# Using Time Profiler in Instruments

## Profile early, profile often

Session 418

Kris Markel

# Agenda

Intro to profiling

Going faster

Doing less

Improving responsiveness

# A Better User Experience

# Profiling

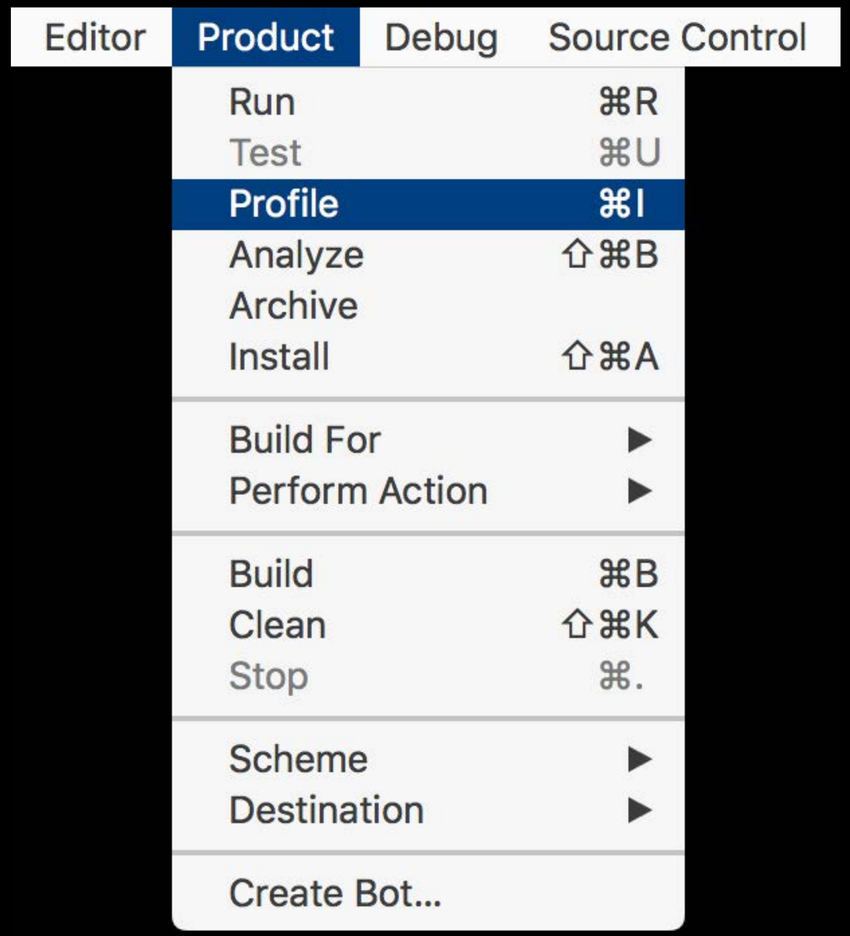
How much and what kind of work is my app doing?

