create multiple, size-bucketed heaps to reduce fragmentation. Use resource size and alignment queries to determine appropriate heap sizes

Resource Aliasing

# Resource Aliasing Multiple resources sharing memory

Resource Heaps allow multiple resources to occupy the same memory location Aliasing resources together reduces total memory usage


### Resource Aliasing Multiple resources sharing memory

Resource Heaps allow multiple resources to occupy the same memory location Aliasing resources together reduces total memory usage

Normal Heap Resources





### Resource Aliasing Multiple resources sharing memory

Resource Heaps allow multiple resources to occupy the same memory location Aliasing resources together reduces total memory usage

Normal Heap Resources

Aliasing Heap Resources





Shadow Map Passes





Main Pass

Post Processing Passes



#### Shadow maps are produced and consumed in consecutive passes

Shadow Map Passes

Time

Main Pass

Post Processing Passes



#### Shadow maps are produced and consumed in consecutive passes





Shadow Map B Shadow Map C

Main Pass

Post Processing Passes



#### Shadow maps are produced and consumed in consecutive passes



Time



#### Shadow maps are produced and consumed in consecutive passes



Time



#### Shadow maps are produced and consumed in consecutive passes Post-processing uses temporary textures internally





#### Shadow maps are produced and consumed in consecutive passes Post-processing uses temporary textures internally





Shadow maps are produced and consumed in consecutive passes Post-processing uses temporary textures internally Contents never used at the same time





let heap = device.newHeap(with:heapDesc)

let shadowMapA = heap.newTexture(with:shadowDesc)

let shadowMapB = heap.newTexture(with:shadowDesc)

let shadowMapC = heap.newTexture(with:shadowDesc)

// Allow heap to reassign shadow map memory to new resources shadowMapA.makeAliasable() shadowMapB.makeAliasable() shadowMapC.makeAliasable()

let postProcessingD = heap.newTexture(with:postProcessingDesc) let postProcessingE = heap.newTexture(with:postProcessingDesc)

let heap = device.newHeap(with:heapDesc)

let shadowMapA = heap.newTexture(with:shadowDesc)

let shadowMapB = heap.newTexture(with:shadowDesc)

let shadowMapC = heap.newTexture(with:shadowDesc)

// Allow heap to reassign shadow map memory to new resources shadowMapA.makeAliasable() shadowMapB.makeAliasable() shadowMapC.makeAliasable()

let postProcessingD = heap.newTexture(with:postProcessingDesc) let postProcessingE = heap.newTexture(with:postProcessingDesc)



let heap = device.newHeap(with:heapDesc)
let shadowMapA = heap.newTexture(with:shadowDesc)
let shadowMapB = heap.newTexture(with:shadowDesc)
let shadowMapC = heap.newTexture(with:shadowDesc)

// Allow heap to reassign shadow map memory to new resources
shadowMapA.makeAliasable()
shadowMapB.makeAliasable()
shadowMapC.makeAliasable()

let postProcessingD = heap.newTexture(with:postProcessingDesc)
let postProcessingE = heap.newTexture(with:postProcessingDesc)



Shadow Map B

Shadow Map C

- let heap = device.newHeap(with:heapDesc)
- let shadowMapA = heap.newTexture(with:shadowDesc)
- let shadowMapB = heap.newTexture(with:shadowDesc)
- let shadowMapC = heap.newTexture(with:shadowDesc)

// Allow heap to reassign shadow map memory to new resources shadowMapA.makeAliasable() shadowMapB.makeAliasable() shadowMapC.makeAliasable()

let postProcessingD = heap.newTexture(with:postProcessingDesc) let postProcessingE = heap.newTexture(with:postProcessingDesc)



Shadow Map B

Shadow Map C

- let heap = device.newHeap(with:heapDesc)
- let shadowMapA = heap.newTexture(with:shadowDesc)
- let shadowMapB = heap.newTexture(with:shadowDesc)
- let shadowMapC = heap.newTexture(with:shadowDesc)

// Allow heap to reassign shadow map memory to new resources shadowMapA.makeAliasable() shadowMapB.makeAliasable() shadowMapC.makeAliasable()

let postProcessingD = heap.newTexture(with:postProcessingDesc) let postProcessingE = heap.newTexture(with:postProcessingDesc)



Shadow Map B

Shadow Map C

- let heap = device.newHeap(with:heapDesc)
- let shadowMapA = heap.newTexture(with:shadowDesc)
- let shadowMapB = heap.newTexture(with:shadowDesc)
- let shadowMapC = heap.newTexture(with:shadowDesc)

// Allow heap to reassign shadow map memory to new resources shadowMapA.makeAliasable() shadowMapB.makeAliasable() shadowMapC.makeAliasable()

let postProcessingD = heap.newTexture(with:postProcessingDesc) let postProcessingE = heap.newTexture(with:postProcessingDesc)

#### Post Processing D

Post Processing E

#### Resource Aliasing Best practices

#### Resource Aliasing Best practices

Calling sequence is important

- Create resources in the order they will be used in a frame
- Interleave with makeAliasable after resource contents are consumed

used in a frame rce contents are consumed

#### Resource Aliasing Best practices

Calling sequence is important

- Create resources in the order they will be used in a frame
- Interleave with makeAliasable after resource contents are consumed

Keep dynamic resources in their own heap

Explicit Command Synchronization



Metal does not track command updates to heap resources

- Modification of resource contents
- Reinterpretation of aliasing resources



Metal does not track command updates to heap resources

- Modification of resource contents
- Reinterpretation of aliasing resources



Metal does not track command updates to heap resources

- Modification of resource contents
- Reinterpretation of aliasing resources

Tell Metal when heap resources are read or updated

Necessary for correct results



Metal does not track command updates to heap resources

- Modification of resource contents
- Reinterpretation of aliasing resources

Tell Metal when heap resources are read or updated

Necessary for correct results



Metal does not track command updates to heap resources

- Modification of resource contents
- Reinterpretation of aliasing resources
- Tell Metal when heap resources are read or updated
- Necessary for correct results

Use GPU fences to explicitly communicate resource dependencies



Pass A Pass B



Commands interact with fences with two operations





Commands interact with fences with two operations

• Update the fence with new timestamp





Commands interact with fences with two operations

- Update the fence with new timestamp
- Wait for GPU to reach the current timestamp





Shadow Vertex Shadow Fragment Main Vertex



Main Fragment

Post Process Compute

Metal commands are submitted in serial order

Shadow Vertex

Shadow Fragment

Main Vertex



Main Fragment

Post Process Compute

Metal commands are submitted in serial order GPUs execute encoded commands in parallel





Main Fragment

Post Process Compute

Metal commands are submitted in serial order GPUs execute encoded commands in parallel

Even across frames





### Synchronizing Heap Updates Incorrect results without synchronization

Two commands write to aliasing heap memory simultaneously





#### Synchronizing Heap Updates Serialize commands with a fence

Shadow Vertex	Main Vertex	
	Shadow Fragment	




Use a fence to serialize access to the aliasing heap resources





Use a fence to serialize access to the aliasing heap resources

Post Process command updates the fence





Use a fence to serialize access to the aliasing heap resources

- Post Process command updates the fence
- Shadow command waits on the fence







Use a fence to serialize access to the aliasing heap resources

- Post Process command updates the fence
- Shadow command waits on the fence







// Create post-processing encoder for frame A let postProcessingEnc = commandBufA.computeCommandEncoder() dispatchPostProcessingFilters(postProcessingEnc) // Tell GPU to update fence when post-processing is complete computeEnc.update(shadowPostProcessingFence) computeEnc.endEncoding()

// Create post-processing encoder for frame A let postProcessingEnc = commandBufA.computeCommandEncoder() dispatchPostProcessingFilters(postProcessingEnc) // Tell GPU to update fence when post-processing is complete computeEnc.update(shadowPostProcessingFence) computeEnc.endEncoding()



// Using Fences To Synchronize Heap Updates Across Commands

let shadowPostProcessingFence = device.newFence()

// Create post-processing encoder for frame A let postProcessingEnc = commandBufA.computeCommandEncoder() dispatchPostProcessingFilters(postProcessingEnc) // Tell GPU to update fence when post-processing is complete computeEnc.update(shadowPostProcessingFence) computeEnc.endEncoding()

// Create post-processing encoder for frame A let postProcessingEnc = commandBufA.computeCommandEncoder() dispatchPostProcessingFilters(postProcessingEnc) // Tell GPU to update fence when post-processing is complete computeEnc.update(shadowPostProcessingFence) computeEnc.endEncoding()

// Create post-processing encoder for frame A
let postProcessingEnc = commandBufA.computeCommandEncoder()
dispatchPostProcessingFilters(postProcessingEnc)
// Tell GPU to update fence when post-processing is complete
computeEnc.update(shadowPostProcessingFence)
computeEnc.endEncoding()

```
Across Commands
ence()
A
CommandEncoder()
gEnc)
ssing is complete
```

// Create post-processing encoder for frame A let postProcessingEnc = commandBufA.computeCommandEncoder() dispatchPostProcessingFilters(postProcessingEnc) // Tell GPU to update fence when post-processing is complete computeEnc.update(shadowPostProcessingFence) computeEnc.endEncoding()

// Create post-processing encoder for frame A let postProcessingEnc = commandBufA.computeCommandEncoder() dispatchPostProcessingFilters(postProcessingEnc) // Tell GPU to update fence when post-processing is complete computeEnc.update(shadowPostProcessingFence) computeEnc.endEncoding()

## Explicit Command Synchronization Best practices

Express dependencies at appropriate granularity

- Do not track static textures
- Track groups of resources with a single fence

### Resource Heaps Sumary

Create resources faster with suballocation Use memory more efficiently with resource aliasing Synchronize heap updates across commands with GPU Fences

Memoryless Render Targets Same Rendering, Less Memory



Memoryless render targets are not backed by system memory



Memoryless render targets are not backed by system memory



Memoryless render targets are not backed by system memory Use for render pass attachments that are not stored



Memoryless render targets are not backed by system memory Use for render pass attachments that are not stored



- Memoryless render targets are not backed by system memory Use for render pass attachments that are not stored Create texture with new storage mode
- MTLStorageModeMemoryless



Memoryless render targets are not backed by system memory Use for render pass attachments that are not stored Create texture with new storage mode MTLStorageModeMemoryless

iOS and tvOS only



## Tile-Based Rendering



### Color Attachment Texture



## Tile-Based Rendering

A7 and later GPUs render to fast GPU tile storage, one tile at a time



GPU Tile Storage



### Color Attachment Texture



## Tile-Based Rendering

A7 and later GPUs render to fast GPU tile storage, one tile at a time

Store actions control whether to copy results to system memory



GPU Tile Storage



### Color Attachment Texture



## Use Case Depth attachments



### GPU Tile Storage



### Color Attachment Texture



## Use Case Depth attachments

Depth attachments are required for depth testing



GPU Tile Storage



### Color Attachment Texture



## Use Case Depth attachments

- Depth attachments are required for depth testing
- If the depth attachment is not stored, make it memoryless



