Power Management

Energy efficient software

Session 711

Ethan Bold
I/O Kit Team

Soren Spies
I/O Kit Team

Power Assertions
DarkWake
Debugging

Power Assertions

DarkWake

Debugging

What Is a Power Assertion?

- They allow user-requested work to complete
- Assertions can prevent idle sleep
- Assertions can prevent idle display sleep
- It’s a hint to OS X

When Not to Use Assertions

- Your work is resumable
- Some frameworks take assertions, so you don't need to
  - NSURLDownload
- User activity prevents idle sleep (mouse, keyboard)
- Remote connections prevent idle sleep (ssh, AFP, SMB)
- Capturing the display prevents idle display sleep

Keeps the System Awake

kIOPMAssertionTypePreventUserIdleSystemSleep

Display is on

Is awake

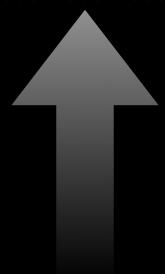
Is asleep

Keeps the System Awake

kIOPMAssertionTypePreventUserIdleSystemSleep

Display is on

Is awake



Is asleep

Keeps the Display Awake

kIOPMAssertionTypePreventUserIdleDisplaySleep

Display is on

Is awake

Is asleep

Keeps the Display Awake

kIOPMAssertionTypePreventUserIdleDisplaySleep

Display is on

Is awake

Is asleep

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

Take an Assertion

```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```

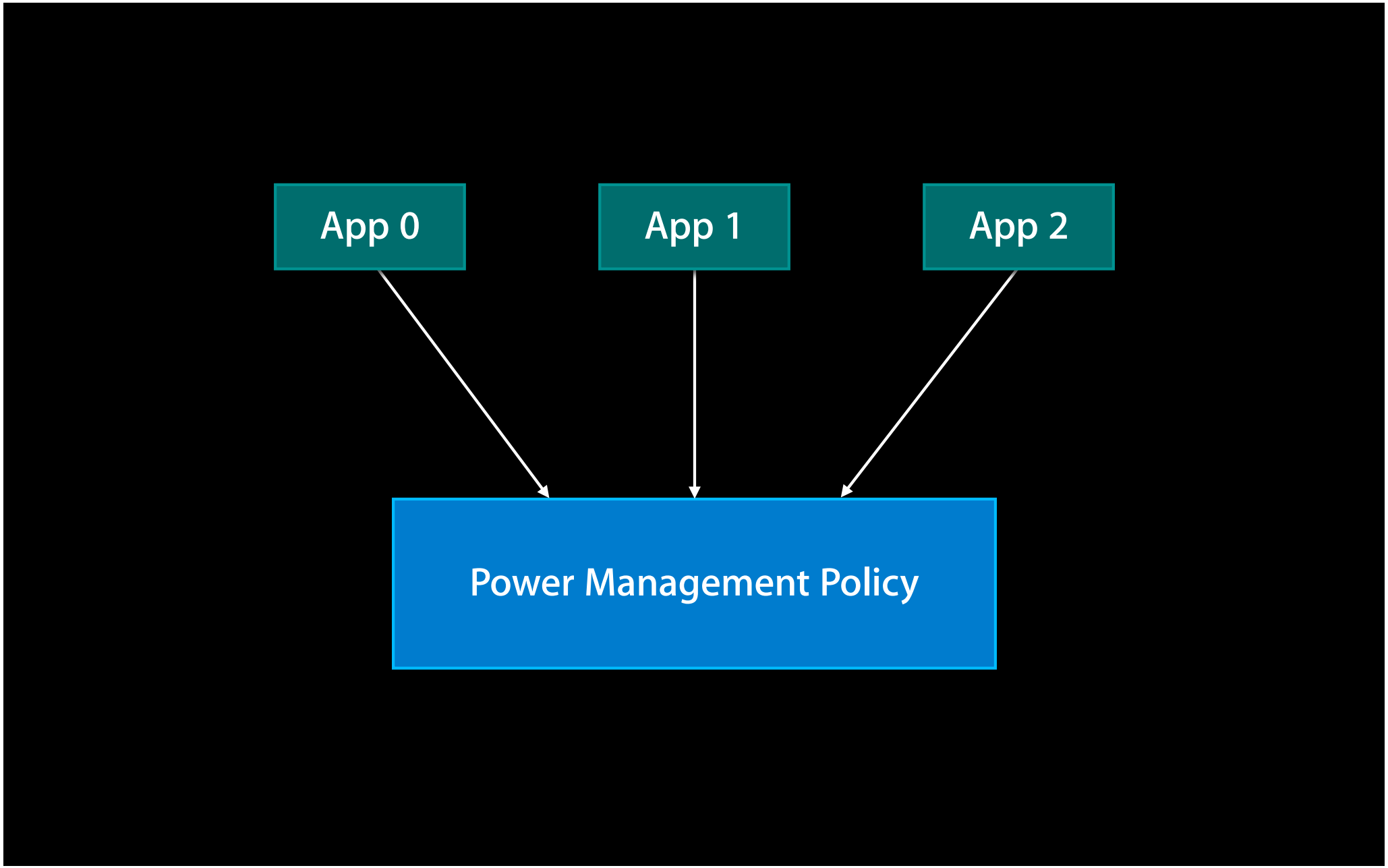
Take an Assertion

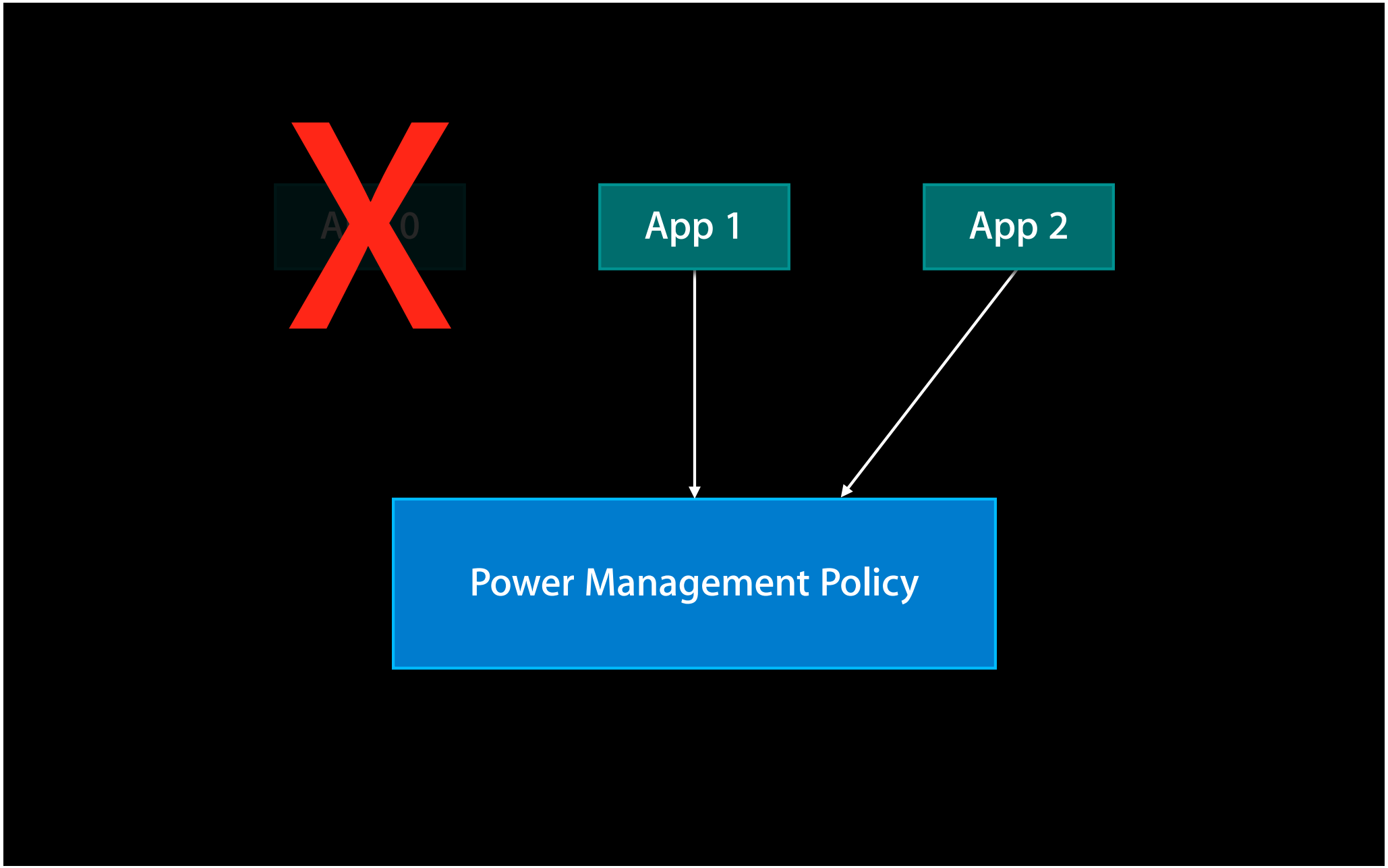
```
#include <IOKit/pwr_mgt/IOPMLib.h>

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

IOPMAssertionCreateWithName(
  kIOPMAssertionTypePreventUserIdleSystemSleep,
  kIOPMAssertionLevelOn,
  CFSTR(“Processing Giant Files”),
  &newAssertion);

IOPMAssertionRelease(newAssertion);
```


App 0
App 1
App 2
Power Management Policy


App 1
App 2
Power Management Policy

Disk I/O and Idle Sleep

- Power assertions prevent idle sleep
- Disk I/O doesn't

What's Going On

```
% pmset -g assertions
Assertion status system-wide:
   PreventUserIdleDisplaySleep    0
   PreventSystemSleep             0
   PreventUserIdleSystemSleep     1
   ExternalMedia                  0
   UserIsActive                   0
   ApplePushServiceTask           0
   BackgroundTask                 0

  pid 29: [0xc0000012f] PreventUserIdleSystemSleep named:"com.apple.metadata.mds"
```

What's Going On

```
% pmset -g assertions
Assertion status system-wide:
   PreventUserIdleDisplaySleep    0
   PreventSystemSleep             0
   PreventUserIdleSystemSleep     1
   ExternalMedia                  0
   UserIsActive                   0
   ApplePushServiceTask           0
   BackgroundTask                 0

  pid 29: [0xc0000012f] PreventUserIdleSystemSleep named:"com.apple.metadata.mds"
```

What's Going On

```
% pmset -g assertions
Assertion status system-wide:
   PreventUserIdleDisplaySleep    0
   PreventSystemSleep             0
   PreventUserIdleSystemSleep     1
   ExternalMedia                  0
   UserIsActive                   0
   ApplePushServiceTask           0
   BackgroundTask                 0

  pid 29: [0xc0000012f] PreventUserIdleSystemSleep named:"com.apple.metadata.mds"
```

What's Going On