# Time Profiler

# Instruments

## Measure, adjust, repeat

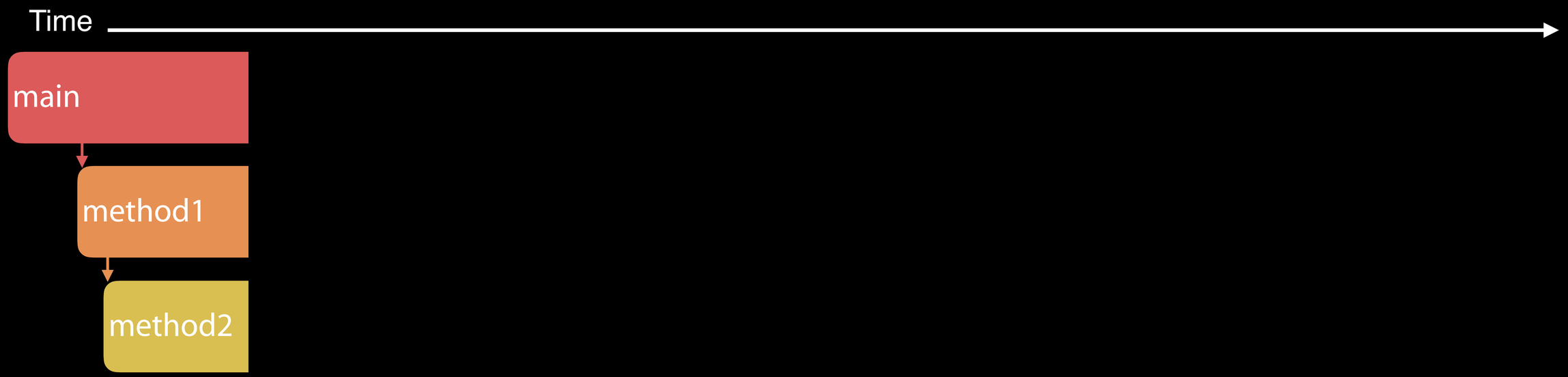
# Getting Started

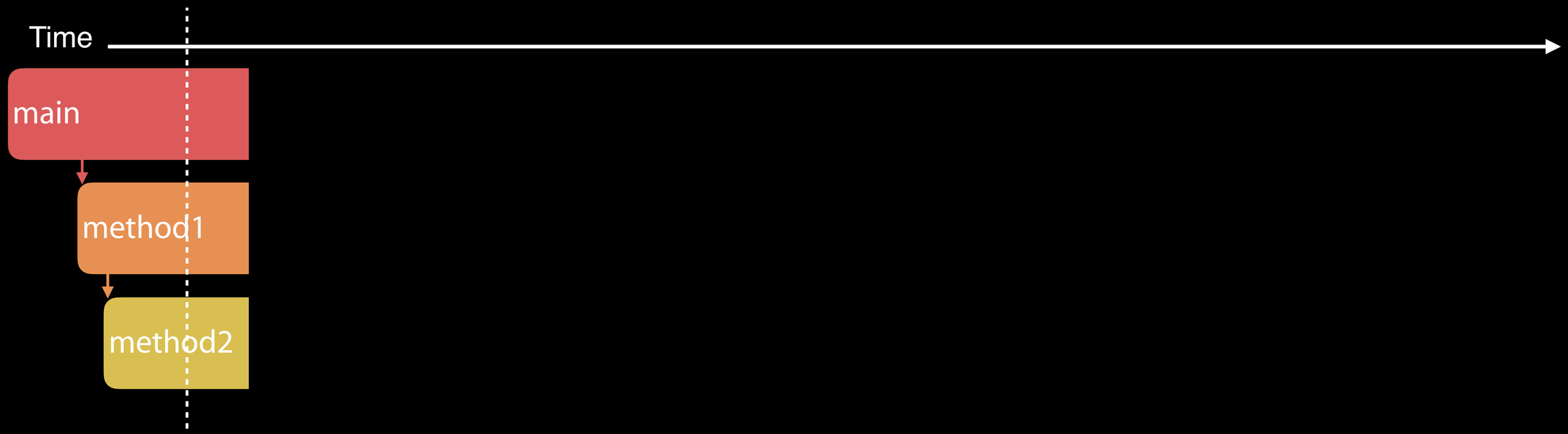


# *Demo*

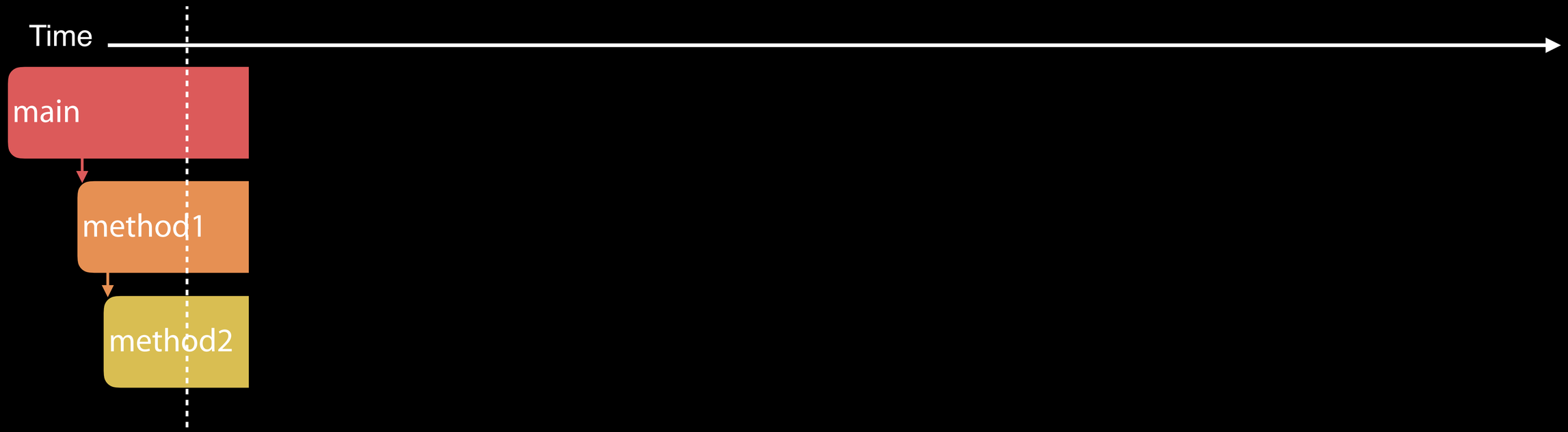
Profiling an app


Time

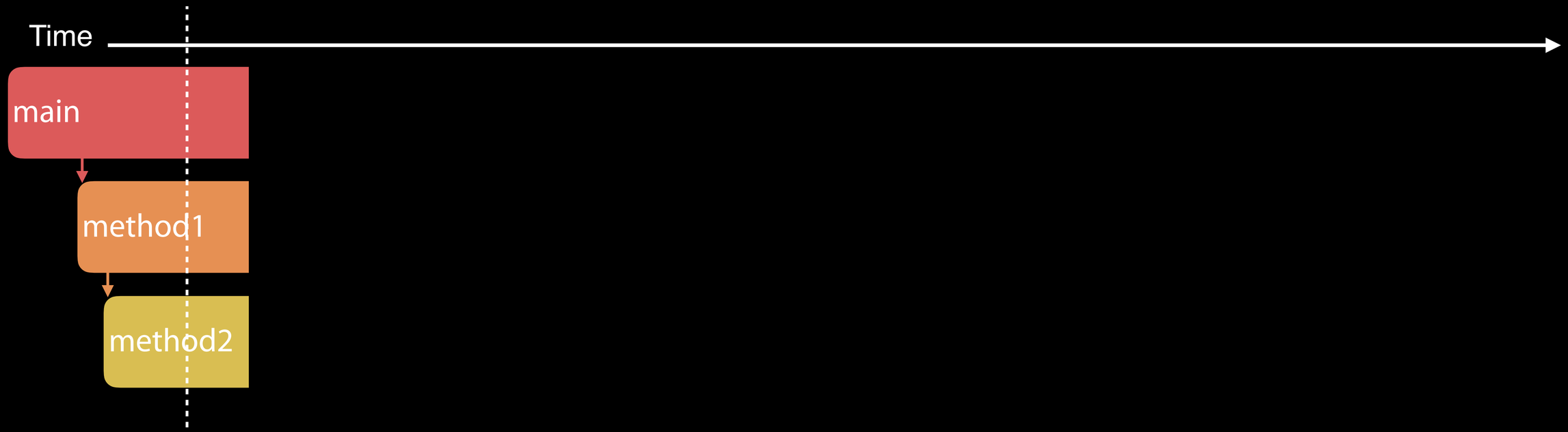

Time
main
method1
method2


Time
main
method1
method2

```
main()
    method1()
        method2()
```

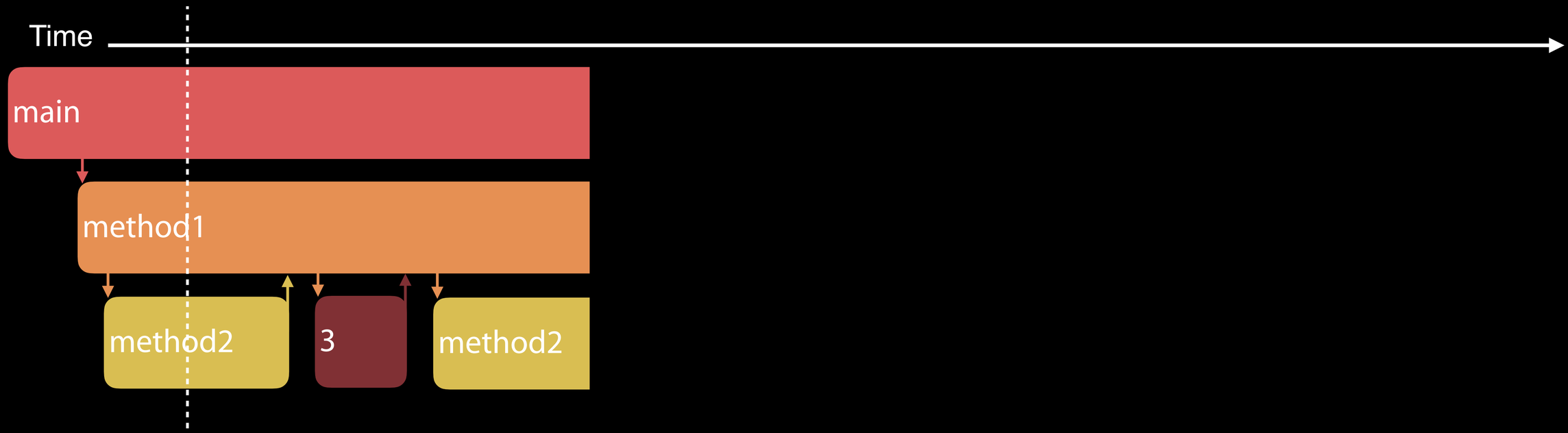


```
main()
    method1()
        method2()
```