GPU Tile Storage



### Color Attachment Texture



## Use Case Multisample color attachments



GPU Tile Storage



### MSAA Color Attachment Texture



### Resolve Color Attachment Texture

## Use Case Multisample color attachments

MSAA attachments are required for multisample rendering



GPU Tile Storage



### MSAA Color Attachment Texture



### Resolve Color Attachment Texture

## Use Case Multisample color attachments

MSAA attachments are required for multisample rendering

If the color attachment is resolved, make the MSAA color attachment memoryless



GPU Tile Storage





### MSAA Color Attachment Texture



### Resolve Color Attachment Texture

## Use Case Memory savings

### Texture Type

### Depth Texture on iPhone 6s Plus 1920x1080

Depth Texture on iPad Pro (12.9-inch) 2732x2048

### 4xMSAA Color Texture on iPhone 6s Plus 1920x1080

4xMSAA Color Texture on iPad Pro (12.9-inch) 2732x2048

### Potential Savings (MB)

	7.9
2	21.3
	81.6
8	35.4

# Improved Tools

Jose Enrique D'Arnaude del Castillo GPU Software Engineer Alp Yucebilgin GPU Software Engineer

## What's New in Metal Tools

Metal System Trace



## What's New in Metal Tools

Metal System Trace GPU Overrides



## What's New in Metal Tools

Metal System Trace GPU Overrides GPU Frame Debugger



## Metal System Trace





### Frame 0 CompletedHandler Frame 3 Frame 1 Frame 2 Frame 3


	🗖 MacBook Pro 🕽 🔍 Metal	DeferredLightingOSX	Run 1 of 1
All Cores	All Processes / Threads	00:07.704 00:07.710 00:07.710	6 00:07.722
	Metal Application	Command buffer 234 shadow g-buffer Bl Re B Ren shadow s. s s	Command buffer 23 shadow g-buffer shadow g. s s.
	Metal User Callbacks		
▶ 😍	Graphics Driver Activity		
► <b>O</b>	Render (GPU 1)	GPU Exec G. GPU Ex	ec g-buffer
▶ 🕝	Scaler		
	Display		
	Vsync		

#### Netal Command Buffer Submissions

Command Buffer	Submission^	Process	Thread	Fram
Command buffer 0	00:01.404.955	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 1	00:01.435.310	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 2	00:01.451.445	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 3	00:01.523.327	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 4	00:01.551.234	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 5	00:01.573.453	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 6	00:01.602.031	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 7	00:01.623.114	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 8	00:01.652.224	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 9	00:01.673.317	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 10	00.01 209 846	MetalDeferredLightingOSX	0va8248	Fram

#### letalDeferredLightingOSX + 🖸 🔳 🖂 80:00:00 00:07.728 00:07.734 00:07.740 00:07.746 00:07.764 00:07.758 00:07.728 00:07.740 00:07.752 00:07.77 E 3 E 3 Command buffer 237 command buffer 236 adow... g-buffer Bl... Ren... Bl... Ren Bl... Ren... Bl... Rend. adow... g-buffer B... Ren... Bl.. adow.. GPU Exec... G. GPU Exec. g-buffer GPU Exec... G. GPU Exec... g-buffer On Display On Display Instrument Detail 🛛 🙆 🕑 Track Display ne ne 1 Style Nested Data $\diamond$ ne 2 ne 2 ne 3 ne 4 ne 5 ne 6 ne 7 ne 8 ne 9 e 10

• • •					📓 Met	alDeferredLightingO	SX						
	💻 MacBook Pro 〉 属 Metall	DeferredLightingOSX			Run 1 of 1	00:00:08					+		
All Cores	All Processes / Threads												
00:07.692	00:07.698 00	:07.704 00:07.7	710 00:07.	716	00:07.722 00	00:07.728	734	00:07.740	00:07.746 00:	07.752 00:	07.758	00:07.764	00:07.77
		Command buffer 234			Command buffer 235			Command buffer 236			Com	mand buffer 237	
			<del></del>		M		4		* * * *	4			
	Motal Application	shadow g-buffer	Bl Re B Ren		shadow g-buffer	Bl Ren Bl Rend.	**	shadow g-buffer	Bl Ren Bl Ren	•	shac	iow g-buffer	B Ren Bl I
		shadow s.	S		shadow g. s s.	s sy		shadow s.	s		shac	low s.	s
	Metal User Callbacks												
▶ �	Graphics Driver Activity		2										
	Graphics Driver Activity												
• 0	Render (GPU 1)	G	GPU Exec G. GPU	ixec g-bi	uffer	GPU Exec G	. GPU Exec	g-buffer		GPJ Exec G. GPU	Exec g-	-buffer	
▶ 🕝	Scaler												
	Display					On Display						On Display	
	Vsync												
ሌ Details 👌 Me	atal Command Buffer Submi	issions							Instrument D	etail		⊘ Ø @	)
Command Buffer	Submission^	Process	Thread		Frame					Tra	ick Display		
Command buffer 0	00:01.404.955	MetalDeferredLighting	OSX 0xa8248		Frame	1				St	/le	Nested Da	ta 🗘
Command buffer 1	00:01.435.310	MetalDeferredLighting	OSX 0xa8248		Frame	2							
Command buffer 2	00:01.451.445	MetalDeferredLighting	OSX 0xa8248		Frame	2							
Command buffer 3	00:01.523.327	MetalDeferredLighting	OSX 0xa8248		Frame	3							
Command buffer 4	00:01.551.234	MetalDeferredLighting	OSX 0xa8248		Frame	4							
Command buffer 5	00:01.573.453	MetalDeferredLighting	OSX 0xa8248		Frame	6							
Command buffer 7	00:01.602.031	MetalDeferredLighting	OSX 0xa8248		Frame	7							
Command buffer 9	00:01.652.224	MetalDeferredLighting	OSX 0xa8248		Frame	8							
Command buffer 9	00:01.673.317	MetalDeferredLighting	OSX 0xa8248		Frame	9							
Command buffer 1	0 00:01 709 846	MetalDeferredLighting	OSX 0ya8248		Frame	10							

				📓 N	letalDeferredLightingOS>	K				
	MacBook Pro 👌 👗 Metall	DeferredLightingOSX		Run 1 of 1	00:00:08				+ 🖸 🔳 📖	
All Cores	All Processes / Threads									
00:07.692	00:07.698 00:	:07.704 00:07.7	10 00:07.716	00:07.722	00:07.728 00:07.73	4 00:07.740	00:07.746 00:	00:07.752	8 00:07.764	00:07.77
▶ 💽		Command buffer 234		Command buffer 23	35	Command buffer	236		Command buffer 237	
		shadow g-buffer	Bl Re B Ren	shadow g-buffe	r Bl Ren Bl Rend	shadow g-buf	er Bl Ren Bl Ren		shadow g-buffer	B Ren Bl R
	Metal Application	shadow s.	S S	shadow g. s	s sy	shadow s.	s sy		shadow s. s	s sy
				s.						
	Metal User Callbacks									
▶ 👽								1		
G	raphics Driver Activity		ļ							
► <b>O</b>	Pender (GPU 1)	G		a buffar	CDIL Evoc C	CPU Exec			a huffer	
	Kender (GPO I)	0	PO EXEC O. OPO EX	sem g-buner	GPU Exec G.	GPO Exec g-buller		SPD EXEC O. GPO EXEC.	g-builei	1
▶ 🗗	Scaler									
	Display				On Display				On Display	
	Vsync									
🕓 Details 👌 Metal	Command Buffer Submi	ssions					Instrument D	etail	○ ② €	
Command Buffer	Submission	Process	Thread	Fran	ne			Track D	isplay	
Command buffer 0	00:01.404.955	MetalDeferredLighting	OSX 0xa8248	Fran	ne 1			Style	Nested Data	۵
Command buffer 1	00:01.435.310	MetalDeferredLighting	OSX 0xa8248	Fran	ne 2					
Command buffer 2	00:01.451.445	MetalDeferredLighting	OSX 0xa8248	Fran	ne 2					
Command buffer 3	00:01.523.327	MetalDeferredLighting	OSX 0xa8248	Fran	ne 3					
Command buffer 4	00:01.551.234	MetalDeferredLighting	OSX 0xa8248	Fran	ne 4					
Command buffer 5	00:01.573.453	MetalDeferredLighting	DSX 0xa8248	Fran	ne 5					
Command buffer 6	00:01.602.031	MetalDeferredLighting	DSX 0xa8248	Fran	ne 6					
Command buffer 7	00:01.623.114	MetalDeferredLighting(	USX 0xa8248	Fran	ne 7					
Command buffer 8	00:01.652.224	MetalDeferredLighting(	DSX 0xa8248	Fran	ne 8					
Command buffer 9	00:01.673.317	MetalDeferredLighting	JSX 0xa8248	Fran	ne 9					
command butter 10	00:01 709 846	MetalDeterredLighting(	12287/18	Fran						

	MacBook Pro > Meta	talDeferredLightingOSX	Run 1 of 1
All	Cores All Processes / Thread	ds	
00:0	00:07.692	00:07.704 00:07.710 00:07.716 00:07.7	722
► <b>(N</b> )		Command buffer 234	mand buffer 23
		shadow g-buffer Bl Re B Ren shad	low g-buffer
	Metal Application	shadow s s s	low g. s
			s.
	Metal User Callbacks		
▶ �	Graphics Driver Activity		
<b>≻ O</b>	Render (GPU 1)	GPU Exec G. GPU Exec g-buffer	
• (7)	Scaler		
	Display		
	Vsync		

#### 💿 Details 👌 Metal Command Buffer Submissions

Command Buffer	Submission^	Process	Thread	Fram
Command buffer 0	00:01.404.955	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 1	00:01.435.310	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 2	00:01.451.445	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 3	00:01.523.327	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 4	00:01.551.234	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 5	00:01.573.453	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 6	00:01.602.031	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 7	00:01.623.114	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 8	00:01.652.224	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 9	00:01.673.317	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 10	00.01 709 846	MatalDeferredLightingOSX	0x28248	Fram

ctalDeleffedelighting00X						
00:00:08				+		
00:07.728 00:07.734	00:07.740 0	0:07.746	07.752	00:07.758	00:07.764	00:07.770
15	Command buffer 236			Ci	ommand buffer 237	
		× × × ×			70 1	
BI Ren BI Rend	shadow g-buffer	Bl Ren Bl Ren	•	sr	nadow g-buffer	B Ren Bl R
s sy	shadow s.	S Sy		sł	nadow s.	S Sy
-						
GPU Exec G. GPU Exec	s g-buffer		GPJ Exec G. (	GPU Exec	g-buffer	
× × .						
On Display					On Display	
		Instrument D	etail		○ ② €	
ne				Track Displa	v	
ne 1				Style	Nested Data	0
ne 2						
ne 2						
ne 3						
ne 4						
ne 6						
ne 7						
ne 8						
ne 9						
ne 10						

