#WWDC18

Metal Game Performance Optimization

Session 612

Guillem Viñals Gangolells, GPU Software Performance
Ohad Frenkel, Game Technologies

Develop awesome games

Develop *technically* awesome games

The Talos Principle

Croteam / Devolver Digital

Features supported in iOS

Physically-based HDR rendering

Dynamic shadows

Real-time reflections

Multi-threaded rendering

Full-screen anti-aliasing

Post processing effects

Profiling Tools

Frame Pacing

Thread Priorities

Thermal States

Unnecessary GPU Work

Profiling Tools

Profile early and often

Know Your Tools

Instruments

Xcode Metal Frame Debugger

Game Performance Template

NEW

Game Performance

Game Performance Template

NEW

Game Performance Template

NEW

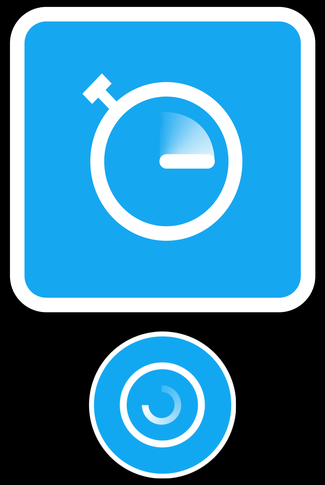
System Trace

Game Performance Template

NEW

System Trace

Time Profiler

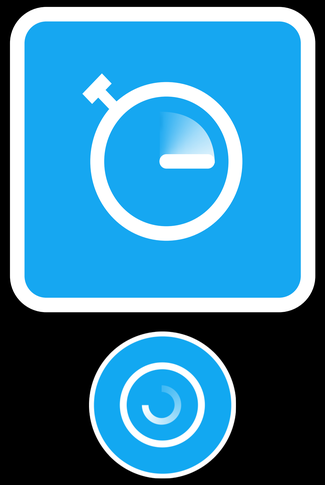


Game Performance Template

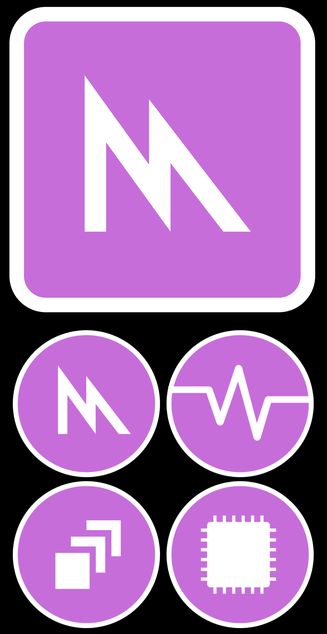
NEW

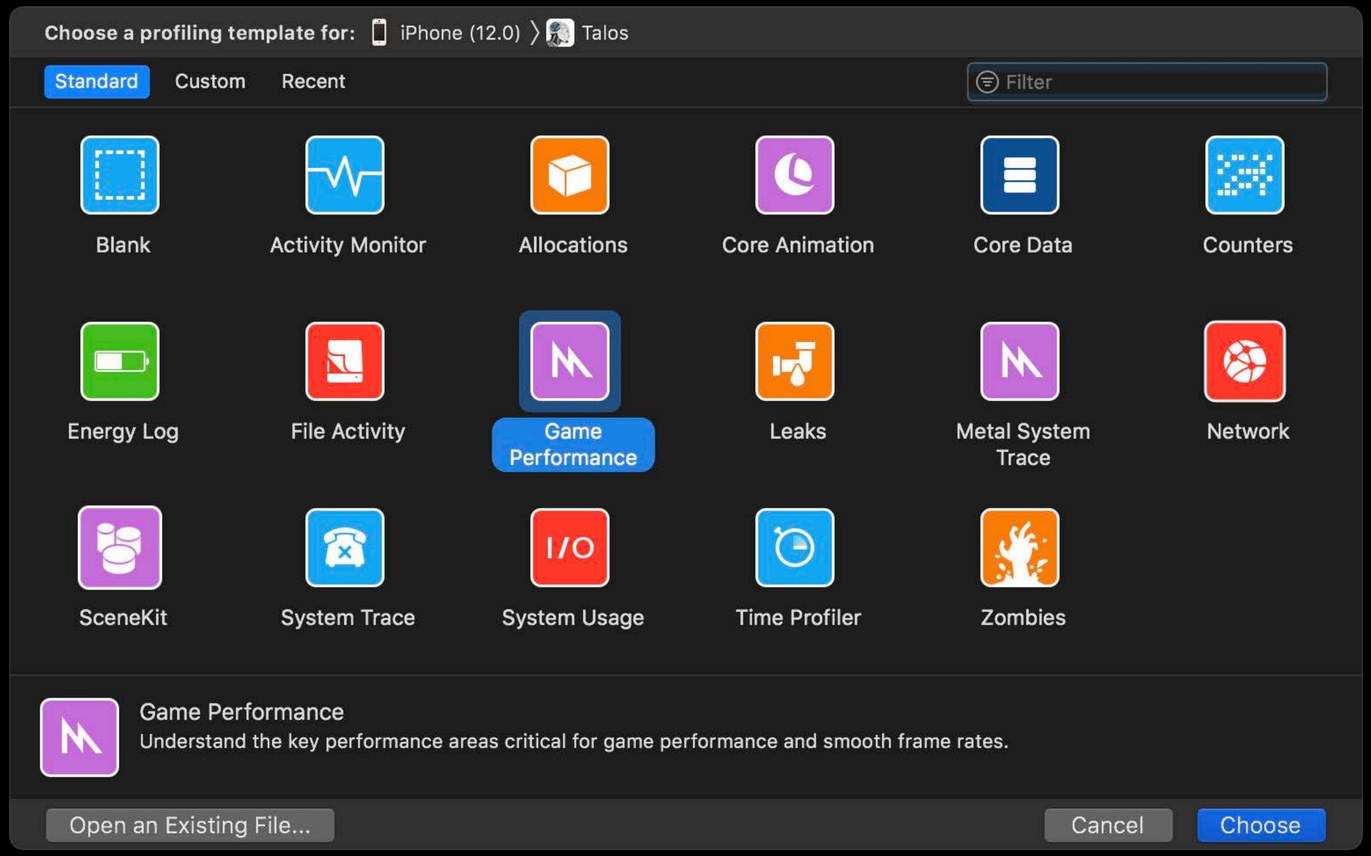
System Trace

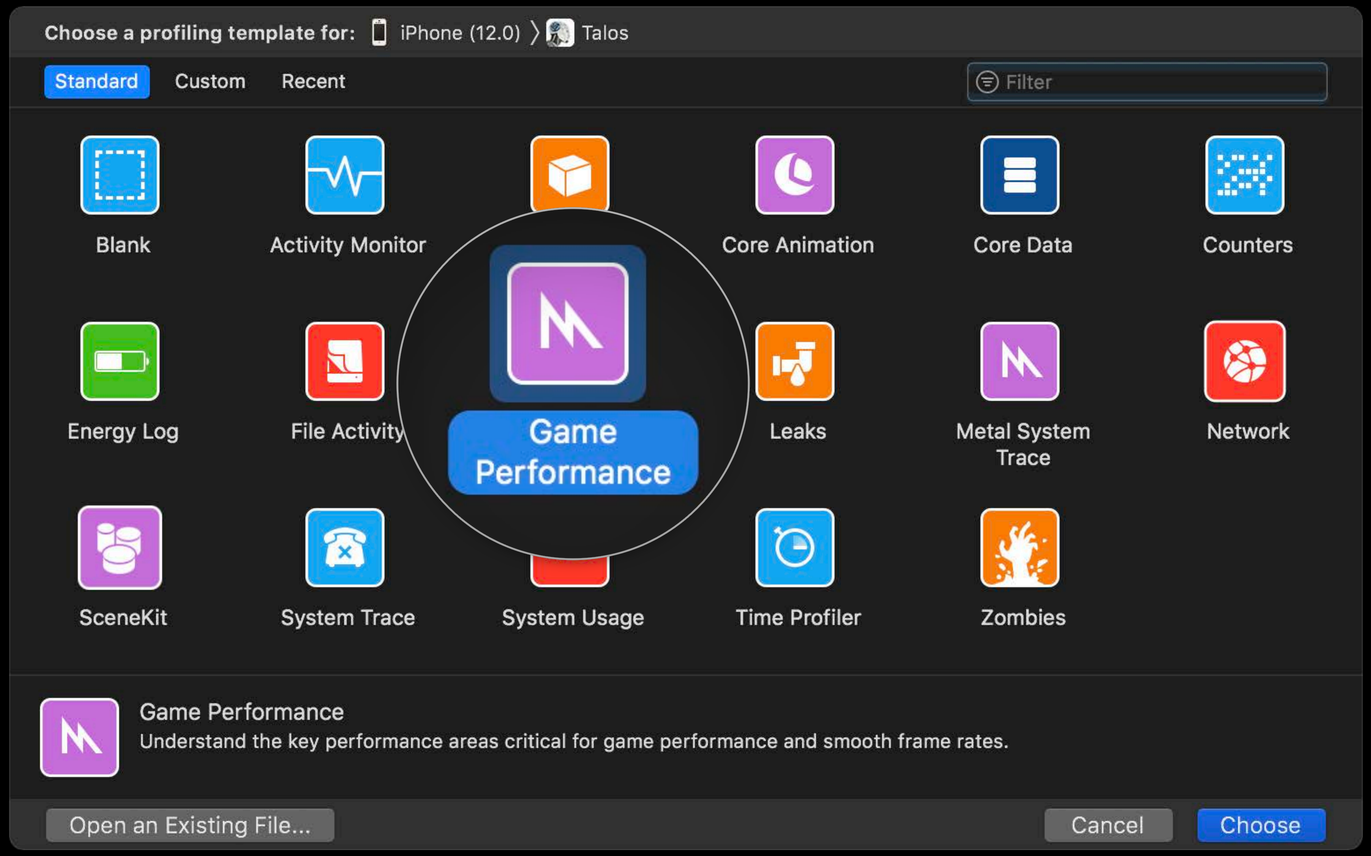
Time Profiler

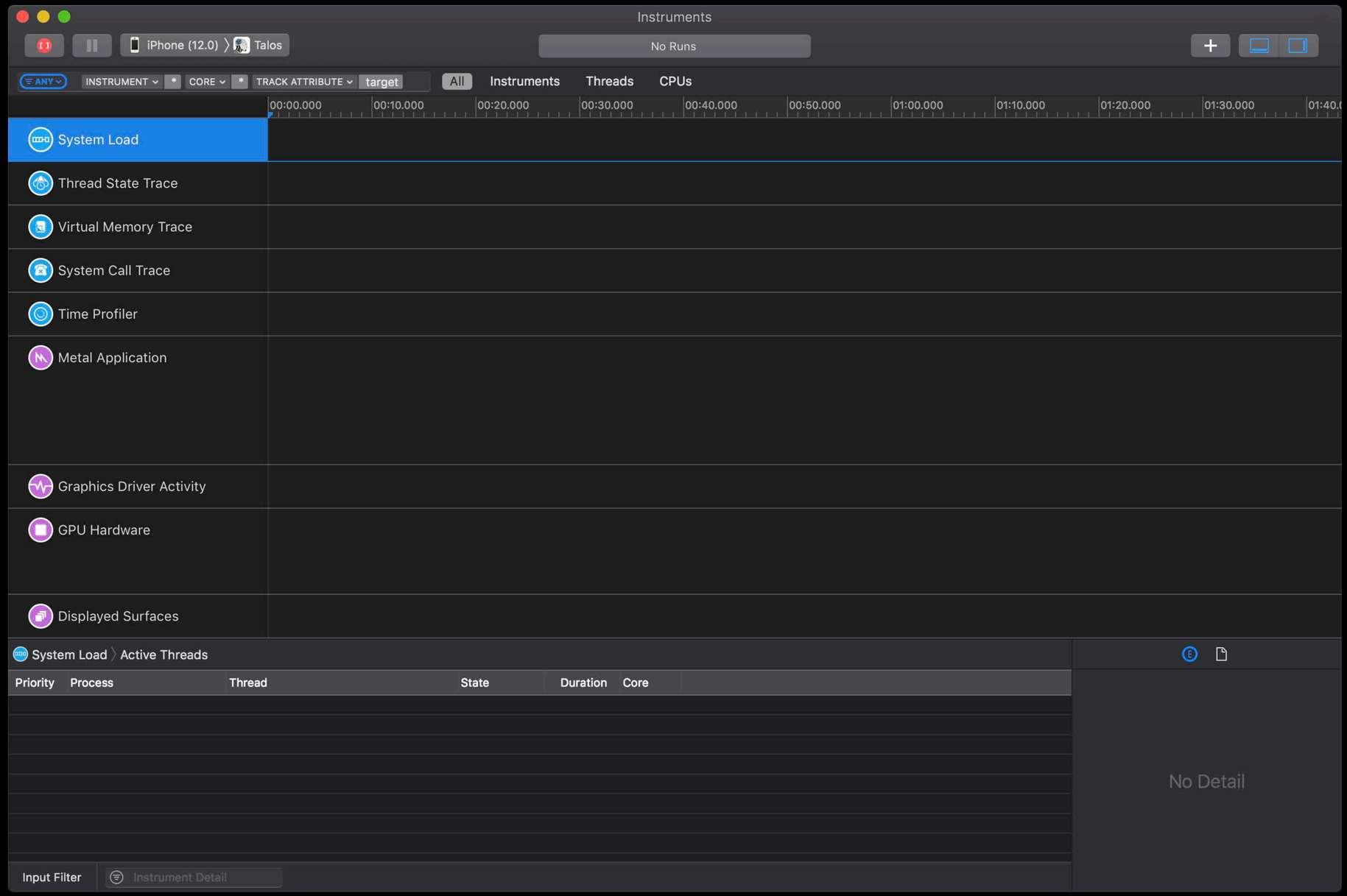


Metal System Trace

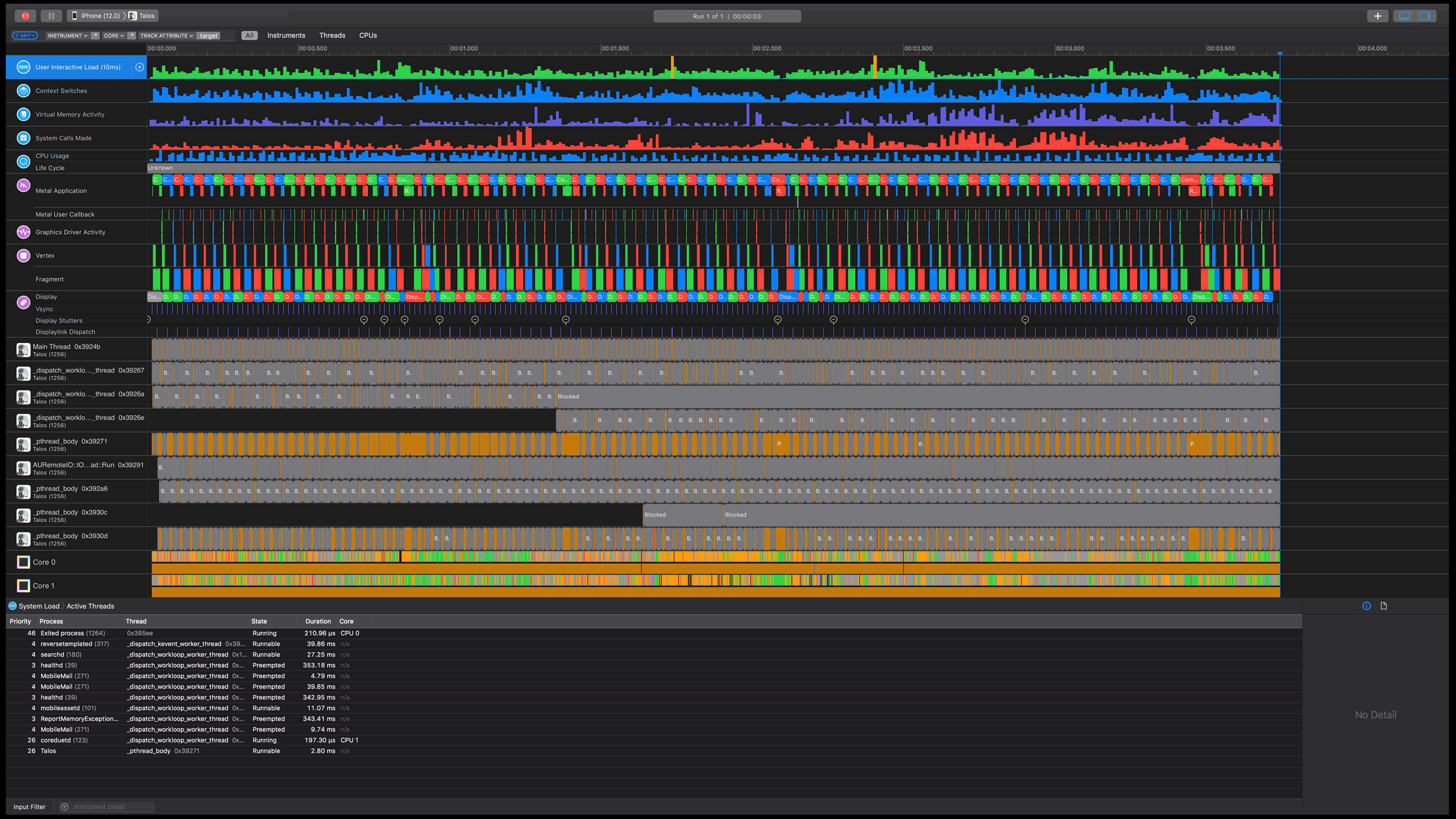



Choose a profiling template for: iPhone (12.0) Talos
Standard
Custom
Recent
Filter
Blank
Activity Monitor
Allocations
Core Animation
Core Data
Counters
Energy Log
File Activity
Game Performance
Leaks
Metal System Trace
Network
SceneKit
System Trace
System Usage
Time Profiler
Zombies
Game Performance
Understand the key performance areas critical for game performance and smooth frame rates.
Open an Existing File...
Cancel
Choose


Choose a profiling template for: iPhone (12.0) Talos
Standard
Custom
Recent
Filter
Blank
Activity Monitor
Core Animation
Core Data
Counters
Energy Log
File Activity
Game Performance
Leaks
Metal System Trace
Network
SceneKit
System Trace
System Usage
Time Profiler
Zombies
Game Performance
Understand the key performance areas critical for game performance and smooth frame rates.
Open an Existing File...
Cancel
Choose


Instruments
iPhone (12.0)
Talos
No Runs
ANY
INSTRUMENT
CORE
TRACK ATTRIBUTE
target
All
Instruments
Threads
CPUs
00:00.000
00:10.000
00:20.000
00:30.000
00:40.000
00:50.000
01:00.000
01:10.000
01:20.000
01:30.000
System Load
Thread State Trace
Virtual Memory Trace
System Call Trace
Time Profiler
