Achieving All-Day Battery Life

Knowing is half the battle

Session 707

Jon Andrews Core OS
Soren Spies Core OS
What Users Expect

Computing Energy

OS Energy Improvements

Developers’ Role

Your Software (Soren)
What Users Expect
All-day battery life

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 a.m.</td>
<td>commute, news</td>
</tr>
<tr>
<td>lunch</td>
<td>email</td>
</tr>
<tr>
<td>break</td>
<td></td>
</tr>
<tr>
<td>dinner</td>
<td>commute, game</td>
</tr>
<tr>
<td>11 p.m.</td>
<td></td>
</tr>
</tbody>
</table>
What Users Expect

All-day battery life

7 a.m. 11 p.m. dinner

low power ~ all day

Background User
What Users Expect
Low power ~ all-day battery life

low power

7 a.m. 11 p.m.
Computing Energy

Fundamentals

Energy = power \times \text{time}

Less power = more time

Battery Life

<table>
<thead>
<tr>
<th>Mode</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>50</td>
</tr>
<tr>
<td>3D Game</td>
<td>100</td>
</tr>
<tr>
<td>Active</td>
<td>150</td>
</tr>
<tr>
<td>Idle (Screen On)</td>
<td>200</td>
</tr>
<tr>
<td>Audio</td>
<td>250</td>
</tr>
<tr>
<td>Standby</td>
<td></td>
</tr>
</tbody>
</table>

Hours

Battery Life
Computing Energy Fundamentals

Energy = power × time

Make your app more like idle

Battery Life

- Max
- 3D Game
- Active
- Idle (Screen On)
- Audio

Hours

- 10
- 20
- 30
- 40
- 50

not all day
all day
Computing Energy Trends

Energy = power × time

Dynamic range growing
- iOS devices getting closer to MacBook

New products more power-efficient
- Yet peak power increases

Battery Life
- Max: not all day
- 3D Game: not all day
- Active: all day
- Idle (Screen On): all day
- Audio: all day

Hours

10 20 30 40 50
Computing Energy

Fixed and dynamic costs

Energy = Power × Time

App Launch

Fixed

Dynamic

App Activities

App Backgrounded

App Suspended
Computing Energy

Standby

- Active
- Inactive
- Idle
- Sleep

- CPU
- GPU
- RAM

- Cellular
- WiFi

- Audio
- Location

- Display
Computing Energy

User idle

- Active
- Inactive
- Idle
- Sleep

- CPU
- GPU
- RAM
- Audio
- Location
- Cellular
- WiFi
- Display
Battery Drain Rate

User idle

Power vs. Time

low power = all day
Battery Drain Rate

User idle

Power vs. Time

low power = all day
Computing Energy

3D game

- Active
- Inactive
- Idle
- Sleep

- Cellular
- WiFi
- CPU
- GPU
- RAM
- Audio
- Location

Display
Battery Drain Rate

3D game

Power vs. Time

low power ~ all day
Battery Drain Rate

3D game

Power vs. Time

low power ~ all day
Energy Optimization Strategy

- Do work less
- Do work later
- Do work efficiently
iOS 9 Improvements

iOS battery life best-in-class
• … our customers want better

iOS 9 is better on existing hardware
• Up to one hour better battery life
• iOS and system apps energy-optimized
Like OS X Mavericks

Eliminated polling in apps, frameworks, drivers, kernel
Quality of service (QoS) for CPU and I/O
Timer coalescing and rate limiting
More efficient CPU power management
“Significant Energy” in battery menu
iOS 9 Improvements

Focused on iPhone
Improve all iOS products
iOS 9 Improvements

Do less work

Optimized sleep timers

Idle power

• Reduce CPU wake-ups

Face-down detection

• Don’t light the screen on notifications
iOS 9 Improvements
Do work later

Defer more work until plugged in
Defer some networking until WiFi
Leverage persistent connection API
iOS 9 Improvements

Do work efficiently

Optimized iOS networking stack for LTE
Optimized power management
Reduced cost of logging
More numerics optimizations
iOS 9 Improvements

User feedback

Per-app battery usage

Environmental factors

Intelligent suggestions
iOS 9 Improvements

User feedback

Per-app battery usage
Environmental factors
Intelligent suggestions

New: Per-app screen and background time
iOS 9 Improvements

User feedback

User-initiated Low Power Mode

- Limit CPU performance
- No Background App Refresh
- No discretionary/background downloads
- No mail fetch
iOS 9 Improvements

