

Faster App

@kevinzhov

About Kevin

- Develop **iOS** App Since 2011
- **Ruby on Rails** Since 2010
- **Design & Linux** Since 2006
- Author of **PRODUCTER PNChart**
- Founder of **Catch Inc**

Fact of Tech

Fact of Tech

- Tech is a **Service**
- Deliver Products which **Make Life Better**
- Dive into the **details**

Startups

Where Good Things
Happens

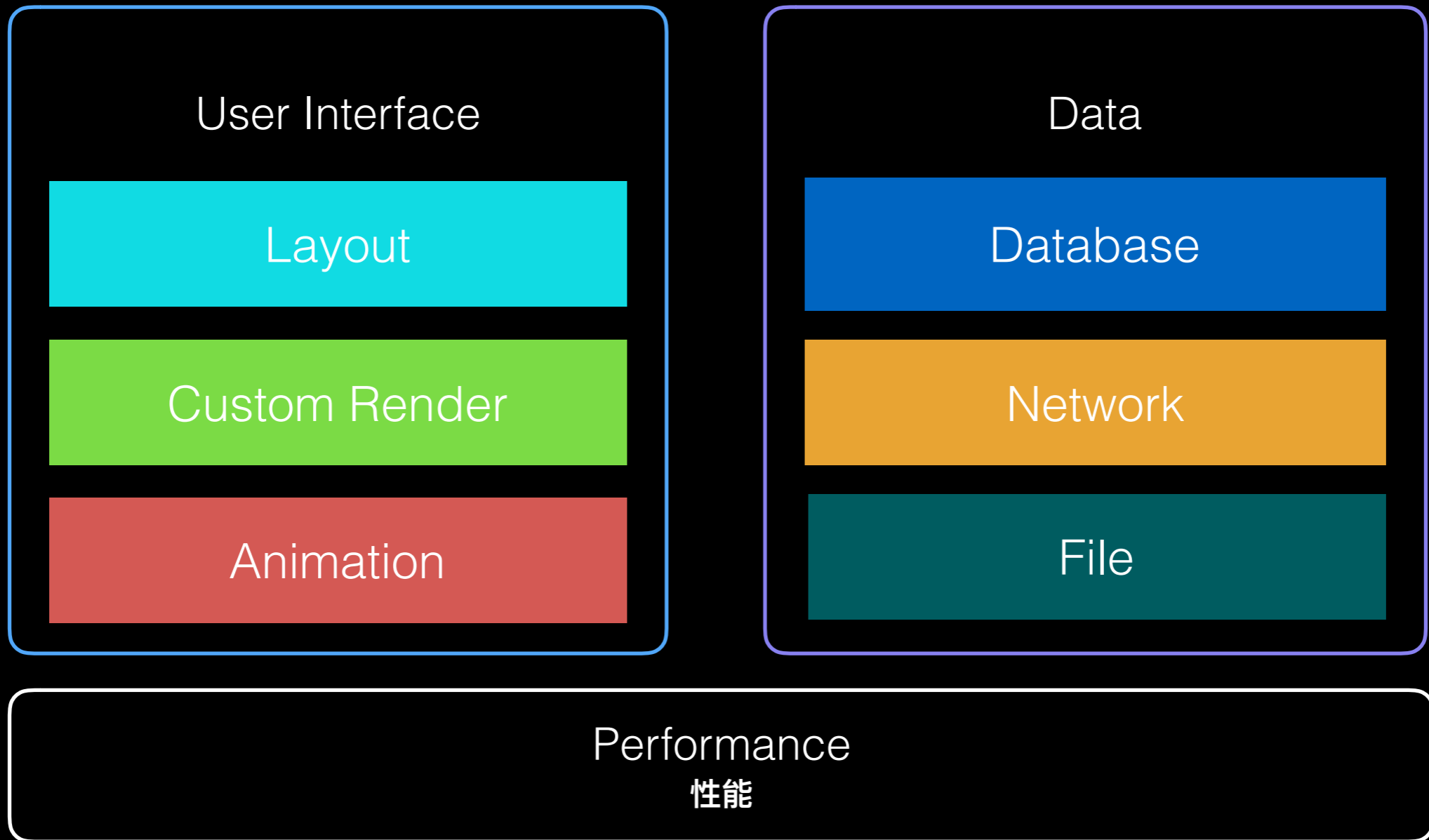
Reality of **Startups**

One Man

- **Agile** 1 - 2 Month
- **Stable** 99.8% Crash Free
- **Adaptable** All Device
- **Save** the Company