Metal Application
Graphics Driver Activity
GPU Hardware
Displayed Surfaces
System Load
Active Threads
Priority
Process
Thread
State
Duration
Core
No Detail
Input Filter
Instrument Detail

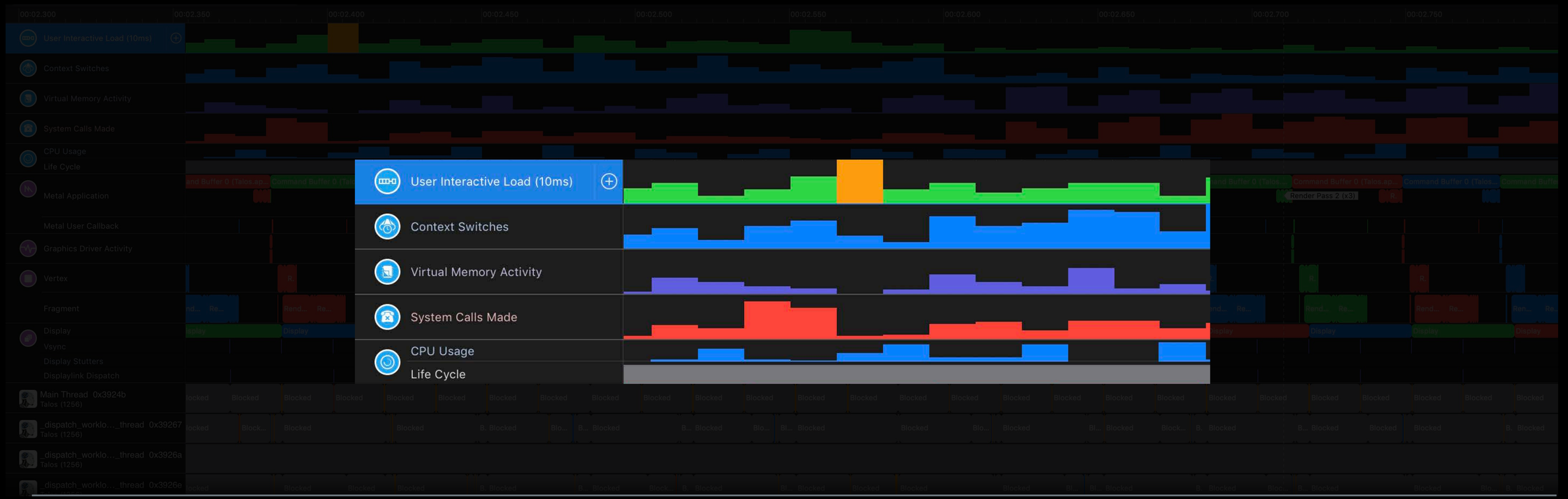

Instruments
iPhone (12.0) 〉 Talos
Run 1 of 1 | --:--:--
Recording in Windowed Mode


iPhone (12.0)
Talos
Run 1 of 1 | 00:00:03
INSTRUMENT
CORE
TRACK ATTRIBUTE
target
All
Instruments
Threads
CPUs
00:00.000
00:00.500
00:01.000
00:01.500
00:02.000
00:02.500
00:03.000
00:03.500
00:04.000
User Interactive Load (10ms)
Context Switches
Virtual Memory Activity
System Calls Made
CPU Usage
Life Cycle
Unknown
Metal Application
Metal User Callback
Graphics Driver Activity
Vertex
Fragment
Display
Vsync
Display Stutters
Displaylink Dispatch
Main Thread 0x3924b
Talos (1256)
_dispatch_worklo..._thread 0x39267
_dispatch_worklo..._thread 0x3926a
_dispatch_worklo..._thread 0x3926e
_pthread_body 0x39271
AURemoteIO::IO...ad::Run 0x39291
_pthread_body 0x392a6
_pthread_body 0x3930c
_pthread_body 0x3930d
Blocked
Core 0
Core 1
System Load
Active Threads
Priority
Process
Thread
State
Duration
Core
46 Exited process (1264) 0x395ee Running 210.96 µs CPU 0
4 reversetemplated (317) _dispatch_kevent_worker_thread 0x39... Runnable 39.86 ms n/a
4 searchd (180) _dispatch_workloop_worker_thread 0x1... Runnable 27.25 ms n/a
3 healthd (39) _dispatch_workloop_worker_thread 0x... Preempted 353.18 ms n/a
4 MobileMail (271) _dispatch_workloop_worker_thread 0x... Preempted 4.79 ms n/a
4 MobileMail (271) _dispatch_workloop_worker_thread 0x... Preempted 39.85 ms n/a
3 healthd (39) _dispatch_workloop_worker_thread 0x... Preempted 342.95 ms n/a
4 mobileassetd (101) _dispatch_workloop_worker_thread 0x... Runnable 11.07 ms n/a
3 ReportMemoryException... _dispatch_workloop_worker_thread 0x... Preempted 343.41 ms n/a
4 MobileMail (271) _dispatch_workloop_worker_thread 0x... Preempted 9.74 ms n/a
26 coreduetd (123) _dispatch_workloop_worker_thread 0x... Running 197.30 µs CPU 1
26 Talos _pthread_body 0x39271 Runnable 2.80 ms n/a
No Detail
Input Filter
Instrument Detail

System Trace and Time Profiler

System Trace and Time Profiler

System Trace and Time Profiler

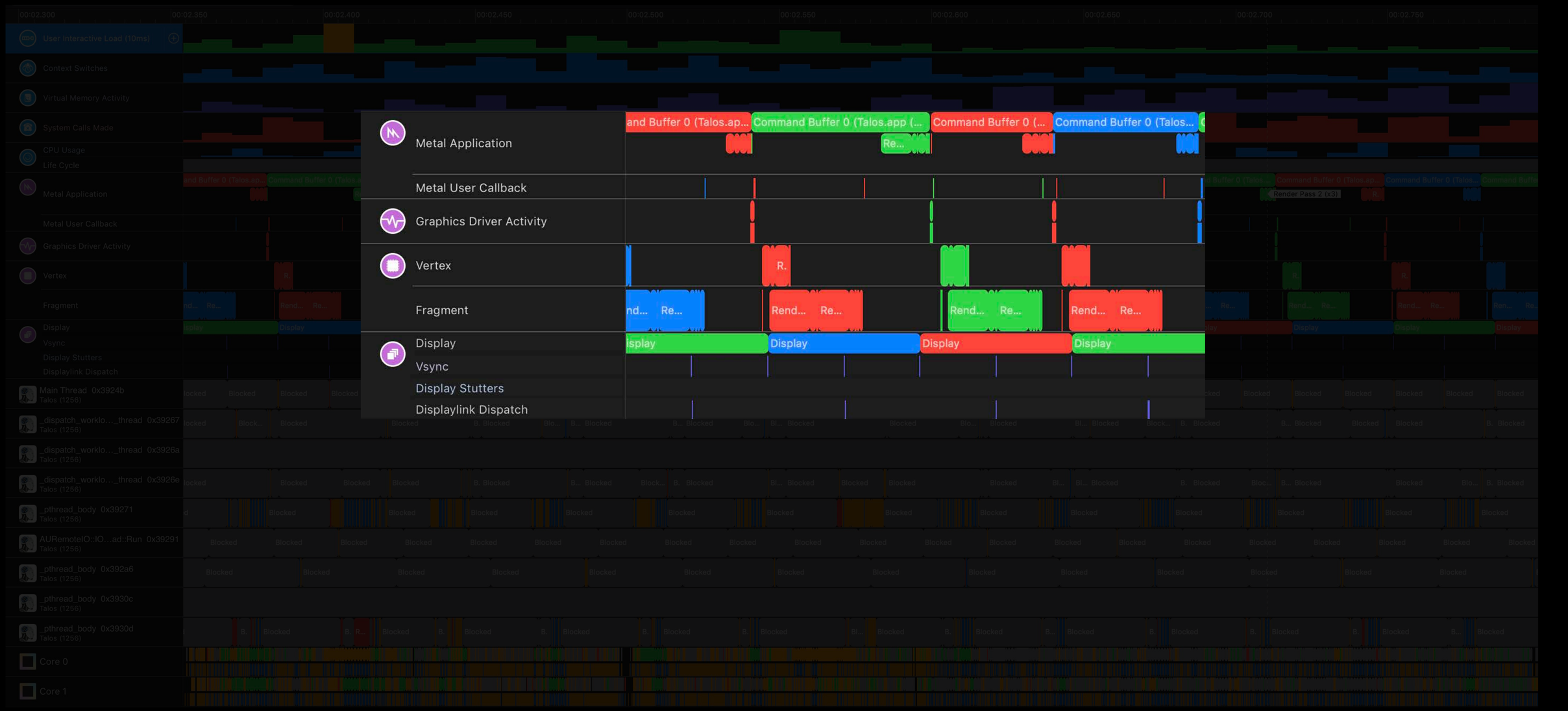


System Trace in Depth — WWDC 2016

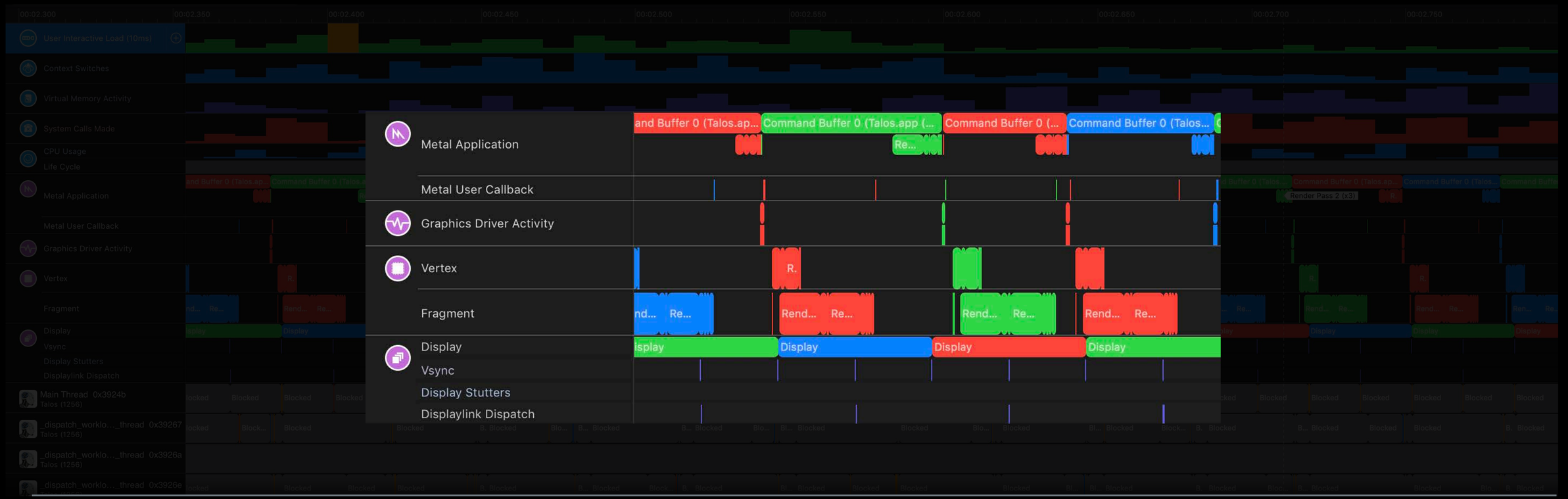
Using Time Profiler in Instruments — WWDC 2016

