
WTF

WTF

Is So Slow In My App?

Greg Stromire

PDX CocoaHeads - May 9, 2018

About Me

Software Engineer @ **Tozny**

- Cryptography and Privacy Company
- I'm *often* developing for iOS

PDX CocoaHeads Member Since **January 2014**

Who has used:

Who has used:

- XCTest?

Who has used:

- `XCTest?`
- `measure?`

Who has used:

- `XCTest`?
- `measure`?
- Benchmarks?

Who has used:

- `XCTest`?
- `measure`?
- Benchmarks?
- *Time Profiler™*?

Agenda

Agenda

1. What this talk is *not* ❌

Agenda

1. What this talk is *not* ❌
2. Notes about "Performance" 

Agenda

1. What this talk is *not* ❌
2. Notes about "Performance" 
3. Example Application 📱

Agenda

1. What this talk is *not* ❌
2. Notes about "Performance" 
3. Example Application 📱
4. *Time Profiler*™ 🕒

Agenda

1. What this talk is *not* ❌
2. Notes about "Performance" 
3. Example Application 📱
4. *Time Profiler*™ 🕒
5. `XCTesting` for Performance ✅

Agenda

1. What this talk is *not* ❌
2. Notes about "Performance" 
3. Example Application 📱
4. *Time Profiler*™ 🕒
5. `XCTesting` for Performance ✅
6. Address Issue and Confirm 💯

Agenda

1. What this talk is *not* ❌
2. Notes about "Performance" 
3. Example Application 📱
4. *Time Profiler*™ 🕒
5. `XCTesting` for Performance ✅
6. Address Issue and Confirm 💯
7. `Check` for Regressions ✨

What this talk is *Not*

What this talk is *Not*

"Performance" as it relates to:

- Network
- Memory
- Battery
- Etc.

Exhaustive List of Possible Performance Issues

A Magic Bullet

When we say Performance we often mean Responsiveness

We've found that if it takes too much longer than a hundred milliseconds, your user starts to feel it.

– WWDC 2015 Session 230

Consider a copy of your `Release` Scheme with wrappers around UX[1]:

```
@IBAction func buttonPressed() {
#if PERFORMANCE
    let before = CFAbsoluteTimeGetCurrent()
#endif
    intensiveOperation() // long running task
#if PERFORMANCE
    let after  = CFAbsoluteTimeGetCurrent()
    print("\"Real\" time of operation: \(after - before)")
#endif
}
```

[1] WWDC 2015 Session 230

Various Performance Tips[2]

[2] 25 iOS App Performance Tips - raywenderlich.com

Various Performance Tips[2]

- Use a `reuseIdentifier` Where Appropriate

[2] 25 iOS App Performance Tips - raywenderlich.com

Various Performance Tips[2]

- Use a `reuseIdentifier` Where Appropriate
- Set Views as Opaque When Possible

[2] 25 iOS App Performance Tips - raywenderlich.com

Various Performance Tips[2]

- Use a `reuseIdentifier` Where Appropriate
- Set Views as Opaque When Possible
- Don’t Block the Main Thread

[2] 25 iOS App Performance Tips - raywenderlich.com

Various Performance Tips[2]

- Use a `reuseIdentifier` Where Appropriate
- Set Views as Opaque When Possible
- Don’t Block the Main Thread
- Size Images to Image Views

[2] 25 iOS App Performance Tips - raywenderlich.com

Various Performance Tips[2]

- Use a `reuseIdentifier` Where Appropriate
- Set Views as Opaque When Possible
- Don’t Block the Main Thread
- Size Images to Image Views
- Use Sprite Sheets

---

[2] 25 iOS App Performance Tips - raywenderlich.com

Various Performance Tips[2]

- Use a `reuseIdentifier` Where Appropriate
- Set Views as Opaque When Possible
- Don’t Block the Main Thread
- Size Images to Image Views
- Use Sprite Sheets
- Avoid Date Formatters Where Possible

---

[2] 25 iOS App Performance Tips - raywenderlich.com

That said...

Beware

Premature Optimization

Premature Optimization

Premature Optimization

- Don't guess, profile!