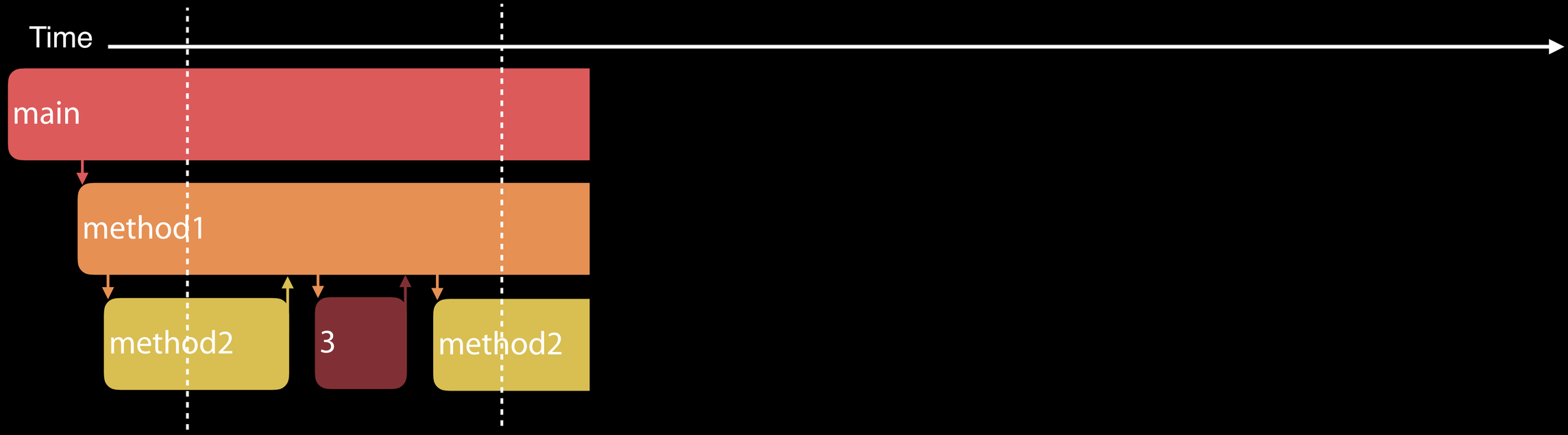

| | |
|---|---|
| 1 | main() |
| 1 | method1() |
| 1 | method2() |

```
main()
    method1()
        method2()
```




| | |
|---|---|
| 1 | main() |
| 1 | method1() |
| 1 | method2() |

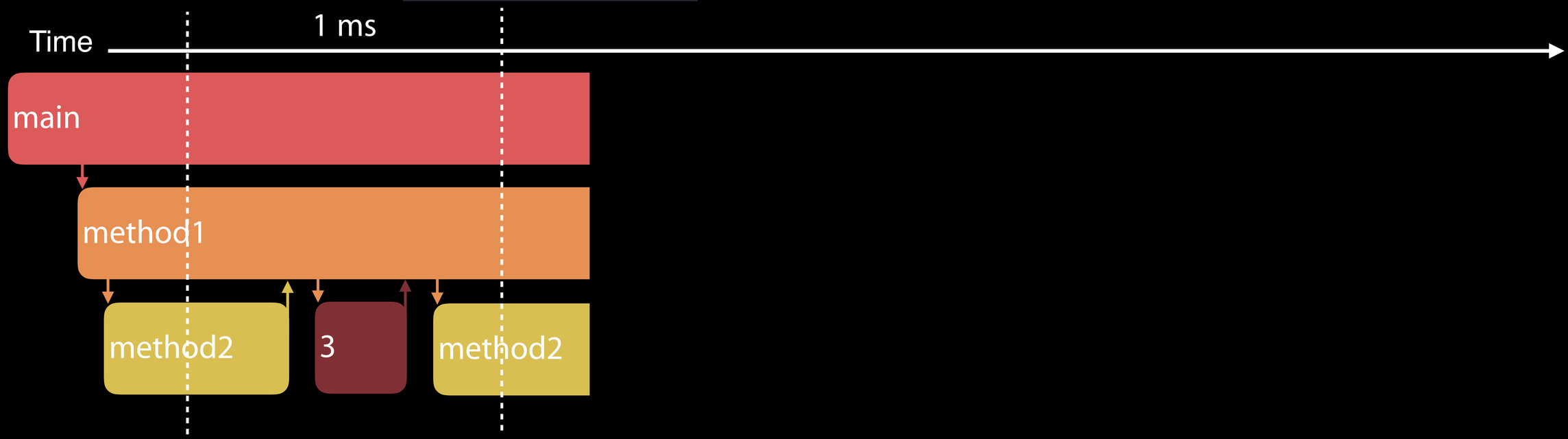
```
main()
    method1()
        method2()
```




| | |
|---|---|
| 1 | main() |
| 1 | method1() |
| 1 | method2() |

```
main()
    method1()
        method2()
```

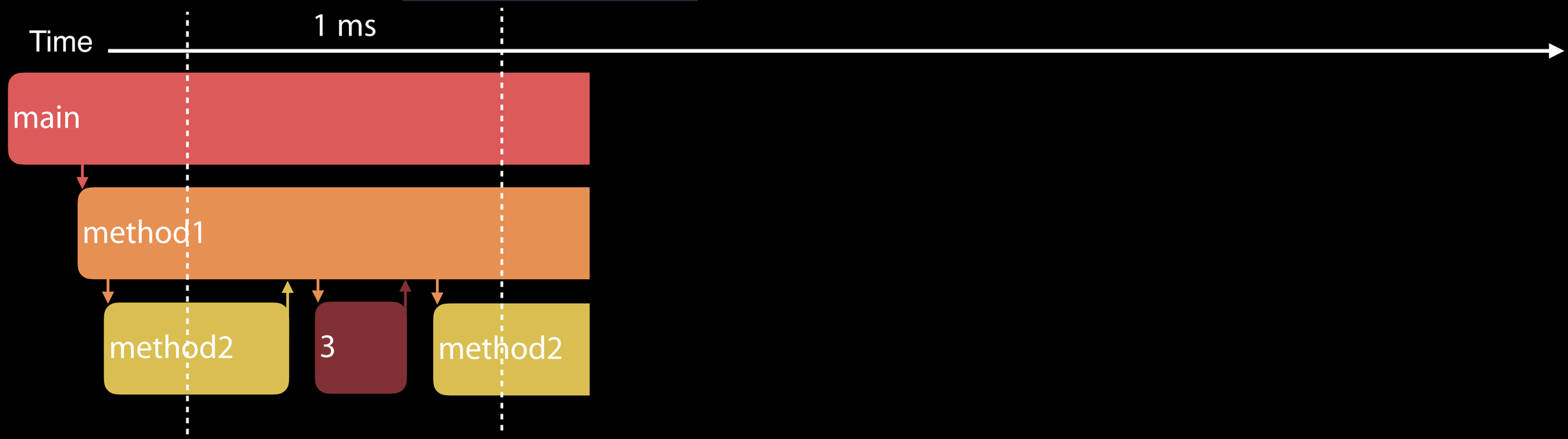
```
main()
    method1()
        method2()
```




| | |
|---|---|
| 1 | main() |
| 1 | method1() |
| 1 | method2() |

```
main()
    method1()
        method2()
```