Let's talk about **it**

Performance



Layout

Layout Blind Spot

External

- iPad
- Rotation
- Call & Recording Bar
- Screen Size

Internal

Layout Blind Spot

External

- iPad
- Rotation
- Call & Recording Bar
- Screen Size

Internal

- Content Change
- Internationalization
- Dynamic Type

Layout Solution

Layout Solution

Programming Frame

- Flexible
- Powerful
- **Complicated**

Autoresizing Masks

- For Simple Interface
- Only External Changes

Hug Apple

AutoLayout

- **Fast** Auto Height Calculate
- Designer **Readable**
- Adapt **All** Situation
- UIView

When to use

Bad

Collection

Heavy Reusable

Animation

Good

Static Interface

Related Views

StackView

AutoLayout

Bad



- Changeable Cell
- Trigger too many needLayout
- FPS drops to 16/s

Solution



- Programming Frame
- UIView.performWithoutAnimation(_:)
- Cache Height & Text Frame

Note

- Don't change view programmatically by geometry property
- Programmatically View should set `translatesAutoresizingMaskIntoConstraints` to NO

Custom **Render**

Custom Render

- Customized Elements
- Complex Content
- Multimedia

Custom Render Solution

UIKit

Core Graphics

TextKit

YYText

Core Text

- Easy to use
- Interaction Friendly
- Flexible
- Powerful & Thread Safe
- **Complicated**
- Full-featured
- Hug Apple

Custom Elements

- Avatar (Image Rect Customized)
- Rich TextView

Decode In Background

Decode

```
let imageRef = self.CGImage
let colorSpace = CGColorSpaceCreateDeviceRGB()
let bitmapInfo = CGBitmapInfo(rawValue: CGImageAlphaInfo.PremultipliedLast.rawValue).rawValue
let contextHolder = UnsafeMutablePointer<Void>()
let context = CGContextCreate(contextHolder,
    CGImageGetWidth(imageRef),
    CGImageGetHeight(imageRef), 8, 0, colorSpace, bitmapInfo)
if let context = context {
    let rect = CGRectMake(0, 0, CGFloat(CGImageGetWidth(imageRef)),
        CGFloat(CGImageGetHeight(imageRef)))
    CGContextDrawImage(context, rect, imageRef)
    let decompressedImageRef = CGContextCreateImage(context)
    return UIImage(CGImage: decompressedImageRef!, scale: scale, orientation: self.imageOrientation)
} else {
    return nil
}
```

Crop In Background

Crop

```
let rect = CGRect(origin: CGPoint(x: 0, y: 0), size: self.size)
UIGraphicsBeginImageContextWithOptions(self.size, false, 1)
UIBezierPath(roundedRect: rect, cornerRadius: cornerRadius).addClip()
drawInRect(rect)
return UIGraphicsGetImageFromCurrentImageContext()
```

Cache

Rich

Carrier

2:59 PM



I am [@kevinzhov](#) and My email is kevinchou.c@gmail.com [#Catch#](#)
you can find my blog at <http://zhovkev.in>



[@ke...zhov](#)

Processing Text

Processing Text

- NSTextStorage processEditing()
- NSRegularExpression
enumerateMatchesInString
- addAttributes to result
- Record range

```

//For Mention

let mentionPattern = "@[^\s: , @]+\$?"

let mentionExpression = try? NSRegularExpression(pattern: mentionPattern,
options: NSRegularExpressionOptions())

if let mentionExpression = mentionExpression {
    mentionExpression.enumerateMatchesInString(self.string, options: NSMatchingOptions(),
range: paragraphRange, usingBlock: { (result, flags, stop) -> Void in

        if let result = result {
            let textValue = (self.string as NSString).substringWithRange(result.range)

            let textAttributes: [String : AnyObject]! =
[NSForegroundColorAttributeName: UIColor.blueColor(),
NSLinkAttributeName: textValue,
RichTextViewDetectedDataHandlerAttributeName: DetectedDataType.Mention.rawValue]

            self.addAttributes(textAttributes, range: result.range )

            self.mentionRanges.append(result.range)
        }

    })
}