```
% pmset -g assertions
Assertion status system-wide:
   PreventUserIdleDisplaySleep    0
   PreventSystemSleep             0
   PreventUserIdleSystemSleep     1
   ExternalMedia                  0
   UserIsActive                   0
   ApplePushServiceTask           0
   BackgroundTask                 0

  pid 29: [0xc0000012f] PreventUserIdleSystemSleep named:"com.apple.metadata.mds"
```

Command Line Power Assertions

```
% caffeinate make
```

IOCancelPowerChange

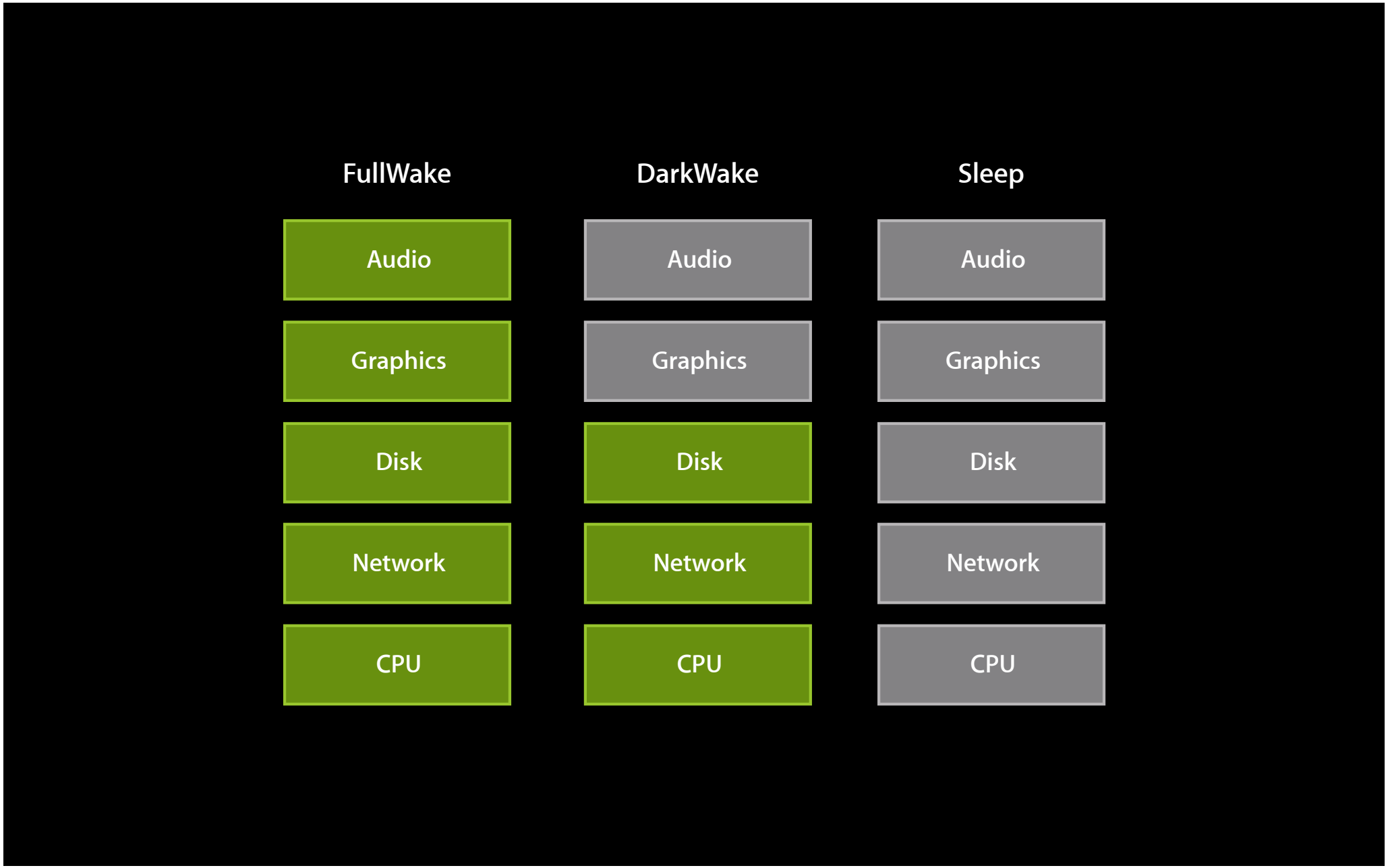
If you use IOCancelPowerChange()

Migrate to IOPMAssertionCreateWithName()

Power Assertions

DarkWake

Debugging


FullWake
DarkWake
Sleep
Audio
Graphics
Disk
Network
CPU
Audio
Graphics
Disk
Network
CPU
Audio
Graphics
Disk
Network
CPU

What Is DarkWake?

- Apple only—Not available to developers
  - OS X suppresses notifications
- Don’t write code for DarkWake
  - Gracefully handle unavailable network, audio, and graphics

DarkWake Timeline

10.6

10.7

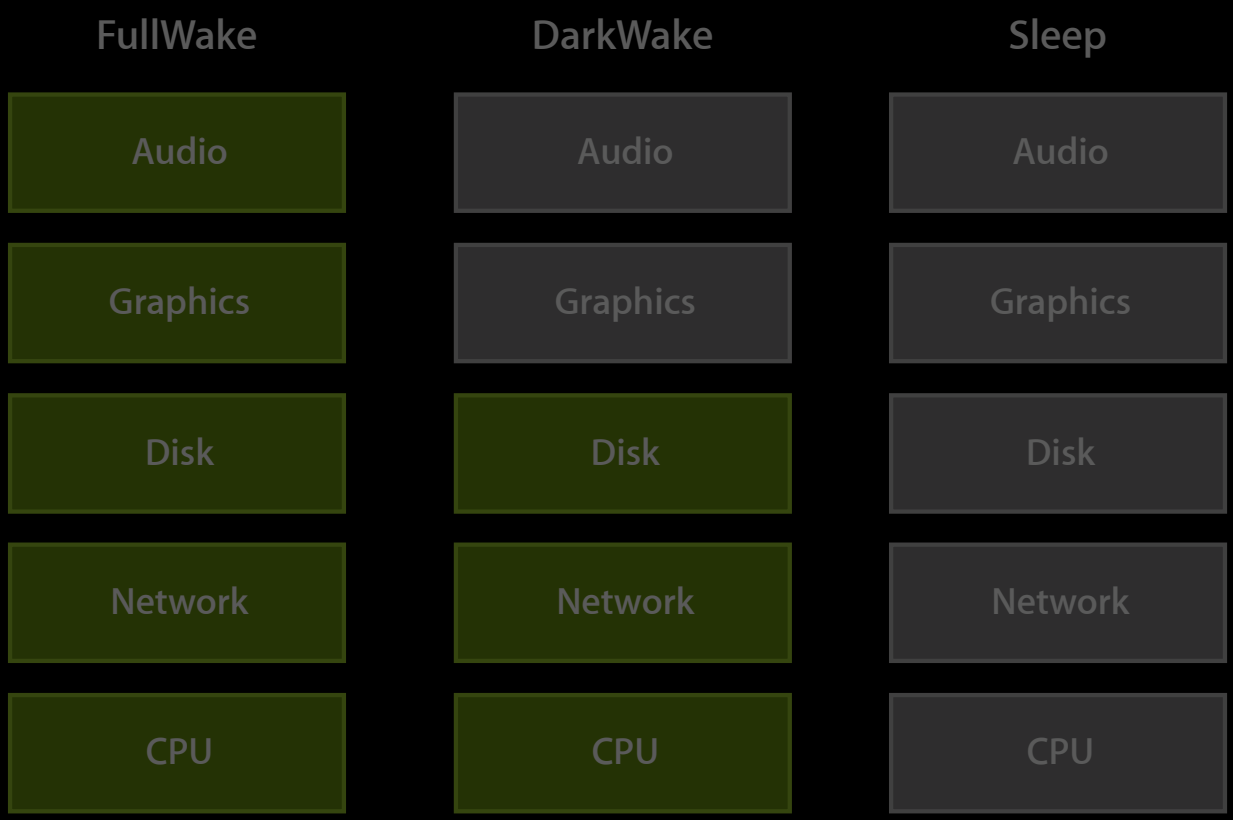


10.8




(Supported Hardware)

DarkWake


FullWake
DarkWake
Sleep
Audio
Audio
Audio
Graphics
Graphics
Graphics
Disk
Disk
Disk
Network
Network
Network
CPU
CPU
CPU

Power Assertions
DarkWake
Debugging

Sleep and Wake

```
% pmset -g log

5/16/12 10:57 PM Sleep             Clamshell Sleep Sleep: Using AC (Charge:11%)          62416 secs
5/16/12 10:58 PM WakeRequests      Clients requested wake events: None
5/17/12 4:18 PM  DarkWake          DarkWake due to EHC1: Using AC (Charge:95%)           19 secs
5/17/12 4:18 PM  Sleep             Clamshell Sleep Sleep: Using AC (Charge:95%)          94 secs
5/17/12 4:20 PM  Wake              Wake due to EC LID0: Using AC (Charge:95%)            2662 secs
```

Sleep and Wake

```
% pmset -g log
```

```
5/16/12 10:57 PM Sleep              Clamshell Sleep Sleep: Using AC (Charge:11%)          62416 secs
5/16/12 10:58 PM WakeRequests       Clients requested wake events: None
5/17/12 4:18 PM  DarkWake           DarkWake due to EHC1: Using AC (Charge:95%)           19 secs
5/17/12 4:18 PM  Sleep              Clamshell Sleep Sleep: Using AC (Charge:95%)          94 secs
5/17/12 4:20 PM  Wake               Wake due to EC LID0: Using AC (Charge:95%)            2662 secs
```

Sleep and Wake

```
% pmset -g log

5/16/12 10:57 PM Sleep             Clamshell Sleep Sleep: Using AC (Charge:11%)          62416 secs
5/16/12 10:58 PM WakeRequests      Clients requested wake events: None
5/17/12 4:18 PM  DarkWake          DarkWake due to EHC1: Using AC (Charge:95%)           19 secs
5/17/12 4:18 PM  Sleep             Clamshell Sleep Sleep: Using AC (Charge:95%)          94 secs
5/17/12 4:20 PM  Wake              Wake due to EC LID0: Using AC (Charge:95%)            2662 secs
```

Sleep and Wake