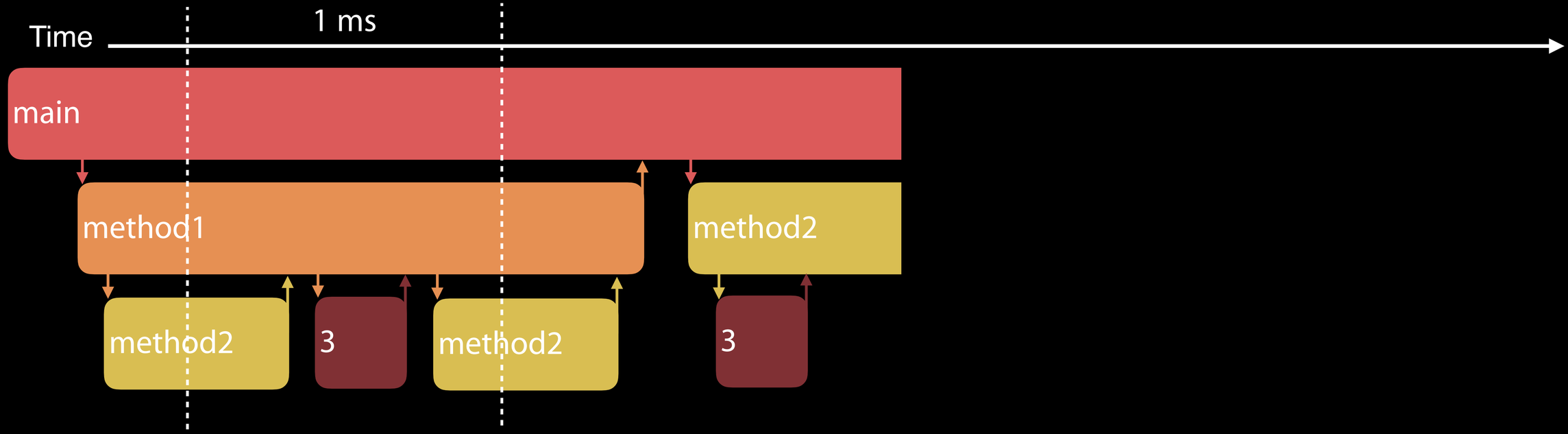
```
main()
    method1()
        method2()
```




| | |
|---|---|
| 2 | main() |
| 2 | method1() |
| 2 | method2() |

```
main()
    method1()
        method2()
```

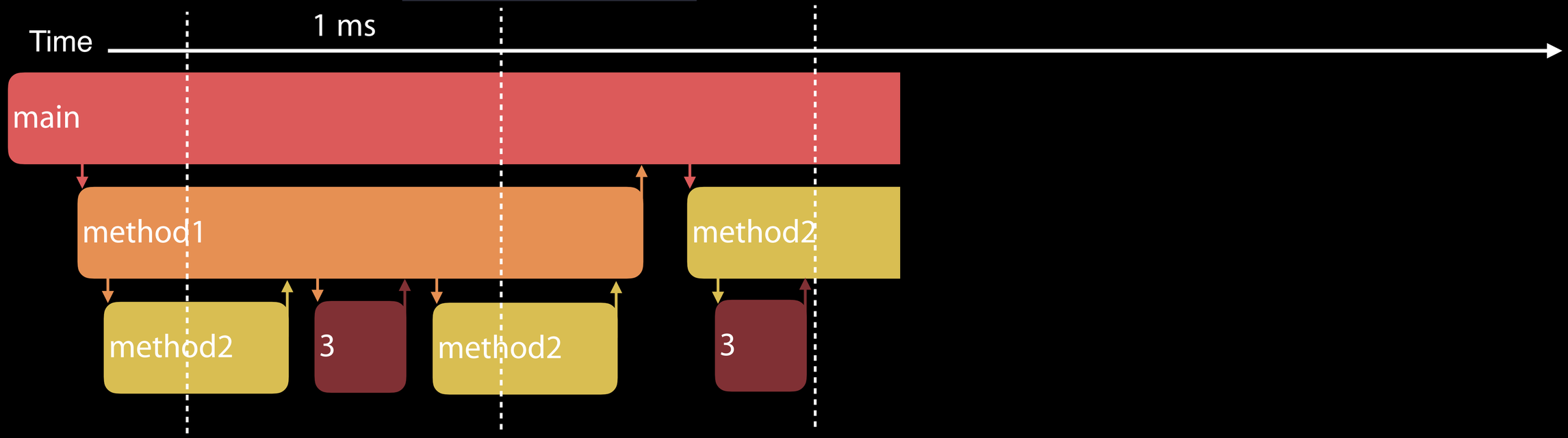
```
main()
    method1()
        method2()
```




| | |
|---|---|
| 2 | main() |
| 2 | method1() |
| 2 | method2() |

```
main()
    method1()
        method2()
```

```
main()
    method1()
        method2()
```

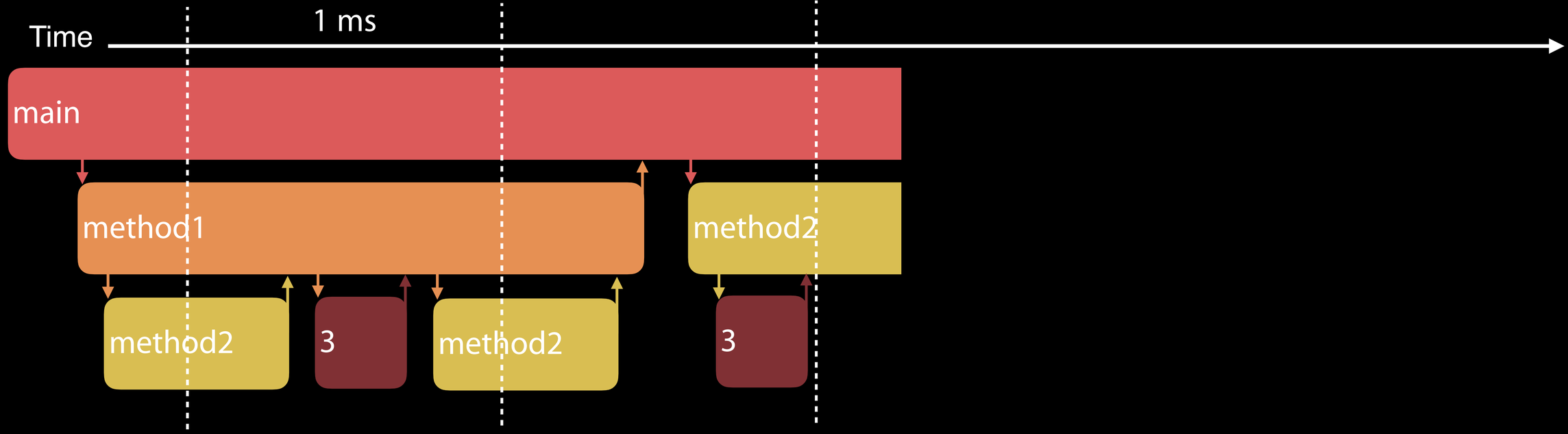



| | |
|---|---|
| 2 | `main()` |
| 2 | `    method1()` |
| 2 | `        method2()` |

```
main()
    method1()
        method2()
```

```
main()
    method1()
        method2()
```

```
main()
    method2()
```