```

NSLayoutManager for **Style**

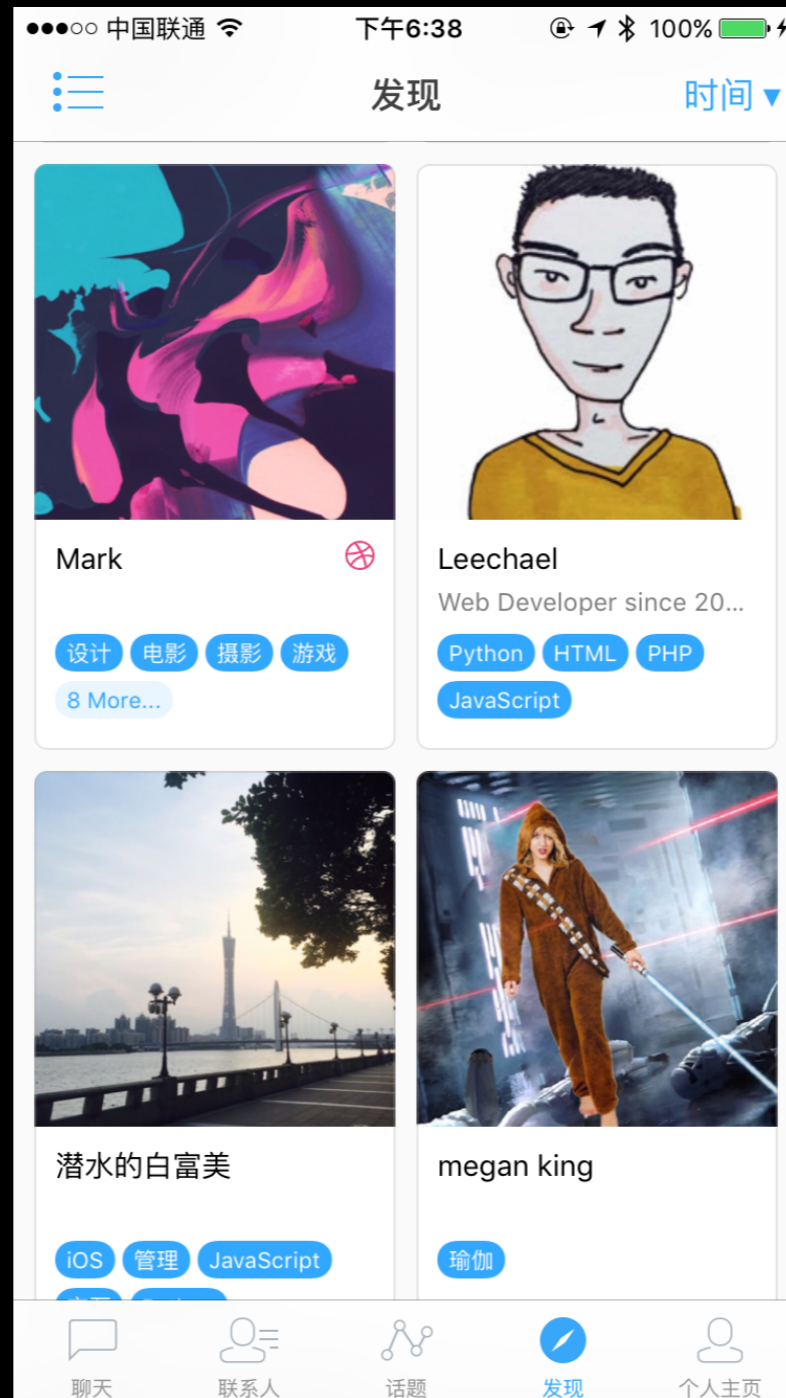
```
layoutManager.enumerateEnclosingRectsForGlyphRange(glyphRange,
withinSelectedGlyphRange: NSRange(NSNotFound, 0),
inTextContainer: textContainer,
usingBlock: { (rect, stop) -> Void in
    var rect = rect

    rect.origin.x += self.textContainerInset.left
    rect.origin.y += self.textContainerInset.top
    rect = UIEdgeInsetsInsetRect(rect, self.tapAreaInsets)

    complete(rect: rect, range: range, stop: true)
})
```