•••					🛓 Meta	alDeferredLigh	tingOSX						
	💶 MacBook Pro 〉 属 Metal	DeferredLightingOSX			Run 1 of 1	00:00:08					+		
All Cores	All Processes / Threads												
00:07.692	00:07.698 00:	07.704 00:07.71	0 00:07.71	16 00:07.	722 00	:07.728	00:07.734	00:07.740	0:07.746 00:	07.752 00:0	07.758	00:07.764	00:07.77
		Command buffer 234		Com	mand buffer 235			Command buffer 236			Com	mand buffer 237	
		shadow g-buffer	31 Re B Ren			Bl Ren Bl	Rend	shadow g-buffer	Bl Ren Bl Ren	4	shad	ow g-buffer	B Ren Bl
	Metal Application	shadow s.	5) <b>S</b> )					shadow s.	s sy			ow s.	s sy
	Metal User Callbacks							-		_			
▶ 🐼	Graphics Driver Activity												
► <b>O</b>	Render (GPU 1)	G	PU Exec G. GPU E	xec g-buffer		GPU Ex	ec G. GPU Exec	g-buffer	G	PU Exec G. GPU E	xec g-	buffer	
▶ 🗗	Scaler												
	Display					On Display						On Display	
	Vsync												
📧 Details 👌 Met	tal Command Buffer Submis	ssions							Instrument De	etail		⊘ ② €	)
Command Buffer	Submission^	Process	Thread		Frame					Tra	ck Display		
Command buffer 0	00:01.404.955	MetalDeferredLightingC	SX 0xa8248		Frame <sup>4</sup>	1				Sty	le	Nested Da	ta 🗘
Command buffer 1	00:01.435.310	MetalDeferredLightingC	<b>SX</b> 0xa8248		Frame 3	2							
Command buffer 2	00:01.451.445	MetalDeferredLightingC	SX 0xa8248		Frame	2							
Command buffer 3	00:01.523.327	MetalDeferredLightingC	OXa8248		Frame	3							
Command buffer 4	00:01.551.234	MetalDeferredLightingC	SX 0xa8248		Frame	4							
Command buffer 5	00:01.573.453	MetalDeferredLightingC	SX 0xa8248		Frame	5							
Command buffer 6	00:01.602.031	MetalDeferredLightingC	OSX 0xa8248		Frame	6							
Command buffer 7	00:01.623.114	MetalDeferredLightingC	OSX 0xa8248		Frame	7							
Command buffer 8	00:01.652.224	MetalDeferredLightingC	OSX 0xa8248		Frame	8							
Command buffer 9	00:01.673.317	MetalDeferredLightingC	OSX 0xa8248		Frame	9							
Command buffer 10	0 00:01 709 846	MetalDeferredLighting(	SX 0va82/18		Frame '	10							

				met 🔤 Met	alDeferredLightingUSX					
	💻 MacBook Pro 👌 属 Meta	DeferredLightingOSX		Run 1 of 1	00:00:08			4		
All Cores	All Processes / Threads									
00:07.692	00:07.698	00:07	.710 00:07.716	00:07.722 00	00:07.728	00:07.740 00	00:07	.752 00:07.758	00:07.764	00:07.77
> 💌		Command buffer 234		Command buffer 235		Command buffer 236			Command buffer 237	
		shadow g-buffer	Bl Re B Ren		Bl Ren Bl Rend	shadow g-buffer	Bl Ren Bl Ren		shadow g-buffer	B Ren Bl F
	Metal Application	shadow s.	S S		s sy	shadow s.	s sy		shadow s.	s sy
	Metal User Callbacks									
▶ �	Graphics Driver Activity									
0	Render (GPU 1)		GPU Exec G. GPU Exe	c g-buffer	GPU Exec G. GPU	J Exec g-buffer	GF	J Exec G. GPU Exec	g-buffer	
▶ 🗗	Scaler									
	Display				On Display				On Display	
	Vsync									
🕓 Details 👌 N	letal Command Buffer Subm	issions					Instrument Det		🛇 🙆 E	
Command Buffer	Submission^	Process	Thread	Frame				Track Dis	olay	
Command buffer	0 00:01.404.955	MetalDeferredLightin	gOSX 0xa8248	Frame	1			Style	Nested Data	\$
Command buffer	1 00:01.435.310	MetalDeferredLightin	gOSX 0xa8248	Frame	2					
Command buffer	2 00:01.451.445	MetalDeferredLightin	gOSX 0xa8248	Frame	2					
Command buffer	3 00:01.523.327	MetalDeferredLightin	gOSX 0xa8248	Frame	3					
Command buffer	4 00:01.551.234	MetalDeferredLightin	gOSX 0xa8248	Frame	4					
Command buffer	5 00:01.573.453	MetalDeferredLightin	gOSX 0xa8248	Frame	5					
Command buffer	7 00:01.602.031	MetalDeferredLightin	GOSX 0x28248	Frame	7					
Command buffer	8 00:01.652.224	MetalDeferredLightin	aOSX 0xa8248	Frame	8					
Command buffer	9 00:01 673 317	MetalDeferredLightin	gOSX 0xa8248	Frame	9					
Command buffer	10 00:01 709 846	MetalDeferredLightin	aOSX 0xa8248	Frame	10					

	💻 MacBook Pro  属 Meta	DeferredLightingOSX	Run 1 of 1
All Cores	All Processes / Threads	:07.704 00:07.710 00:07.716	00:07.722
		Command buffer 234	Command buffer 23
		shadow g-buffer Bl Re B Ren	
	Metal Application	shadow s. s s	
	Metal User Callbacks		
▶ �	Graphics Driver Activity		
• 0	Render (GPU 1)	GPU Exec G. GPU Exec	g-buffer
▶ 🗗	Scaler		
	Display		
	Vsync		

#### 💿 Details 👌 Metal Command Buffer Submissions

Command Buffer	Submission^	Process	Thread	Fram
Command buffer 0	00:01.404.955	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 1	00:01.435.310	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 2	00:01.451.445	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 3	00:01.523.327	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 4	00:01.551.234	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 5	00:01.573.453	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 6	00:01.602.031	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 7	00:01.623.114	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 8	00:01.652.224	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 9	00:01.673.317	MetalDeferredLightingOSX	0xa8248	Fram
Command buffer 10	00.01 709 846	MatalDeferredLightingOSX	0x28248	Fram

etalDeferredLightingOSX				
00:00:08			+ [	
00:07.728 00:07.734	00:07.740 00:07.74	46 00:07.752	00:07.758	00:07.764 00:07.77
	Command buffer 236		Comm	nand buffer 237
Bl Ren Bl Rend	shadow g-buffer Bl	Ren Bl Ren	shado	w g-buffer B Ren Bl F
s sy	shadow s. s	sy		w s s sy
1				
		4		
GPU Exec G. GPU Exec	g-buffer	GPJ Exec	G. GPU Exec g-b	puffer
On Display				On Display
	1		-	
		) Instrument Detail		⊗ Ø €
			Track Display	Nexted Data
ne 2			Style	
ne 2				
ne 3				
ne 4				
ne 5				
ne 6				
ne 9				
pe 10				

iOS







iOS tvOS macOS





NEW



#### Resource events





				Instrument Detail		⊙ 🧿 €	
ID Label	Туре	Num Bytes	Process		Track Display		
532606790295 n/a	DataBuffer	n/a	MetalDeferredLightingOSX		Style	Nested Data	0
532606790296 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790297 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790298 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790299 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790300 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790301 MTLIOAccelResourcePool	DataBuffer	n/a	MetalDeferredLightingOSX				
528311819786 n/a	DataBuffer	16384	MetalDeferredLightingOSX				
528311819786 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790299 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790297 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				

#### Resource events





				Instrument Detail		⊙ 🧿 €	
ID Label	Туре	Num Bytes	Process		Track Display		
532606790295 n/a	DataBuffer	n/a	MetalDeferredLightingOSX		Style	Nested Data	0
532606790296 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790297 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790298 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790299 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790300 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790301 MTLIOAccelResourcePool	DataBuffer	n/a	MetalDeferredLightingOSX				
528311819786 n/a	DataBuffer	16384	MetalDeferredLightingOSX				
528311819786 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790299 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				
532606790297 n/a	DataBuffer	n/a	MetalDeferredLightingOSX				

Resource events

MacBook Pro All Cores All Processe 00:01.379 Metal Appl Metal User Cal ► O Render (C ▶ 🗗 Details ) Resource Events Timestamp^ Event 00:01.394.791 CreateResource 00:01.394.870 CreateResource 00:01.394.947 CreateResource 00:01.395.015 CreateResource 00:01.395.102 CreateResource 00:01.395.174 CreateResource 00:01.395.297 CreateResource 00:01.413.169 PageOnResource 00:01.414.338 ZeroFillResource 00:01.417.341 ZeroFillResource

00:01.420.334 ZeroFillResource 532606790297 n/a

			Attail Metal Deferred L						
<sub>Ă</sub> Meta	IDeferredLightingOSX		Run 1 of 1 00:00:08	Run 1 of 1 00:00:08					
Threads									
00:01.3	86 00:01.393 00:01	.400 00:01.407	00:01.414	00:01.421 00:01.428	00:01.435 00:01.442 0	00:01.449 00:01.456 00:01.463			
ation	Command buffer 0 shado g-buffer B. Re shado s. s	B. Re s.		Command buffer 1 shado g-buffer B. Re B. Re shado s. s s	Command buffer 2 shado g-buffer B. Re B. shado s. s s	Re Shad			
acks									
tivity		Command buffer (							
PU 1)	P. Pa	ige Pag	e Page	Page O	. Page O Page	. R Blit Enco			
caler									
splay			-			On Display			
sync					Instrument Detail				
	ID Label	Туре	Num Bytes	Process		Track Display			
5326 5326	06790295 n/a 06790296 n/a	DataBuffer DataBuffer	n/a n/a	MetalDeferredLightingOSX MetalDeferredLightingOSX		Style Nested Data 🗘			
5326	06790298 n/a	DataBuffer	n/a	MetalDeferredLightingOSX					
5326	06790299 n/a	DataBuffer	n/a	MetalDeferredLightingOSX					
5326	06790300 n/a 06790301 MTLIOAccelResourcePool	DataBuffer	n/a n/a	MetalDeferredLightingOSX MetalDeferredLightingOSX					
5283	11819786 n/a	DataBuffer	16384	MetalDeferredLightingOSX					
5283	11819786 n/a	DataBuffer	n/a	MetalDeferredLightingOSX					
5326	06790299 n/a	DataBuffer	n/a	MetalDeferredLightingOSX					