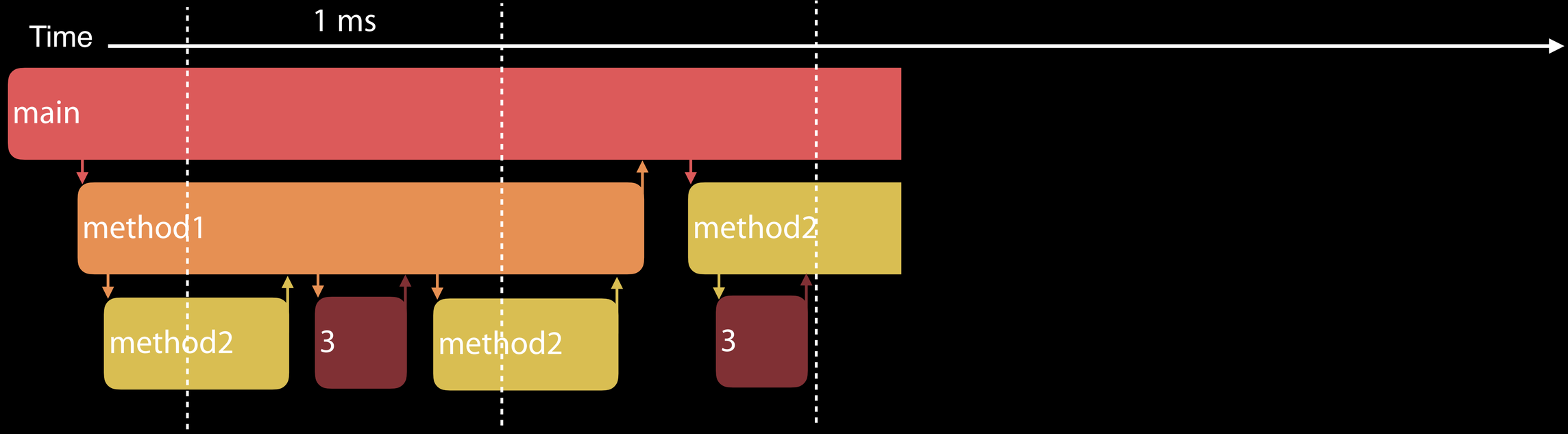



| | |
|---|---|
| 2 | main() |
| 2 | method1() |
| 2 | method2() |

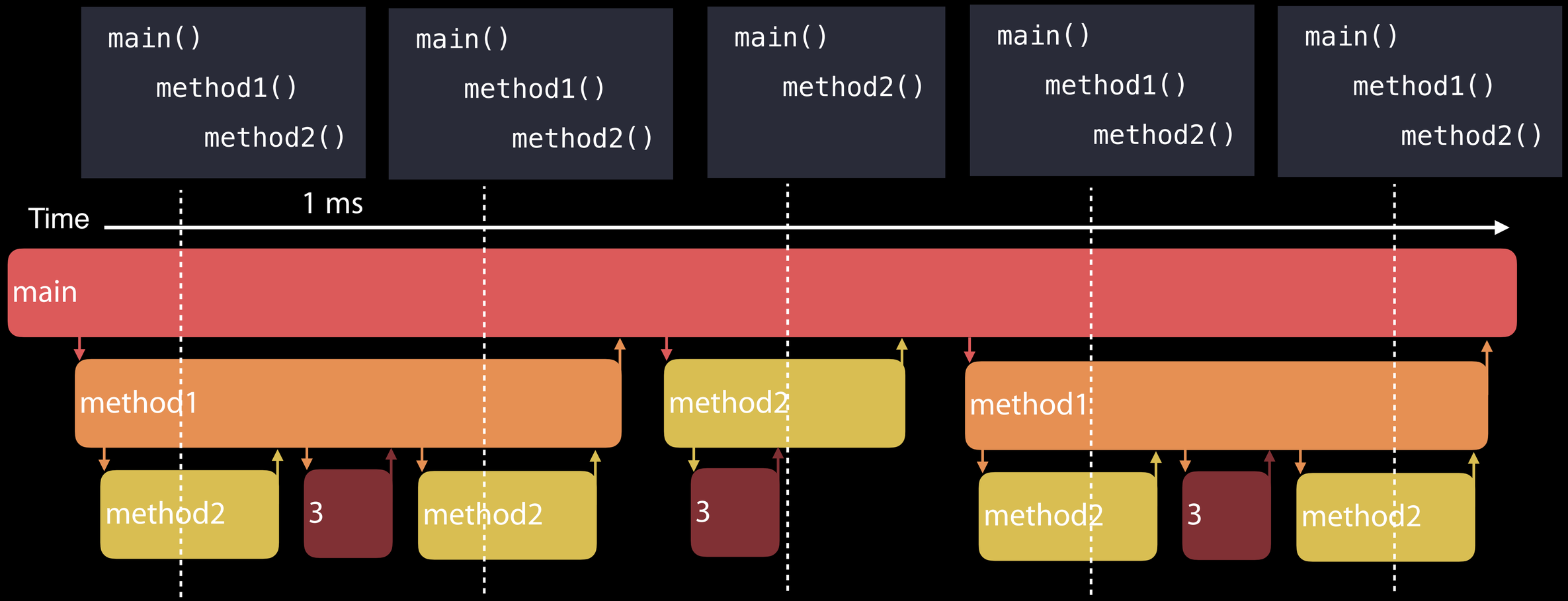
```
main()
    method1()
        method2()
```

```
main()
    method1()
        method2()
```

