

SwiftのAccelerateフレームワーク

宇佐見公輔 (usami-k)

2024-05-31

Mobile Act OSAKA | 3

自己紹介

- iOSアプリ開発エンジニア
- Mobile Act OSAKAの発起人
- 2013～2021：フェンリル
- 2022～現在：ゆめみ

Sponsors

ダイアモンドプラン



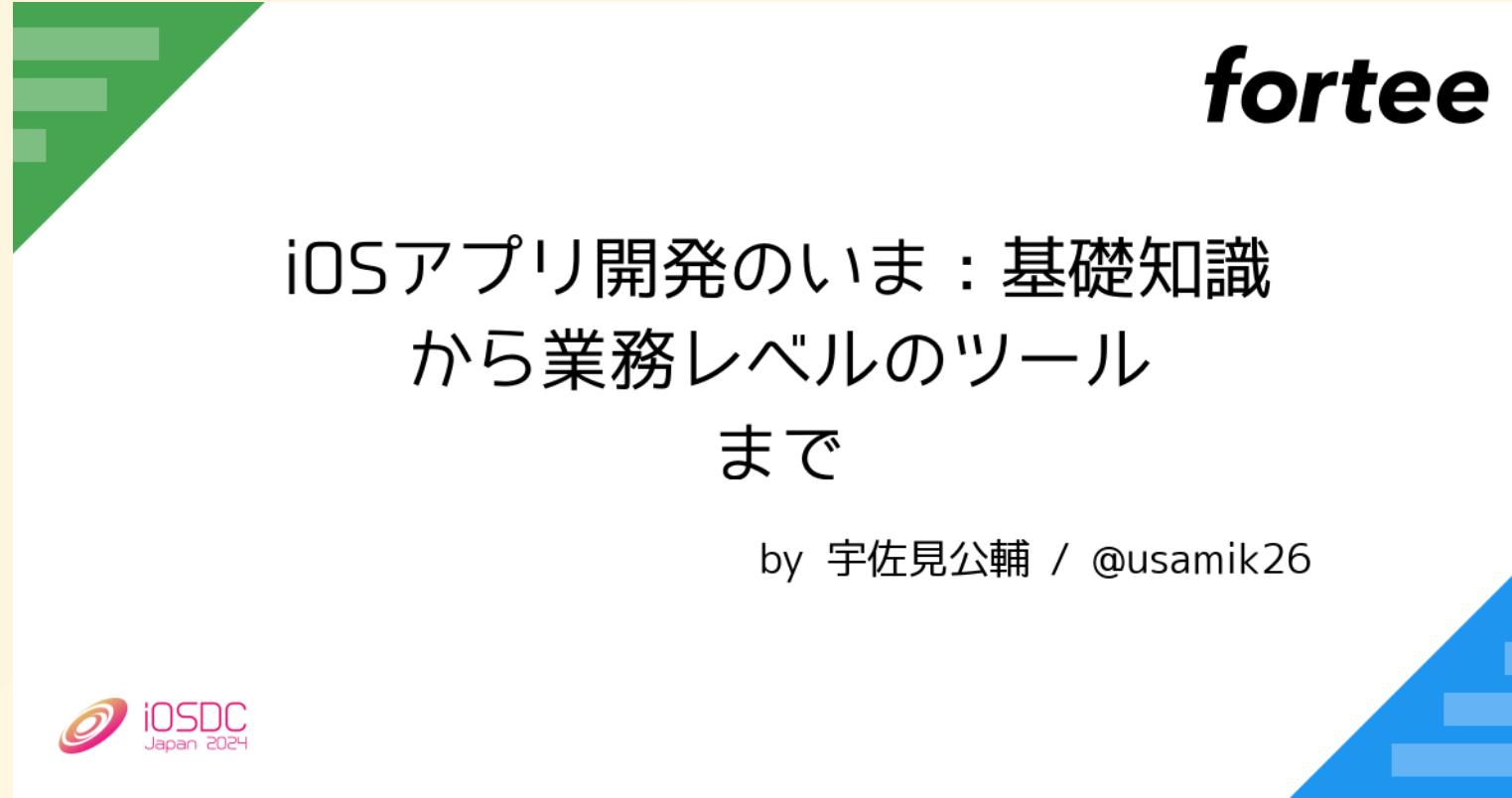
近況(1)