MetalDeferredLightingOSX

DataBuffer

n/a

Resource events Debug groups



		📓 Metal	DeferredLightingOSX					
ro 👌 属 MetalD	DeferredLightingOSX	Run 1 of 1	00:00:08				+	
ses / Threads								
.370	00:02.372 00:02.374	00:02.376	00:02.378 00:02.38	00:00	2.382	00:02.384	00:02.386	00:02.388
	Command buffer 33							Comman
Versligstige	shadow buffer	g-buffer	Blit Encoder 3	Blit Encoder 3 Render Encoder 4		Render Encoder 6		shadow
Application	shadow buffer pass	g-buffe structure	sync copies	sun	sync copies	fairy		shadow
		skybox						
Callbacks							1	
ver Activity								
ler (GPU 1)	GPU Executi GPU Execu	ution				sh Re	e g-buffer	
Scaler								
Display			On Display					
Vsync								
Buffer Submis	ssions		· · · · · · · · · · · · · · · · · · ·		Instrume	ent Detail	o 🙆 E	
Submission^	Process Thread	Frame					Track Display	

Frame 24

Frame 25

Frame 26

Frame 27

Frame 28

Frame 29

Frame 30

Frame 31

Frame 32

Frame 33

Frame 34

Nested Data

Style

Resource events Debug groups



	MetalDeferredLighting	JOSX		
MetalDeferredLightingOSX	Run 1 of 1 00:00:08			
Threads		-		88
372 00:02.374		00:02.376	00:02	.378
Command buffer 33				adow
shadow buffer	g-bu	ffer		
shadow buffer pass	g-bu	iffe structure		
	skyb	ox		

.252.372	MetalDeferredLightingOSX	0xa8248	Frame 28
.274.518	MetalDeferredLightingOSX	0xa8248	Frame 29
.302.171	MetalDeferredLightingOSX	0xa8248	Frame 30
.324.098	MetalDeferredLightingOSX	0xa8248	Frame 31
.359.667	MetalDeferredLightingOSX	0xa8248	Frame 32
.385.532	MetalDeferredLightingOSX	0xa8248	Frame 33
.408.123	MetalDeferredLightingOSX	0xa8248	Frame 34

Resource events Debug groups Multi GPU (macOS)



					📓 Me	alDeferredLightingOSX_N	MultiGPU						
💊 Meta	IDeferredLig	htingOSX			Run 1 c	f 1 00:00:04					+		
hreads													
0	00:04	.084	00:04.088	00	0:04.092	00:04.096 00:04	.100 0	0:04.104	00:04.108	00:04.112	00:0	4.116 00:0	4.120
	Con	nmand buffer 9	6				Command buffer	97				Command buffer 9	3
	sha	idow buffer	g-buffer	Blit Re	nde Blit Render.		shadow buffer	g-buffer	Blit E Render	Blit Render		shadow buffer	-buffer
tion	sha	idow buffer	g-buffer pass	syn s.	sync f.		shadow buffer	g-buffer pass	sync s.	syn f		shadow buffer g	-buffer pass
		s	s st					s str				s	str
acks													
ivity						Ĭ						Ī	
J 2)	GP	U Execution					GPU Execution					GPU Execution	
01)	paer 3	Ken Biit End	oder b			R. g-butter			Blit Encoder 3	Ren GPU EX	CUTION	R. g-butter	
play					Display								
sync													
									Instrume	nt Detail		🛇 🙆 E	
<b>∨</b> T	otal Durat	Min Duration	Avg Duration	Max Durati	Std Dev Du						Track Display	1	
813	3.29 s	42.07 µs	4.05 ms	16.94 ms	4.52 ms						Style	GPU Hardware	٥
/11	2.06 s	42.07 μs	2.89 ms	16.94 ms	3.55 ms						-		
102	1.24 5	11.59 115	12.131115	12.17 115	65.19 μ5								

Resource events Debug groups Multi GPU (macOS)



Resource events Debug groups Multi GPU (macOS) Scaler (iOS)



			MetalDeferredLighting						
Meta	alDeferredLighting	Run 3 of 3	00:00:04				+		
ds									
0	0:04.146.600 00:04.147.200 00:04.147.800	00:04.148.400	00:04.149	00:04.149.600	00:04.150.200	00:04.150.800	00:04.151.400	00:04.152	00:04.1
ation	Command Buffer 0 s. g-buffer g-buffer pass light accumulation stencil		Com R						
ation									
acks				Sche					1
ation									
tivity			Command Buf shadow g	C.					
ertex							shadow bu	g-buffer	
ment							Render Encoder 0	shadow	buffer
caler			Scaler						
splay	Display								
/sync									
					(				

	Instrument Detail		🛇 🔯 E	
End Time	Duration Process	Track Disp	lay	
00.04.040.091	10.07 ms backboardd (58)	01.1		~
00:04.063.359	16.67 ms backboardd (58)	Style	Display Events	~
00:04.080.006	16.65 ms backboardd (58)			
00:04.096.693	16.69 ms backboardd (58)			
00:04.113.341	16.65 ms backboardd (58)			
00:04.130.048	16.71 ms backboardd (58)			
00:04.146.711	16.66 ms backboardd (58)			
00:04.163.357	16.65 ms backboardd (58)			
00:04.150.069	849.50 μs MetalDeferredLighting			
00:04.180.082	16.73 ms backboardd (58)			

Resource events Debug groups Multi GPU (macOS) Scaler (iOS)



		📓 Me	etalDeferredLighting	]				
🔄 Meta	alDeferredLighting	Run 3 of 3	00:00:04			+ 0		
ds								
. I 0	0:04.146.600 00:04.147.200 00:04.147.800 Command Buffer 0	00:04.148.400	00:04.149 Com]	00:04.149.600 00:04.150.2	00 00:04.150.800	00:04.151.400	00:04.152	00:04.1
ation	s. g-buffer g-buffer pass light accumulation		R					
	stencil							
cation								
backs				Sche				
lation								
tivity			Command Buf shadow g					
ertex/						shadow bu	g-buffer	
Iment						Render Encoder 0	shadow b	uffe
Scaler	-		Scaler					
splay	Display							
/sync								
					Instrument Detail		9 😰 E	
00.0	End Time Duration Process					Track Display		
00:0	04.063.359 16.67 ms backboardd (58)					Style	Display Events	0
00:0	04.080.006 16.65 ms backboardd (58)							
00:0	04.096.693 16.69 ms backboardd (58)							

•													Ma Ma
	📗 💻 MacBook Pro 👌 🝌 Meta	lDef	err	edLighti	ngOSX	)					F	Run 2	of 2
All	Cores All Processes / Threads	\$											
20	00:04.240 00:04.260		1		04.28	)	12 12	00:04.3	00	00:	04.320		<u>к</u> е е
► 💌	Metal Application	uf		Commar s. g s.	d buff			Command s. s.	buffer 3	9 0 0			
	Metal User Callbacks												
<b>\$</b>	Graphics Driver Activity												
► O	Render (GPU 1)	. P.	P.	g-buff	er	ві	Bl			g-buffer	Bl	Bl	g-b
Þ	Scaler												
	Display				On Dis	play				On Disp	olay		On Dis
	Vsync												
<b>D</b>	CPU					Ĩ.							
Þ 🖸	File Activity: Stack Depth												
► 🕜	All Heap & Anonymous VM												

Display			On Display		On Display	On Display		On Display	On Display			On Display	On Display	On E
	Vsync													
Þ	CPU		-							1				
▶ 🛃	File Activity: Stack Depth													
► <b>(</b> )	All Heap & Anonymous VM													
🚯 Details 👌 Me	etal Command Buffer Sul	omissions						C	) Instrument I	Detail		0 0	E	
Command Buffer	Submissio	n^ Process		Thread		Frame					Track I	Display		
Command buffer '	16 00:04.692.2	69 MetalDeferre	edLightingOSX	CVDisplayLink::runIO	Thread 0xeb7a0	Frame 17					Style	Nest	ed Data	٢
Command buffer '	17 00:04.720.6	97 MetalDeferre	edLightingOSX	CVDisplayLink::runlO	Thread 0xeb7a0	Frame 18								
Command buffer '	18 00:04.743.5	57 MetalDeferre	edLightingOSX	CVDisplayLink::runlO	Thread 0xeb7a0	Frame 19								
Command buffer '	19 00:04.770.6	81 MetalDeferre	edLightingOSX	CVDisplayLink::runlO	Thread 0xeb7a0	Frame 20								
Command buffer 2	20 00:04.793.2	24 MetalDeferre	edLightingOSX	CVDisplayLink::runIO	Thread 0xeb7a0	Frame 21								
Command buffer 2	21 00:04.820.5	92 MetalDeferre	edLightingOSX	CVDisplayLink::runlO	Thread 0xeb7a0	Frame 22								
Command buffer :	00:04.843.6	27 MetalDeferre	edLightingOSX	CVDisplayLink::runlO	Thread 0xeb7a0	Frame 23								
Command buffer 2	23 00:04.878.6	91 MetalDeferre	edLightingOSX	CVDisplayLink::runIO	Thread 0xeb7a0	Frame 24								
Command buffer 2	24 00:04.903.8	86 MetalDeferre	edLightingOSX	CVDisplayLink::runlO	Thread 0xeb7a0	Frame 25								
Command buffer 2	25 00:04.928.7	62 MetalDeferre	edLightingOSX	CVDisplayLink::runlO	Thread 0xeb7a0	Frame 26								



NEW



				Me
	MacBook Pro 👌 🝌 Metal	DeferredLightingOSX		Run 2 of 2
All Cores	All Processes / Threads	00104.000		200
			00:04.300 00:04	
	Metal Application	f S. g S. J.	Command buffer 3 s. s.	
	Metal User Callbacks			
► <b>₩</b>	Graphics Driver Activity			
► <b>O</b>	Render (GPU 1)	P. P. g-buffer Bl B	g-buffer	Bl Bl g-bu
▶ 🗗	Scaler			
	Display	On Display		On Dis
	Vsync			
► ( <u>③</u> <		CPU		
Fil Fil	e Activity: Stack Depth			
	Heap & Anonymous VM			
🕟 Details 〉 Meta	I Command Buffer Submi	ssions		
Command Buffer	Submission	Process	Thread	Frame
Command buffer 16	00:04.692.269	MetalDeferredLightingOSX	CVDisplayLink::runIOThread 0xe	eb7a0 Fram
Command buffer 17	00:04.720.697	MetalDeferredLightingOSX	CVDisplayLink::runIOThread 0xe	eb7a0 Fram
Command buffer 18	00:04.743.557	MetalDeferredLightingOSX	CVDisplayLink::runIOThread 0xe	eb7a0 Fram
Command buffer 19	00:04.770.681	MetalDeferredLightingOSX	CVDisplayLink::runIOThread 0xe	eb7a0 Fram

00:04.793.224 MetalDeferredLightingOSX CVDisplayLink::runIOThread 0xeb7a0 Fram

00:04.820.592 MetalDeferredLightingOSX CVDisplayLink::runIOThread 0xeb7a0 Fram

00:04.843.627MetalDeferredLightingOSXCVDisplayLink::runIOThread 0xeb7a0Fram00:04.878.691MetalDeferredLightingOSXCVDisplayLink::runIOThread 0xeb7a0Fram