```
% pmset -g log

5/16/12 10:57 PM Sleep             Clamshell Sleep Sleep: Using AC (Charge:11%)          62416 secs
5/16/12 10:58 PM WakeRequests      Clients requested wake events: None
5/17/12 4:18 PM  DarkWake          DarkWake due to EHC1: Using AC (Charge:95%)           19 secs
5/17/12 4:18 PM  Sleep             Clamshell Sleep Sleep: Using AC (Charge:95%)          94 secs
5/17/12 4:20 PM  Wake              Wake due to EC LID0: Using AC (Charge:95%)            2662 secs
```

Sleep and Wake

```
% pmset -g log

5/16/12 10:57 PM Sleep             Clamshell Sleep Sleep: Using AC (Charge:11%)          62416 secs
5/16/12 10:58 PM WakeRequests      Clients requested wake events: None
5/17/12 4:18 PM  DarkWake          DarkWake due to EHC1: Using AC (Charge:95%)           19 secs
5/17/12 4:18 PM  Sleep             Clamshell Sleep Sleep: Using AC (Charge:95%)          94 secs
5/17/12 4:20 PM  Wake              Wake due to EC LID0: Using AC (Charge:95%)            2662 secs
```

Sleep and Wake

```
% pmset -g log

5/16/12 10:57 PM Sleep             Clamshell Sleep Sleep: Using AC (Charge:11%)          62416 secs
5/16/12 10:58 PM WakeRequests      Clients requested wake events: None
5/17/12 4:18 PM  DarkWake          DarkWake due to EHC1: Using AC (Charge:95%)           19 secs
5/17/12 4:18 PM  Sleep             Clamshell Sleep Sleep: Using AC (Charge:95%)          94 secs
5/17/12 4:20 PM  Wake              Wake due to EC LID0: Using AC (Charge:95%)            2662 secs
```

Assertions

```
% pmset -g log

6/5/12 4:22:29 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Created PreventUserIdleSystemSleep "AirDrop"
6/5/12 4:22:41 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Released PreventUserIdleSystemSleep "AirDrop"
6/6/12 8:51:54 AM PDT    Assertions           PID 5681(AddressBookSour)
    Created PreventUserIdleSystemSleep "Address Book Source Sync"
6/6/12 8:52:16 AM PDT    Assertions           PID 5681(AddressBookSour)
    Released PreventUserIdleSystemSleep "Address Book Source Sync"
```

Assertions

```
% pmset -g log

6/5/12 4:22:29 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Created PreventUserIdleSystemSleep "AirDrop"
6/5/12 4:22:41 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Released PreventUserIdleSystemSleep "AirDrop"
6/6/12 8:51:54 AM PDT    Assertions           PID 5681(AddressBookSour)
    Created PreventUserIdleSystemSleep "Address Book Source Sync"
6/6/12 8:52:16 AM PDT    Assertions           PID 5681(AddressBookSour)
    Released PreventUserIdleSystemSleep "Address Book Source Sync"
```

Assertions

```
% pmset -g log

6/5/12 4:22:29 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Created PreventUserIdleSystemSleep "AirDrop"
6/5/12 4:22:41 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Released PreventUserIdleSystemSleep "AirDrop"
6/6/12 8:51:54 AM PDT    Assertions           PID 5681(AddressBookSour)
    Created PreventUserIdleSystemSleep "Address Book Source Sync"
6/6/12 8:52:16 AM PDT    Assertions           PID 5681(AddressBookSour)
    Released PreventUserIdleSystemSleep "Address Book Source Sync"
```

Assertions

```
% pmset -g log

6/5/12 4:22:29 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Created PreventUserIdleSystemSleep "AirDrop"
6/5/12 4:22:41 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Released PreventUserIdleSystemSleep "AirDrop"
6/6/12 8:51:54 AM PDT    Assertions           PID 5681(AddressBookSour)
    Created PreventUserIdleSystemSleep "Address Book Source Sync"
6/6/12 8:52:16 AM PDT    Assertions           PID 5681(AddressBookSour)
    Released PreventUserIdleSystemSleep "Address Book Source Sync"
```

Assertions