おもしろ同人誌バザール、技書博、技術書典に、技術同人誌を個人で出展しました



近況(2)

iOSDC Japan 2024
パンフレット原稿採択
されました（執筆中）
トークも応募する予定



fortee

iOSアプリ開発のいま : 基礎知識
から業務レベルのツール
まで

by 宇佐見公輔 / @usamik26



Accelerateフレームワークとは

Accelerateフレームワークとは

- Appleが提供する数値計算ライブラリ群
- CPUのベクタ演算機能を活用
- iOS 4 / macOS 10.3 / tvOS 9 / watchOS 2 / visionOS 1

<https://developer.apple.com/jp/accelerate/>

<https://developer.apple.com/documentation/accelerate/>

Accelerate vs Metal

- CPUのベクタ演算機能を活用 : Accelerate
- GPUのベクタ演算機能を活用 : Metal
- 両者は機能が重なっている
- 実のところ、GPUが使えるならMetalのほうが効率が良い

<https://developer.apple.com/jp/metal/>

<https://developer.apple.com/documentation/metal/>

Accelerateフレームワークの機能

- 機械学習 : BNNS
- 画像処理 : vImage
- デジタル信号処理 : vDSP
- ベクトルに対する関数演算 : vForce
- 線形代数 : BLAS / LAPACK / Sparse Solvers

BNNS

SwiftのAccelerateフレームワーク

q

BNNS

- BNNS=Basic Neural Network Subroutines
- CPU上でニューラルネットワークの演算を高速に実行

BNNSよりも以下を使うほうが良い。

- Core ML
 - 内部でMetalやAccelerateを使っている
- Metal Performance Shaders Graph (MPS Graph)
 - GPU上でニューラルネットワークの演算を高速に実行

vImage

vImage

- CPU上で画像処理を高速に実行

vImageよりも以下を使うほうが良い。

- Core Image
 - 内部でMetalやAccelerateを使っている
- Metal
 - GPU上で画像処理を高速に実行

vDSP

vDSP

- DSP=Digital Signal Processing
- 主に音声処理で使われる
- 高速フーリエ変換 (FFT)
- デジタルフィルタリング
- その他、ベクトルに対する算術演算 (加減乗除)

音声関連

- 再生・録音
 - Media Player フレームワーク
 - AVFoundation フレームワーク
- 音声処理
 - Core Audio : デバイス入出力
 - vDSP : 信号処理

vDSPの事例

同僚が「iPhoneで楽器チューナを作つてみる」という記事を書いています。

「ゆめみ大技林 '22」は無償で頒布中です。

「'23」「'23 (2)」「'24」も頒布中。



vForce

vForce

ベクトルに対する関数演算

```
let num = 1_000_000
let input = (0..<num).map { Float($0) }
```

```
// vForceを使っていない計算
let output = input.map { sqrt($0) }
```

```
// vForceを使った計算
let output = vForce.sqrt(input)
```

vForceの書きかた2種類

UnsafePointer を使う書きかた (C言語由来)

```
var num = Int32(1_000_000)
var input = (0..<num).map { Float($0) }
var output = [Float](repeating: 0, count: input.count)
vvsqrdf(&output, &input, &num)
```