00:04.928.762 MetalDeferredLightingOSX CVDisplayLink::runIOThread 0xeb7a0 Fran

Fran

00:04.903.886 MetalDeferredLightingOSX CVDisplayLink::runIOThread 0xeb7a0

Command buffer 20

Command buffer 21

Command buffer 22

Command buffer 23 Command buffer 24

Command buffer 25

etalDeferredLig	htingOSX												
00:00:09									+	0			
00:04.340 Command buffe	er 4	360 9 0 0 0 0	00:04.3	80 Con s.	00:04 nmand b g	4.400 		2:04.420 Command S. g S.		00:04.440 Comman S g		00:0/ Comma s g	4.460 and b
uffer Bl Bl	g-buffer	Bl Bl	g-buffer	B B		g-buffer	Bl G	9 g-buffe	G	G	g-buffer	G (	3)
play			On I	Display					On Displa	ıy	On Displa	y	On D
					Inst	rument Deta				0	<b>3</b> (E)		
e								Track	Display				
e 17								Style		Nest	ted Data		٢
e 18													
e 20													
e 21													
e 22													
e 24													
e 25													
e 26													

• •								Me Me
	📗 💻 Mac	:Book Pro 👌 属 Metal[	eferredLi	ghtingOSX				Run 2 of 2
All Co	ores All	Processes / Threads						
0	00:04.240	00:04.260		00:04.280	)	00:04.300	00:04.3	20
			f Com	mand buff.		Command buffer 3		
		Metal Application	S.			s.		
	Met	tal User Callbacks					-	
▶ �	Graph	ics Driver Activity						
► O		Render (GPU 1)	P. P. g-l	buffer	Bl	Bl	g-buffer B	I BI g-b
• 🕣		Scaler						
		Display		On Dis	play			On Dis
		Vsync						
► (O)		CPU						
Þ 🖸		File Activity:	Stack	Depth				
► <b>(</b> )	All Heap	& Anonymous VM						
No Details	s 〉 Metal Con	nmand Buffer Submi	ssions					
Command E	Buffer	Submission	Process			Thread		Fram
Command b	ouffer 16	00:04.692.269	MetalDef	- erredLighti	ngOSX	CVDisplayLink::run	IOThread Oxeb	7a0 Fram
Command b	ouffer 17	00:04.720.697	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread Oxeb	7a0 Fram
Command b	ouffer 18	00:04.743.557	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread Oxeb	7a0 Fram
Command b	ouffer 19	00:04.770.681	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread 0xeb	7a0 Fram
Command b	ouffer 20	00:04.793.224	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread Oxeb	7a0 Fram
Command b	ouffer 21	00:04.820.592	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread 0xeb	7a0 Fram
Command b	ouffer 22	00:04.843.627	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread 0xeb	7a0 Fram
Command b	ouffer 23	00:04.878.691	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread 0xeb	7a0 Fram
Command b	ouffer 24	00:04.903.886	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	OThread Oxeb	7a0 Fram
Command b	ouffer 25	00:04.928.762	MetalDef	erredLighti	ngOSX	CVDisplayLink::run	IOThread Oxeb	7a0 Fram

talDeferredLightingOSX					
00:00:09			+ 0		
00:04.340 00:04.360	00:04.380 00:04	.400 00:04.	420 00	:04.440	00:04.460
Command buffer 4	Command b	Command b	command b	Command b	Command b
s	s. g	s g	. g	i g	s g
		s			s
		· 11			1
	1				
ffer Bl Bl g-buffer Bl Bl g	-buffer B B	g-buffer Bl G	g-buffer G (	G g-buffer	G G
lay	On Display On Dis		On Display	On Display	On D
	Instru	ument Detail		0 <b>0</b> E	
			Track Display		
17			Style	Nested Data	\$
18					
: 19 : 20					
21					
22					
24					
25					
26					

• • •								MetalDe	eferredLigh	ntingOSX										
	II 🔍 N	MacBook Pro 👌 🛼 Meta	DeferredLighti	ngOSX			Run 2 d	of 2	00:00:09								+			
All Co	res	All Processes / Threads	\$																	
20	00:04.24	0 00:04.260	00	:04.280	00:04.300	00:0	4.320	00:04	4.340	00:04.3	360	00:04.380	0	0:04.400		00:04.420		00:04.440		00:04.460
			uf Comman	d buff	Command buf	fer 3		Со	mmand buffe	er 4			Command 1	b Corr	mand b	Commar	nd b	Command	b	Command b
		Metal Application	S. g		s. s.			s s.					s. g			s. g		s g s		s g s] [] []
		Metal User Callbacks		1					[			1		1						
▶ 🕜	Gra	aphics Driver Activity																I		
► O		Render (GPU 1)	. P. P. g-buff	er Bl	ві	g-buffer	Bl Bl	g-buffer	Bl Bl	g-buffer	ві ві	g-buffer	В В	g-buffe	er Bl	G g-buff	er G	G g-	buffer	G G
▶ 🗗		Scaler									<u> </u>									
		Disalar																		
—		Display		On Display			ay	n Display				Un Dis	play O				On Disp	lay	On Display	
		Vsync																		
► <b>(</b> )		CPU																		
	File	Activity: Stack Depth																		
Þ 🕜		All Heap & Al	nonymous	VM													JN .			
🕓 Details	) Metal C	Command Buffer Subm	nissions											nstrument D	)etail			0 0	(E)	
Command B	uffer	Submission	Process		Thread			Frame								Trac	k Display			
Command b	uffer 16	00:04.692.269	9 MetalDeferre	dLightingOSX	CVDisplayLink:	runIOThread 0	xeb7a0	Frame 17								Style	e 🦲	Neste	d Data	۵
Command b	uffer 17	00:04.720.697	7 MetalDeferre	dLightingOSX	CVDisplayLink:	:runIOThread 0	xeb7a0	Frame 18												
Command b	uffer 18	00:04.743.557	<ul> <li>MetalDeferre</li> <li>MetalDeferre</li> </ul>	dLightingOSX	CVDisplayLink:	runiOThread 0	xeb/a0	Frame 19								_				
Command b	ouffer 20	00:04.770.08	4 MetalDeferre	dLightingOSX	CVDisplayLink:	runIOThread 0	xeb7a0	Frame 21												
Command b	ouffer 21	00:04.820.592	2 MetalDeferre	dLightingOSX	CVDisplayLink:	runIOThread 0	xeb7a0	Frame 22												
Command b	ouffer 22	00:04.843.627	7 MetalDeferre	dLightingOSX	CVDisplayLink:	runIOThread 0	xeb7a0	Frame 23												
Command b	ouffer 23	00:04.878.697	1 MetalDeferre	dLightingOSX	CVDisplayLink:	runIOThread 0	xeb7a0	Frame 24												
Command b	ouffer 24	00:04.903.886	6 MetalDeferre	dLightingOSX	CVDisplayLink:	runIOThread 0	xeb7a0	Frame 25												
Command b	uffer 25	00:04.928.762	2 MetalDeferre	dLightingOSX	CVDisplayLink:	runIOThread 0	xeb7a0	Frame 26												

	MacBook Pro 👌 属 Metal	DeferredLightingOSX	Ru	Me In 2 of 2
All Cores	All Processes / Threads	CPU Usage		
00:04.24	40 00:04.260	00:04.280	00:04.300 00:04.320	
Core 0				
Core 1 (logical)				
Core 2				
Core 3 (logical)				
Core 4				
Core 5 (logical)				
Core 6				
Core 7 (logical)				
		f Command buff	Command buffer 3	
	Metal Application	s. g	s. g.,	
	Metal User Callbacks			
🚯 Details 🔪 Metal	Command Buffer Submis	ssions		
Command Buffer	Submission^	Process	Thread	Fram
Command buffer 16	00:04.692.269	MetalDeferredLightingOSX	CVDisplayLink::runIOThread 0xeb7a0	Fram
Command buffer 17	00:04.720.697	MetalDeferredLightingOSX	CVDisplayLink::runIOThread 0xeb7a0	Fram

ommand burler to	00.04.092.209	MetalDererredLighting05X	CVDIsplayLink::runiOThread	0xeb/a0	FIG
ommand buffer 17	00:04.720.697	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 18	00:04.743.557	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 19	00:04.770.681	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 20	00:04.793.224	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 21	00:04.820.592	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 22	00:04.843.627	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 23	00:04.878.691	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 24	00:04.903.886	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran
ommand buffer 25	00:04.928.762	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fran



• • •					M
MacBoo	ok Pro 👌 属 Metal	DeferredLightingOSX			Run 2 of 2
All Cores All Pro	ocesses / Threads	CPU Usage			
00:04.240	00:04.260	00:04.280	00:04.300	00:0	4.320
Core 0	Core 0				
Core 1 (logical)	Core 1 (lo	gical)			
Core 2	001011(10	9.04.7			
Core 3 (logical)	Core 2				
Core 4	Core 3 (lo	ogical)			
Core 5 (logical)				-	
Core 6	Core 4				
Core 7 (logical)	Core 5 (lo	ogical)			
	Core 6				
M	Core 7 (lo	gical)			
Metal					
🕓 Details 👌 Metal Comm				uf Cor	nmand buff
Command Buffer	<u> </u>				
Command buffer 16		Metal A	pplication		9
Command buffer 17				s.	
Command buffer 18					
Command buffer 20					
Command buffer 21	00:04.820.592	MetalDeferredLightingOSX	CVDisplavLink::ru	unIOThread 0x	eb7a0 Fram
Command buffer 22	00:04.843.627	MetalDeferredLightingOSX	CVDisplayLink::ru	InIOThread 0x	eb7a0 Fram
Command buffer 23	00:04.878.691	MetalDeferredLightingOSX	CVDisplayLink::ru	InIOThread 0x	eb7a0 Fram
Command buffer 24	00:04.903.886	MetalDeferredLightingOSX	CVDisplayLink::ru	InIOThread 0x	eb7a0 Fram
Command buffer 25	00:04.928.762	MetalDeferredLightingOSX	CVDisplayLink::ru	InIOThread 0x	eb7a0 Fram

etalDeferredLig	htingOSX					
00:00:09				+ 0		
00:04.340	00:04.360	00:04.380	00:04,400	00:04.420	00:04.440	00:04.46
					_	
					-	
					command b	Command
					g	s g
						S
	Command buf	fer 3	Co	mmand buffer 4	<b>(3)</b>	
		(Chiefe		1		
	S.	g	S		Nested Data	٢
	s.		S			
	-					
e 22 e 23						
e 24						
e 25						
6 20						