Metal System Trace

Metal System Trace



Metal System Trace



Metal Performance Optimization Techniques — WWDC 2015

Metal 2 Optimization and Debugging — WWDC 2017

Thread States View

NEW

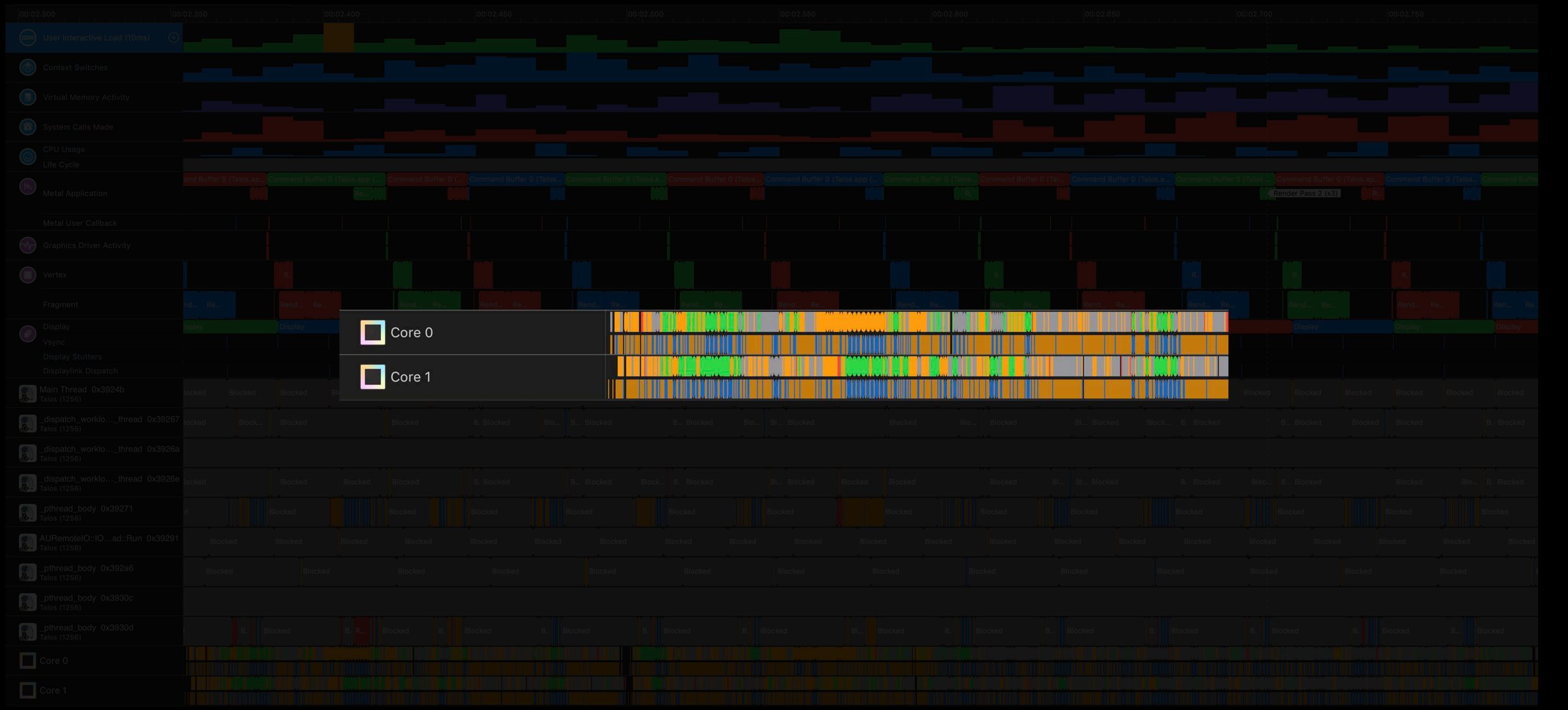
Thread States View

NEW

CPU


00:02.300
00:02.350
00:02.400
00:02.450
00:02.500
00:02.550
00:02.600
00:02.650
00:02.700
00:02.750
User Interactive Load (10ms)
Context Switches
Virtual Memory Activity
System Calls Made
CPU Usage
Life Cycle
Metal Application
Metal User Callback
Graphics Driver Activity
Vertex
Fragment
Display
Vsync
Display Stutters
Displaylink Dispatch
Render Pass 2 (x3)
Main Thread 0x3924b
Talos (1256)
_dispatch_worklo…_thread 0x39267
_dispatch_worklo…_thread 0x3926a
_dispatch_worklo…_thread 0x3926e
_pthread_body 0x39271
AURemoteIO::IO…ad::Run 0x39291
_pthread_body 0x392a6
_pthread_body 0x3930c
_pthread_body 0x3930d
Core 0
Core 1

CPU


00:02.300
00:02.350
00:02.400
00:02.450
00:02.500
00:02.550
00:02.600
00:02.650
00:02.700
00:02.750
User Interactive Load (10ms)
Context Switches
Virtual Memory Activity
System Calls Made
CPU Usage
Life Cycle
Metal Application
Render Pass 2 (x3)
Metal User Callback
Graphics Driver Activity
Vertex
Fragment
Display
Vsync
Display Stutters
Displaylink Dispatch
Main Thread 0x3924b
Talos (1256)
_dispatch_worklo..._thread 0x39267
Talos (1256)
_dispatch_worklo..._thread 0x3926a
Talos (1256)
_dispatch_worklo..._thread 0x3926e
Talos (1256)
_pthread_body 0x39271
Talos (1256)
AURemoteIO::IO...ad::Run 0x39291
Talos (1256)
_pthread_body 0x392a6
Talos (1256)
_pthread_body 0x3930c
Talos (1256)
_pthread_body 0x3930d
Talos (1256)
Core 0
Core 1
Core 0
Core 1

Frame Pacing


B
A


B
A


B
A


B
A

40 FPS

30 FPS

Micro Stuttering

Inconsistent frame rate

- Frame time higher than display refresh interval
- Game logic timing errors

Naive Approach

Present as fast as possible

```
// Render Scene
...
// Get drawable and present
if let drawable = view.currentDrawable {
    // Render Final Pass
    ...
    commandBuffer.present(drawable)
}
commandBuffer.commit()
```

Naive Approach

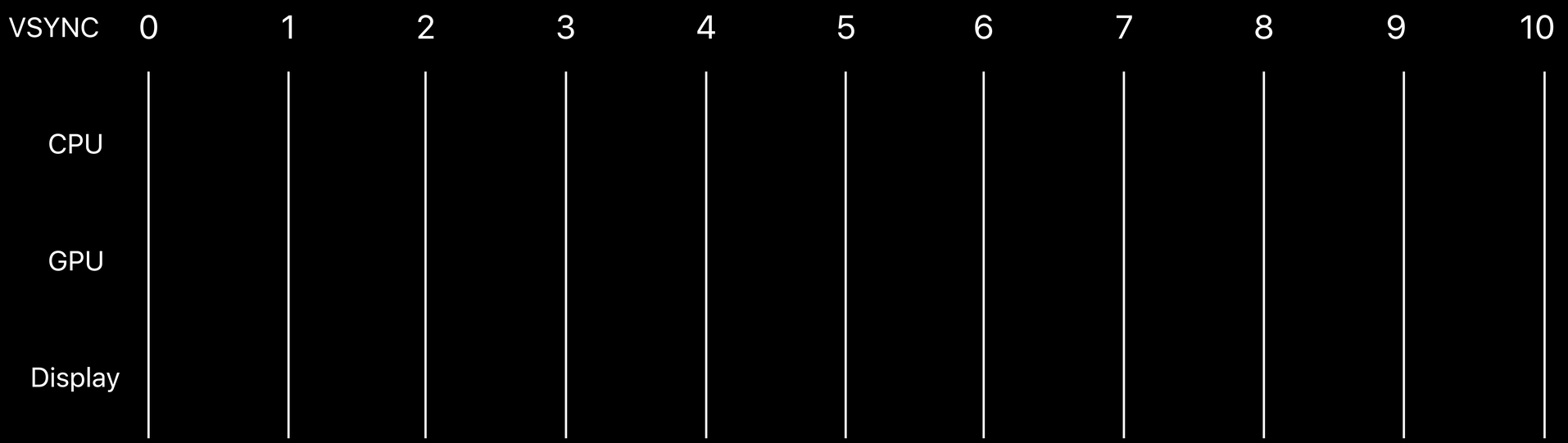
Present as fast as possible

```
// Render Scene
...
// Get drawable and present
if let drawable = view.currentDrawable {
    // Render Final Pass
    ...
    commandBuffer.present(drawable)
}
commandBuffer.commit()
```

Do not usleep() to pace frames!