Complex Content
but **reused** a lot

Complex Content



Core Graphics

- UIBezierPath for **bounds**
- **NSParagraphStyle** NSString & drawInRect
- Convert to Image & Cache
- Boost FPS from 27~ 47 to 57fps


```
UIGraphicsBeginImageContextWithOptions(CGSize(), false, UIScreen.mainScreen().scale)
    //// Text Drawing
    let textRect = CGRectMake(0, 0, 0, 14)
    let textTextContent = NSString(string: skillLocal)
    let textStyle = NSParagraphStyle.defaultParagraphStyle().mutableCopy() as! NSMutableParagraphStyle
    textStyle.alignment = .Center

    let rectanglePath = UIBezierPath(roundedRect: CGRectMake(), cornerRadius:)
    fillColor.setFill()
    rectanglePath.fill()
    skillLabels.append(rect)
    textTextContent.drawInRect(rect, withAttributes: textFontAttributes)
    let backgroundImage = UIGraphicsGetImageFromCurrentImageContext()
    UIGraphicsEndImageContext()
```

Multimedia

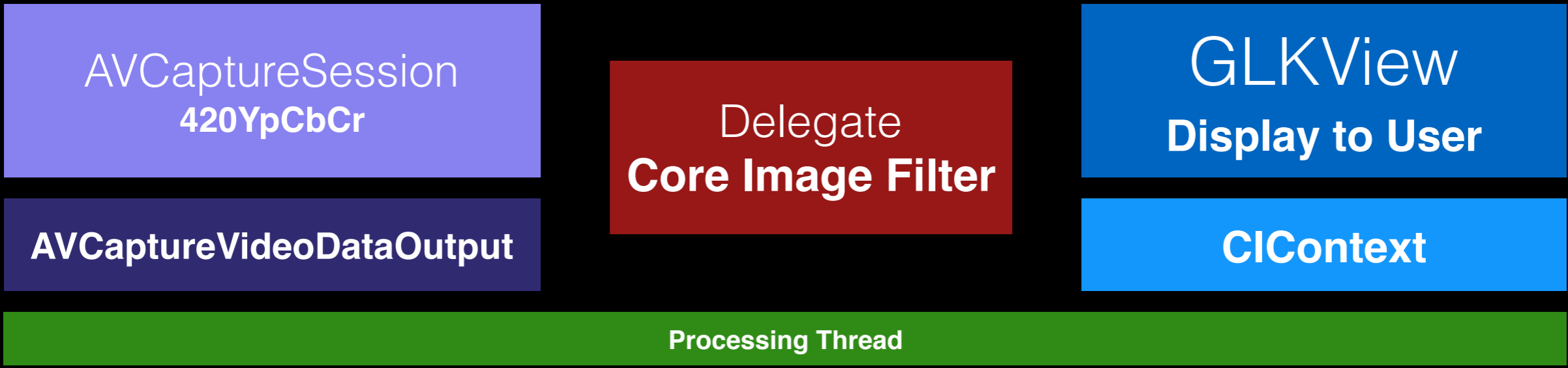
Multimedia

- **Live** Video Filter
- Video **background** playback

Live Image Filter

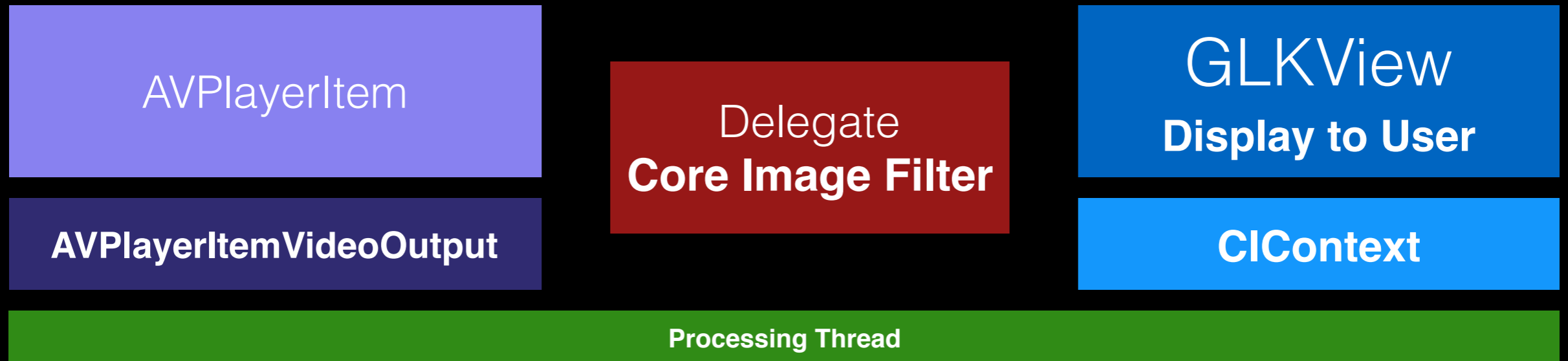
- Capture in separated threads
- Processing by Core Image
- Preview on GLKView