Developer feedback

iOS energy gauge in Xcode
Location instrument
Developers’ Role

Example: Video playback

Full-screen video optimization
- Reduce backlight
- Adjust gamma to compensate

Overlay UI disables

Power delta is small, but videos often long
Developers’ Role

OS X

MacBook
• Thermally constrained (no fan)
• Works best with bursts
• QoS is critical for responsiveness

Optimize
• Existing OS X gauge & instruments
• Prioritize work with NSOperation/GCD
Developers’ Role

Understanding

Energy = power × time

Hardware has a large dynamic range

Low power for all-day battery life
Reducing Software Energy
Your code vs. users’ batteries

Soren Spies, Core OS
The Battery Is for the User

CPU and GPU Energy

Achieving Low (Average) Power for All-Day Battery Life

iOS Energy Consumption

Energy-Aware Development
The Battery Is for the User

Strategy

Do less work, less often

• Eliminate polling, timers
• Respond to user, then absolute idle
The Battery Is for the User

Strategy

Do less work, less often
- Eliminate polling, timers
- Respond to user, then absolute idle

Do work later
- Does the user need this NOW?
The Battery Is for the User

Strategy

Do less work, less often
• Eliminate polling, timers
• Respond to user, then absolute idle

Do work later
• Does the user need this NOW?

Do work efficiently
• Batch work into user-driven bursts
• Optimize power and time ~ lower energy
The Battery Is for the User

Strategy

Do less work, less often
- Eliminate polling, timers
- Respond to user, then absolute idle

Do work later
- Does the user need this NOW?

Do work efficiently
- Batch work into user-driven bursts
- Optimize power and time ~ lower energy

All-day battery life assumes ~10% load
The Battery Is for the User
User should control energy use

Power vs. Time

~5% Power = display on
The Battery Is for the User
User should control energy use

Power vs. Time

~10% Power = all day

~5% Power = display on
CPU and GPU Energy

Biggest consumers on OS X

Power vs. Time

- CPU (Mac)
- GPU (Mac)
- Display

low power = all day
CPU and GPU Energy

Biggest consumers on OS X

Power vs. Time

- >10% average
- ~ partial day
- low power = all day

- Orange: CPU (Mac)
- Yellow: GPU (Mac)
- Green: Display
CPU and GPU Energy
Wake-ups are expensive

Power vs. Time

low power = all day
CPU and GPU Energy
Faster = less energy

Power vs. Time

naive
Faster = less energy

CPU and GPU Energy

Power vs. Time

efficient

naive
CPU and GPU Energy
Faster = less energy

Power vs. Time

parallel  efficient  naive
CPU and GPU Energy

Faster = less energy

---

Power vs. Time

- **OpenCL/GPU**
- **parallel**
- **efficient**
- **naive**
CPU and GPU Energy

Faster = less energy

Power vs. Time

OpenCL/GPU

parallel

efficient

naive

lowest power = highest cost
CPU and GPU Energy

Why?

Power vs. Time

naive
CPU and GPU Energy

Less time with fixed costs

Power vs. Time

naive

fixed costs
CPU and GPU Energy

Less time with fixed costs

Power vs. Time

OpenCL/GPU
- parallel
- efficient
- naive

fixed costs
CPU and GPU Energy

Average power

naive
CPU and GPU Energy

Average power

naive
CPU and GPU Energy

Average power

Is it low? No!

low power = all day

naive
CPU and GPU Energy

Average power

Is it low?

OpenCL/GPU

naive

efficient

parallel

low power= all day
iOS Energy Consumption

Low-hanging fruit

Networking
• Beware high, fixed costs

Location
• Don’t leave it leaking

Background operation
• Don’t delay sleep
iOS Networking
High, fixed costs

Power vs. Time

overhead

low power = all day

overhead
iOS Networking

High, fixed costs

Power vs. Time

- Low power = all day
- Average > low = partial day

Overhead
iOS Networking

Batch work to minimize overhead

Power vs. Time

coalesce?

overhead

low power = all day

overhead

Batch work to minimize overhead
iOS Networking

Batch work to minimize overhead

Power vs. Time

coalesced!

overhead

low power = all day
iOS Networking

Batch work to minimize overhead

Power vs. Time

coalesced!

overhead

average < low

= all-day!

low power = all day

low power = all day
Use network less

• Design it down or out
• Once a minute is a lot
iOS Networking

Optimization

Use network less
- Design it down or out
- Once a minute is a lot

Use network later
- Can it wait?
- Background update, URLSession
iOS Networking
Optimization

Use network less
• Design it down or out
• Once a minute is a lot

Use network later
• Can it wait?
• Background update, NSURLSession

Use network efficiently
• Batch work
• Use notifications... sparingly
iOS Location
Precision ~ power