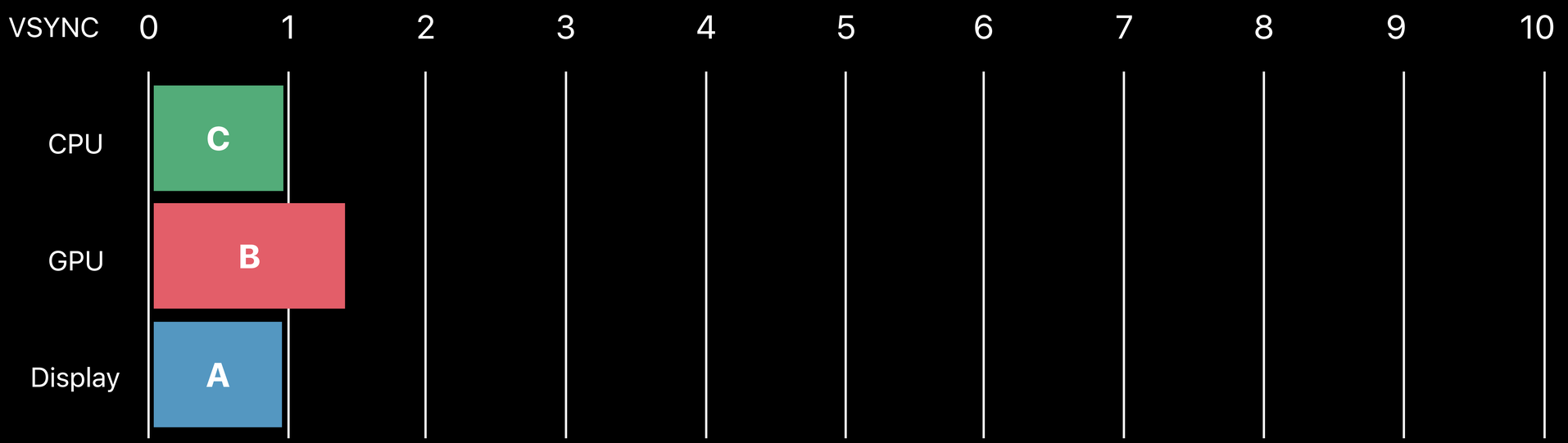
Naive Approach

Present as fast as possible



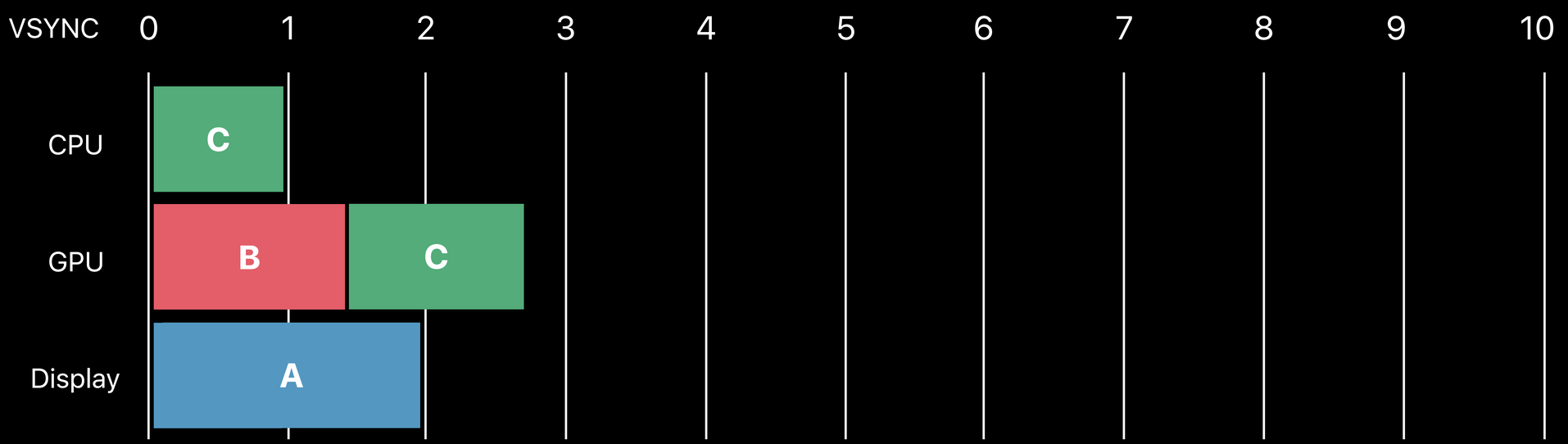
Naive Approach

Present as fast as possible



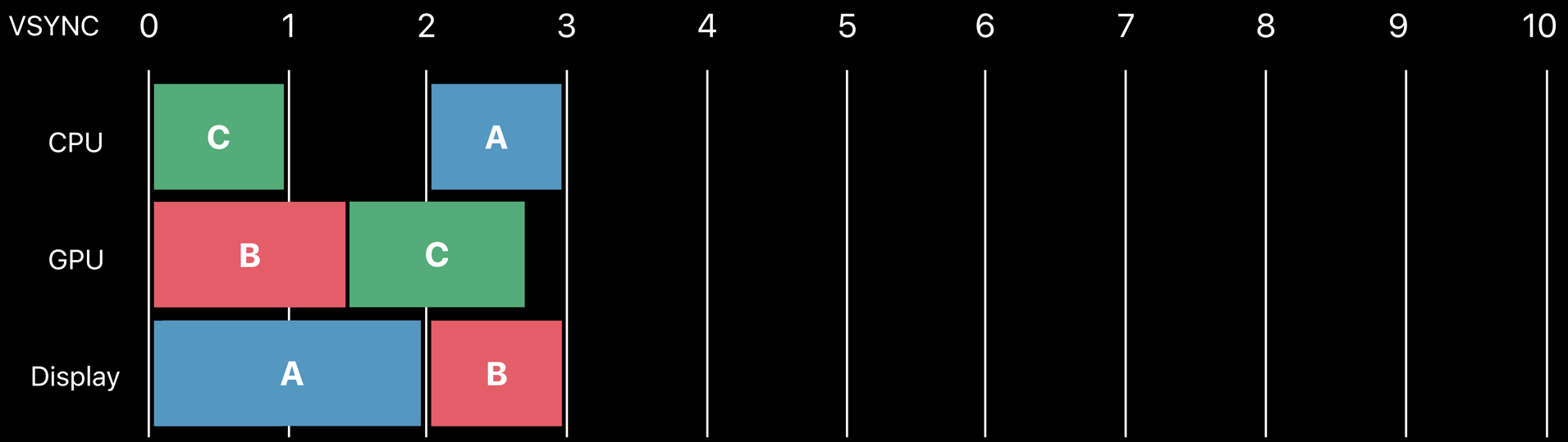
Naive Approach

Present as fast as possible



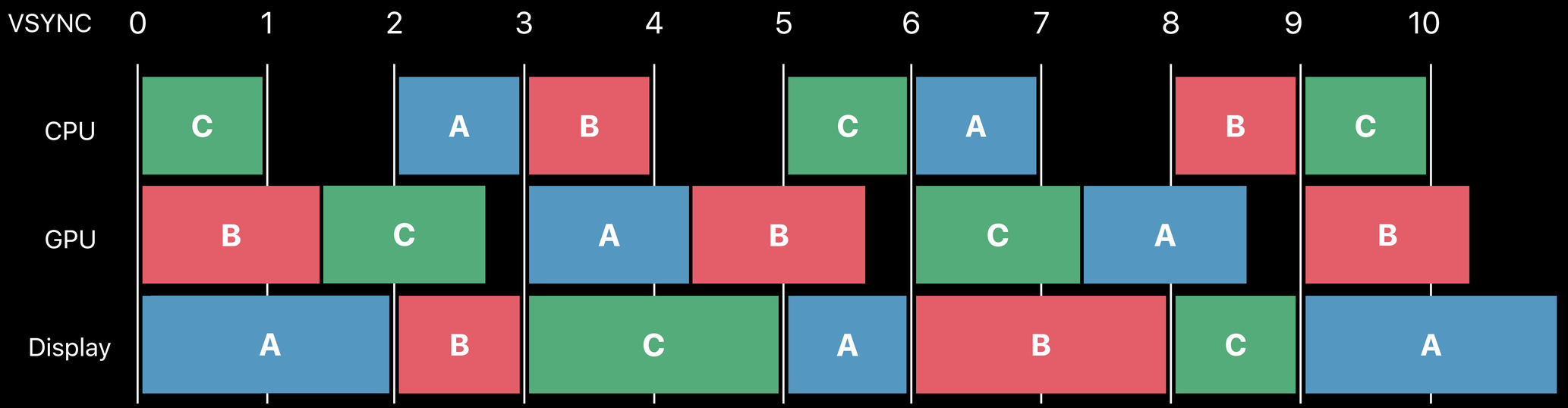
Naive Approach

Present as fast as possible



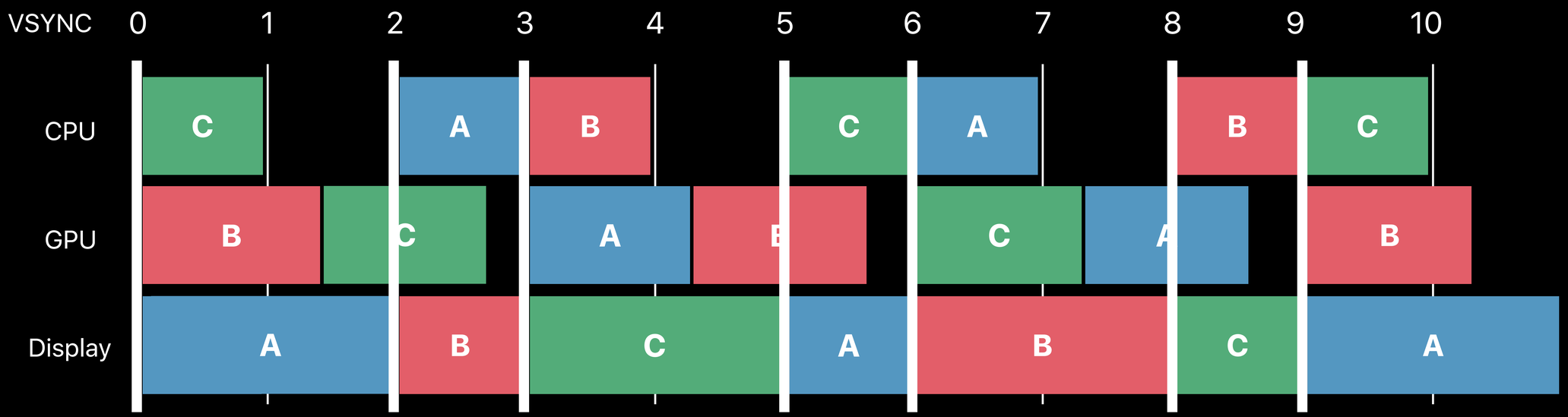
Naive Approach

Present as fast as possible



Naive Approach

Present as fast as possible



Demo

Best Practice

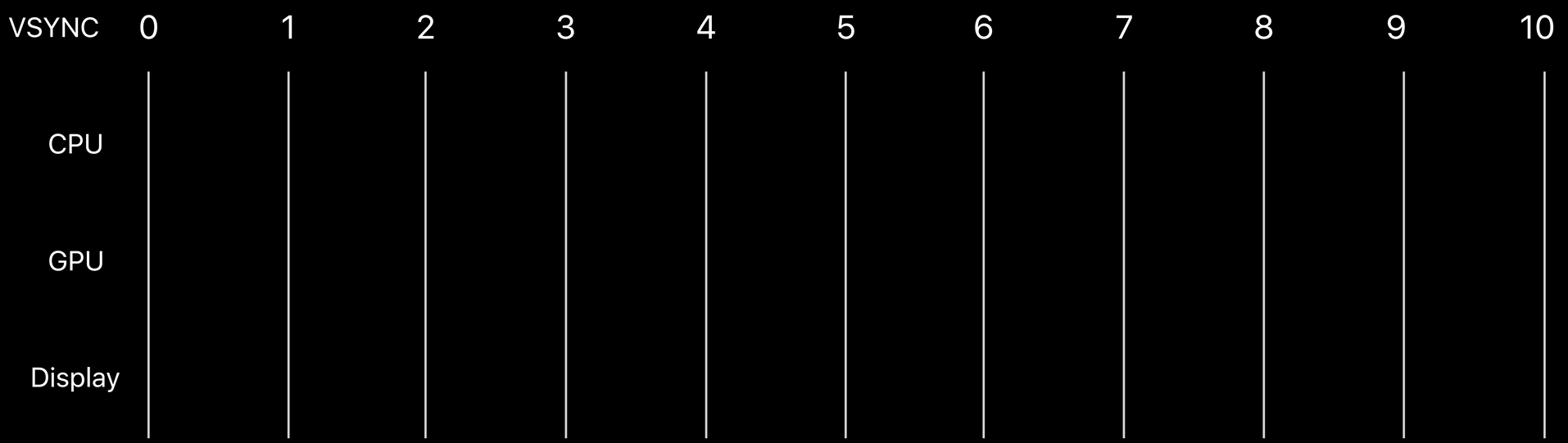
Target explicit frame rate

Use the following APIs (iOS 10.3+)

- `MTLDrawable addPresentedHandler`
- `MTLCommandBuffer presentDrawable afterMinimumDuration`
- `MTLCommandBuffer presentDrawable atTime`

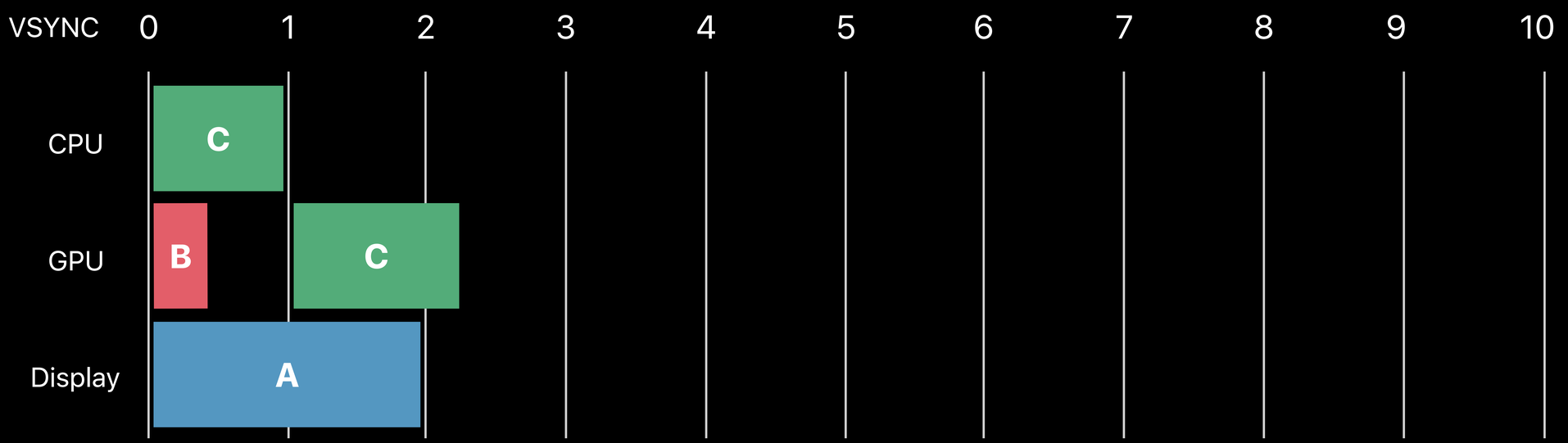
Explicit Frame Pacing

Minimum frame duration of 33ms



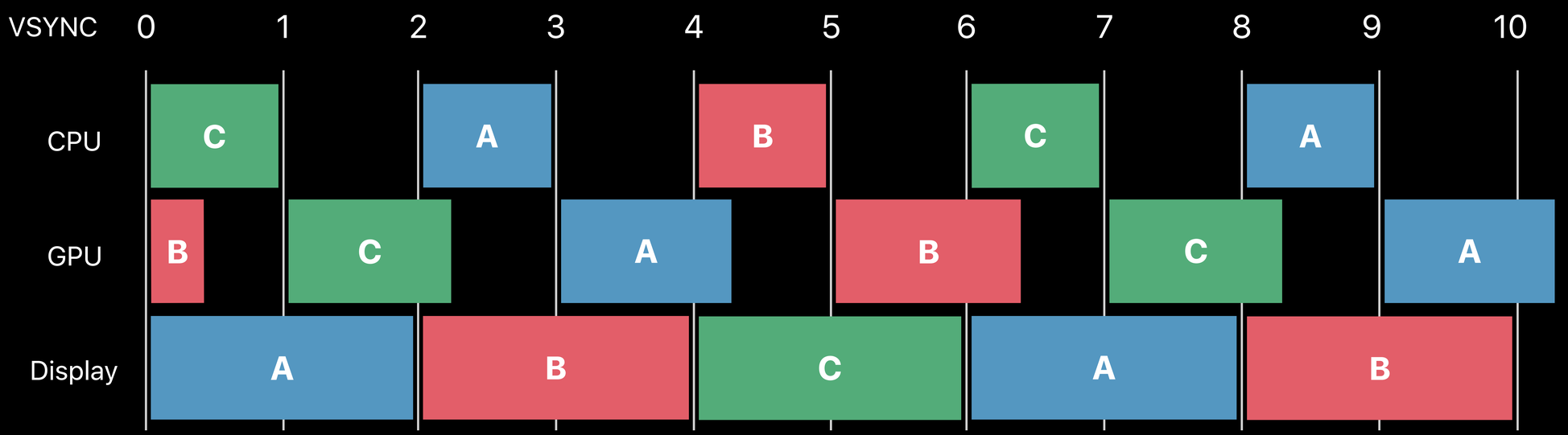
Explicit Frame Pacing

Minimum frame duration of 33ms



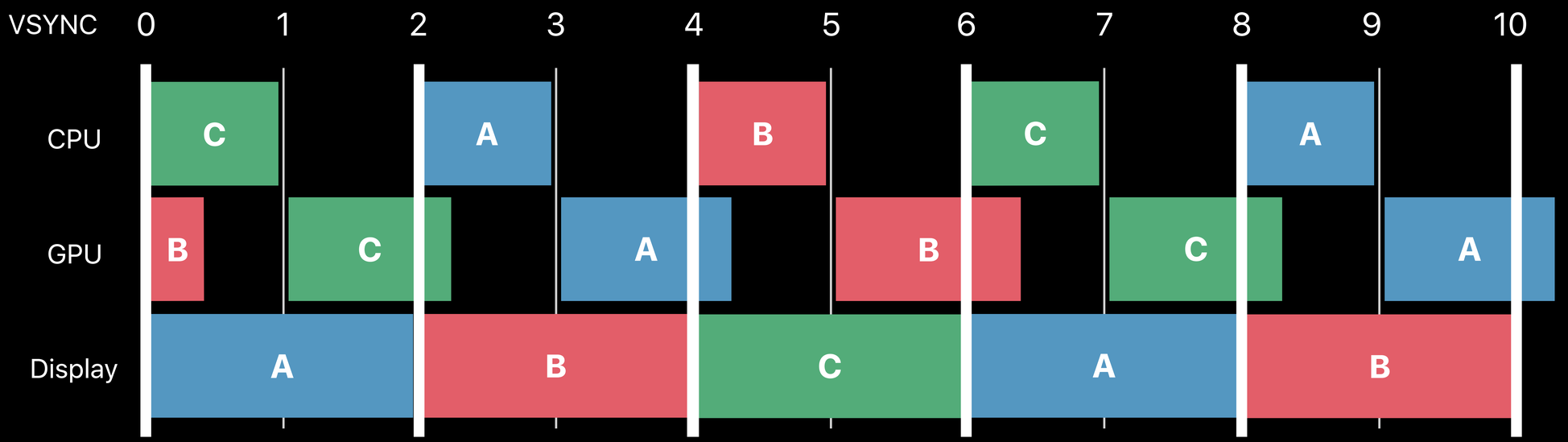
Explicit Frame Pacing

Minimum frame duration of 33ms



Explicit Frame Pacing

Minimum frame duration of 33ms



Explicit Frame Pacing

Minimum frame duration of 33ms

```
// Render Scene
...
// Get drawable and present at 30 FPS
if let drawable = view.currentDrawable {
    // Render Final Pass
    ...
    let duration = 33.0 / 1000.0  // Duration of 33 ms
    commandBuffer.present(drawable, afterMinimumDuration: duration)
}
commandBuffer.commit()
```

Explicit Frame Pacing

Minimum frame duration of 33ms

```
// Render Scene
...
// Get drawable and present at 30 FPS
if let drawable = view.currentDrawable {
    // Render Final Pass
    ...
    let duration = 33.0 / 1000.0  // Duration of 33 ms
    commandBuffer.present(drawable, afterMinimumDuration: duration)
}
commandBuffer.commit()
```