```
% pmset -g log

6/5/12 4:22:29 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Created PreventUserIdleSystemSleep "AirDrop"
6/5/12 4:22:41 PM PDT    Assertions           PID 163(NetworkBrowserA)
    Released PreventUserIdleSystemSleep "AirDrop"
6/6/12 8:51:54 AM PDT    Assertions           PID 5681(AddressBookSour)
    Created PreventUserIdleSystemSleep "Address Book Source Sync"
6/6/12 8:52:16 AM PDT    Assertions           PID 5681(AddressBookSour)
    Released PreventUserIdleSystemSleep "Address Book Source Sync"
```

Energy Efficient Software

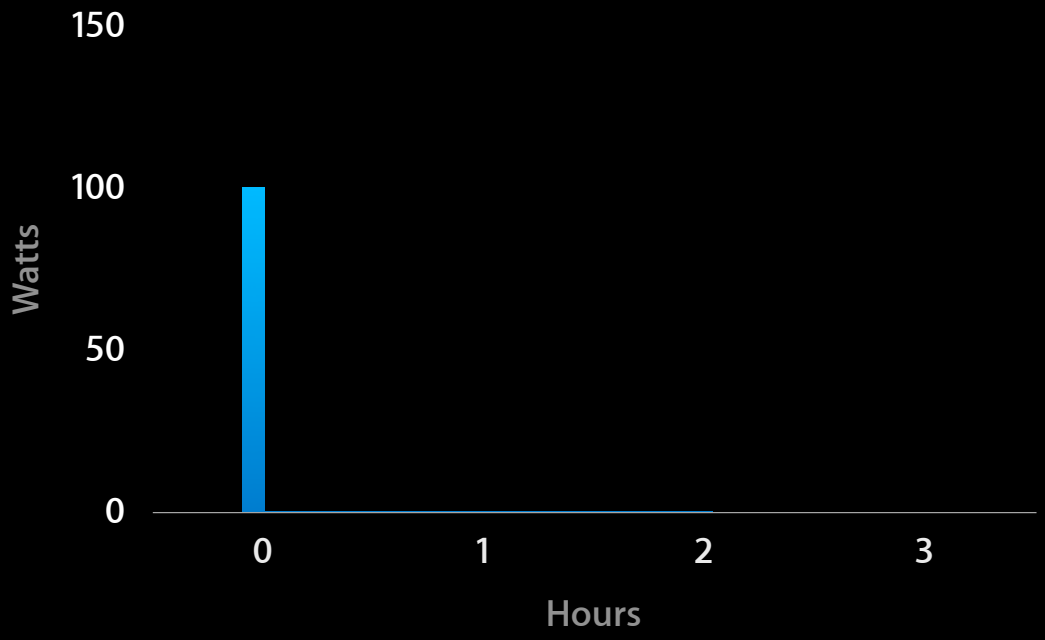
Soren Spies, I/O Kit Team
Core OS Energy Guru

Energy Efficient Software

- Energy 101
- Rationale
- Energy vs. utility
- Software principles and techniques

Energy 101

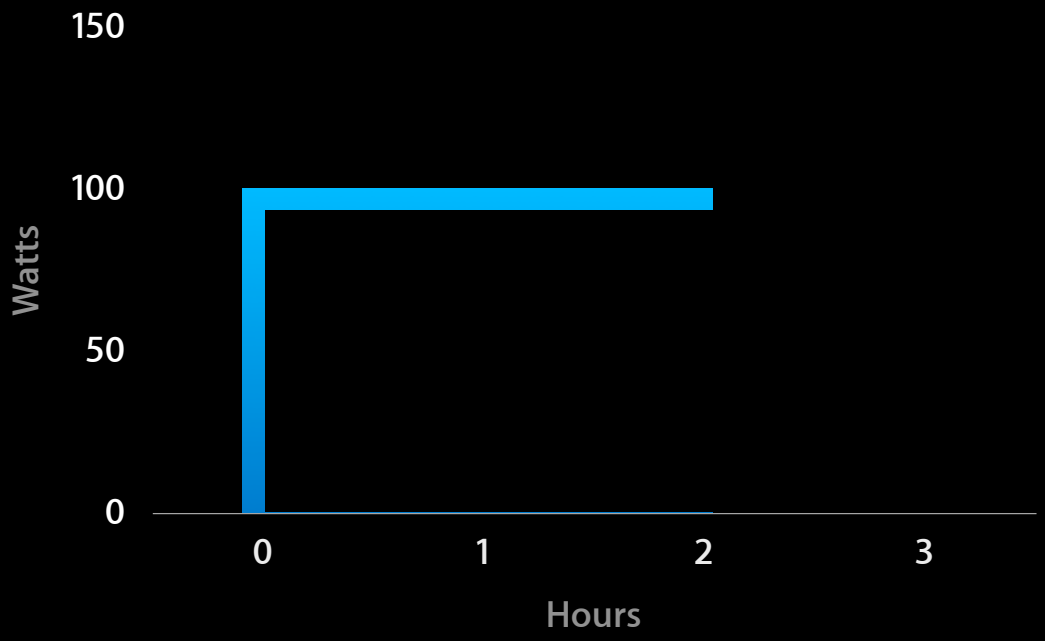
Energy = power x time



100W

Energy 101

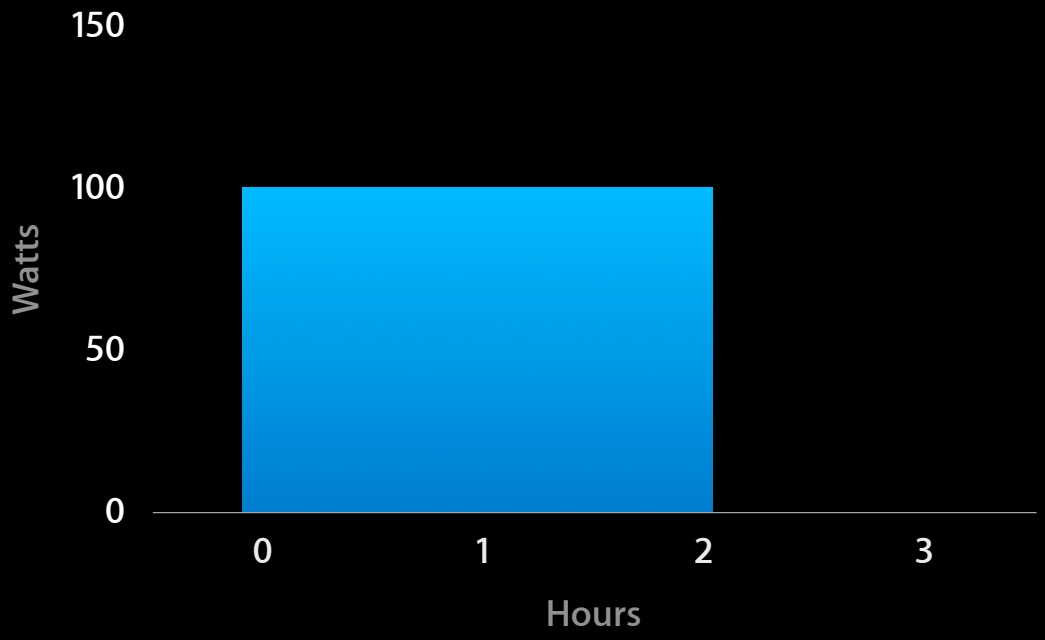
Energy = power x time



100W x 2h

Energy 101

Energy = power x time



100W x 2h = 200 Wh

Rationale

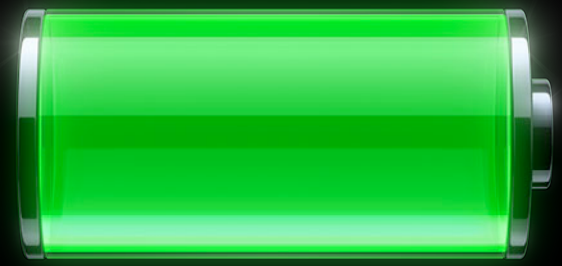
Why software energy efficiency?

- User experience!
  - Battery life
  - Thermals/heat
  - Acoustics/fan noise
- Tread lightly

Rationale

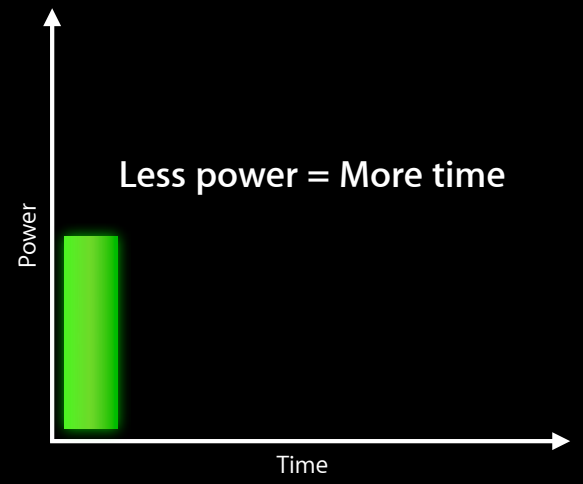