Power vs. Time

- Precise
- Imprecise

- Low power = all day

- Background
- Off

- iOS Location
- Display
iOS Location

Optimization

Use location less

• Don’t call `startUpdatingLocation()` until needed
• Call `stopUpdatingLocation()` as soon as possible
iOS Location

Optimization

Use location less

- Don’t call `startUpdatingLocation()` until needed
- Call `stopUpdatingLocation()` as soon as possible
- For single-fix, iOS 9 introduce `requestLocation()`
- `locManager.allowsBackgroundLocationUpdates = false`
Use location less

- Don’t call `startUpdatingLocation()` until needed
- Call `stopUpdatingLocation()` as soon as possible
- For single-fix, iOS 9 introduce `requestLocation()`
- `locManager.allowsBackgroundLocationUpdates = false`

Use location efficiently

- Lower accuracy = lower power
- `allowDeferredLocationUpdatesUntilTraveled:timeout:`
iOS Background Operation

Cut it out!

Power vs. Time

unseen
notification

low power = all day

Energy not delivering user benefit.

low power = all day

Energy not delivering user benefit.
iOS Background Operation

Optimization

`startBackgroundTask()` keeps device awake
- Only start for non-trivial user work

Call `endBackgroundTask()`
- As soon as possible
- In all cases

Delegate indeterminate networking to OS
The Ideal App
User-driven, fast, and idle

Power vs. Time

launch idle! user idle!

low power = all day
How Is My App Doing?

Xcode now highlights

• CPU
• Networking
• Location
• Background

Instruments support

Come to our next talk
Think Energy
Everyday workflow

Design

• Plan to work less/later/efficiently
• User expectation given app function
Think Energy

Everyday workflow

Design

• Plan to work less/later/efficiently
• User expectation given app function

Implementation

• Yours, Apple’s, third-party code
Think Energy

Everyday workflow

Design
  • Plan to work less/later/efficiently
  • User expectation given app function

Implementation
  • Yours, Apple’s, third-party code

Test
  • Verify correct behavior
Additional Optimizations
Doing more with less

Careful background updates

Notifications – use PushKit, including VOIP

Display brightness – leave to user

Drawing – 2014 talk

Energy Guide!
Energy = Power × Time
Low Power ~ All-day Battery
Do Less Work
Do It Later
Do It Efficiently
More Information

Documentation and Videos

iOS Energy Guide

OS X Energy Guide

Writing Energy Efficient Code, Parts 1 & 2 (2014)
http://developer.apple.com/videos
More Information

Technical Support
Apple Developer Forums
http://developer.apple.com/forums

Developer Technical Support
http://developer.apple.com/support/technical

General Inquiries
Paul Danbold, Core OS Evangelist
danbold@apple.com
<table>
<thead>
<tr>
<th>Related Sessions</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debugging Energy Issues</td>
<td>Nob Hill</td>
<td>Wednesday 10:00AM</td>
</tr>
<tr>
<td>Networking with NSURLSession</td>
<td>Pacific Heights</td>
<td>Thursday 9:00AM</td>
</tr>
<tr>
<td>Low Energy, High Performance: Compression and Accelerate</td>
<td>Nob Hill</td>
<td>Thursday 10:00AM</td>
</tr>
<tr>
<td>What's New in Core Location</td>
<td>Pacific Heights</td>
<td>Thursday 1:30PM</td>
</tr>
<tr>
<td>Advanced NSOperations</td>
<td>Presidio</td>
<td>Friday 9:00AM</td>
</tr>
<tr>
<td>Building Responsive and Efficient Apps with GCD</td>
<td>Nob Hill</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>Performance on iOS and watchOS</td>
<td>Presidio</td>
<td>Friday 11:00AM</td>
</tr>
</tbody>
</table>
## Related Labs

We’re here for you!

<table>
<thead>
<tr>
<th>Lab</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power and Performance Lab</td>
<td>Frameworks Lab B</td>
<td>Wednesday 1:30PM</td>
</tr>
<tr>
<td>Networking Lab</td>
<td>Frameworks Lab E</td>
<td>Thursday 10:00AM</td>
</tr>
<tr>
<td>Core Location Lab</td>
<td>Frameworks Lab A</td>
<td>Thursday 2:30PM</td>
</tr>
<tr>
<td>Power and Performance Lab</td>
<td>Frameworks Lab C</td>
<td>Friday 12:00PM</td>
</tr>
<tr>
<td>Networking Lab</td>
<td>Frameworks Lab B</td>
<td>Friday 1:30PM</td>
</tr>
</tbody>
</table>