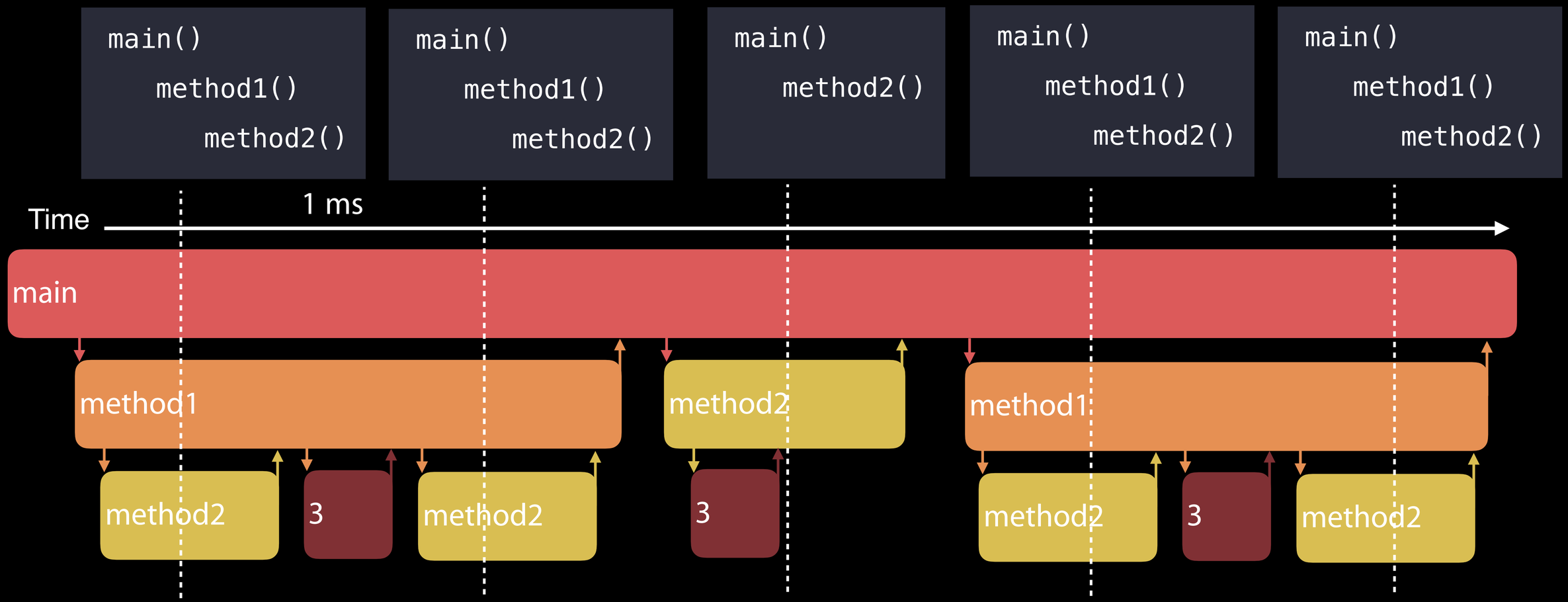
```
main()
    method2()
```




| | |
|---|---|
| 3 | main() |
| 2 | method1() |
| 2 | method2() |
| 1 | method2() |




| | |
|---|---|
| 3 | main() |
| 2 | method1() |
| 2 | method2() |
| 1 | method2() |




| | |
|---|---|
| 5 | main() |
| 4 | method1() |
| 4 | method2() |
| 1 | method2() |

# The Call Tree

| | |
|---|---|
| 5 | main() |
| 4 | method1() |
| 4 | method2() |
| 1 | method2() |

# The Call Tree

Doesn’t measure duration

| | |
|---|---|
| 5 | main() |
| 4 | method1() |
| 4 | method2() |
| 1 | method2() |

# The Call Tree

Doesn’t measure duration

Aggregates samples into a useful summary

| | |
|---|---|
| 5 | main() |
| 4 | method1() |
| 4 | method2() |
| 1 | method2() |

# The Call Tree

Doesn’t measure duration

Aggregates samples into a useful summary

- Long running vs. repetitive

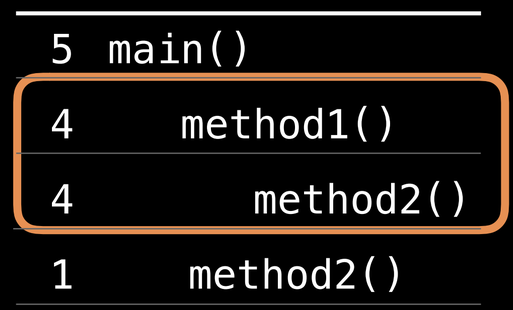
| | |
|---|---|
| 5 | main() |
| 4 | method1() |
| 4 | method2() |
| 1 | method2() |

# The Call Tree

Doesn’t measure duration

Aggregates samples into a useful summary

- Long running vs. repetitive



# The Call Tree

Doesn’t measure duration

Aggregates samples into a useful summary

- Long running vs. repetitive
- Focuses on CPU usage

| | |
|---|---|
| 5 | main() |
| 4 | method1() |
| 4 | method2() |
| 1 | method2() |

# The Call Tree

Doesn’t measure duration

Aggregates samples into a useful summary

- Long running vs. repetitive

Focuses on CPU usage

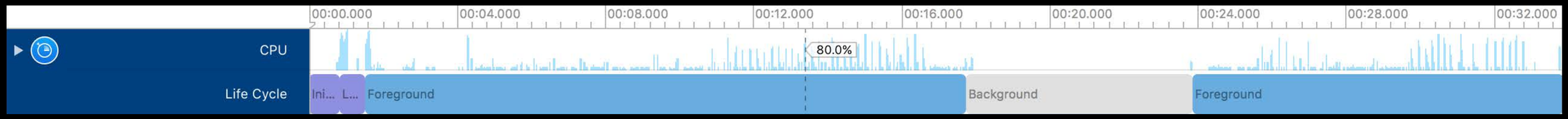
- Doesn’t capture everything

| | |
|---|---|
| 5 | main() |
| 4 | method1() |
| 4 | method2() |
| 1 | method2() |

# *Demo*

## Going faster and doing less

# Starting the Investigation




Look for the unexpected

- Trackpad friendly
- Drag to apply a filter
- Draggable filter edges
- Option-drag to zoom in control-drag to zoom out
- Data values on hover

# What We Did

# What We Did

Going faster

# What We Did

Going faster

- Focused on an area of high CPU usage

# What We Did

Going faster

- Focused on an area of high CPU usage
- Examined the call tree, looking for where the work was happening

# What We Did

Going faster

- Focused on an area of high CPU usage
- Examined the call tree, looking for where the work was happening
- Walked back to our code

# What We Did

Going faster

- Focused on an area of high CPU usage
- Examined the call tree, looking for where the work was happening
- Walked back to our code
- Inspected our code

# What We Did

Going faster

- Focused on an area of high CPU usage
- Examined the call tree, looking for where the work was happening
- Walked back to our code
- Inspected our code
- Made it faster

# What We Did

Going faster

- Focused on an area of high CPU usage
- Examined the call tree, looking for where the work was happening
- Walked back to our code
- Inspected our code
- Made it faster
- Verified the changes

# What We Did

Going faster

- Focused on an area of high CPU usage
- Examined the call tree, looking for where the work was happening
- Walked back to our code
- Inspected our code
- Made it faster
- Verified the changes
- Saved the user's time

# What We Did

# What We Did

Doing less

# What We Did

Doing less

- Focused on low, but unexpected CPU usage

# What We Did

Doing less

- Focused on low, but unexpected CPU usage
- Examined the call tree

# What We Did

Doing less

- Focused on low, but unexpected CPU usage
- Examined the call tree
- Determined the frameworks involved

# What We Did

Doing less

- Focused on low, but unexpected CPU usage
- Examined the call tree
- Determined the frameworks involved
- Stopped doing unnecessary work

# What We Did

Doing less

- Focused on low, but unexpected CPU usage
- Examined the call tree
- Determined the frameworks involved
- Stopped doing unnecessary work
- Verified the changes

# What We Did

Doing less

- Focused on low, but unexpected CPU usage
- Examined the call tree
- Determined the frameworks involved
- Stopped doing unnecessary work
- Verified the changes
- Improved battery life