Thread Priorities


B
A


B
A

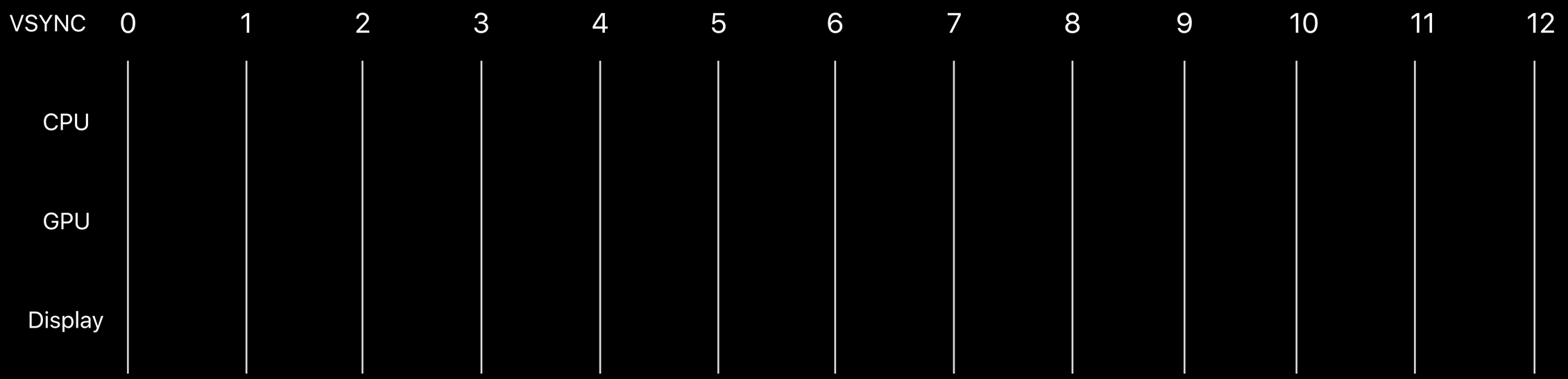
Thread Stalling

Render thread gets preempted due to low priority

- Priority decay
- Priority inversion

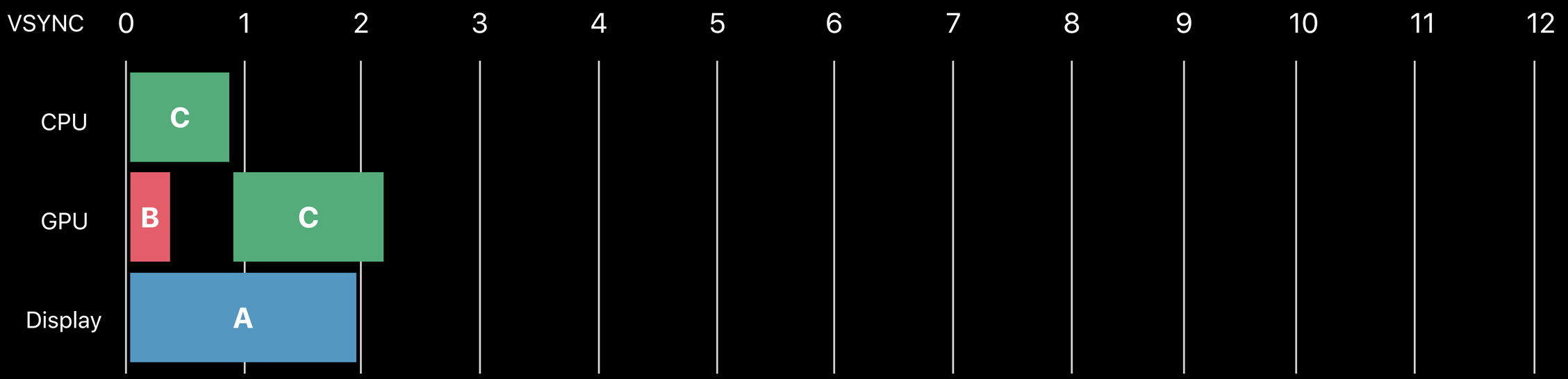
Naive Approach

Render thread gets preempted due to low priority



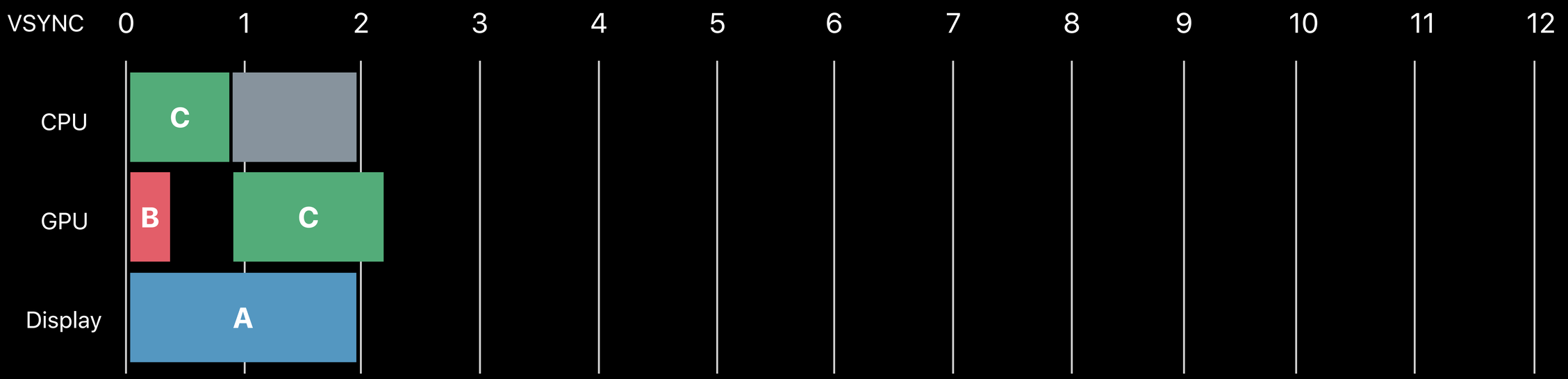
Naive Approach

Render thread gets preempted due to low priority



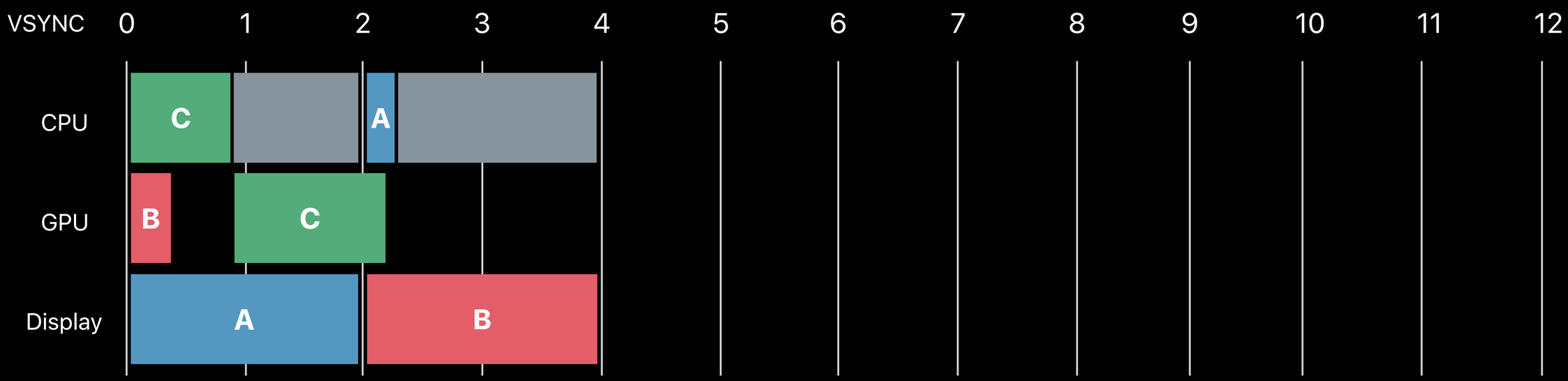
Naive Approach

Render thread gets preempted due to low priority



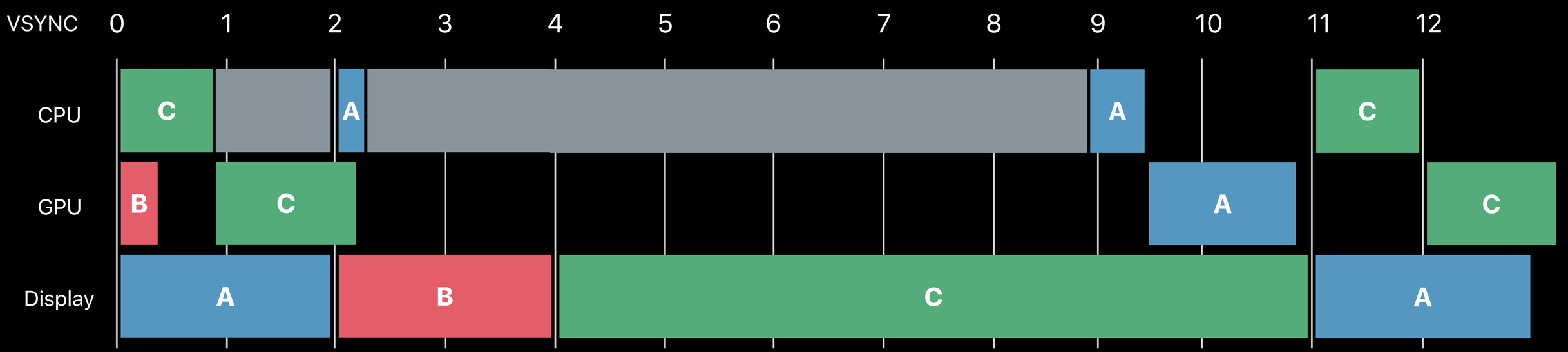
Naive Approach

Render thread gets preempted due to low priority



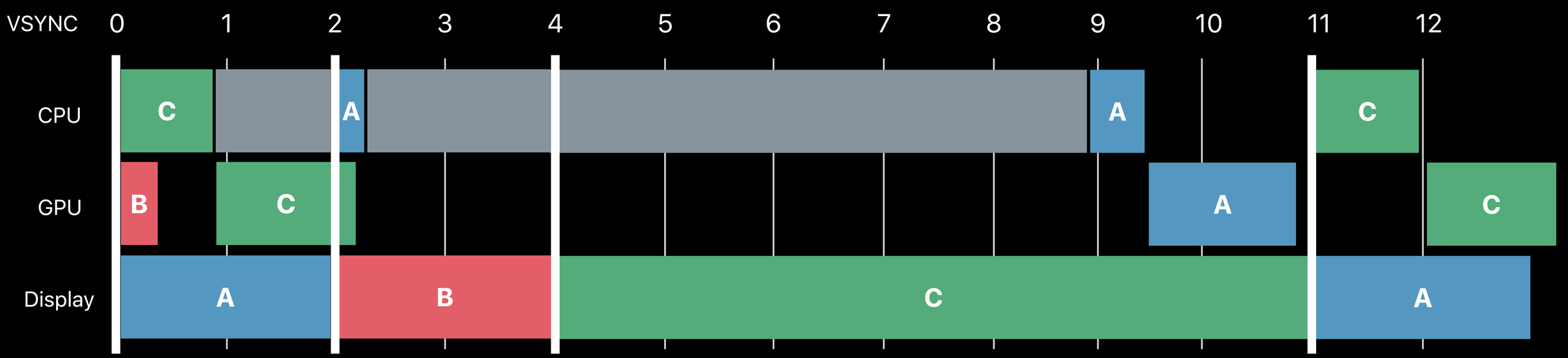
Naive Approach

Render thread gets preempted due to low priority



Naive Approach

Render thread gets preempted due to low priority



Demo

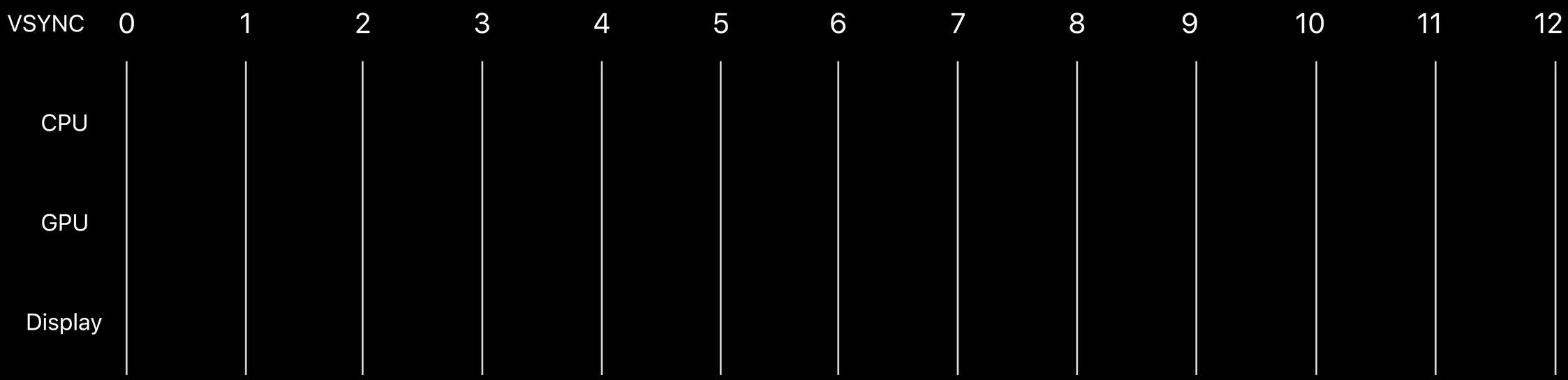
Best Practice

Configure the render thread

- Priority 45
- Opt out of Quality of Service

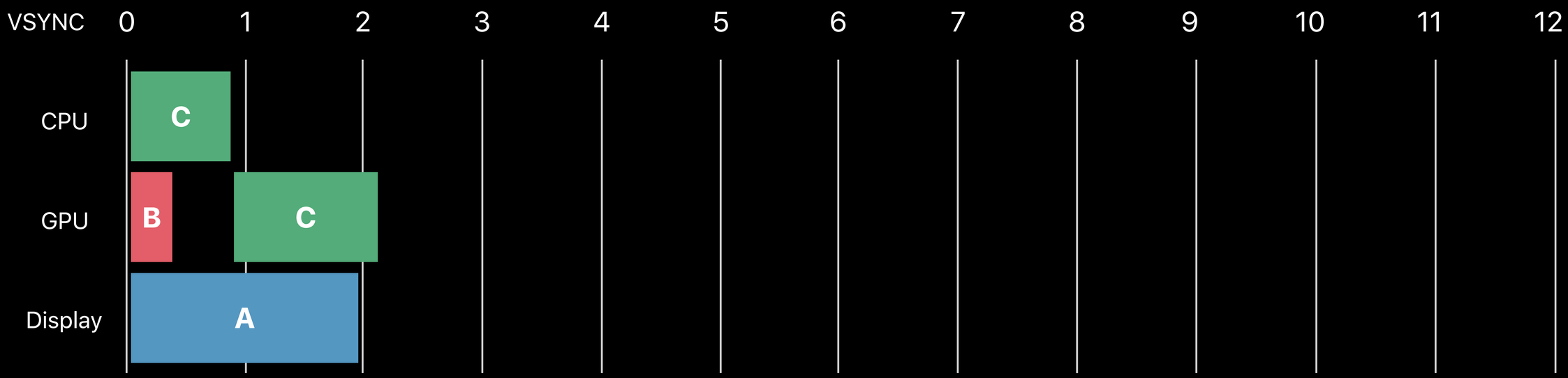
Correct Thread Priority

Priority set to 45 and no Quality of Service



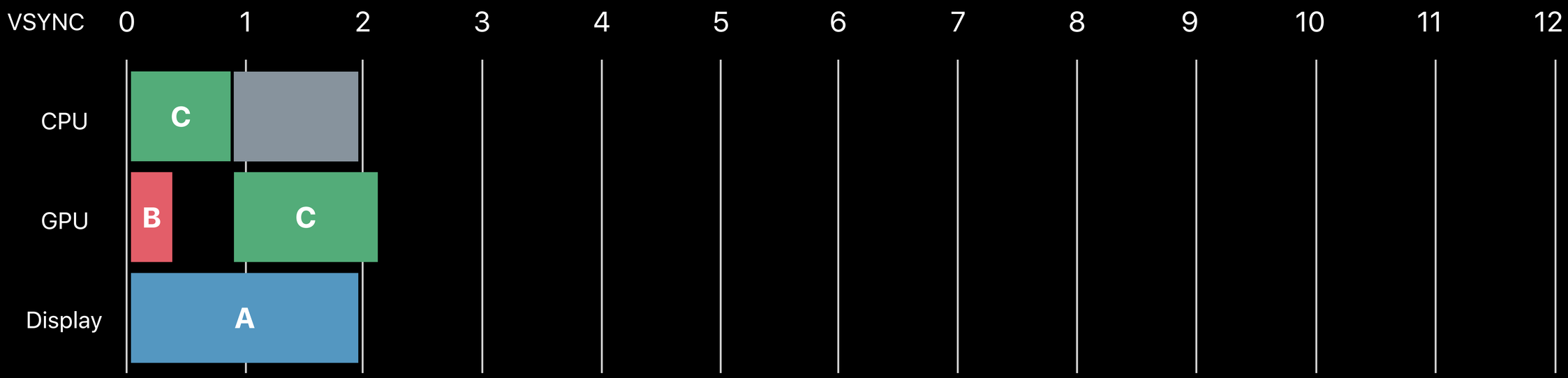
Correct Thread Priority

Priority set to 45 and no Quality of Service



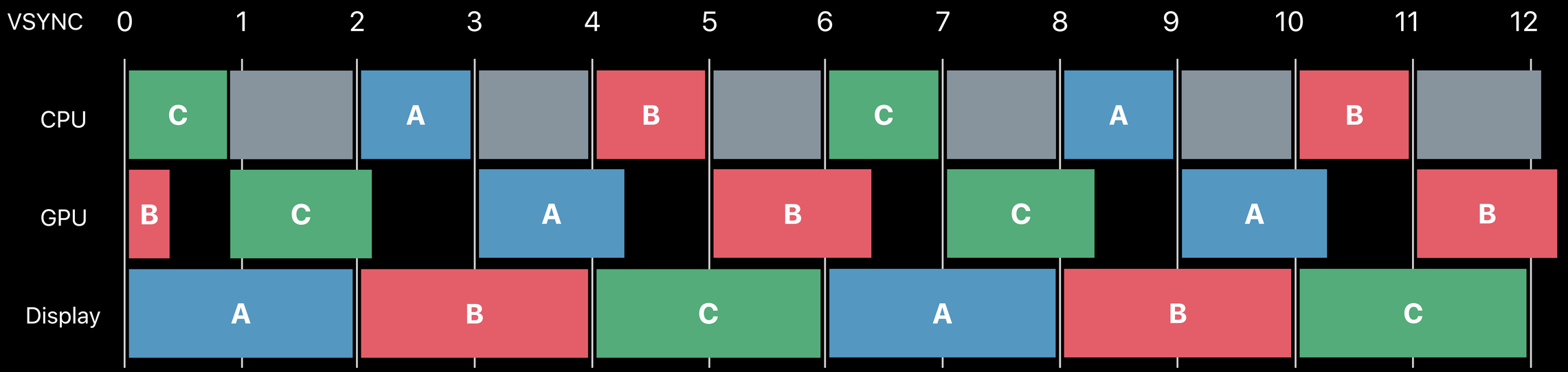
Correct Thread Priority

Priority set to 45 and no Quality of Service



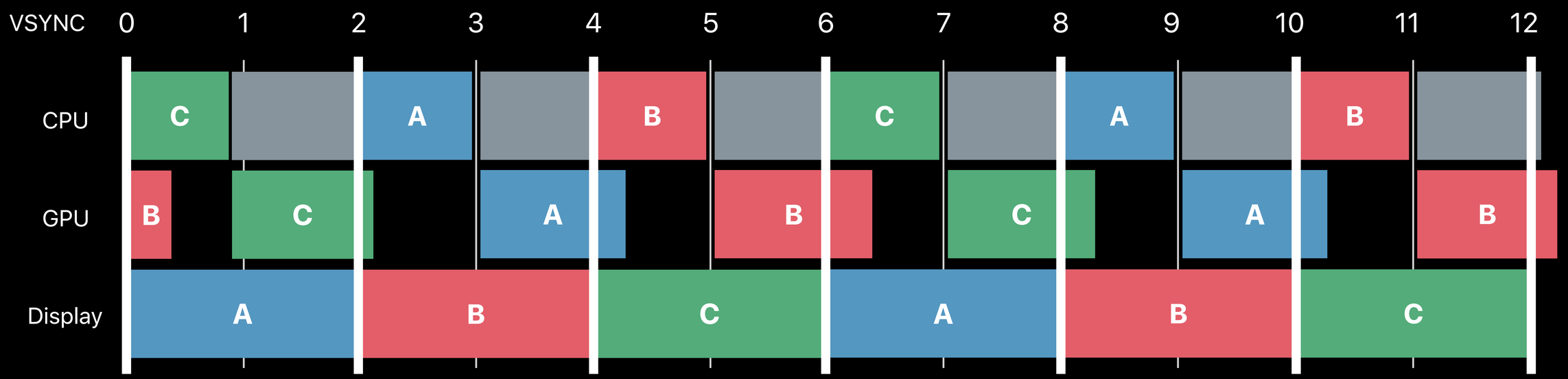
Correct Thread Priority

Priority set to 45 and no Quality of Service



Correct Thread Priority

Priority set to 45 and no Quality of Service



Correct Thread Priority

Priority set to 45 and no Quality of Service

```
...
r = pthread_attr_init(&attr);

r = pthread_attr_setschedpolicy(&attr, SCHED_RR);  // Opt out of Quality of Service
struct sched_param param = {.sched_priority = 45}; // Configure priority 45
r = pthread_attr_setschedparam(&attr, &param);     // Set priority

r = pthread_create(&posixThreadID, &attr, &PosixThreadMainRoutine, NULL);
r = pthread_attr_destroy(&attr);
...
```

Correct Thread Priority

Priority set to 45 and no Quality of Service

```
...
r = pthread_attr_init(&attr);

r = pthread_attr_setschedpolicy(&attr, SCHED_RR);  // Opt out of Quality of Service
struct sched_param param = {.sched_priority = 45}; // Configure priority 45
r = pthread_attr_setschedparam(&attr, &param);     // Set priority

r = pthread_create(&posixThreadID, &attr, &PosixThreadMainRoutine, NULL);
r = pthread_attr_destroy(&attr);
...
```

Thermal States

Design for sustained performance

Thermal Throttling

Impact on system performance

- High device temperature
- Low power mode enabled

Best Practice

Adjust the workload to the system state