									R. Matell	to the second second	Li- OOV												
			v 142 0.16.8						MetalDe	aterrealigi	itingUSX												
		MacBook Pro > 🙀 Meta	DeferredLightir	ngOSX				Run	2 of 2	00:00:09									+				
All	Cores	All Processes / Threads									The second second											1	
20	0	/0:04.240 00:04.260	00.	;04.280	, <u>, e e e e</u>	00:04.300	00:0	/4.320	0:00	4.340	00:04.3	60 	00:00	/4.380	00:0	J4.400	00	0:04.420		00:04.440		00:04	.460
► 🕟		Metal Application	uf S. g S.	d buff	Co s.	mmand buffe	ır 3 g		Co s.	mmand buffe	- 4	g			:ommand b s. g s.	Comm s g s	nand b	Commar s. g s.	nd b	Command s g s		Commar s g s	d b
		Metal User Callbacks		)																			
▶ �		Graphics Driver Activity							1											1			
► <b>O</b>		Render (GPU 1)	. P. P. <mark>g-buff</mark> r	er B	и ві		g-buffer	BI BI	g-buffer	Bl Bl	g-buffer	Bl Bl	g-buff	er B	В	g-buffer	Bl (	G g-buff	er G	G g-	ouffer	G G	
▶ 🕝		Scaler																					
		Display		On Disp			On Disr	ter:	On Display					On Dieplay	On	Disslay			On Disp		On Displa	2015	
		עמוקכוס		On Dispie	ly			Ay	On Display					In Display		Jispiay			UII Disp	lay	)n Dispia	y	
		Vsync																					
► (Ö)		CPU																					
▶ 🖸		File Activity: Stack Depth																					
Þ 🕜		All Heap & Anonymous VM																					
N Det	ails $\rangle$	Metal Command Buffer Subr	nissions												Inr	strument De	etail			00	E		
Commar	nd Buffe	er Submission	<ul> <li>Process</li> </ul>		Thr	ead			Frame						<u><u>k</u></u>			Trac	k Display				
Comma	nd buff	er 16 00:04.692.26	9 MetalDeferre	adLightinç	gOSX CV	DisplayLink::	:runlOThread (	Jxeb7a0	Frame 17	h								Style	9	Neste	d Data		٥
Comma	nd buffe	er 17 00:04.720.697	/ MetalDeferre	dLighting	JOSX CVL	JisplayLink::	.runIOThread 0	Jxeb7a0	Frame 18														
Comman	id buffe	er 18 00:04.743.557	/ MetalDeferre	dLighting	JOSX CVE	JisplayLink::r	runIOThread 0	xeb7a0	Frame 19														
Comman	id buffe	ar 19 00:04.770.681	/ MetalDeferrer	dLighting	JOSX CVE	JisplayLink::r	runIOThread 0	xeb7a0	Frame 20														
Comma	d butte	r 20 00:04.793.224	MetalDeferre		OSX CVL	DisplayLink:	/uniOThread (	xeb/au	Frame 21														
Comma	a burre	/r 22 00:04.843.62	7 MetalDeferre	adl ighting	OSX CV	DisplayLink:	runiOThread (	Aveb7a0	Frame 23														
Comma	nd buff	er 23 00:04.878.69	1 MetalDeferre	AdLighting	aOSX CV	DisplayLink:	runIOThread (	)xeb7a0	Frame 24														
Transferration of	ACC STRUCTURES	AND THE PARTY AND A PARTY AND	A MARTINE MARTINE CONTRACTOR	A THE REAL PROPERTY.	A CONTRACTOR OF THE OWNER	1. Provense	A STREET, A STREET, ST	100000000	State and the State of States														,
Commai	nd buffe	ər 24 00:04.903.88t	3 MetalDeferre	dLighting	OSX CVL	DisplayLink::	runIOThread 0.	xeb7a0	Frame 25														

• • •								MetalDe	eferredLig	htingOSX											
	🔲 MacBook Pro 👌 属 Meta	DeferredLighti	ingOSX				Run f	2 of 2	00:00:09									+ 0			
All Cores	All Processes / Threads																				
0 (	00:04.240 00:04.260	or	0:04.280	r	00:04.300	00:0/	4.320	00:0	4.340	00:0/	4.360	00:04.7	380	00:0	4.400	00	:04.420	00:0	4.440	01	0:04.460
		Comma	ad buff		mmand buffer	2		<u>FEELEE</u>	menand buf		TTTT		Com	mand h	Com	rrr r	Command		mmend b	Cor	amond h
	Metal Application	s. g		s.		g		S		1 4	g		s. g.		s g		s. g	200 S 0 S		S (	
	Metal User Callbacks								ļ					/							
▶ 🕜	Graphics Driver Activity																				
► <b>O</b>	Render (GPU 1)	. P. P. <mark>g-buff</mark>	er Bl.	в!		g-buffer	BI BI	g-buffer	Bi Bi	g-buffer	Bl Bl	g-buffer	B B		g-buffer	Bl G	g-buffer	G G	g-buffe	G	G
▶ 🗗	Scaler																				
	Display		On Display			On Displa	y	On Display				On	Display	On Di	splay			On Display	On Di	splay	On D
	Vsync																				
▶ 🙆	CPU																				
	File Activity: Stack Depth																				
▶ 🕜	All Heap & Anonymous VM																				
🕓 Details 👌	Metal Command Buffer Subm	issions											ſ	lnstr	rument De	tail		6	) 🙆 (E		
Command Buffe	er Submission^	Process		Threa	ad			Frame									Track D	)isplay			
Command buff	er 16 00:04.692.26§	MetalDeferre	∋dLightingC	JSX CVDi	isplayLink::ru	InIOThread 0x	keb7a0	Frame 17									Style		Nested Dat	ίa	0
Command buffe	er 17 00:04.720.697	MetalDeferre	dLightingC	JSX CVDi	isplayLink::ru	nIOThread 0x	keb7a0	Frame 18													
Command buffe	er 18 00:04.743.557	MetalDeferrer	dLightingO	JSX CVDi	splayLink::ru	nIOThread 0x/	.eb7a0	Frame 19													
Command buffe	er 19 00:04.770.681	MetalDeferrer	dLightingO	ISX CVDir	splayLink::rur	nIOThread 0xr	.eb7a0	Frame 20													
Command buffe	er 20 00:04.793.224	MetalDeferrer	dLightingO	SX CVDI	splayLink::rur	nIOThread Oxe	eb7a0	Frame 21													
Command buff	3r 21     00:04.820.592       00:04.843.62	MetalDeferre		SX CVDR	splayLink::rui	AlOThread Uxe	eb/au	Frame 22													
Command buff	or 22 00:04.878.691	MetalDeferre	alighting	SX CVDI	SplayLinku	INOThread 0	207a0	Frame 24													
Command buff	er 24 00:04.903.88f	MetalDeferre	edLighting(	USX CVD	isolavLink::ru	inIOThread 0y	xeb7a0	Frame 25													
Command buff	er 25 00:04.928.767	2 MetalDeferre	adLighting	JSX CVD	isplayLink::ru	unIOThread 0y	xeb7a0	Frame 26													

		Metal Application	uf Command bu s. g s.
		Metal User Callbacks	
► <b>③</b> ► <b>O</b>		Graphics Driver Activity	
	► O	Render (GPU 1)	. P. P. g-buffer
► ©	▶ 🗇	Scaler	
► <b>©</b>		Display	On
Note the Comman		Vsync	
Commar Commar Commar	▶ 🗿	CPU	
Commar Commar Commar Commar		File Activity: Stack Depth	
Commar Commar Commar	Þ 🕜	All Heap & Anonymous VM	



		Metal Application	uf Command bu s. g s.
		Metal User Callbacks	
► <b>③</b> ► <b>O</b>		Graphics Driver Activity	
	► O	Render (GPU 1)	. P. P. g-buffer
► ©	▶ 🗇	Scaler	
► <b>©</b>		Display	On
Note the Comman		Vsync	
Commar Commar Commar	▶ 🗿	CPU	
Commar Commar Commar Commar		File Activity: Stack Depth	
Commar Commar Commar	Þ 🕜	All Heap & Anonymous VM	



		Metal Application	uf Command bu s. g s.
		Metal User Callbacks	
► <b>③</b> ► <b>O</b>		Graphics Driver Activity	
	► O	Render (GPU 1)	. P. P. g-buffer
► ©	▶ 🗇	Scaler	
► <b>©</b>		Display	On
Note the Comman		Vsync	
Commar Commar Commar	▶ 🗿	CPU	
Commar Commar Commar Commar		File Activity: Stack Depth	
Commar Commar Commar	Þ 🕜	All Heap & Anonymous VM	

![](_page_243_Figure_2.jpeg)

		Metal Application	uf Command bu s. g s.
		Metal User Callbacks	
► <b>③</b> ► <b>O</b>		Graphics Driver Activity	
	► O	Render (GPU 1)	. P. P. g-buffer
► ©	▶ 🗇	Scaler	
► <b>©</b>		Display	On
Note the Comman		Vsync	
Commar Commar Commar	▶ 🗿	CPU	
Commar Commar Commar Commar		File Activity: Stack Depth	
Commar Commar Commar	Þ 🕜	All Heap & Anonymous VM	

![](_page_244_Figure_2.jpeg)

•••			Ma Ma
	MacBook Pro > 🔥 Meta	DeferredLightingOSX	Run 2 of 2
All Co	All Processes / Threads		
.192	00:09.198 00:09.204	00:09.210 00:09.216 00:09.222	00:09.228
		Command buffer 192	Command buffer 193
	Motal Application	shadow g-buffer B Re B. Re	shado g-buffer B
	Metal Application	shado s. s	shado s. s
	Metal User Callbacks		
▶ 😯	Graphics Driver Activity		
► O	Render (GPU 1)	GPU Exec G. GPU Exec g-buffer	GI
▶ 🔁	Scaler		
	Display		
	Vsync		
▶ (0)	CPU	RR R	® ® ® ® ® ®
▶ 💽	File Activity: Stack Depth		
▶ 🕜	All Heap & Anonymous VM		
🕓 Details	> Metal Command Buffer Subm	issions	
Command E	Buffer Submission^	Process Thread	Fram

Command Buffer	Submission^	Process	Thread		Fram
Command buffer 182	00:09.021.404	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 183	00:09.038.229	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 184	00:09.054.835	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 185	00:09.071.433	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 186	00:09.091.758	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 187	00:09.120.807	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 188	00:09.138.427	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 189	00:09.154.806	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 190	00:09.172.720	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 191	00:09.192.612	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram

	Instrument Detail		o 🙆 E	
e		Track Displa	у	
e 183		Style	Nested Data	0
e 184		-		
e 185				
e 186				
e 187				
e 188				
e 189				
e 190				
e 191				
e 192				

R	PP	R	<b>RR</b>	P	R	R	R	P	R	R	R	R	R	R	R	R	R	R	R	R	R	P	R	R	R	

00:09.234 00:09.	240 00:09.2	46	00:09.252	00:09.258	3	00:09.264	3 6 3 6	00:09.270	12 11 12 11	00:09.276
	Command buffer 194				Command	buffer 195				
3 Re B Re	shado g-buffer	B. Re B.	Ren		shado g	g-buffer	B Re	B Re		
s	shado s.	S S			shado	S.	s	S		
								~		
GPU Exec G. GPU Exec g-	-buffer		GPU Exec G. GF	PU Exec	g-buffe	€r		GF	U Exec	G. GPU Ex

letalDeferredLig	ghtingOSX	

00:00:09

+ 🖸 🗏 🖾 🗖 🗖

R

![](_page_245_Picture_9.jpeg)