# Responsiveness

# Responsiveness

# Responsiveness

The main thread does all the UI work

# Responsiveness

The main thread does all the UI work

- Run loop waiting for events

# Responsiveness

The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance

# Responsiveness

The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain

# Responsiveness

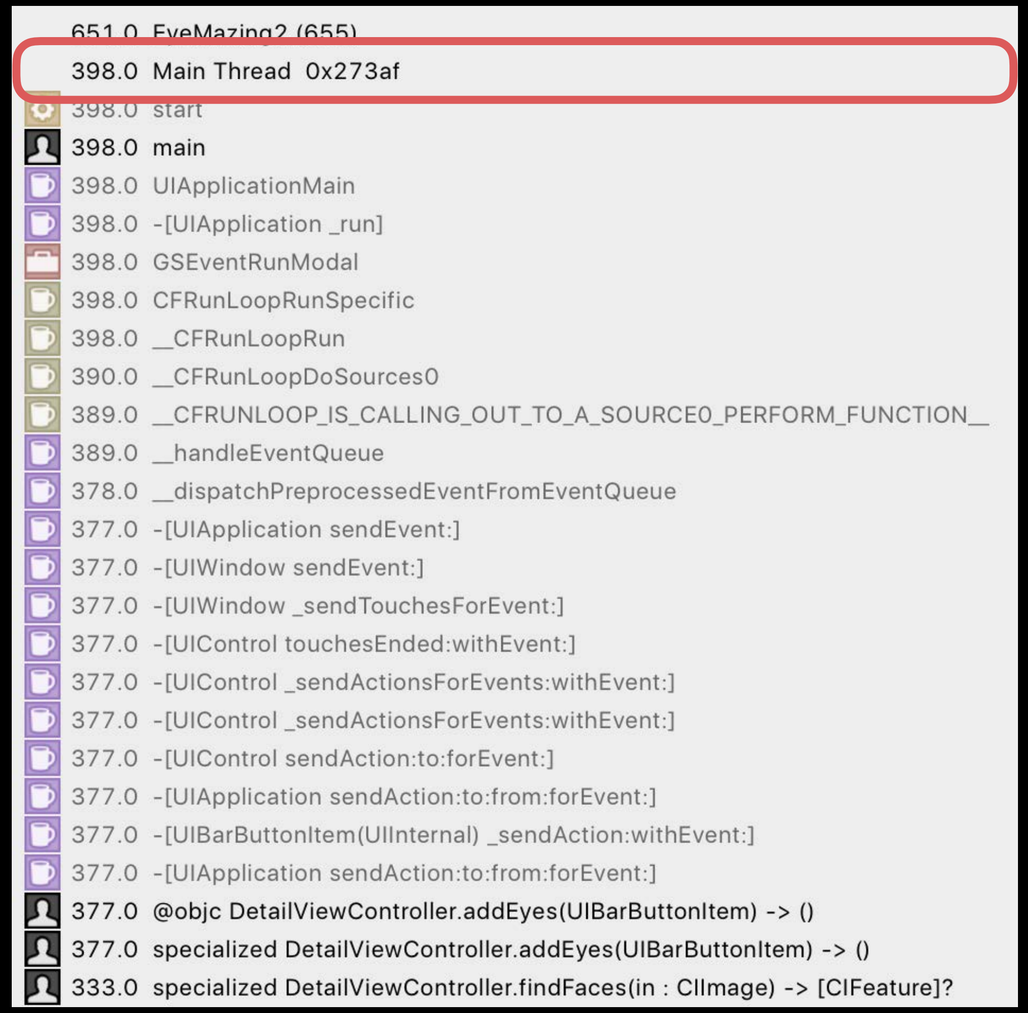
The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain
- Your code gets invoked

# Responsiveness

The main thread does all the UI work

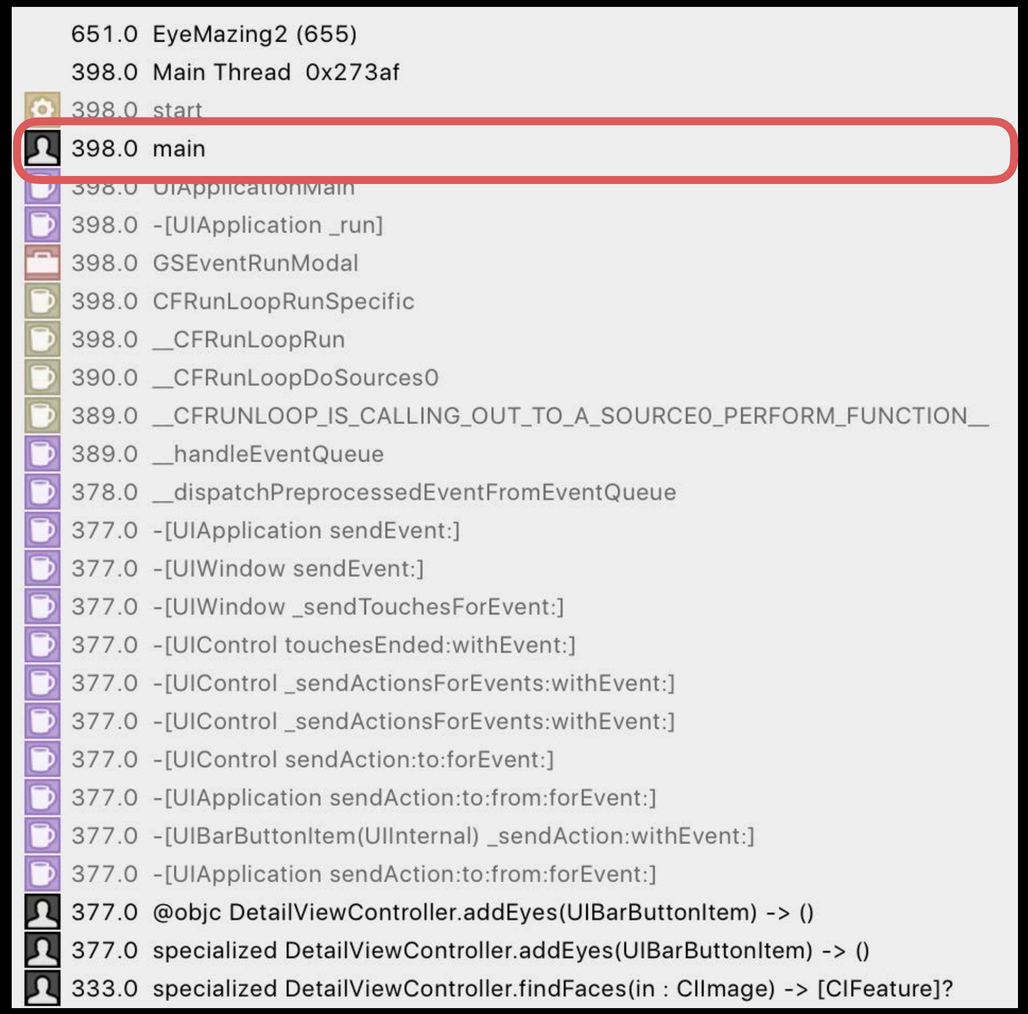
- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain
- Your code gets invoked



# Responsiveness

The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain
- Your code gets invoked



# Responsiveness

The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain
- Your code gets invoked

# Responsiveness

The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain
- Your code gets invoked