Use the following APIs

- (iOS 11.0+) `NSProcessInfo thermalState`
- (iOS 9.0+) `NSProcessInfo lowPowerModeEnabled`
- (iOS 10.3+) `MTLCommandBuffer GPUStartTime/GPUEndTime`

```
// Determine thermal state

switch ProcessInfo.processInfo.thermalState {
case .fair:
    // Thermals are fair
    // Consider taking proactive measures to prevent higher thermals
case .serious:
    // Thermals are highly elevated
    // Help the system by taking corrective action
case .critical:
    // Thermals are extremely elevated
    // Help the system by taking immediate corrective action
default:
    // Thermals are okay
    // Go about your business
}
```

```swift
// Determine thermal state

switch ProcessInfo.processInfo.thermalState {
case .fair:
    // Thermals are fair
    // Consider taking proactive measures to prevent higher thermals
case .serious:
    // Thermals are highly elevated
    // Help the system by taking corrective action
case .critical:
    // Thermals are extremely elevated
    // Help the system by taking immediate corrective action
default:
    // Thermals are okay
    // Go about your business
}
```

```swift
// Determine thermal state

switch ProcessInfo.processInfo.thermalState {
case .fair:
    // Thermals are fair
    // Consider taking proactive measures to prevent higher thermals
case .serious:
    // Thermals are highly elevated
    // Help the system by taking corrective action
case .critical:
    // Thermals are extremely elevated
    // Help the system by taking immediate corrective action
default:
    // Thermals are okay
    // Go about your business
}
```

```
// Determine thermal state

switch ProcessInfo.processInfo.thermalState {
case .fair:
    // Thermals are fair
    // Consider taking proactive measures to prevent higher thermals
case .serious:
    // Thermals are highly elevated
    // Help the system by taking corrective action
case .critical:
    // Thermals are extremely elevated
    // Help the system by taking immediate corrective action
default:
    // Thermals are okay
    // Go about your business
}
```

```swift
// Determine thermal state

switch ProcessInfo.processInfo.thermalState {
case .fair:
    // Thermals are fair
    // Consider taking proactive measures to prevent higher thermals
case .serious:
    // Thermals are highly elevated
    // Help the system by taking corrective action
case .critical:
    // Thermals are extremely elevated
    // Help the system by taking immediate corrective action
default:
    // Thermals are okay
    // Go about your business
}
```