Battery life

- Powerful, dynamic machines
  - 10:1 maxed:idle ~ 10:1 idle:sleep
- Finite batteries
  - Time, heat, utility



Rationale

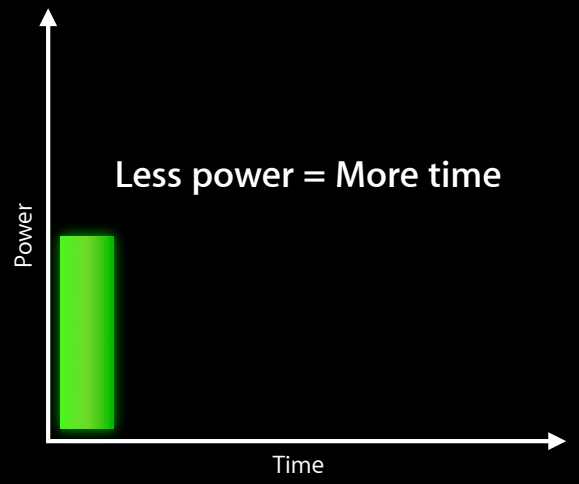
Battery life



Rationale

Battery life

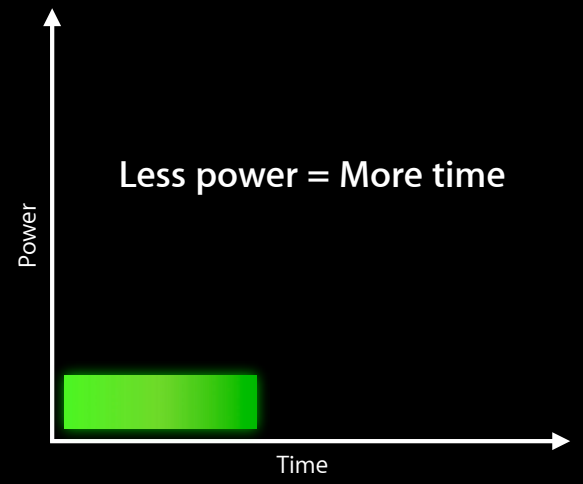
- Make energy last!
- "0.1% CPU" can raise idle power 10%
  - Costs a Mac 30–45 min of battery life



Rationale

Battery life

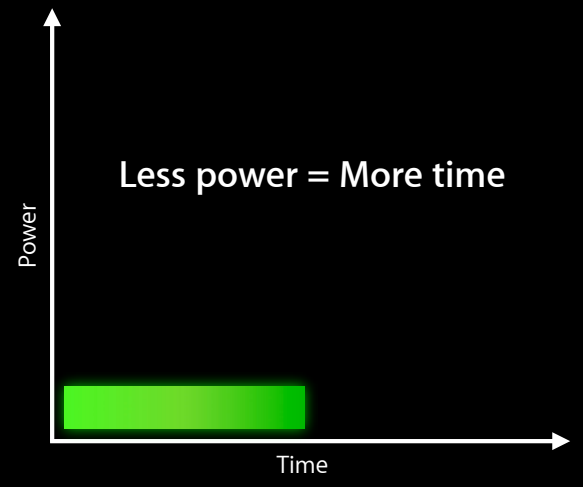
- Make energy last!
- "0.1% CPU" can raise idle power 10%
  - Costs a Mac 30–45 min of battery life



Rationale

Battery life

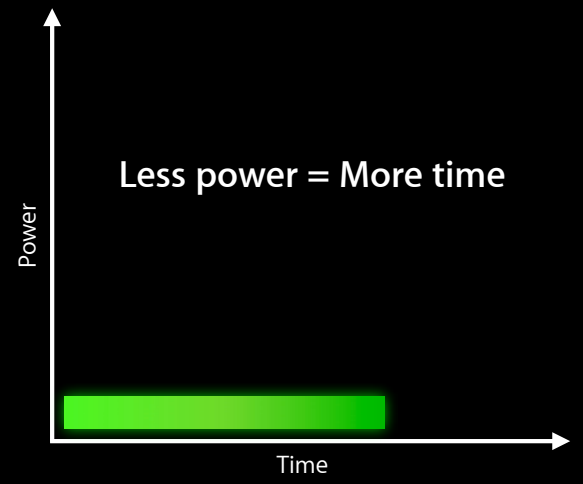
- Make energy last!
- "0.1% CPU" can raise idle power 10%
  - Costs a Mac 30–45 min of battery life



Rationale

Battery life

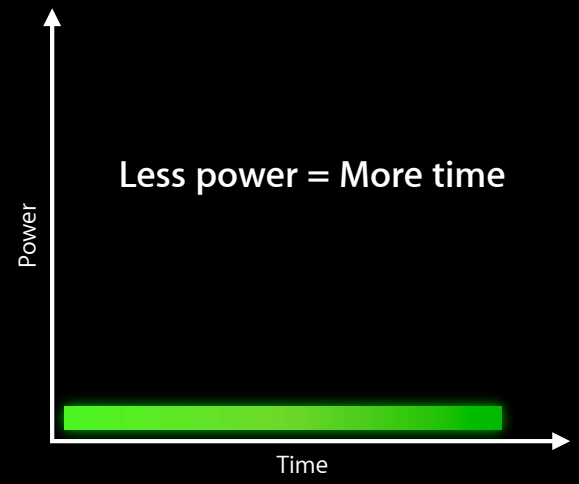
- Make energy last!
- "0.1% CPU" can raise idle power 10%
  - Costs a Mac 30–45 min of battery life



Rationale

Battery life

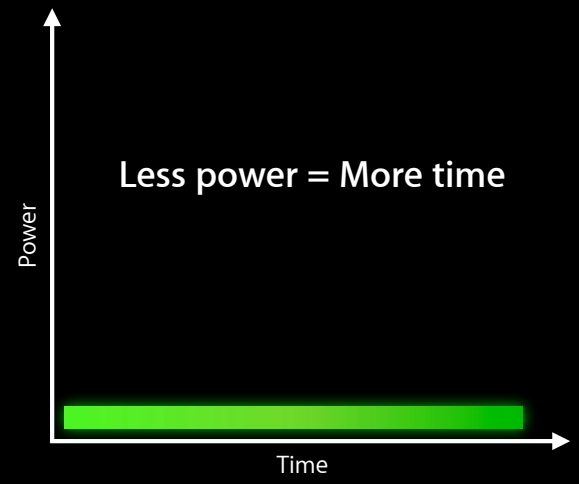
- Make energy last!
- "0.1% CPU" can raise idle power 10%
  - Costs a Mac 30–45 min of battery life



Rationale

Battery life

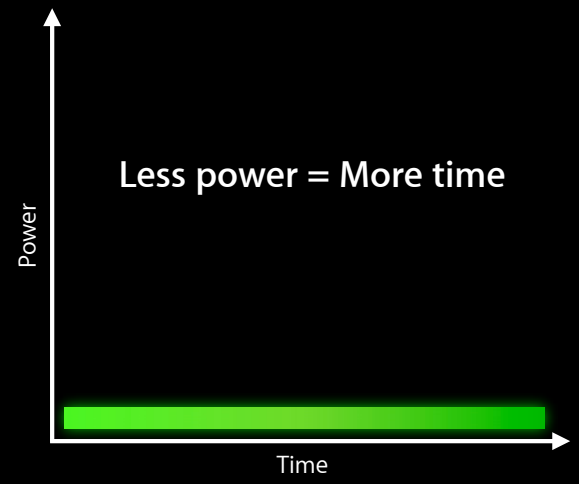
- Make energy last!
- "0.1% CPU" can raise idle power 10%
  - Costs a Mac 30–45 min of battery life