# Responsiveness

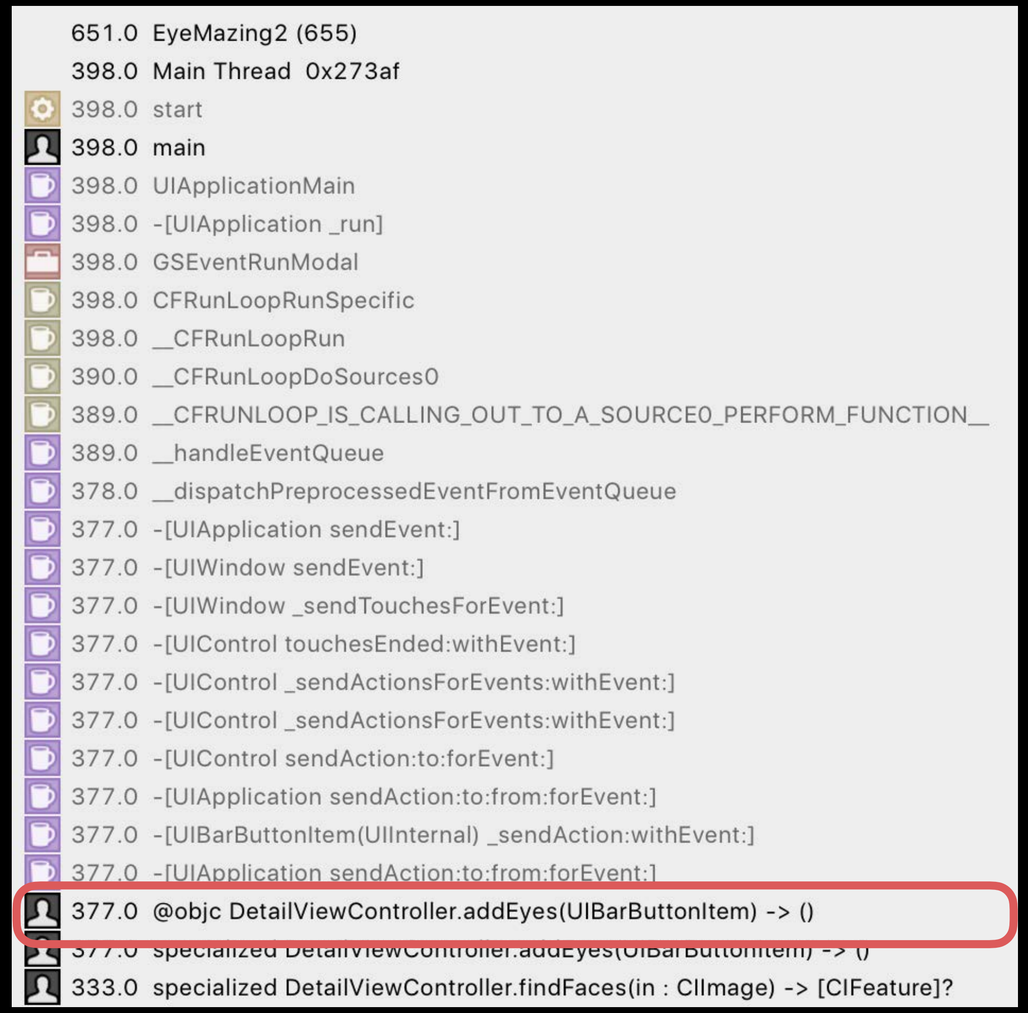
The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain
- Your code gets invoked

# Responsiveness

The main thread does all the UI work

- Run loop waiting for events
- Sends events to your UIApplication instance
- Passes through the responder chain
- Your code gets invoked



# Responsiveness

# Responsiveness

When busy, the main thread can’t process events

# Responsiveness

When busy, the main thread can’t process events

- The queue backs up

# Responsiveness

When busy, the main thread can’t process events

- The queue backs up
- Stuttering and hiccups

# Responsiveness

When busy, the main thread can’t process events

- The queue backs up
- Stuttering and hiccups
- App becomes unresponsive

# Responsiveness

When busy, the main thread can’t process events

- The queue backs up
- Stuttering and hiccups
- App becomes unresponsive

Keep the main thread free

# *Demo*

## Improving responsiveness

# What We Did

# What We Did

Responsiveness

# What We Did

Responsiveness

- Examined the CPU spikes

# What We Did

Responsiveness

- Examined the CPU spikes
- Focused on the main thread

# What We Did

Responsiveness

- Examined the CPU spikes
- Focused on the main thread
- Identified non-UI work happening on the main thread

# What We Did

Responsiveness

- Examined the CPU spikes
- Focused on the main thread
- Identified non-UI work happening on the main thread
- Distributed the work across multiple threads

# What We Did

Responsiveness

- Examined the CPU spikes
- Focused on the main thread
- Identified non-UI work happening on the main thread
- Distributed the work across multiple threads
- Verified the changes

# What We Did

Responsiveness

- Examined the CPU spikes
- Focused on the main thread
- Identified non-UI work happening on the main thread
- Distributed the work across multiple threads
- Verified the changes
- Achieved a better user experience

# Regarding Optimization

# Regarding Optimization

The fixes were simple, but added complexity

# Regarding Optimization

The fixes were simple, but added complexity

By finding them early in the development process, there's more time to verify correctness

# Regarding Optimization

The fixes were simple, but added complexity

By finding them early in the development process, there's more time to verify correctness

Profile early, profile often!

# Regarding Optimization

The fixes were simple, but added complexity

By finding them early in the development process, there's more time to verify correctness

Profile early, profile often!

Sometimes get big gains for little effort

# Best Practices

Always profile release builds

Always profile on the device

Run with old devices

Use large data sets where it makes sense

- Look for poorly scaling code ($O(n^2)$, etc.)

# Summary

# Summary

If you want …

# Summary

If you want …

- The best experience for your users across all the devices you support

# Summary

If you want …

- The best experience for your users across all the devices you support
  - A faster application

# Summary

If you want …

- The best experience for your users across all the devices you support
  - A faster application
  - Better battery life

# Summary

If you want …

- The best experience for your users across all the devices you support
  - A faster application
  - Better battery life
  - More responsive UI

# Summary

If you want …

- The best experience for your users across all the devices you support
  - A faster application
  - Better battery life
  - More responsive UI

Then you should …

# Summary

If you want …

- The best experience for your users across all the devices you support
  - A faster application
  - Better battery life
  - More responsive UI

Then you should …

- Profile early

# Summary

If you want …

- The best experience for your users across all the devices you support
  - A faster application
  - Better battery life
  - More responsive UI

Then you should …

- Profile early
- Profile often

# More Information

https://developer.apple.com/wwdc16/418

# Related Sessions

| | | |
|---|---|---|
| Optimizing App Startup Time | Mission | Wednesday 10:00AM |
| System Trace in Depth | Nob Hill | Thursday 9:00AM |
| Thread Sanitizer and Static Analysis | Mission | Thursday 10:00AM |
| Optimizing I/O for Performance and Battery Life | Nob Hill | Friday 11:00AM |
| Concurrent Programming with GCD in Swift 3 | Pacific Heights | Friday 4:00PM |
| Unified Logging and Activity Tracing | Nob Hill | Friday 5:00PM |
| Profiling in Depth | | WWDC 2015 |

# Labs

| | | |
|---|---|---|
| Xcode Open Hours | Developer Tools Lab B | Friday 3:00PM |
| Swift Open Hours | Developer Tools Lab A | Friday 3:00PM |

WWDC16