Adjust the Workload

Target sustainable framerate

Reduce the resolution

Simplify the shadow maps

Use smaller textures

Decrease the level of detail (LOD) for geometry

Simplify post-processing and effects

Unnecessary GPU Work

Ohad Frenkel, Game Technologies

Wasted GPU Time

Waste of power and GPU budget

- Large resources
- Unused GPU work

Best Practice

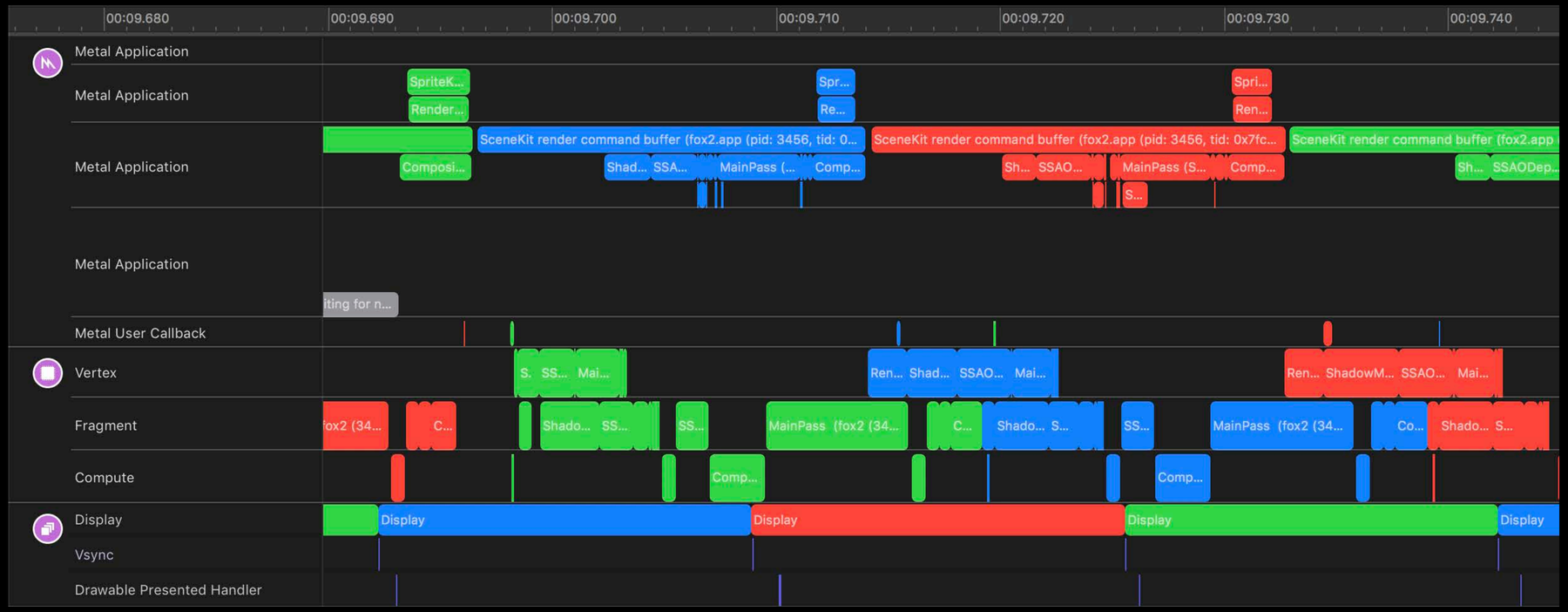
Profile the GPU

- Understand the cost of every rendering feature
- Remove excessive work

Metal System Trace

Accurate timing for Vertex, Fragment, and Compute work

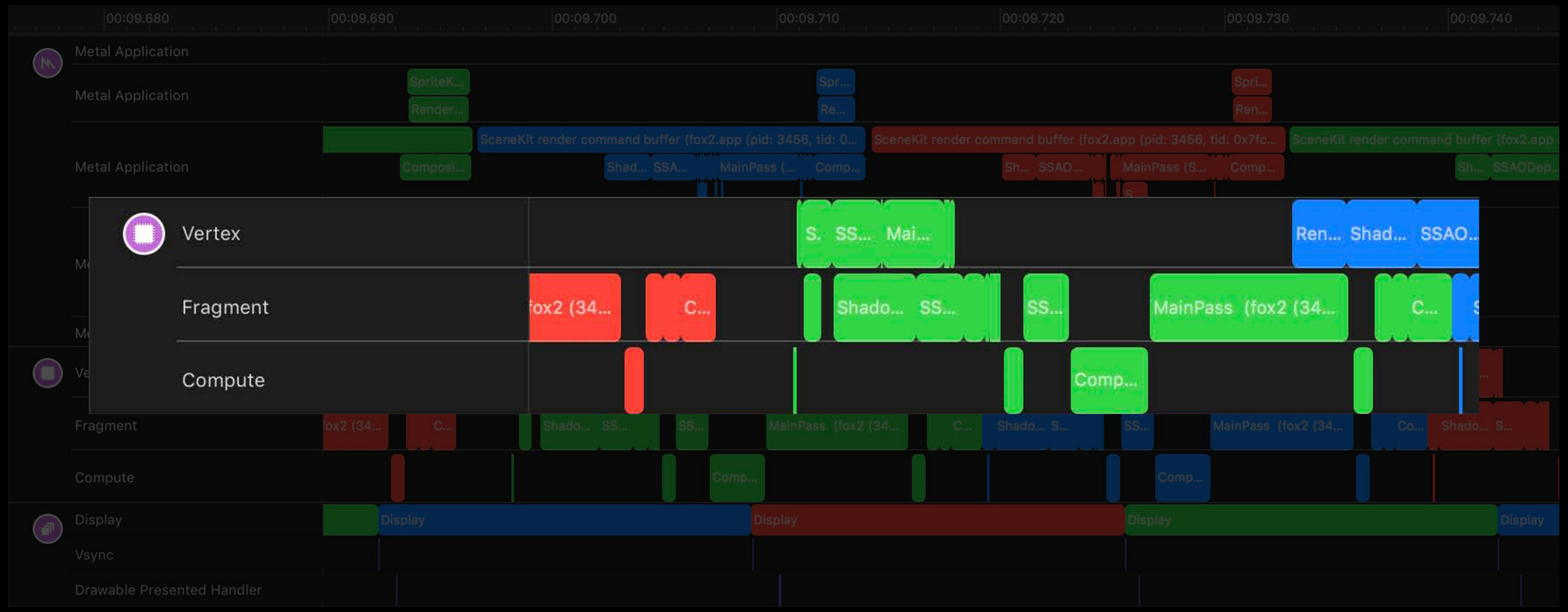
Ideal to measure GPU budget



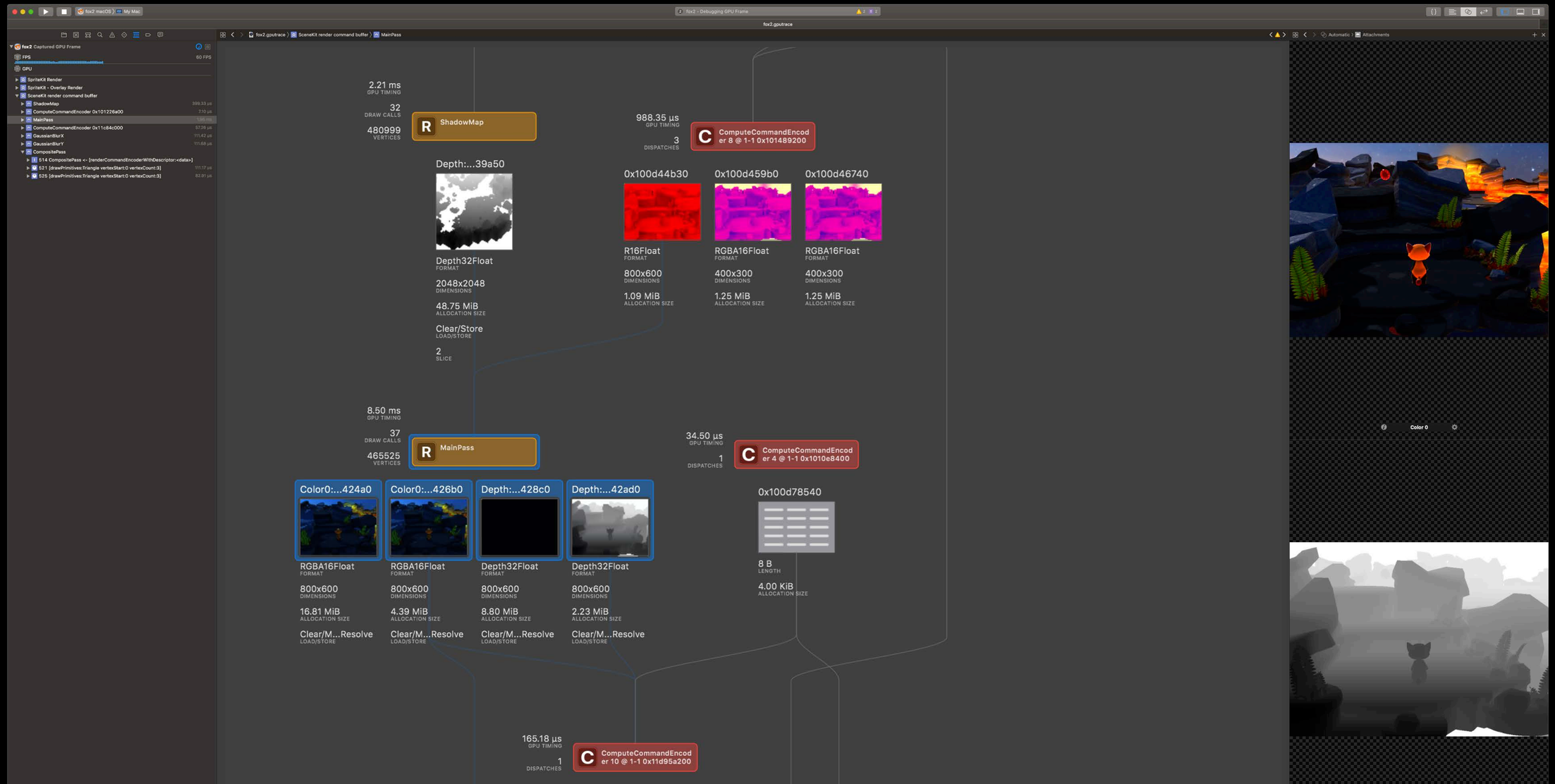
Metal System Trace

Accurate timing for Vertex, Fragment, and Compute work

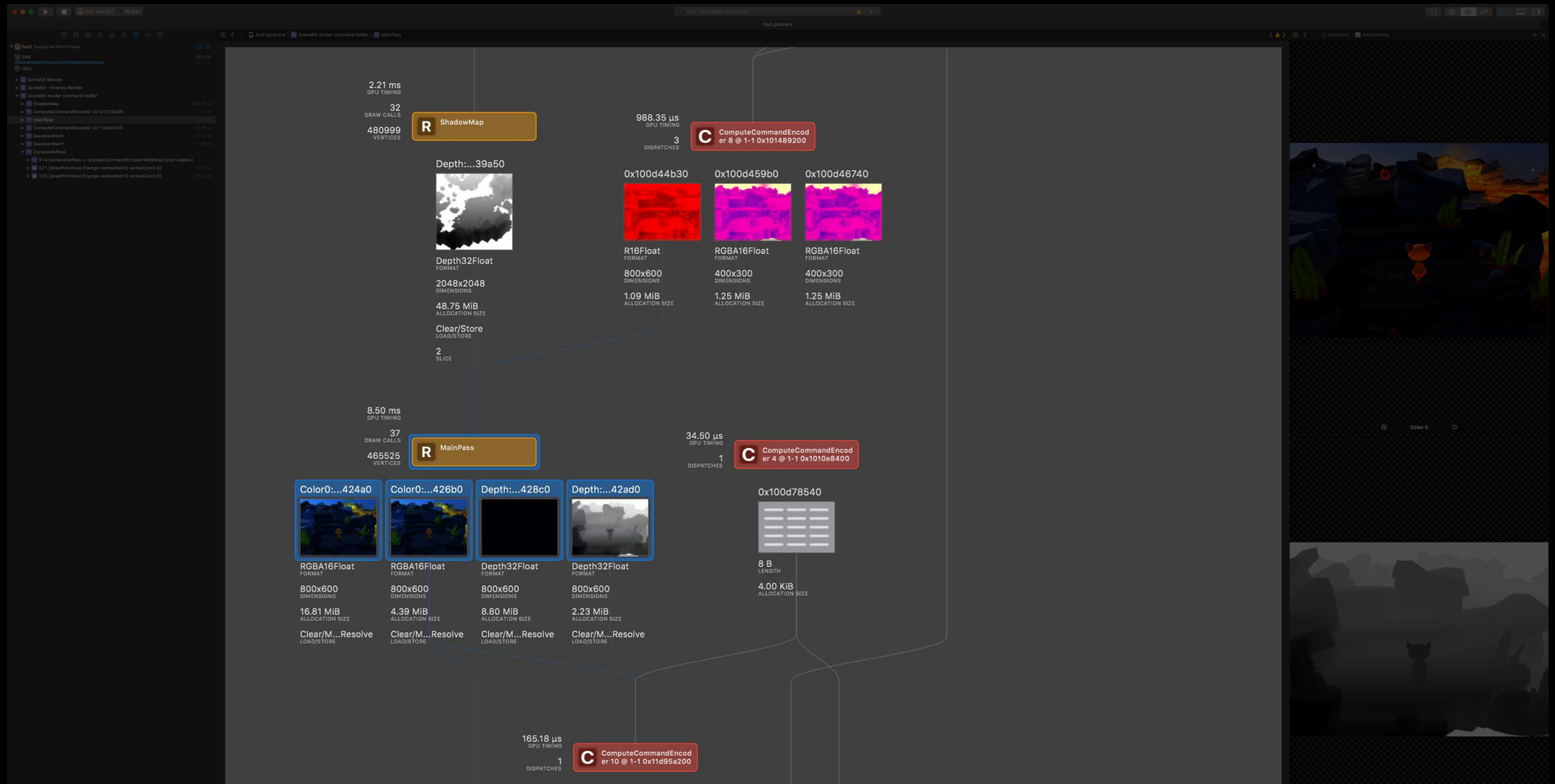
Ideal to measure GPU budget



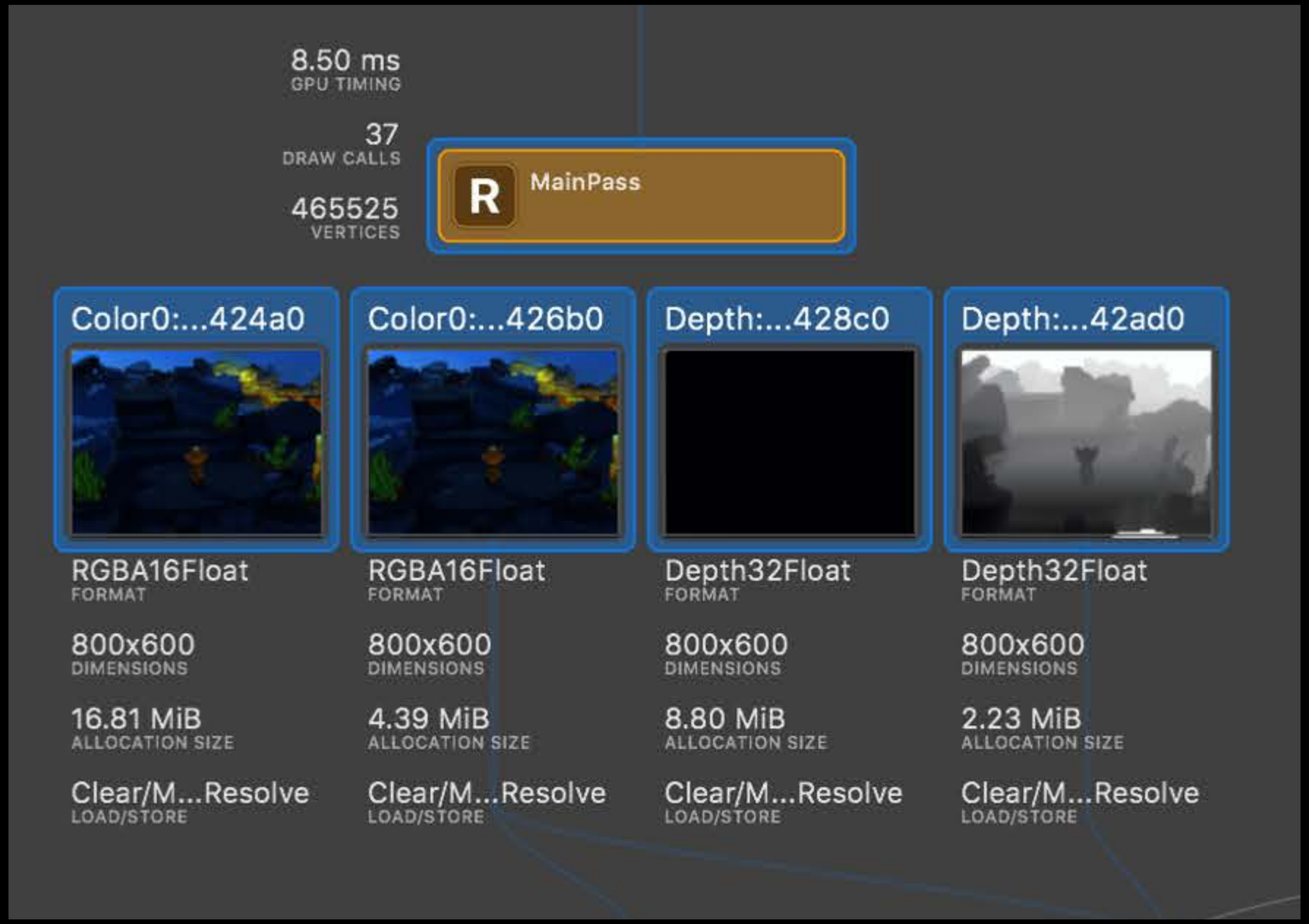
Dependency Viewer



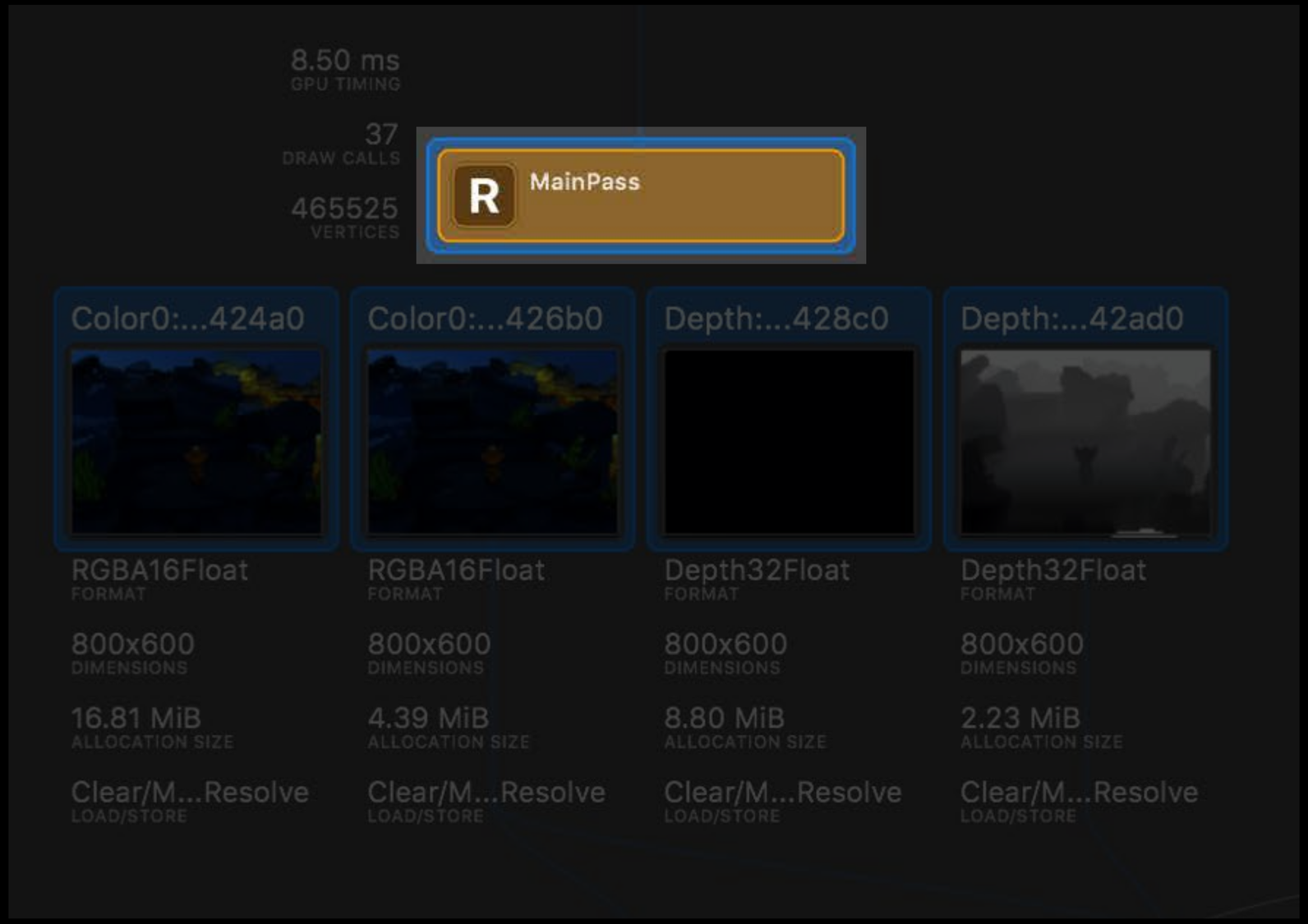
Dependency Viewer