Premature Optimization

- Don't guess, profile!
- Be systematic!

Example App

Example App

Secret message application.
Encrypts 🔒 and decrypts 🔑 user-supplied text.
Uses large file of most common passwords

- Simplified version of a real problem I encountered
- Anyone heard of `libsodium`[3]?

[3] libsodium


WTF

WTF Is So Slow In My App?

WTF Is So Slow In My App?

1. Am I using a slow algorithm?

WTF Is So Slow In My App?

1. Am I using a slow algorithm?
2. Is this library implementation slow?

WTF Is So Slow In My App?

1. Am I using a slow algorithm?
2. Is this library implementation slow?
3. Are the instance creations expensive?

WTF Is So Slow In My App?

1. Am I using a slow algorithm?
2. Is this library implementation slow?
3. Are the instance creations expensive?
4. Is this a result of moving around the `String` (i.e. Value type)?

WTF Is So Slow In My App?

1. Am I using a slow algorithm?
2. Is this library implementation slow?
3. Are the instance creations expensive?
4. Is this a result of moving around the `String` (i.e. Value type)?
5. Something else?

WTF Is So Slow In My App?

- ~~Use a `reuseIdentifier` Where Appropriate~~
- ~~Set Views as Opaque When Possible~~
- **Don’t Block the Main Thread**
- ~~Size Images to Image Views~~
- ~~Use Sprite Sheets~~
- ~~Avoid Date Formatters Where Possible~~

Time Profiler™

Time Profiler™

Initial Findings:

- Understanding String Representations in Swift
- Strings, characters, and performance in Swift—a deep dive

Time Profiler™

Can be in-app or in-tests

Allows inspection and drill-down

Heaviest stack trace on the right panel

Hold down `option` and click disclosure triangle

Click to go to calling code

Did we identify any issues?

XCTesting for Performance

XCTesting for Performance

XCTesting for Performance

- `measure`, `measureMetrics`, `startMeasuring`, `stopMeasuring`

XCTesting for Performance

- `measure`, `measureMetrics`, `startMeasuring`, `stopMeasuring`
- Baseline and STDDEV

`XCTest`ing for Performance

- `measure`, `measureMetrics`, `startMeasuring`, `stopMeasuring`
- Baseline and STDDEV
  - Baselines are *per device*

`XCTest`ing for Performance

- `measure`, `measureMetrics`, `startMeasuring`, `stopMeasuring`
- Baseline and STDDEV
  - Baselines are *per device*
  - Can commit baselines to project repo

Addressing the Issue

Addressing the Issue

- Base64URL 🐌

Addressing the Issue

- Base64URL 🐌
- Swift-Sodium 🐎

Addressing the Issue

- Base64URL 🐌
- Swift-Sodium 🐎
- Fix issue and Run Tests ✅

Addressing the Issue

- Base64URL 🐌
- Swift-Sodium 🐎
- Fix issue and Run Tests ✅
- Compare Baselines 💯

Checking for Regressions

Checking for Regressions

- Encoding random data -- does this new implementation match?

Checking for Regressions

- Encoding random data -- does this new implementation match?
- I should test with a diverse set of cases to raise my confidence.

Checking for Regressions

- Encoding random data -- does this new implementation match?
- I should test with a diverse set of cases to raise my confidence.
- But how?

Checking for Regressions

- Encoding random data -- does this new implementation match?
- I should test with a diverse set of cases to raise my confidence.
- But how?
- Enter `SwiftCheck`

`SwiftCheck`

- Based on `QuickCheck` from Haskell
- Generates 100 widely variable test cases
- Also **shrinks** the failing cases!
- General use testing tool -- I'm using it here for regressions
- Setup: `Arbitrary`, `forAll`, etc.

SwiftChecking for Regressions

Lessons From Our Performance Journey

1. Profile Early, Profile Often, and Let that Inform Optimizations
2. `XCTest` for Performance: `measure` and Baselines
3. Drill Deep with TimeProfiler™
4. Fix the Issue and Update Baselines
5. Check for Regressions with `SwiftCheck`

Thanks!

Questions?