



Marcel Weiher

iOS and macOS[™] Performance Tuning

Cocoa®, Cocoa Touch®, Objective-C®, and Swift[™]





Opened in November 2010, the **Sheikh Zayed Bridge** offers a stunning entryway to the city of Abu Dhabi, United Arab Emirates. Designed by the visionary architect Zaha Hadid, the bridge appears to onlookers as a series of rising and falling concrete waves, reminiscent of the region's nearby sand dunes. Hadid's "waves," reaching 64 meters at their peak, appear to propel themselves toward the city. Her design's extraordinary energy is reinforced by dynamic nighttime lighting, making the bridge an unforgettable local landmark.

iOS and macOS Performance Tuning: Cocoa, Cocoa Touch, Objective-C, and Swift

Table of Contents

Cover

Title Page

Copyright Page

Contents

About the Author

Introduction

1 CPU: Principles

 A Simple Example

 The Perils of(Micro-)Benchmarking

 More Integer Summing

 Swift

 Other Languages

 The Power of Hybrids

 Trends

 Cost of Operations

 Computational Complexity

 Summary

2 CPU: Measurement and Tools

 Command-Line Tools

 top

Table of Contents

time

sample

Xcode Gauges

Instruments

Setup and Data Gathering

Profiling Options

Basic Analysis

Source Code

Data Mining I:Focus

Data Mining II:Pruning

Internal Measurement

Testing

Dtrace

Optimization Beyondthe Call of Duty

Summary

3 CPU: Pitfalls and Techniques

Representation

Primitive Types

Strings

Objects

Accessors

Public Access

Object Creation and Caching

Mutability and Caching

Lazy Evaluation

Caching Caveats

Pitfall:Generic (Intermediate) Representations

Arrays and Bulk Processing

Table of Contents

Dictionaries

Messaging

IMP Caching

Forwarding

Uniformity and Optimization

Methods

Pitfall:CoreFoundation

Multicore

Threads

Work Queues

Mature Optimization

4 CPU Example: XML Parsing

An HTML Scanner

Mapping Callbacks to Messages

Objects

Objects,Cheaply

Evaluation

Tune-Ups

Optimizing the Whole Widget:MAX

MAX Implementation

Summary

5 Memory: Principles

The Memory Hierarchy

Mach Virtual Memory

Heap and Stack

Stack Allocation

Heap Allocation with malloc()

Table of Contents

Resource Management

- Garbage Collection

- Foundation Object Ownership

- Tracing GC

- Automatic Reference Counting

- Process-Level Resource Reclamation

Summary

6 Memory: Measurement and Tools

- Xcode Gauges

- Command-Line Tools

 - top

 - heap

 - leaks and malloc_debug

- Internal Measurement

- Memory Instruments

 - Leaks Instrument

 - Allocations

 - VM Tracker

 - Counters/PMEvents

Summary

7 Memory: Pitfalls and Techniques

- Reference Counting

 - Avoiding Leaks

- Foundation Objects vs. Primitives

- Smaller Structures

 - But What About Y2K?

 - Compression

Table of Contents

Purgeable Memory

Memory and Concurrency

Architectural Considerations

Temporary Allocations and Object Caching

NSCache and libcache

Memory-Mapped Files

madvise

iOS Considerations

Optimizing ARC

Summary

8 Memory Example: FilterStreams

Unix Pipes and Filters

Object-Oriented Filters

DescriptionStream

Eliminating the Infinite Recursion from description

Stream Hierarchy

Summary

9 Swift

Swift Performance:Claims

Language Characteristics

Benchmarks

Assessing Swift Performance

Basic Performance Characteristics

Collections

Larger Examples

nginx HTTP Parser

FreddyJSONParser

Table of Contents

Image Processing

Observations

Compile Times

Type Inference

Generics Specialization

Whole-Module Optimization

Controlling Compile Times

Optimizer-Oriented Programming

A Sufficiently Smart Compiler

The Death of Optimizing Compilers

Practical Advice

Alternatives

Summary

10 I/O: Principles

Hardware

Disk Drives

Solid-State Disks

Network

Operating System

Abstraction:Byte Streams

FileI/O

The Network Stack

Summary

11 I/O: Measurement and Tools

Negative Space:top and time

Summary Information:iostat and netstat

Instruments

Table of Contents

Detailed Tracing:fs_usage

Summary

12 I/O: Pitfalls and Techniques

Pushing Bytes with NSData

 A Memory-Mapping Anomaly

 How Chunky?

Unixy I/O

Network I/O

 Overlapping Transfers

 Throttling Requests

 Data Handling

 Asynchronous I/O

 HTTP Serving

Serialization

 Memory Dumps

 A Simple XML Format

 Property Lists

 Archiving

 Serialization Summary

CoreData

 Create and Update in Batches

 Fetch and Fault Techniques

 Object Interaction

 Sub setting

 Analysis

SQLite

 Relational and Other Databases

Event Posting

Table of Contents

Hybrid Forms

Segregated Stores

Summary

13 I/O: Examples

iPhone Game Dictionary

Fun with Property Lists

A Binary Property List Reader

Lazy Reading

Avoiding Intermediate Representations

Comma-Separated Values

Public Transport Schedule Data

Stops

Stop Time Lookup

Stop Time Import

Faster CSV Parsing

Object Allocation

Push vs. Pull

Keys of Interest

Parallelization

Summary

14 Graphics and UI: Principles

Responsiveness

Software and APIs

Quartz and the PostScript Imaging Model

OpenGL

Metal

Graphics Hardware and Acceleration

Table of Contents

From Quartz Extreme to Core Animation

Summary

15 Graphics and UI: Measurement and Tools

CPU Profiling with Instruments

Quartz Debug

Core Animation Instrument

When the CPUs Not theP roblem

What Am I Measuring?

Summary

16 Graphics and UI: Pitfalls and Techniques

Pitfalls

Techniques

Too Much Communication Slows Down Installation

The Display Throttle

Working with the Display Throttle

Installers and Progress Reporting Today

Overwhelming an iPhone

Its Just an Illusion

Image Scaling and Cropping

Thumbnail Drawing

How Definitely Not to Draw Thumbnails

How to Not Really Draw Thumbnails

How to Draw Non-Thumbnails

Line Drawing on iPhone

Summary

17 Graphics and UI: Examples

Beautiful Weather App

Table of Contents

An Update

Fun with PNG

Brainstorming

Data Points to JPEG

A Measuring Hiccup

JPNG and JPJP

A Beautiful Launch

Wunderlist3

Wunderlist2

Overall Architecture

URLs and In-Process REST

An Eventually Consistent Asynchronous Data Store

RESTOperation Queues

A Smooth and Responsive UI

Wunderlist in Short

Summary

Index