• • •								ľ	M
	MacBook Pro > 👰 Meta	alDeferredLightin	ngOSX					Run 2 d	of 2
All C	Cores All Processes / Threads	;							
.192	00:09.198 00:09.204	00:09.210		00:09.216		0:09.222	,	0:09.228	a r
		Command buffe	er 192				Comman	d buffer 19	93
	Metal Application	shadow g-bu	iffer	B Re B.	Re		shado	g-buffer	В.
	metal Application	shado s.		s s			shado	s.	s
	Metal User Callbacks								
► 🐨	Graphics Driver Activity								
► O	Render (GPU 1)	GPU	I Exec	G. GPU Exec.	g-l	ouffer			G
	Scaler								
	Display								
	Vsync								
Þ (Ö)	CPU	RR R					R		R
	File Activity: Stack Depth								
	All Heap & Anonymous VM								
N Detai	ils 〉Metal Command Buffer Subm	nissions							

Command Buffer	Submission^	Process	Thread		Fram
Command buffer 182	00:09.021.404	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 183	00:09.038.229	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 184	00:09.054.835	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 185	00:09.071.433	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 186	00:09.091.758	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 187	00:09.120.807	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 188	00:09.138.427	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 189	00:09.154.806	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 190	00:09.172.720	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram
Command buffer 191	00:09.192.612	MetalDeferredLightingOSX	CVDisplayLink::runIOThread	0xeb7a0	Fram

R0       B       R0       g       B       R0       C								
Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter         Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Strutter       Image: Struter       Image: Strutter       <	Re B	Re	shado g-buffer	B. Re B Ren	sha	ado g-buffer	B Re B Re	
U Exec. C. C. OP J Exec. g-buffer GPU Exec. C. C. OP U Exec. g. buffer OP J Exec. C. OP U C Exec. C. C. OP J Exec. g-buffer GPU Exec. C. OP U Exec. g. buffer OP J Exec. C. OP U C Exec. C. C. OP J Exec. g-buffer GPU Exec. C. OP U C Exec. C. C. OP J Exec. G. OP U Exec. G. OP U C Exec. C. C. OP J Exec. C. OP U C Exec. C. C. OP J Exec. C. OP U C Exec. C. C. OP J Exec. C. OP U C Exec. C. C. OP J Exec. C. OP U C Exec. C. C. OP J Exec. C. OP U C Exec. Exec. C. OP U	S		shado s.	S S	sha		S S	
U Exoc 0. GP I Exoc g-buffer GPU Exoc G. GPU Exoc GPU Exo					-			
U Exec G. OP J Exec g-buffer GPU Exec G. OPU Exec g-buffer 0P J Exec G. OPU I 2 @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @		1		n	1			1
U Exec G. GP J Exec g-buffer GPU Exec G. GPU Exec g-buffer GP J Exec G. GPU Exec GPU								
U Exec G. GP J Exec g-buffer GPU Exec g-buffer GP U Exec g-buffer GP U Exec G. GPU I Exec GPU I Exec G. GPU I Exec GP								
U Exec., G. GP J Exec., GR J E				-			4	
U Exec. G. GP / Exec. g-buffer GPU Exec. g-buffer GPU Exec. g-buffer GPU Exec. g. GPU I I Exec. g. GPU I I I I I I I I I I I I I I I I I I I								
• • • • • • • • • • • • • • • • • • •	U Exec	G. GPIJ Exec	g-buffer	GPU Exec (	G. GPU Exec			U Exec G. GPU E
Image: Serie Seri							<u> </u>	
Image: Serie Seri								
P       P								
P @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @								
Instrument Detail       Image: Comparison of the comparison of								
Instrument Detail       Image: Comparison of the comparison of								
Instrument Detail       Image: Construction of the second o	PRR	RRRR	R R R R R R R R	R R R R R R R R	RRRR			R
Instrument Detail								
Instrument Detail   Image: Distrument Detail								
Instrument Detail Instrument Detail IRAS								
Instrument Detail   Instrument Detail        Instrument Detail								
Instrument betan     Track Display       183     Style       184     Nested Data       185     Nested Data       186     Nested Data       187     Nested Data       188     Nested Data       190     Nested Data       191     Nested Data					rument Detail			5
Item     Track Display       183     Style       184     Nested Data       185     Nested Data       186     Nested Data       187     Nested Data       188     Nested Data       190     Nested Data       191     Nested Data								
Style     Nested Data       183     Style       184     185       186     187       188     189       189     190       191     191       192     192	)					Track Di	splay	
185	e 183					Style	Nested Da	ata 🗘
186         187         188         189         190         191         192	185							
2 187 2 188 2 189 2 190 2 191 2 192 1 0 0 1 0 1	e 186							
2 188 2 189 2 190 2 191 2 192 100	e 187							
2 189 2 190 2 191 2 192	e 188							
e 190 e 191 e 192 	e 189							
e 191 e 192	e 190							
192	e 191							
	102							
	192							

00:09.252

00:09.246

ommand buffer 194

00:09.258

00:09.264

etalDeferredLightingOSX

00:09.240

00:00:09

00:09.234

+ 🖸 🚍 🛄 🔲

00:09.270

00:09.276

![](_page_246_Picture_5.jpeg)

![](_page_247_Figure_1.jpeg)

![](_page_248_Figure_1.jpeg)

![](_page_249_Figure_1.jpeg)

#### Performance Observations

![](_page_250_Picture_1.jpeg)

Timestamp^	Narrative
00:14.121.430	GPU spent 4.75 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.127.680	"Render Encoder 0" took 6.83 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.136.490	GPU spent 3.11 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.140.182	GPU spent 4.85 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.152.551	"Render Encoder 0" took 7.73 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.162.695	GPU spent 2.93 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.166.304	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.168.131	"Render Encoder 0" took 7.72 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.177.821	GPU spent 2.87 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.180.735	"Render Encoder 0" took 7.96 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.181.442	GPU spent 4.77 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.189.894	GPU spent 3.04 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.193.590	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2.

00:13.640 00:13.645 00:13.650 00:13.655 00:13.660 00:13.665 mmand Buffer 0 Command Buffer 0 Render Encoder 0 lender Encoder 0 nder Enc.. ender Encoder 0 der Encoder 0 Re Display 🕒 v Instrument Detail

.20 ms)

.12 μs)

2.20 ms)

.12 μs)

2.20 ms)

.12 μs)

2.20 ms)

.12 μs)

.20 ms)

![](_page_250_Picture_34.jpeg)

#### Performance Observations

![](_page_251_Picture_1.jpeg)

Timestamp^	Narrative
00:14.121.430	GPU spent 4.75 ms executing Fragment for "Render Encoder 0" (Upper bound 2
00:14.127.680	"Render Encoder 0" took 6.83 ms to complete (Upper bound 724.08 $\mu s$ )
00:14.136.490	GPU spent 3.11 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.140.182	GPU spent 4.85 ms executing Fragment for "Render Encoder 0" (Upper bound 2
00:14.152.551	"Render Encoder 0" took 7.73 ms to complete (Upper bound 724.08 $\mu s$ )
00:14.162.695	GPU spent 2.93 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.166.304	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2
00:14.168.131	"Render Encoder 0" took 7.72 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.177.821	GPU spent 2.87 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.180.735	"Render Encoder 0" took 7.96 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.181.442	GPU spent 4.77 ms executing Fragment for "Render Encoder 0" (Upper bound 2
00:14.189.894	GPU spent 3.04 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.193.590	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2

00:13.640 00:13.645 00:13.650 00:13.655 00:13.660 00:13.665 mmand Buffer 0 Command Buffer 0 Render Encoder 0 lender Encoder 0 nder Enc.. ender Encoder 0 der Encoder 0 Re Display 🕒 v Instrument Detail

.20 ms)

.12 μs)

2.20 ms)

.12 μs)

2.20 ms)

.12 μs)

2.20 ms)

.12 μs)

.20 ms)

![](_page_251_Picture_20.jpeg)


Timestamp^	Narrative
00:14.121.430	GPU spent 4.75 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.127.680	"Render Encoder 0" took 6.83 ms to complete (Upper bound 724.08 µs)
00:14.136.490	GPU spent 3.11 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.140.182	GPU spent 4.85 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.152.551	"Render Encoder 0" took 7.73 ms to complete (Upper bound 724.08 µs)
00:14.162.695	GPU spent 2.93 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.166.304	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.168.131	"Render Encoder 0" took 7.72 ms to complete (Upper bound 724.08 µs)
00:14.177.821	GPU spent 2.87 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.180.735	"Render Encoder 0" took 7.96 ms to complete (Upper bound 724.08 µs)
00:14.181.442	GPU spent 4.77 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.189.894	GPU spent 3.04 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.193.590	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2.

### .12 μs) 2.20 ms) .12 μs) .20 ms)

2.20 ms)

.12 μs)

.12 μs) 2.20 ms)

.20 ms)

NEW



All Co	res Al	l Processes / Threads					
.185	00	00:14.190	5 00:14.:	200	T. T. F	00:14.205	ĩĩ
						Command B	uffer
		Metal Application	icoder 0	11			F
		Metal Application					
		Metal User Callbacks					
▶ 🎸		Shader Compilation					
		Graphics Driver Activity					
Þ		Vertex			Re	nder Enc	
		Fragment	0 R. R				Rei
		Scaler					
		Display			Display		
		Vsync					
😵 Deta	ails 〉Ob	servations					
Tim	nestamp^	Narrative					
00:	14.203.049	Surface was displayed for 1	8.84 ms				
00:	14.204.510	GPU spent 2.92 ms executi	ng Vertex for "Rende	r Encod	er 0" (Up	per bound	167.
00:	14.208. <mark>0</mark> 99	GPU spent 4.76 ms executi	ng Fragment for "Rer	nder End	oder 0"	(Upper bou	ind 2.
00:	14.208.494	"Render Encoder 0" took 5.	88 ms to complete (l	Jpper b	ound 724	4. <mark>08 μs</mark> )	
00:	14.216.715	"Render Encoder 0" took 9.	88 ms to complete (l	Jpper b	ound 724	4.08 μs)	
00:	14.219.879	GPU spent 2.91 ms executi	ng Vertex for "n/a" (l	Jpper bo	ound 79.	92 µs)	
00:	14.223.523	GPU spent 4.75 ms executi	ng Fragment for "Rer	nder End	oder 0"	(Upper bou	ind 2.
00:	14.231.322	"Render Encoder 0" took 4.	11 ms to complete (l	Jpper b	ound 724	4.08 μs)	
00:	14.235.832	"Render Encoder 0" took 7.	07 ms to complete (l	Jpper b	ound 724	4.08 μs)	
00:	14.243.844	GPU spent 2.97 ms executi	ng Vertex for "Rende	r Encod	er 0" (Up	per bound	167.
00:	14.247.236	"Render Encoder 0" took 6.	50 ms to complete (l	Jpper b	ound 724	4.08 μs)	
00:	14.247.505	GPU spent 4.78 ms executi	ng Fragment for "Rer	nder End	oder 0"	(Upper bou	ind 2.
00:	14.254.728	GPU spent 3.48 ms executi	ng Vertex for "Rende	r Encod	er 0" (Up	per bound	167.