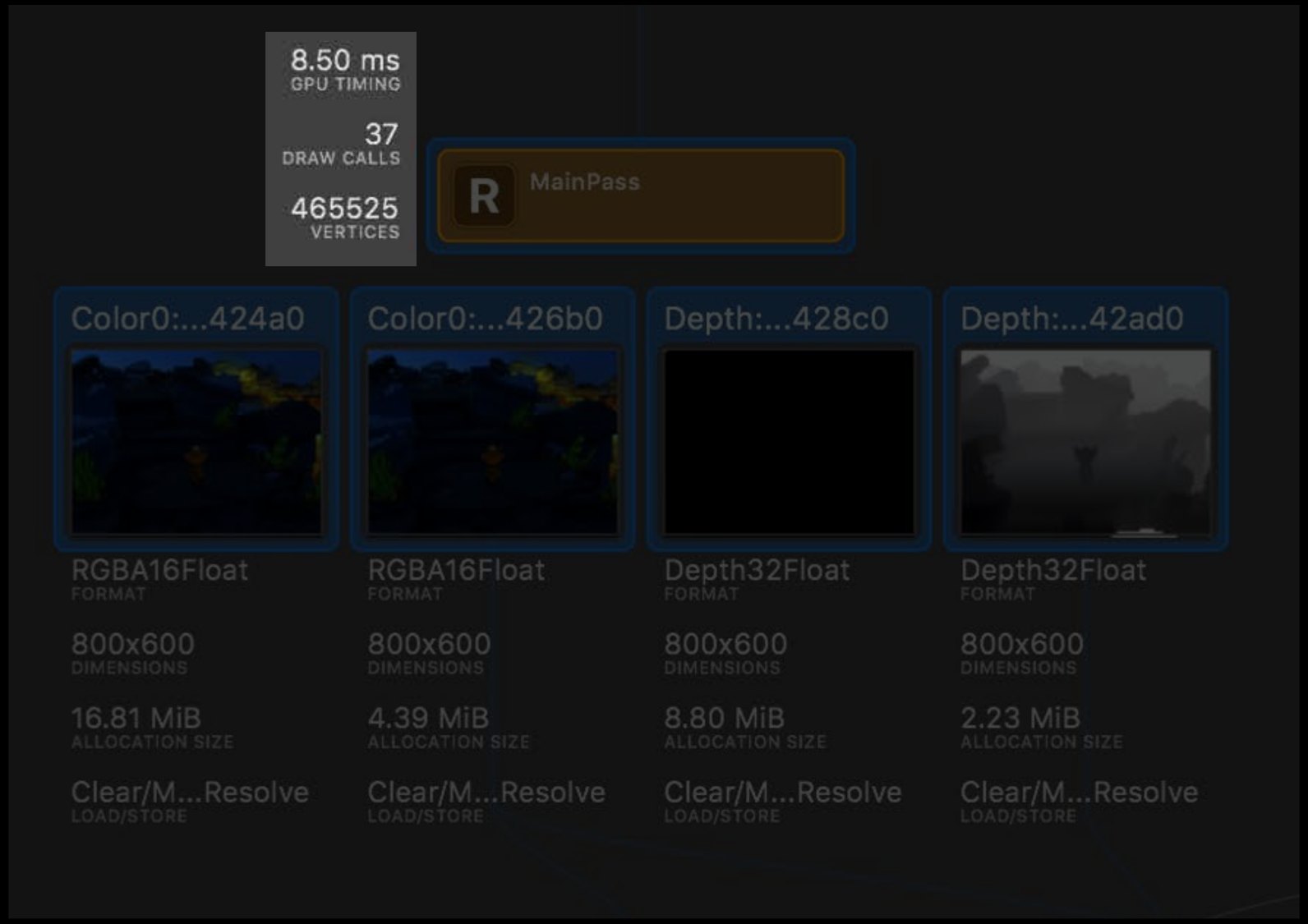
Dependency Viewer



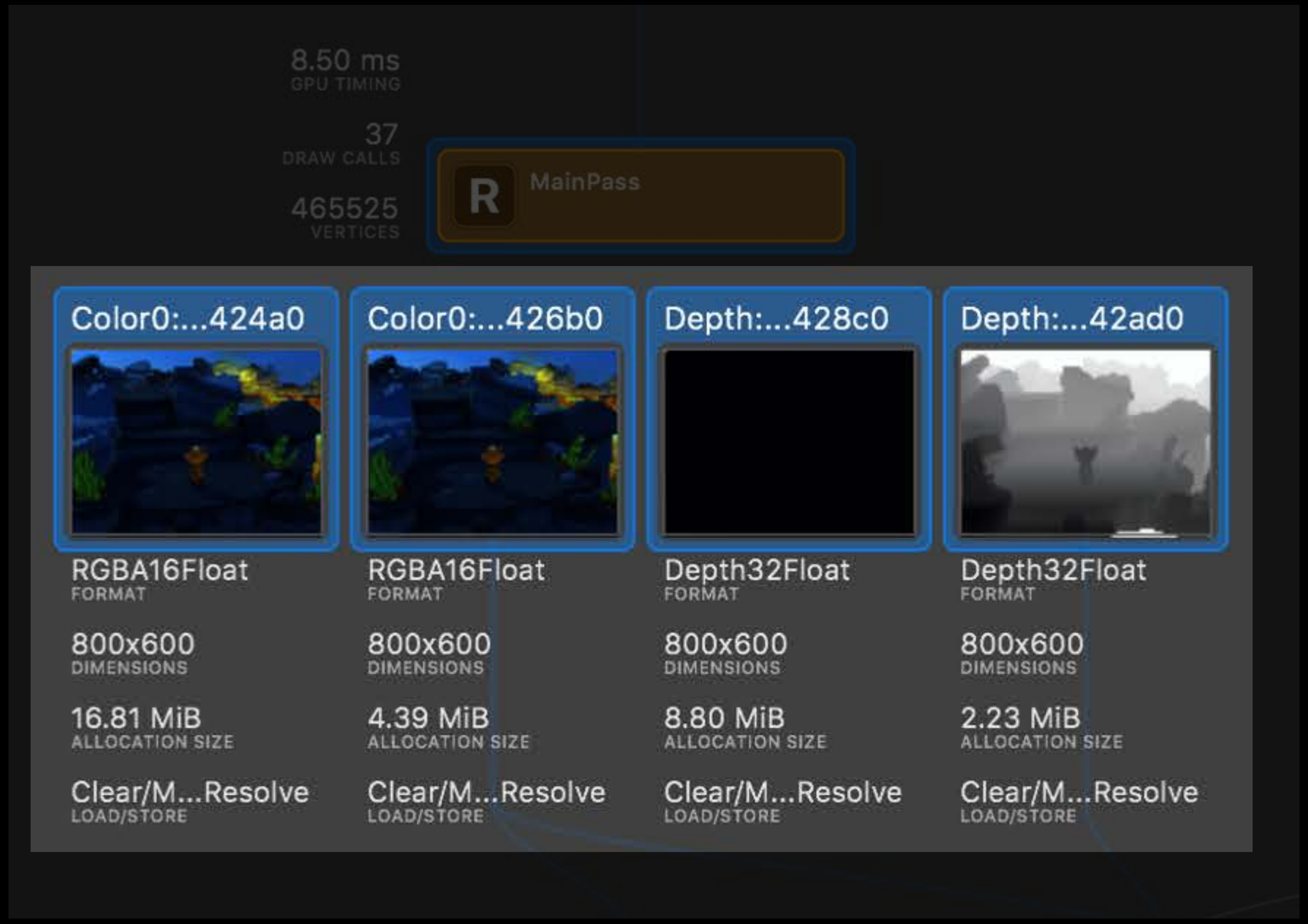
Dependency Viewer



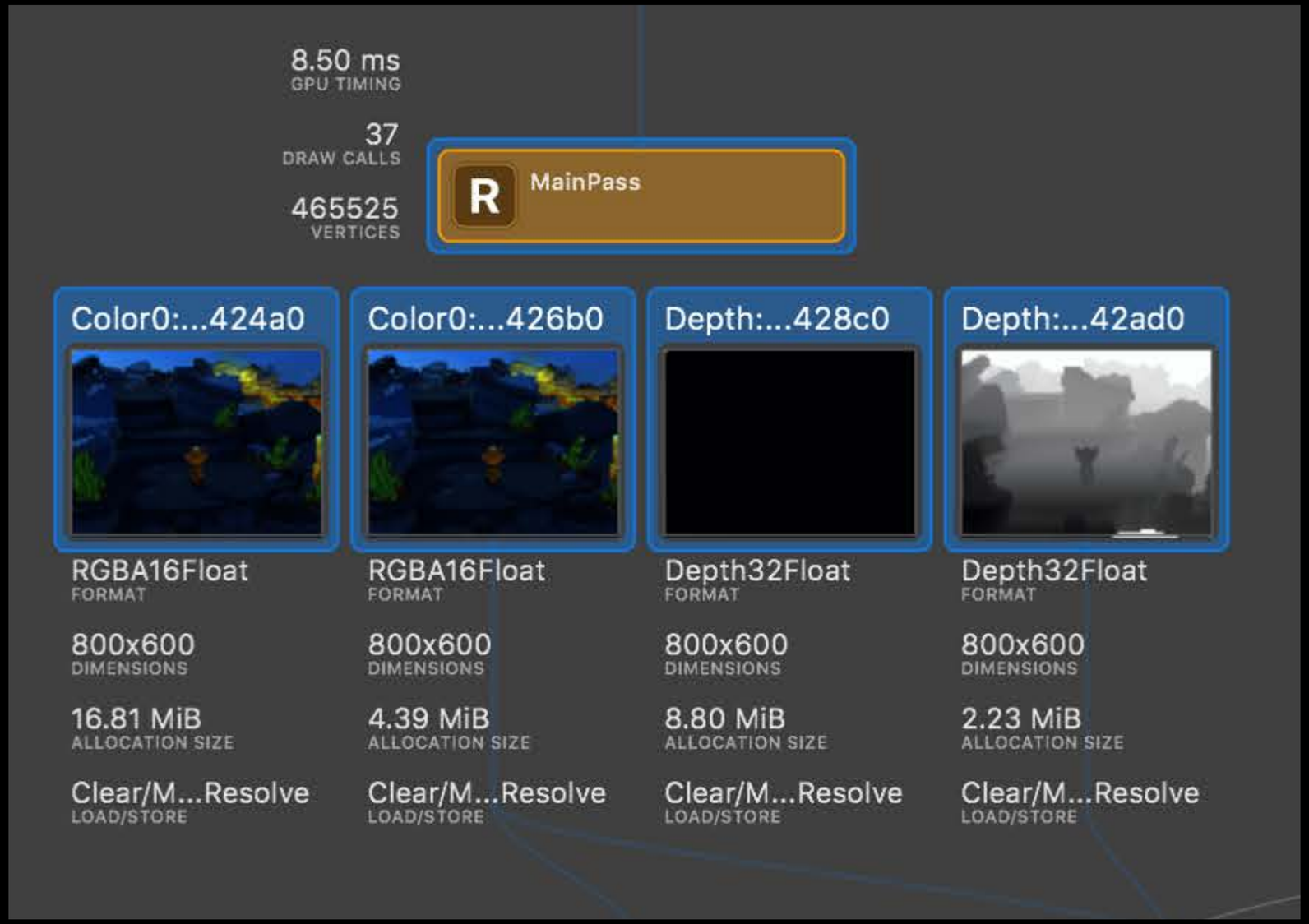
Dependency Viewer



Dependency Viewer



Dependency Viewer



Demo

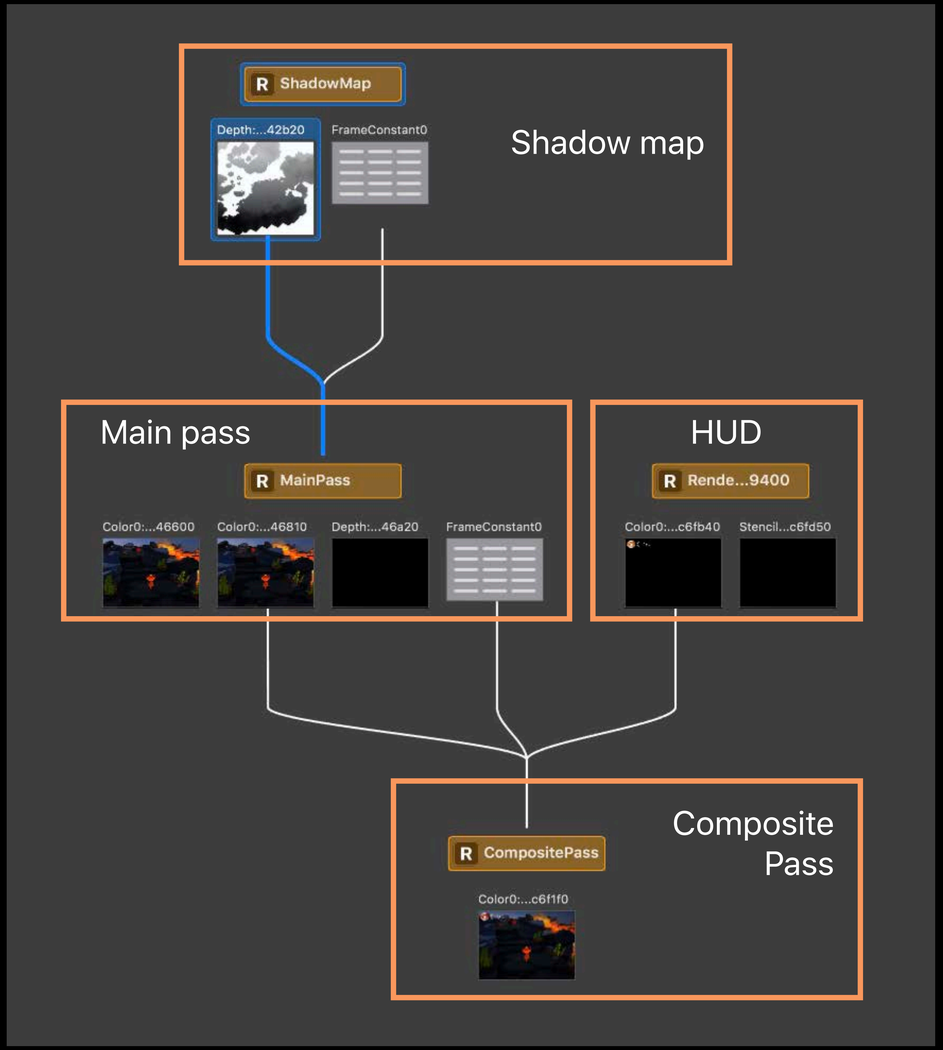
Finding Hidden Complexity

Shadow map

Main pass

HUD

Composite pass



Finding Hidden Complexity

Cascaded shadow maps (3 passes)

SSAO (5 passes)

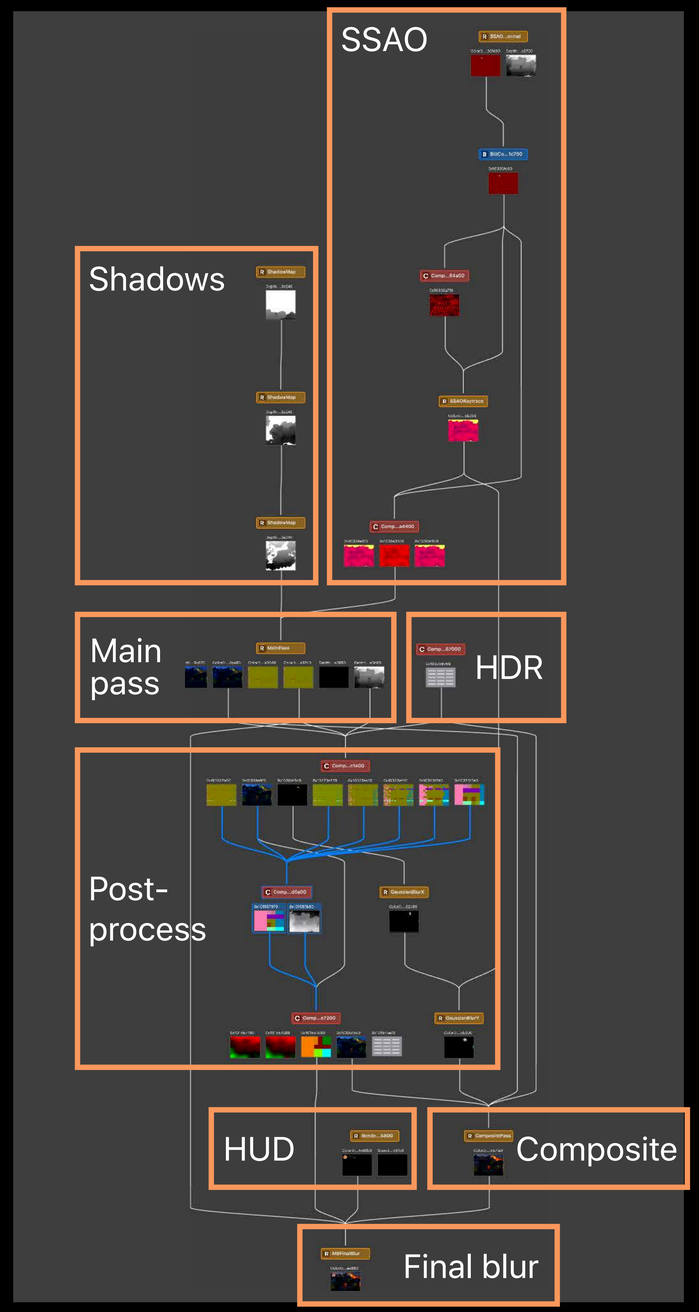
Main pass

HDR

Post-process (5 passes)

Composite pass

And more...



Profile!

Take-Away

Profile early and often

Target a consistent frame rate

Set the correct thread priorities

Adapt to system load and thermals

Don’t submit unnecessary work to the GPU

More Information

https://developer.apple.com/wwdc18/612

Metal Shader Debugging and Profiling		WWDC 2018
Metal Debugging and Profiling Lab	Technology Lab 5	Fri 12:00 PM
Metal for Game Developers		WWDC 2018


WWDC18