Rationale

Battery life

- Make energy last!
- "0.1% CPU" can raise idle power 10%
  - Costs a Mac 30–45 min of battery life



Energy vs. Utility

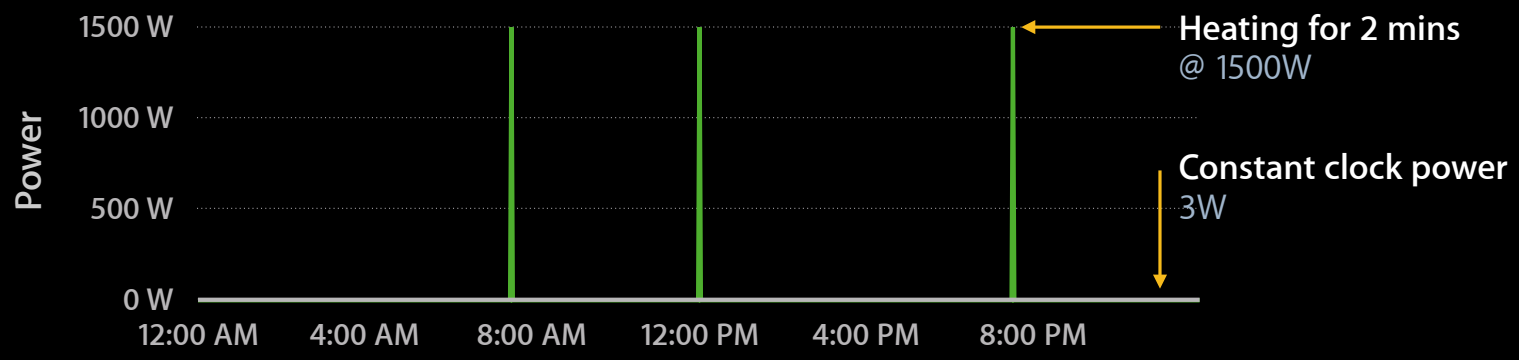
Microwave energy

Microwave Energy

- A microwave has two functions
  - Heating your food (~1500W)
  - Displaying the time (~3W)

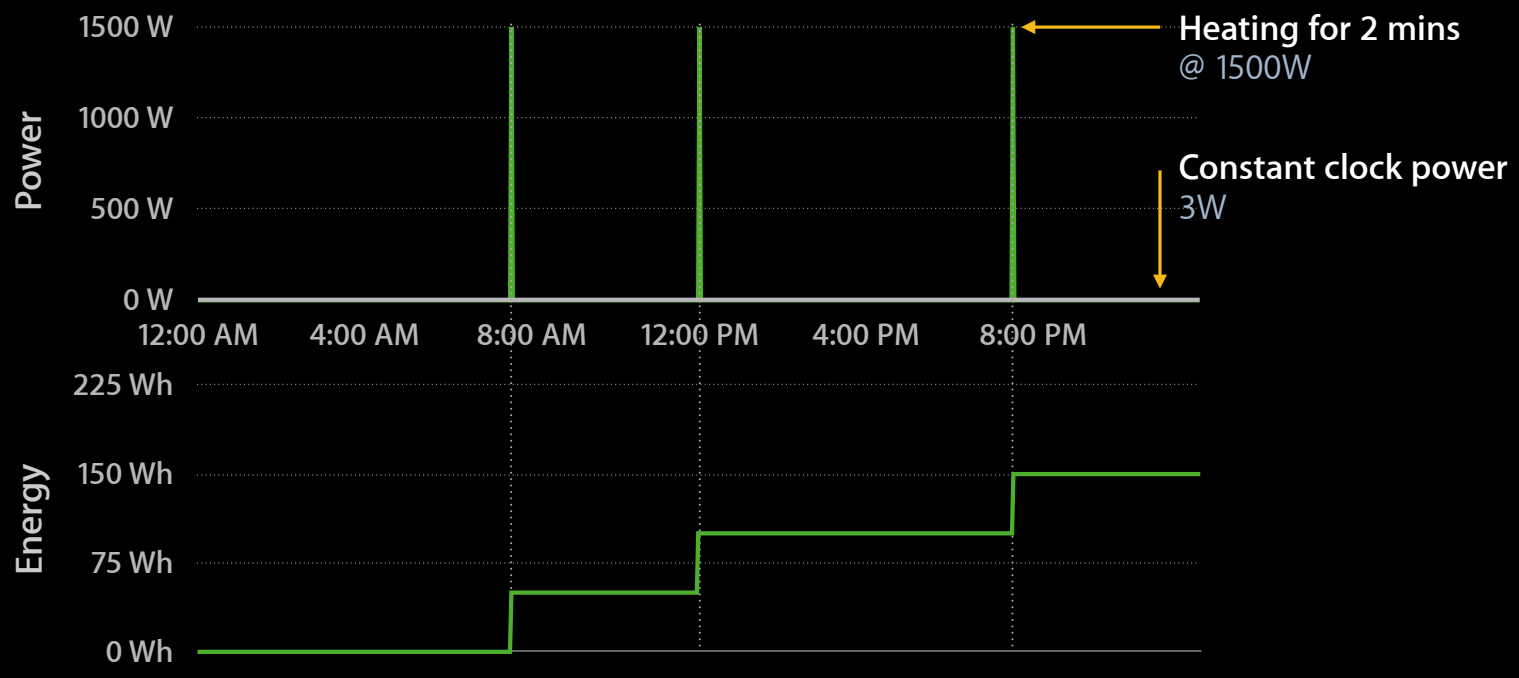
Microwave Energy

Heating vs. clock



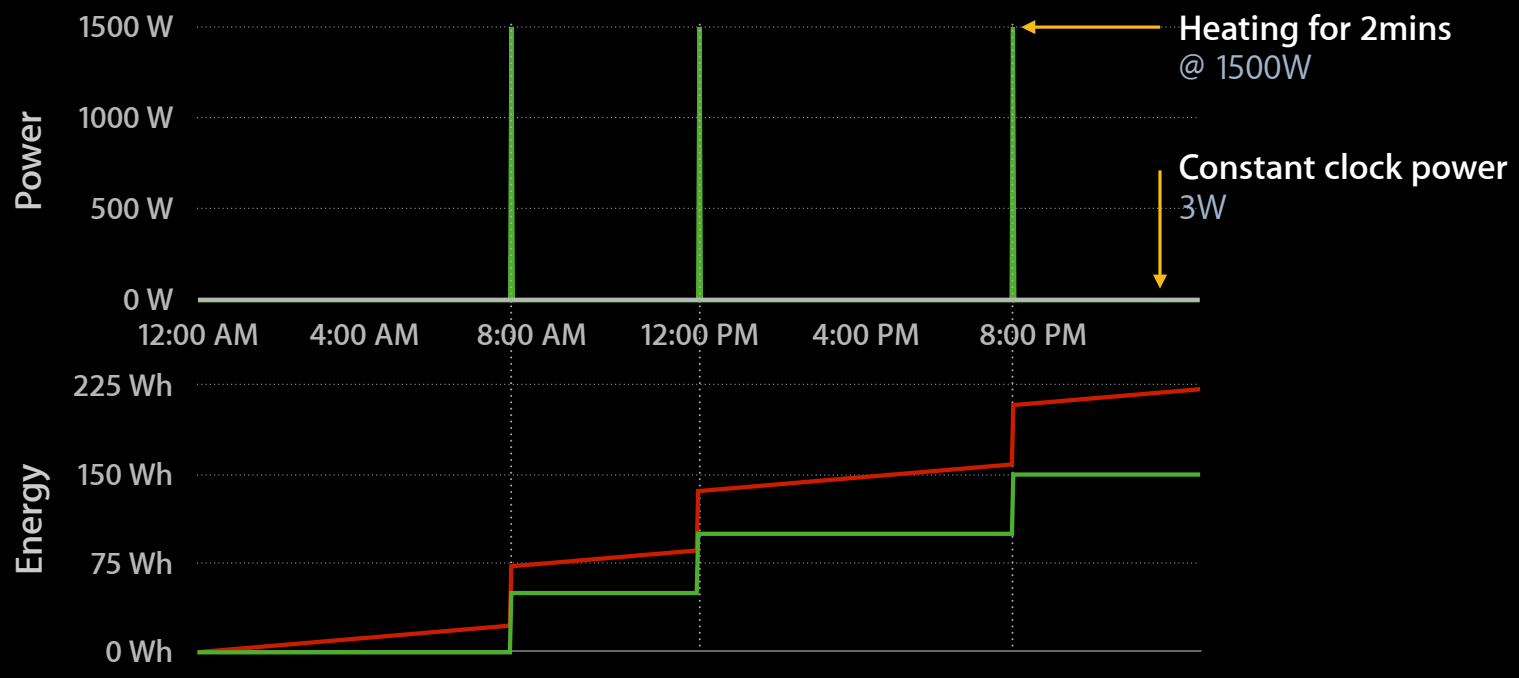
Microwave Energy

Energy = power x time



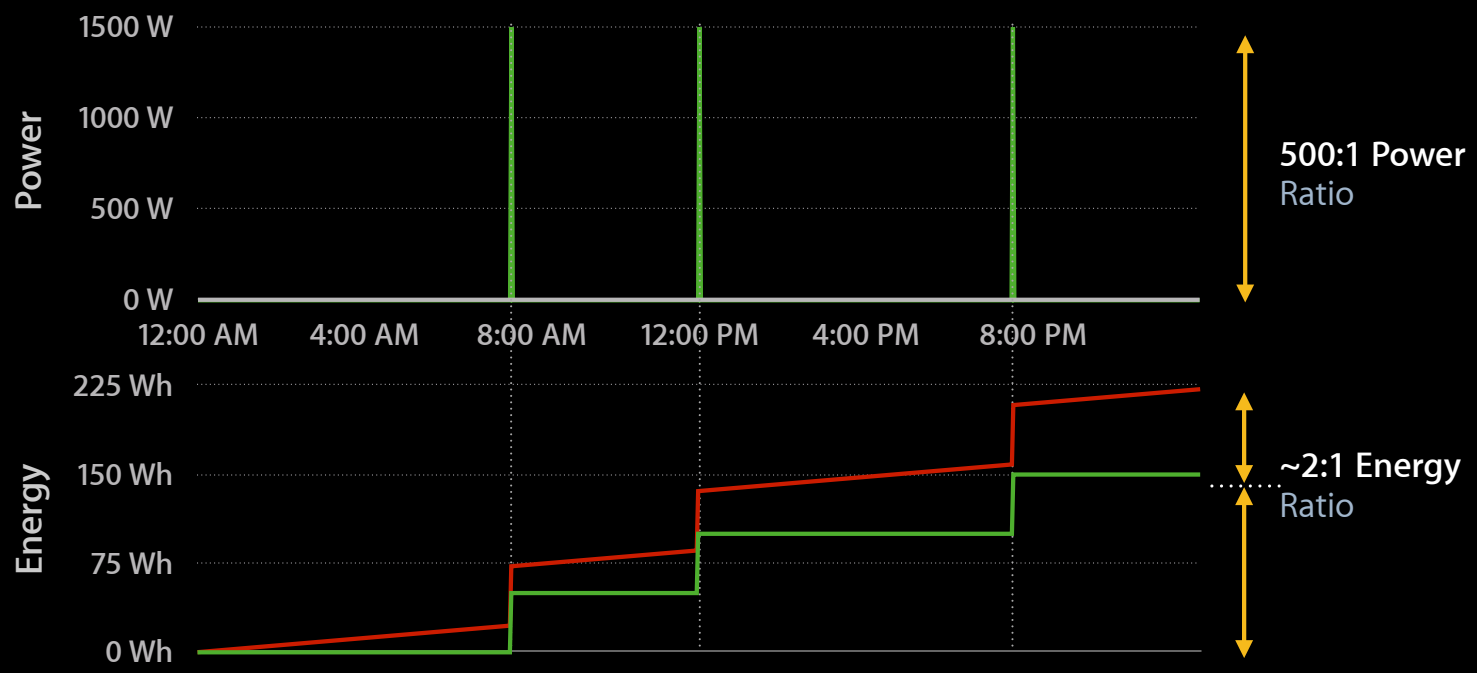
Microwave Energy

1500W x 6m + 3W x 1440m



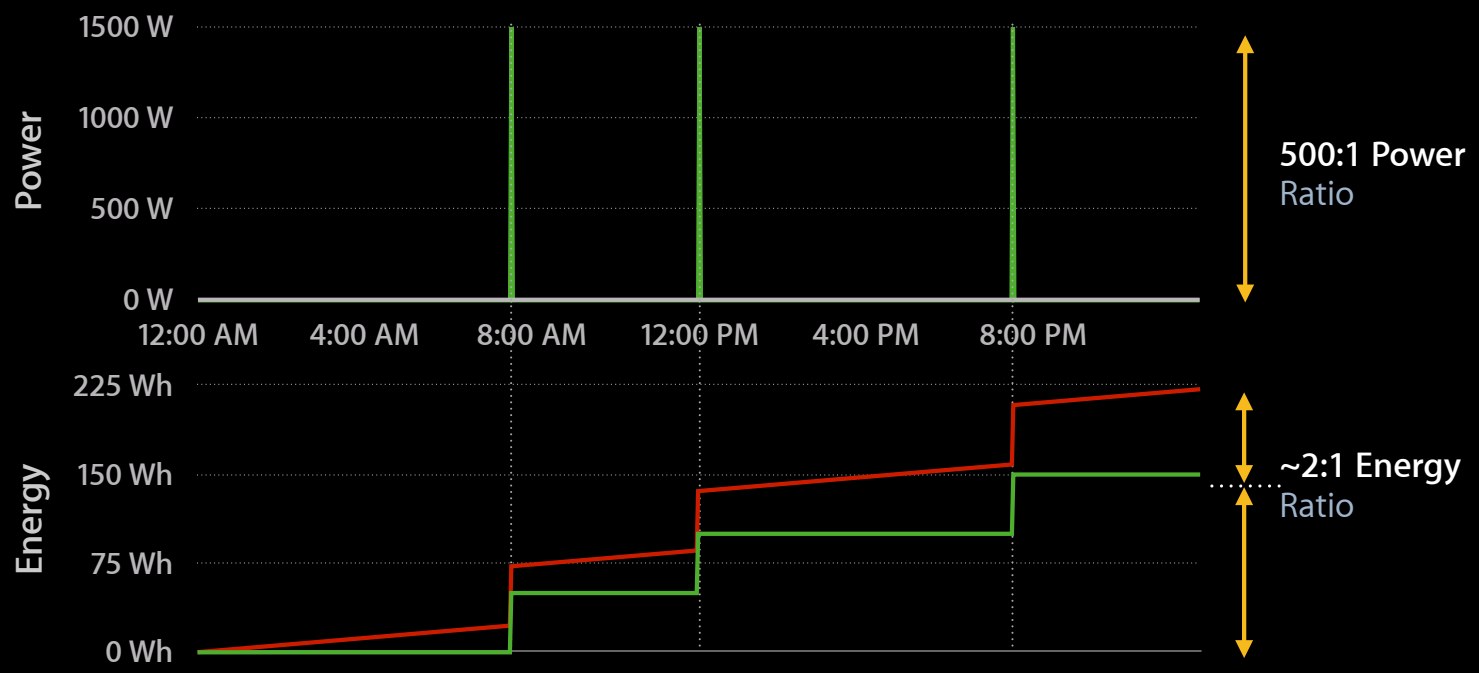
Microwave Energy

1500W x 6m + 3W x 1440m



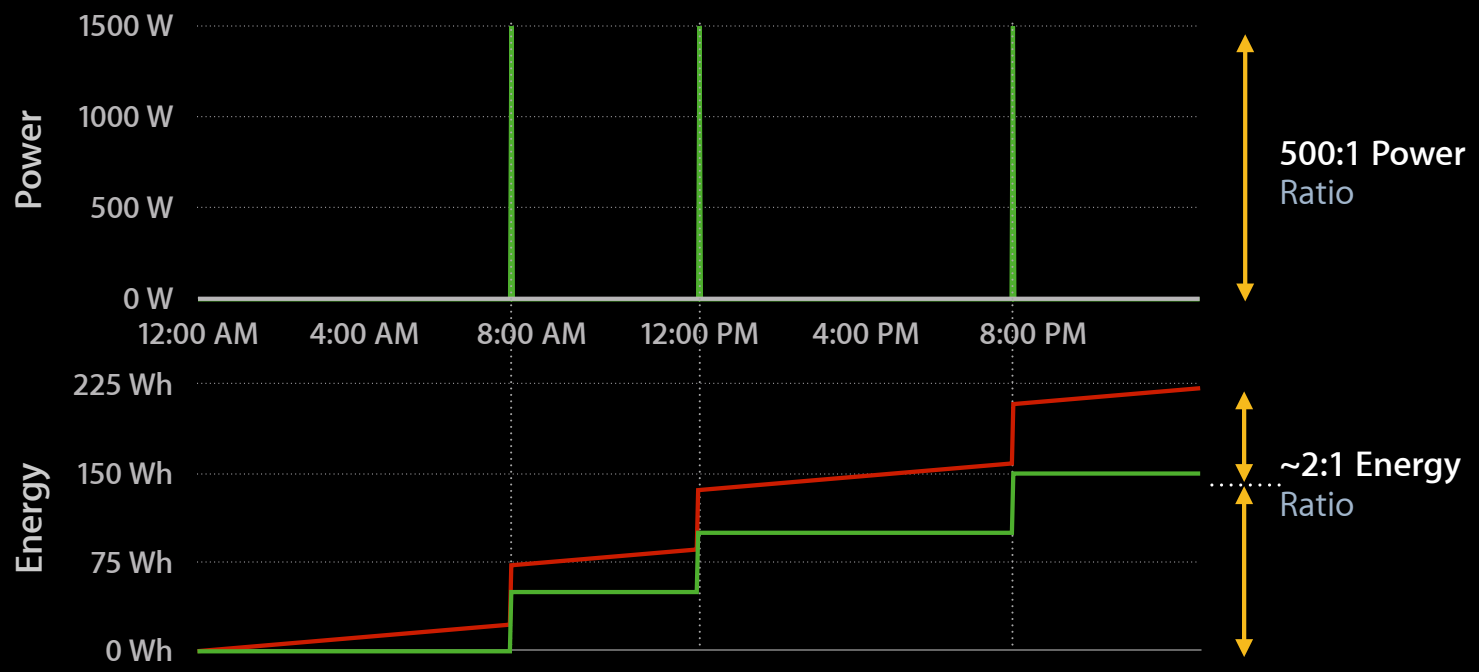
Microwave Energy

1500W x 6m + 3W x 1440m



Microwave Energy

1500W x 6m + 3W x 1440m



Energy vs. Utility

- User wants food heated
- Background activity costs

Principles and Techniques

Energy efficient software

Principles

Energy efficient software

- Absolute idle
- Extreme efficiency
- Turn off the lights!

Absolute Idle

- Block for useful work
  - System API
- Remove periodic activity
  - Animations
  - Network polling
  - Timers!

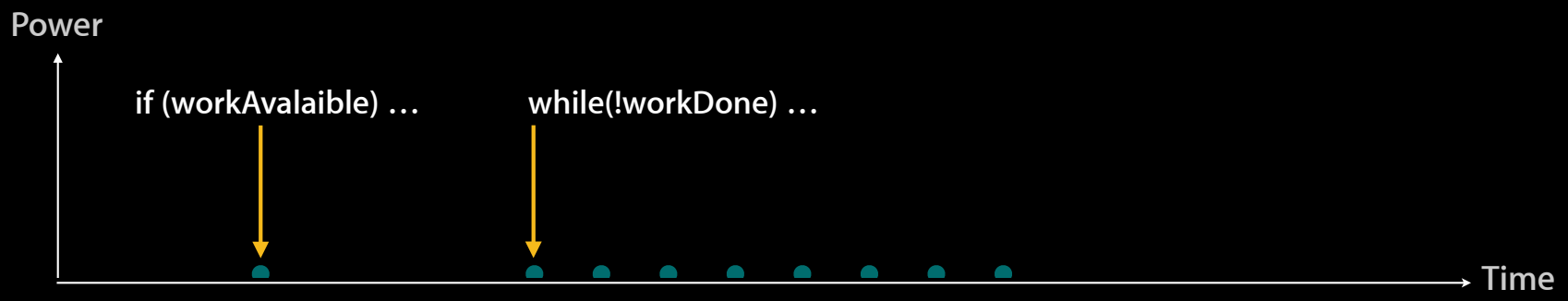
Absolute Idle

Naive polling