.12 μs) 2.20 ms)

2.20 ms)

.12 μs)

2.<mark>20 ms)</mark>

.12 µs)

All Cores	All Processes / Threads				
.185	00:14.190 00	):14.195	00:14.2	200	00:14.205
					Command Buffer 0
	Metal Applicati	on ico	oder 0	11111	Render E
	Metal Applicati	on			
	Metal User Callbac	ks			
▶ 🐼	Shader Compilati	on			
	Graphics Driver Activ	ity			
► O	Vert	ex			Render Enc
	Fragme	ent 0	R		Render End
A	Sca	ler			
	Displ	lay		Disp	lay
	Vsy	nc			
😵 Details	> Observations				
Timesta	amp^ Narrative				
00:14.20	03.049 Surface was displayed	d for 18.	84 ms		
00:14.20	04.510 GPU spent 2.92 ms e	xecuting	Vertex for "Rende	r Encoder 0	" (Upper bound 167.12 μs)
00:14.20	08.099 GPU spent 4.76 ms e	xecuting	Fragment for "Ren	nder Encode	er 0" (Upper bound 2.20 ms)
00:14.20	8.494 "Render Encoder 0" to	ook 5.88	ms to complete (L	Jpper bound	d 724.08 μs)
00:14.21	6.715 "Render Encoder O" to	ook 9.88	ms to complete (L	Jpper bound	d 724.08 μs)
00:14.21	9.879 GPU spent 2.91 ms e	xecuting	Vertex for "n/a" (U	Jpper bound	d 79.92 μs)
00:14.22	3.523 GPU spent 4.75 ms e	xecuting	Fragment for "Ren	nder Encode	er 0" (Upper bound 2.20 ms)
00:14.23	31.322 "Render Encoder 0" to	ook 4.11	ms to complete (L	Jpper bound	d 724.08 μs)
00:14.23	35.832 "Render Encoder 0" to	ook 7.07	ms to complete (L	Jpper bound	d 724.08 μs)
00:14.24	13.844 GPU spent 2.97 ms e	xecuting	Vertex for "Rende	r Encoder 0	" (Upper bound 167.12 μs)
00:14.24	17.236 "Render Encoder 0" to	ook 6.50	ms to complete (L	Jpper bound	d 724.08 μs)
00:14.24	17.505 GPU spent 4.78 ms e	xecuting	Fragment for "Ren	nder Encode	er 0" (Upper bound 2.20 ms)
00:14.25	64.728 GPU spent 3.48 ms e	xecuting	Vertex for "Rende	r Encoder 0	" (Upper bound 167.12 µs)





0:11.004	00:11.007	00:11.010	00:11.013	00:11.0	16	00:11.019
			Command Bi	uffer 47		
			Command Bu			
			Render Enco			
	Render Encode	er O		Re	Rend	
				• Instru	ument Detail	
6.46.46.64.6						
ouild time						
0 ms)						
0						
2 μs)						
0 ms)						
2 µs)						



00:11.047.875 "Render Encoder 0" took 8.19 ms to complete (Upper bound 724.08 μs)

00:11.061.533 "Render Encoder 0" took 7.25 ms to complete (Upper bound 724.08 μs)

00:11.004 00:11.007 00:11.010 00:11.013 00:11.016 00:11.01 ● v Instrument Detail

00:11.058.032 GPU spent 2.98 ms executing Vertex for "Render Encoder 0" (Upper bound 167.12 μs)

00:11.062.353 GPU spent 4.92 ms executing Fragment for "Render Encoder 0" (Upper bound 2.20 ms)

All Cores All Processes / Threads								
	00:14.841		00:14.844		00:14.847		1.15	00:14.850
						Co	mmand	Buffer 9
		Metal App	lication		1	111111 Re	nder En	coder 0
	1						(1) Million ( Sect.)	
		Metal App	lication					
		Metal User Ca	llbacks					
r w		Shader Com	pilation					
	-				1			
					С.			
		Graphics Driver	Activity					
					R.	-		
			<b>.</b>					
P U			Vertex					
		Fra	agment					
			Scaler					
			Display					
			Vsync					
😵 Det	ails 👌 Ob	servations						
Tin	nestamp^	Narrative						
00:	14.849.935	GPU spent 2.89	ms executi	ng Vertex for "n	/a" (Upper l	bound 79.	92 µs)	
00:	14.853.557	GPU spent 5.10	ms executi	ng Fragment for	"Render Er	ncoder 0"	(Upper	bound 2.2
00:	14.859.192	"Render Encoder	0" took 6.	01 ms to compl	ete (Upper	bound 72	4.08 µs	:)
00:	14.866.129	"Render Encoder	0" took 4.	03 ms to compl	ete (Upper	bound 72	4.08 μs	;)
00:	14.867.172	GPU spent 2.91	ms executi	ng Vertex for "n	/a" (Upper I	bound 79.	92 µs)	hound Q (
00:	14.8/0.767	"Render Encoder	0" took 7	34 ms to compl	ete (Unner	bound 72		
00:	14.873.727	"Render Encoder	0" took 4	57 ms to compl	ete (Upper	bound 72	4.00 µs	-) -)
00:	14.883.085	GPU spent 2.91	ms executi	ng Vertex for "n	/a" (Upper I	bound 79.	92 µs)	
00:	14.886.585	GPU spent 4.91	ms executi	ng Fragment for	"Render Er	ncoder 0"	(Upper	bound 2.2
00:	14.890.509	"Render Encoder	0" took 6.	34 ms to compl	ete (Upper	bound 72	4.08 μs	;)
00:	14.897.942	"Render Encoder	0" took 5.	73 ms to compl	ete (Upper	bound 72	4.08 μs	;)
00:	14.898.805	GPU spent 2.88	ms <mark>execu</mark> ti	ng Vertex for "n	/a" (Upper l	bound 79.	92 µs)	

00:14 952	00:14 85	6	00:14 850		00:14 962
00.14.853	00.14.85		00.14.859		00,14.862
	Command Buff	fer 0			
			Render Encode	er O	
Render	Encoder 0	F	Ren	Render	
Scaler					
Display					
				● Instrum	ent Detail

.20 ms)

.20 ms)

All Co	All Cores All Processes / Threads							
1 1 1	00:14.841		00:14.844		00:14.847		00:14.850	
						Command	Buffer 9	
		Metal Application			1	Render Er	ncoder 0	
	1.7							
		Metal App	lication					
		Metal User Ca	llbacks					
		Shader Com	pilation					
					C			
		Graphics Driver	Activity					
					R			
							-	
► O			Vertex					
		Fra	agment					
A			Scaler					
			Display					
			Vsync					
😵 Det	ails ) Obs	servations						
Tin	nestamp^	Narrative						
00:	14.849.935	GPU spent 2.89	ms executii	ng Vertex for "n	/a" (Upper bo	und 79.92 μs)	(	
00:	14.853.557	GPU spent 5.10	ms executir	ng Fragment for	"Render Enco	oder 0" (Uppe	r bound 2.2	
00:	14.859.192	"Render Encoder	0" took 6.0	01 ms to comple	ete (Upper bo	und 724.08 µ	s)	
00:	14.866.129	"Render Encoder	0" took 4.0	3 ms to comple	ete (Upper bo	und 724.08 µ	s)	
00:	14.867.172	GPU spent 2.91	ms executir	ng Vertex for "n,	/a" (Upper bo	und 79.92 µs)		
00:	14.870.767	GPU spent 4.71	ms executii	ng Fragment for	"Render Enco	oder 0" (Upper	r bound 2.2	
00:	14.873.727	"Render Encoder	0" took 7.3	34 ms to comple	ete (Upper bo	und 724.08 µ	s)	
00:	14.882.438	"Render Encoder	0" took 4.	o/ ms to comple	ete (Upper bo	und 724.08 µ	s)	
00:	14.883.085	GPU spent 2.91	ms executio	ng Vertex for "n/	a" (Upper bo	una 79.92 µs)	have 10 d	
00:	14.886.585	GPU spent 4.91	ms execution	ig Fragment for	Render Enco	oder U" (Upper	bound 2.2	
00:	14.890.509	"Render Encoder	0" took 6.3	34 ms to comple	ete (Upper bo	una 724.08 µ	s)	
00:	14.897.942	Render Encoder	U 100K 5.	s ms to comple	ete (Opper bo	una 724.08 µ	5)	
00:	14.898.805	GPU spent 2.88 i	ms execution	ng Vertex for "n/	/a" (Upper bo	und 79.92 µs)		



20 ms)



Timestamp^	Narrative
00:14.121.430	GPU spent 4.75 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.127.680	"Render Encoder 0" took 6.83 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.136.490	GPU spent 3.11 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.140.182	GPU spent 4.85 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.152.551	"Render Encoder 0" took 7.73 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.162.695	GPU spent 2.93 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.166.304	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.168.131	"Render Encoder 0" took 7.72 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.177.821	GPU spent 2.87 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.180.735	"Render Encoder 0" took 7.96 ms to complete (Upper bound 724.08 $\mu$ s)
00:14.181.442	GPU spent 4.77 ms executing Fragment for "Render Encoder 0" (Upper bound 2.
00:14.189.894	GPU spent 3.04 ms executing Vertex for "Render Encoder 0" (Upper bound 167.
00:14.193.590	GPU spent 4.80 ms executing Fragment for "Render Encoder 0" (Upper bound 2.

00:13.640 00:13.645 00:13.650 00:13.655 00:13.660 00:13.665 mmand Buffer 0 Command Buffer 0 Render Encoder 0 ender Encoder 0 nder Enc.. ender Encoder 0 der Encoder 0 Display 🕒 v Instrument Detail

.20 ms)

.12 μs)

2.20 ms)

.12 μs)

2.20 ms)

.12 μs)

2.20 ms)

.12 μs)

.20 ms)

### Demo GPU Frame Debugger

### What's New in Metal Tools

Metal System Trace GPU Overrides GPU Frame Debugger



### What's New in Metal Tools

Metal System Trace GPU Overrides GPU Frame Debugger

Extended Validation



### What's New in Metal Tools

Metal System Trace GPU Overrides GPU Frame Debugger

- Extended Validation
- Metal Library Projects



Resource Heaps and Memoryless Render Targets

Tessellation

Improved Tools

Resource Heaps and Memoryless Render Targets

Tessellation

Improved Tools

Resource Heaps and Memoryless Render Targets

Improved Tools

Function Specialization and Function Resource Read-Writes

Wide Color and Texture Assets

Additions to Metal Performance Shaders

Tessellation

# More Information https://developer.apple.com/wwdc16/604

### Related Sessions

Adopting Metal, Part 1

Adopting Metal, Part 2

What's New in Metal, Part 2

Advanced Metal Shader Optimization

Nob Hill	Tuesday 1:40PM
Nob Hill	Tuesday 3:00PM
Pacific Heights	Wednesday 1:40PM
Pacific Heights	Wednesday 3:00PM



Metal Lab

### Graphics, Games, and Media Lab B Thursday 12:00PM