Video Flow



Video **Background** Playback

Video Background



Animation

Animation

- Antialiasing **allowsEdgeAntialiasing** or **Transparent Edge**
- **CGRectIntegral()** set Frame for better antialiasing performance
- Layer **shadowPath**
- Perspective **m34 -1.0/eyePosition**
- **allowsGroupOpacity**
- **shouldRasterize** for layer which **not** redrawn

Database

Data Needs

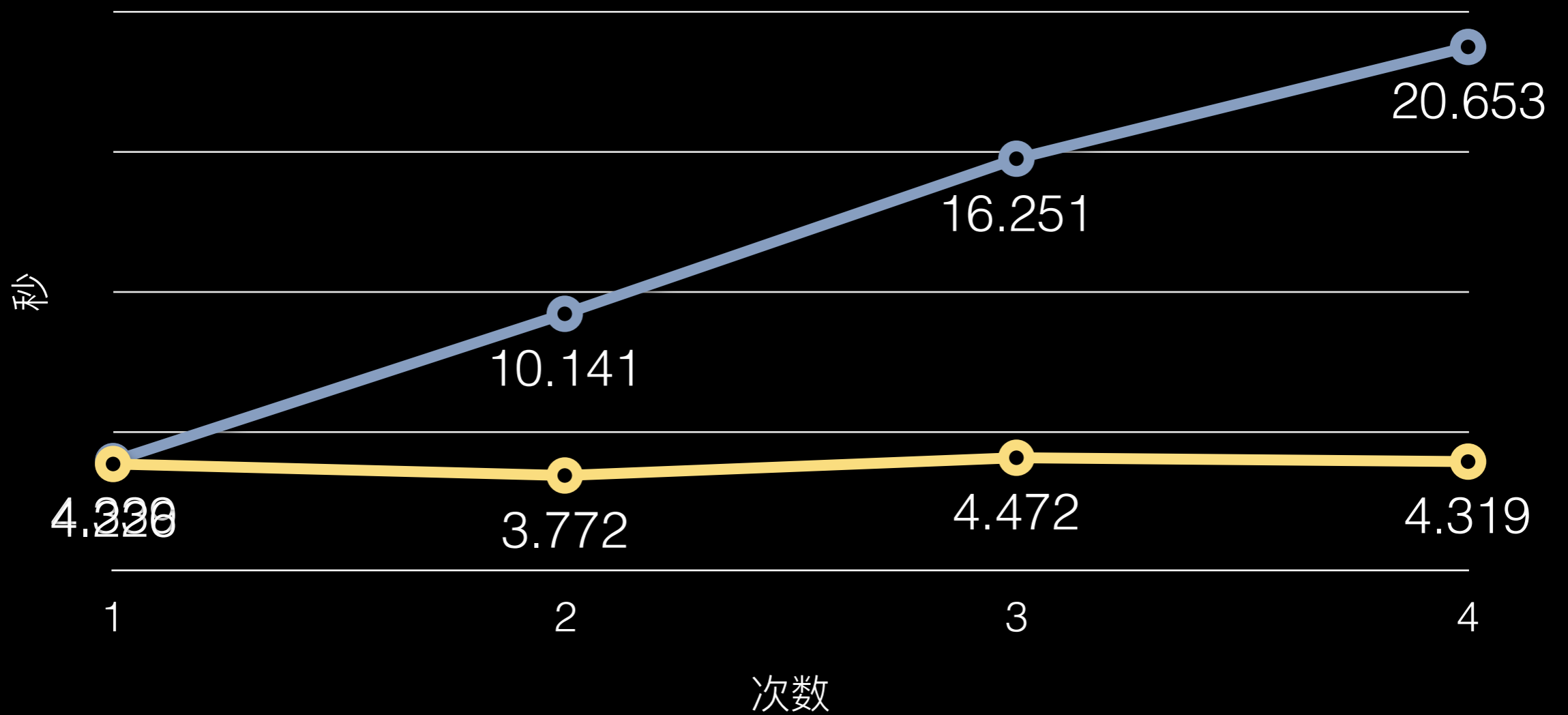
- Threading
- I/O Performance
- Query Performance
- Migration
- Model Relationship