```
while(sleep(1)) {
    if (workAvailable) {
         launchWorkOnHelperThread();
         while(!workDone) {
             nanosleep(<100ms>);
         }
         workIsDone();
     }
}
```

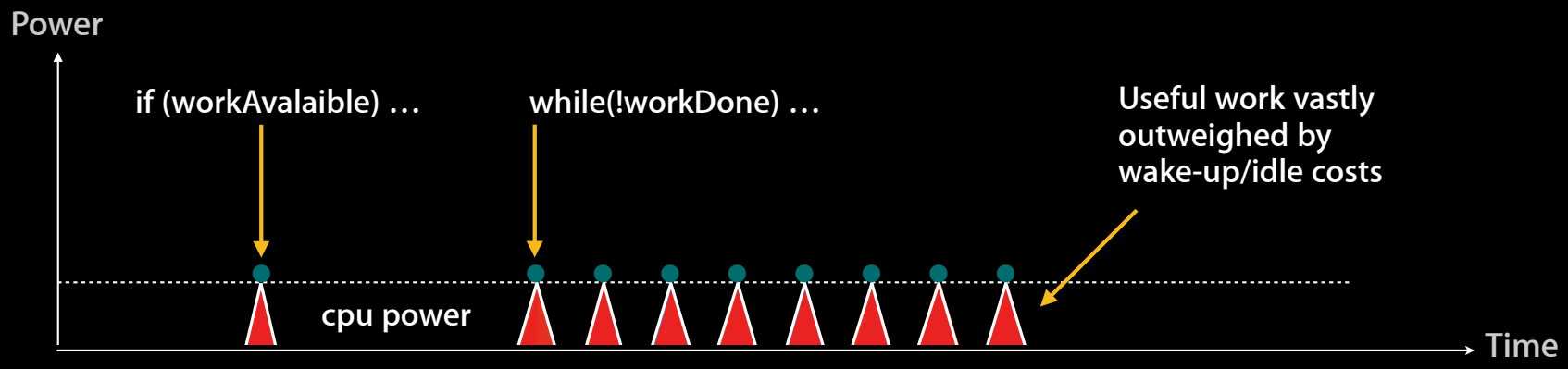
Absolute Idle

CPU wakes expensive



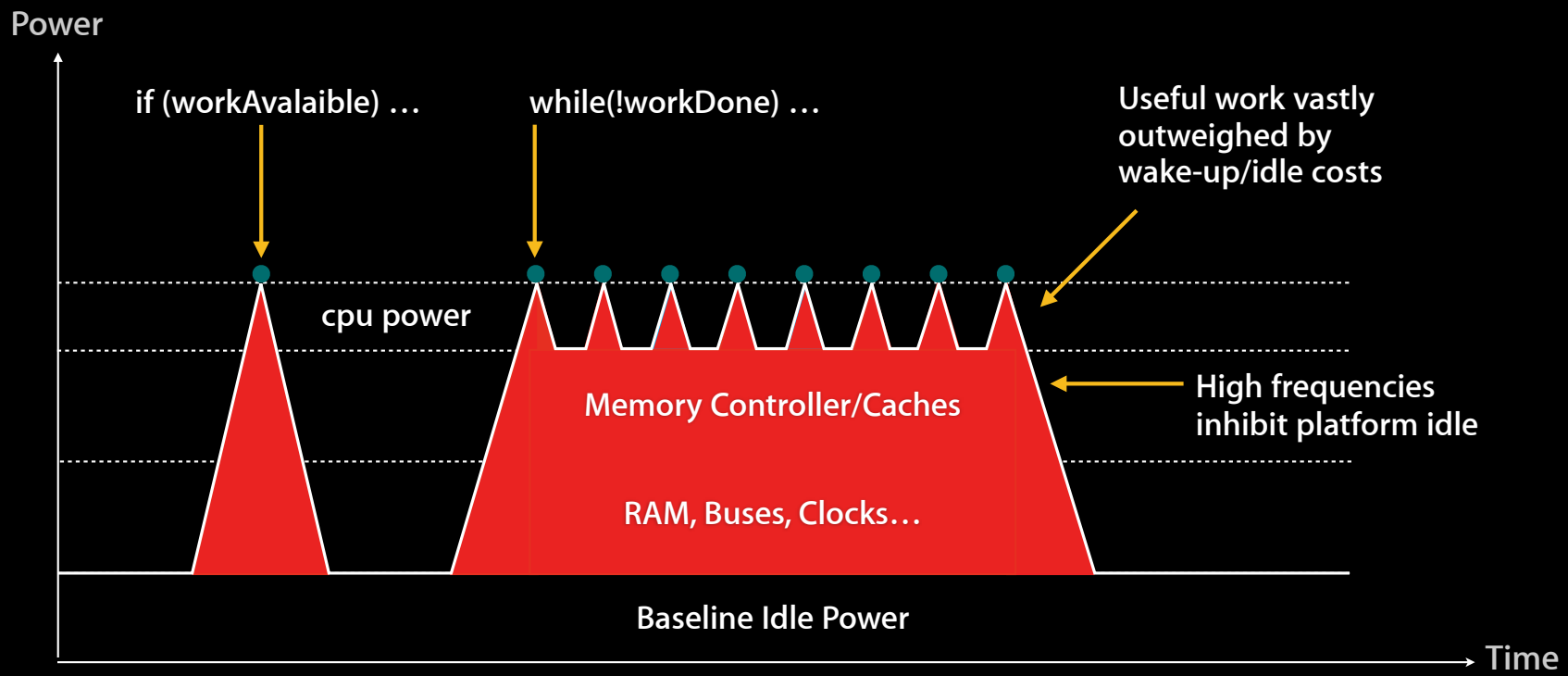
Absolute Idle

CPU wakes expensive



Absolute Idle

CPU wakes expensive



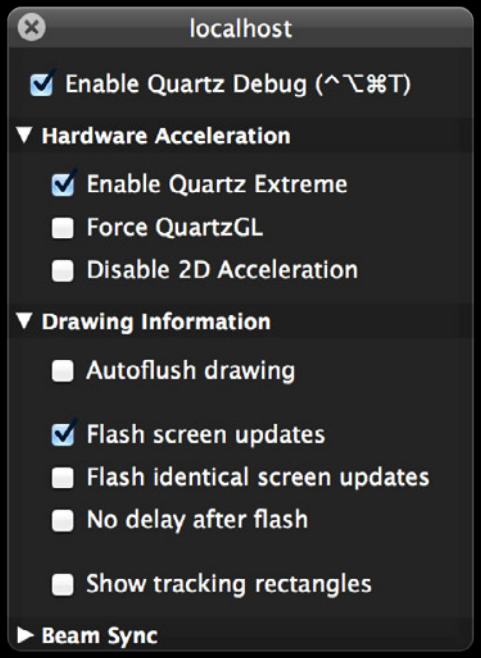
Eliminating Idle Energy Leaks

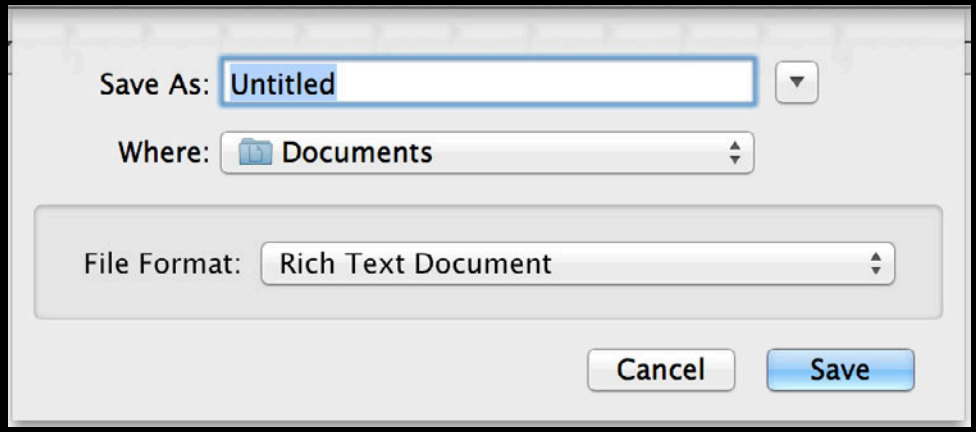
Tools, tips, and techniques

- Unexpected drawing?
- CPU time
- System calls
- Function calls

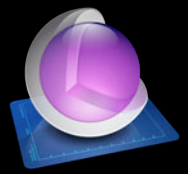
Eliminating Idle Energy Leaks

Unexpected drawing



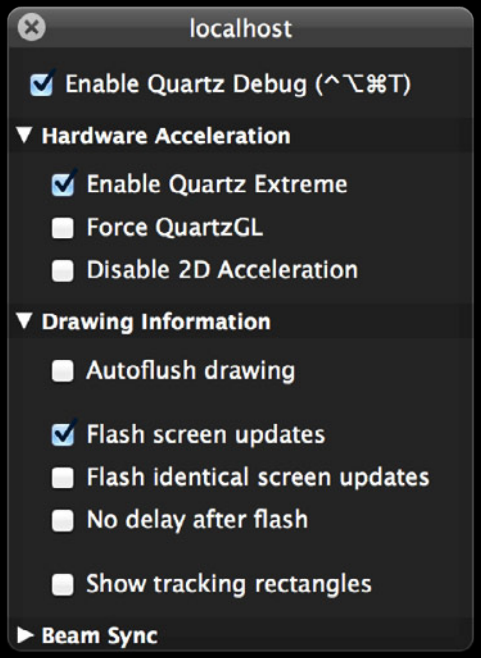


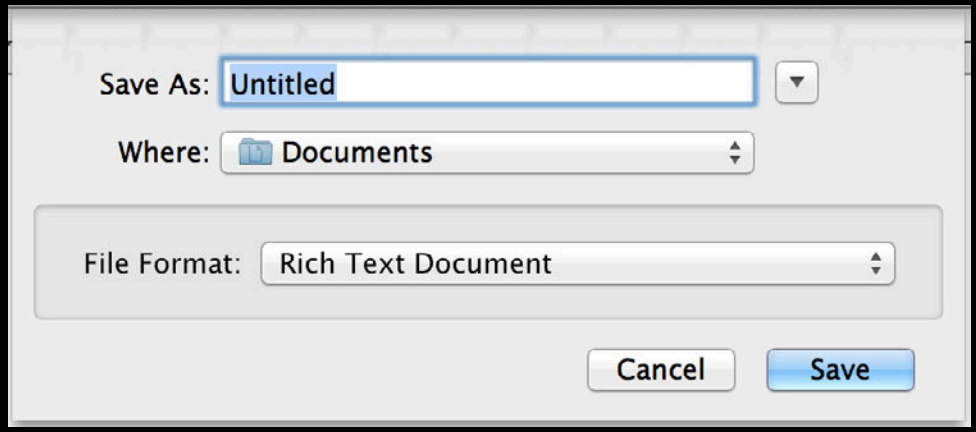
Core Animation



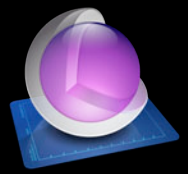
Eliminating Idle Energy Leaks

Unexpected drawing





Core Animation



Eliminating Idle Energy Leaks

Accumulating CPU time?