UnsafePointer を使わない書きかた

```
let output = vForce.sqrt(input)
```

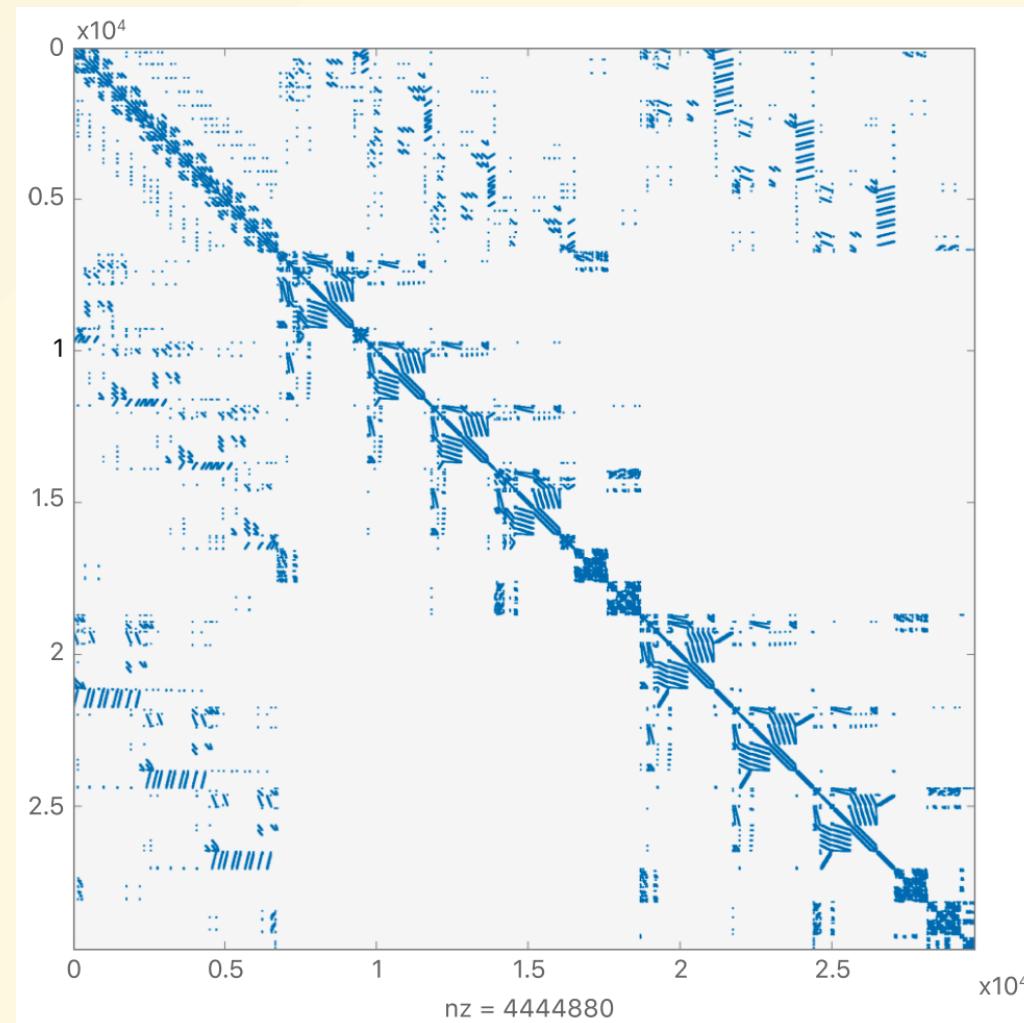
BLAS / LAPACK / Sparse Solvers

BLAS / LAPACK

- BLAS=Basic Linear Algebra Subroutines
 - 線形代数の数値演算ライブラリのAPI仕様
- LAPACK=Linear Algebra PACKage
 - BLASの上位パッケージ
- BLASやLAPACKのAPIを実装したライブラリが多く存在している
 - Accelerateもそういった実装のひとつ

Sparse Solvers

- 線形代数の数値演算で、疎行列の扱いに特化したライブラリ
 - 疎行列 = 多くの成分が0の行列
- 行列のサイズが大きくとも効率的に計算できる
- 扱う行列が疎行列かどうかは開発者が判断して、ライブラリを選択する



Accelerateのまとめ・活用箇所

Accelerateのまとめ

- ベクトルや行列に対する数値演算に適している
 - デジタル信号処理 : vDSP
 - ベクトルに対する関数演算 : vForce
 - 線形代数 : BLAS / LAPACK / Sparse Solvers
- 機械学習や画像処理ができるが最適解ではない
 - それらはCore ML、Core Image、Metalが良い

Accelerateの活用箇所

Apple製のライブラリで以下のものにAccelerateが活用されている

- Apple Archive (ファイル圧縮)
- Spatial (軽量な3D数学ライブラリ)
- Quadrature (定積分の近似計算)