CoreData or Realm

Insert 100k

● Realm 0.96.3

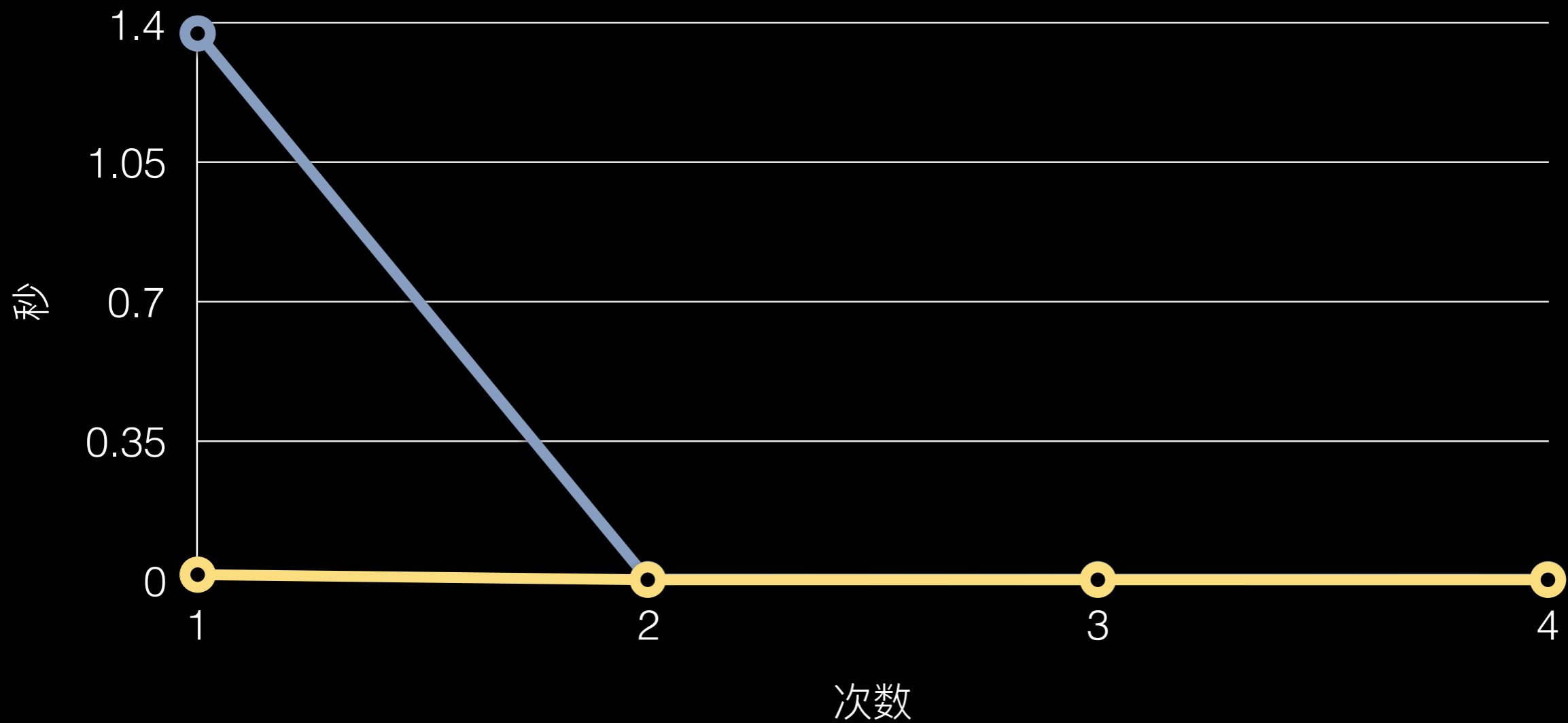
● CoreData iOS 9.2



Query from 400k

● Realm 0.96.3

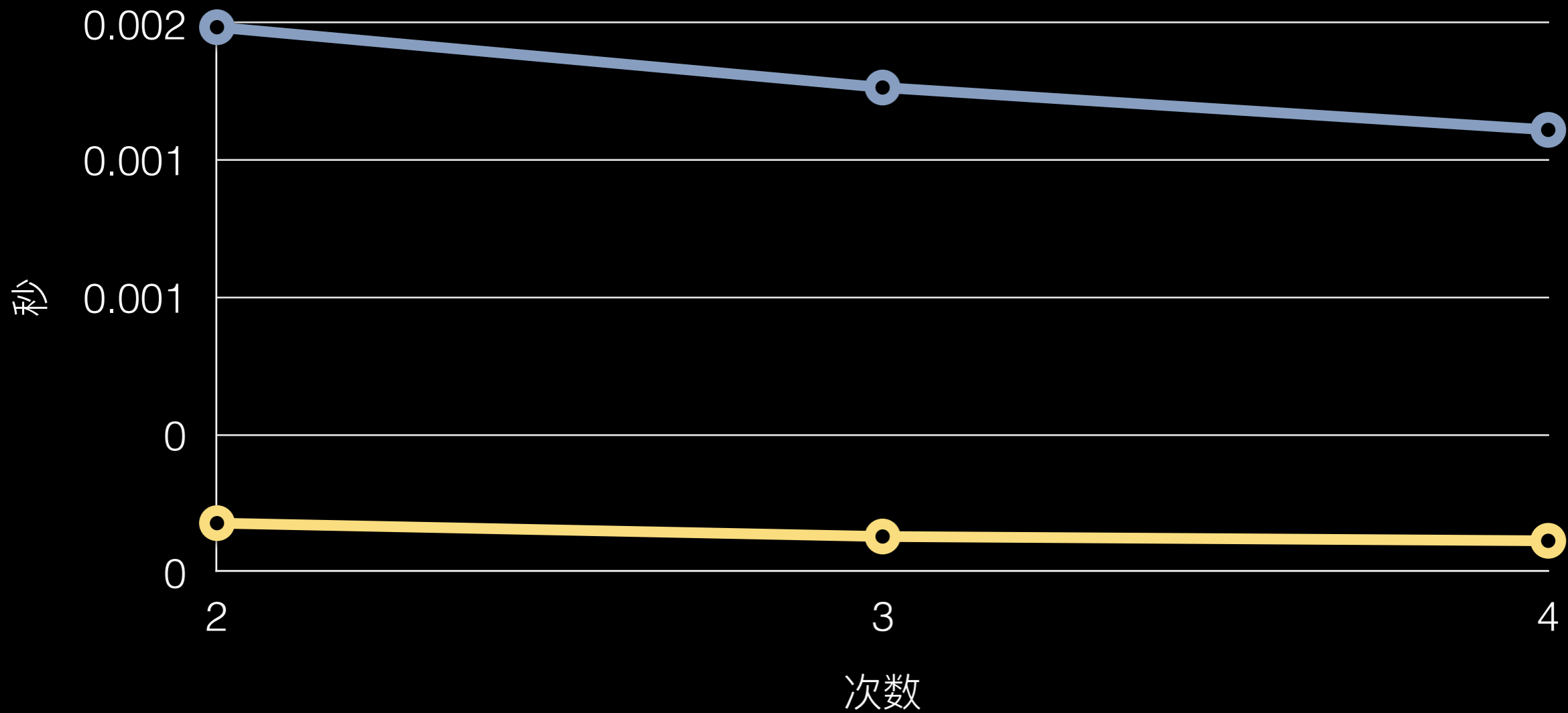
● CoreData iOS 9.2



Query from 400k

● Realm 0.96.3

● CoreData iOS 9.2



CoreData

- Build in Support
- Package Size
- GUI Support

Realm

- Great Threading Support
- Faster
- But **Fat**

Threading

CoreData
Notification

Realm
Sync When Commit

use Batch update

Network

Cache Solution

NSURLConnection

NSURLCache

ETag
HTTP ETag Field

Cache-Control

Enable **Gzip**

Performance vs Function

JSON

Manual

YYModel

Mantel

protobuf

“Happy app developing”

-@kevinzhov