```
$ top -a -pid <target>
PID    COMMAND      %CPU TIME     #TH   #WQ   #POR #MRE RPRVT RSHRD RSIZE  VPRVT
24920  TextEdit     0.0  00:00.00 1     0     21+  26+  712K+ 852K+ 1136K+ 9648K+

$ sample TextEdit
Sampling process 12572 for 10 seconds with 1 millisecond of run time between samples
...
```

Activity Monitor

Time Profiler

Eliminating Idle Energy Leaks

Accumulating CPU time?

```
$ top -a -pid <target>
PID    COMMAND      %CPU TIME     #TH   #WQ   #POR #MRE RPRVT RSHRD RSIZE  VPRVT
24920  TextEdit     0.0  00:00.00 1     0     21+  26+  712K+ 852K+ 1136K+ 9648K+

$ sample TextEdit
Sampling process 12572 for 10 seconds with 1 millisecond of run time between samples
...
```

Activity Monitor

Time Profiler

Eliminating Idle Energy Leaks

File system tickles

- fs_usage <target>
  - [... nothing? ...]

File Activity

System Usage

Eliminating Idle Energy Leaks

sc_usage(8) collates system calls

```
$ sudo sc_usage -e <target>

TYPE                              NUMBER        CPU_TIME   WAIT_TIME
-------------------------------------------------------------------------------
System          Idle                                     00:08.150(00:00.954)
System          Busy                                     00:01.889(00:00.055)
<target>        Usermode                       00:00.000

mk_timer_arm                       28(4)       00:00.000
psynch_cvwait                      20(2)       00:00.000  00:18.229(00:02.044)
mach_msg_trap                      19          00:00.000  00:05.066
kevent                             12          00:00.000  00:05.994
gettimeofday                        6(1)       00:00.000
workq_kernreturn                    3          00:00.000  00:08.049
```

Eliminating Idle Energy Leaks

sc_usage(8) collates system calls

```
$ sudo sc_usage -e <target>

TYPE                              NUMBER        CPU_TIME    WAIT_TIME
-------------------------------------------------------------------------------
System          Idle                                      00:08.150(00:00.954)
System          Busy                                      00:01.889(00:00.055)
<target>        Usermode                      00:00.000

mk_timer_arm                       28(4)        00:00.000
psynch_cvwait                      20(2)        00:00.000   00:18.229(00:02.044)
mach_msg_trap                      19           00:00.000   00:05.066
kevent                             12           00:00.000   00:05.994
gettimeofday                        6(1)        00:00.000
workq_kernreturn                    3           00:00.000   00:08.049
```

Eliminating Idle Energy Leaks

sc_usage(8) collates system calls

```
$ sudo sc_usage -e <target>

TYPE                            NUMBER        CPU_TIME    WAIT_TIME
-------------------------------------------------------------------------------
System          Idle                                      00:08.150(00:00.954)
System          Busy                                      00:01.889(00:00.055)
<target>        Usermode                      00:00.000

mk_timer_arm                    28(4)         00:00.000
psynch_cvwait                   20(2)         00:00.000   00:18.229(00:02.044)
mach_msg_trap                   19            00:00.000   00:05.066
kevent                          12            00:00.000   00:05.994
gettimeofday                    6(1)          00:00.000
workq_kernreturn                3             00:00.000   00:08.049
```

Eliminating Idle Energy Leaks

Digging in

```
$ sudo dtrace –n ‘syscall::gettimeofday:entry
     /execname == “myApp”/ { ustack() }’

$ sudo dtrace –n ‘pid123:::entry’
```

Eliminating Idle Energy Leaks

Digging in

```
$ sudo dtrace –n ‘syscall::gettimeofday:entry
     /execname == “myApp”/ { ustack() }’

$ sudo dtrace –n ‘pid123:::entry’
```

Eliminating Idle Energy Leaks

Digging in

```
$ sudo dtrace –n ‘syscall::gettimeofday:entry
     /execname == “myApp”/ { ustack() }’

$ sudo dtrace –n ‘pid123:::entry’
```

System Trace

System Calls

Extreme Efficiency

- Faster completion ~= less energy
  - Use Apple-optimized library code
- Expose significant work via threads/GCD/NSOperation
- setpriority(2) w/DARWIN_BG for background threads
- Batch “maintenance” operations with work for user

Extreme Efficiency

Accelerate FFT

- Setup…Operate…Destroy

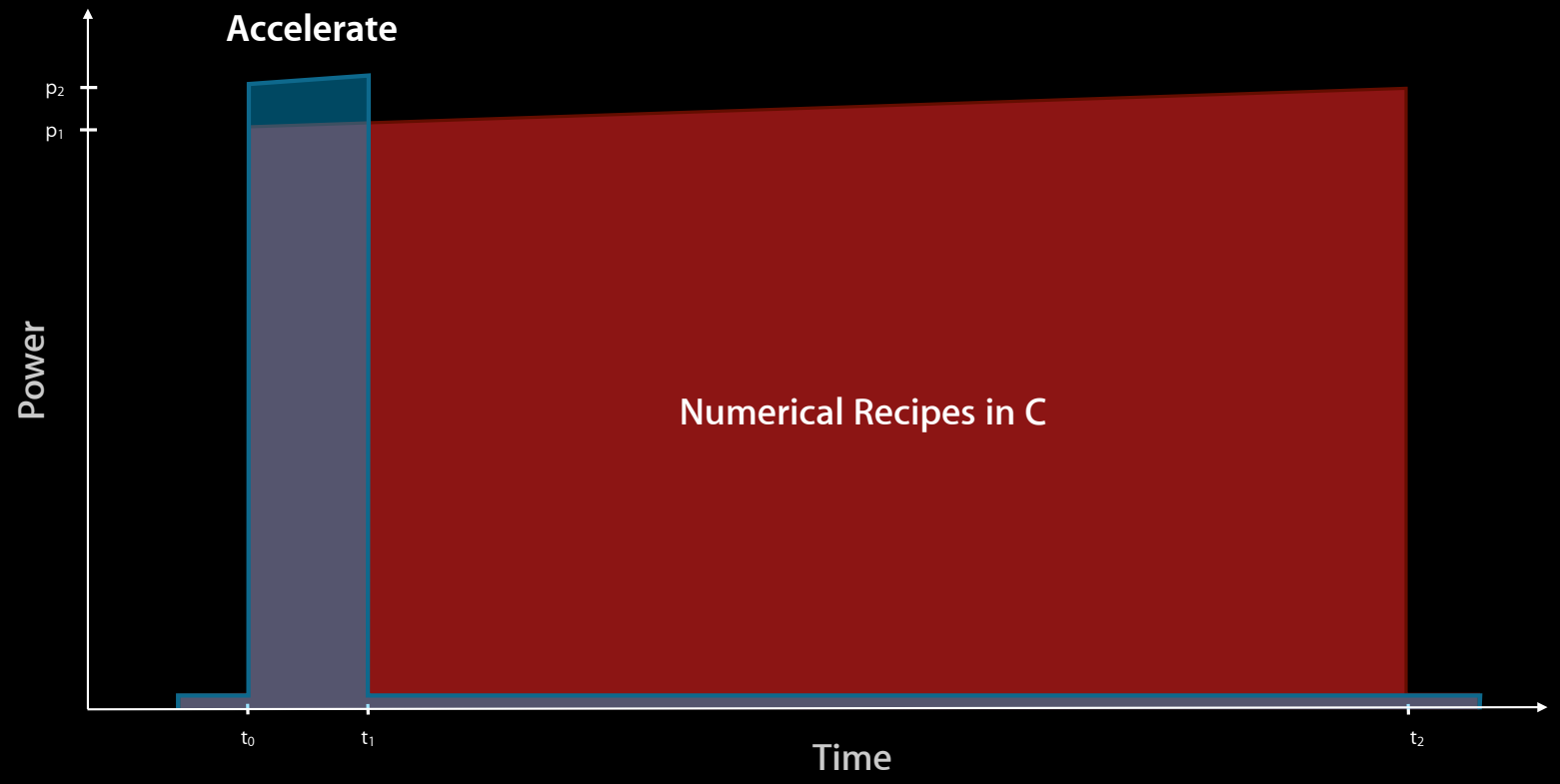
```
#include <Accelerate/Accelerate.h>

DSPSplitComplex data;
const int log2n = 10;

// Once at start:
FFTSetup setup = vDSP_create_fftsetup(log2n, FFT_RADIX2);
...
    vDSP_fft_zip(setup, &data, 1, log2n, FFT_FORWARD);
...
// Once at end:
vDSP_destroy_fftsetup(setup);
```

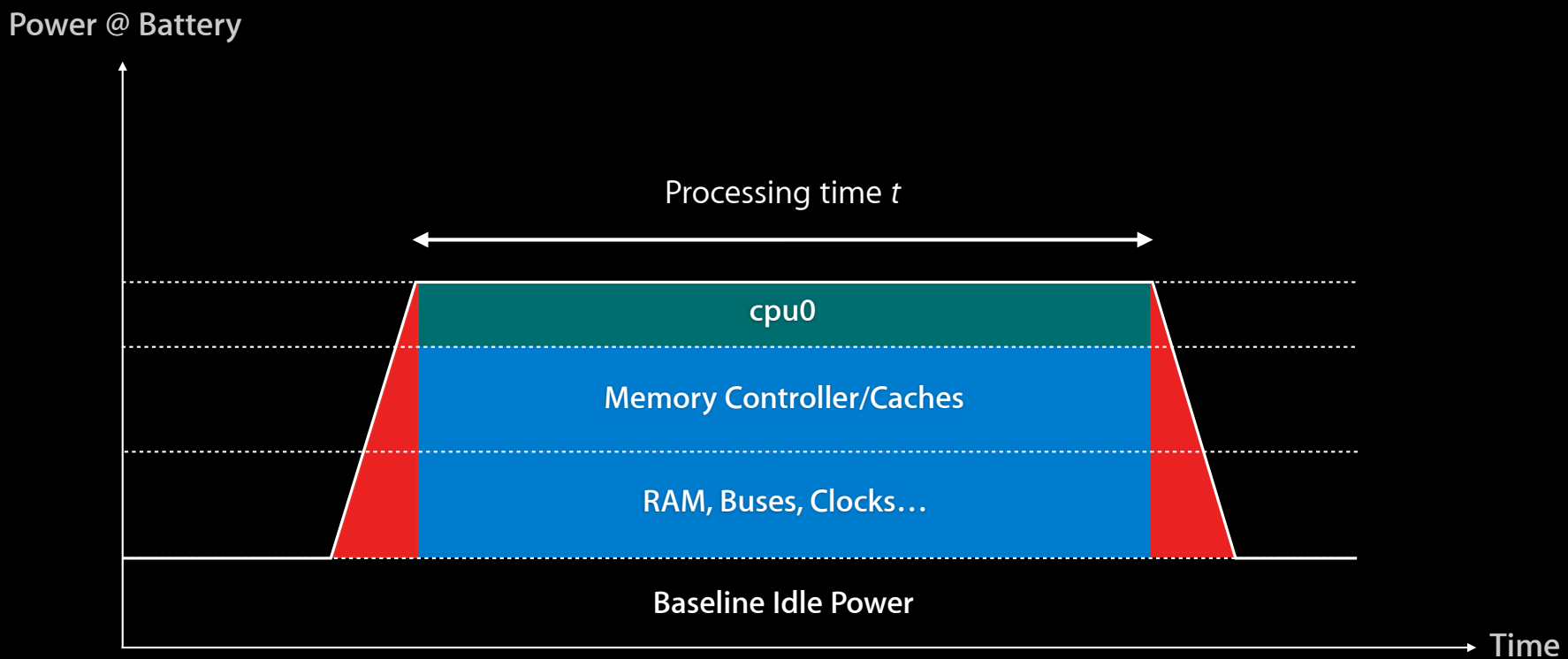
Extreme Efficiency

Accelerate vs. textbook C



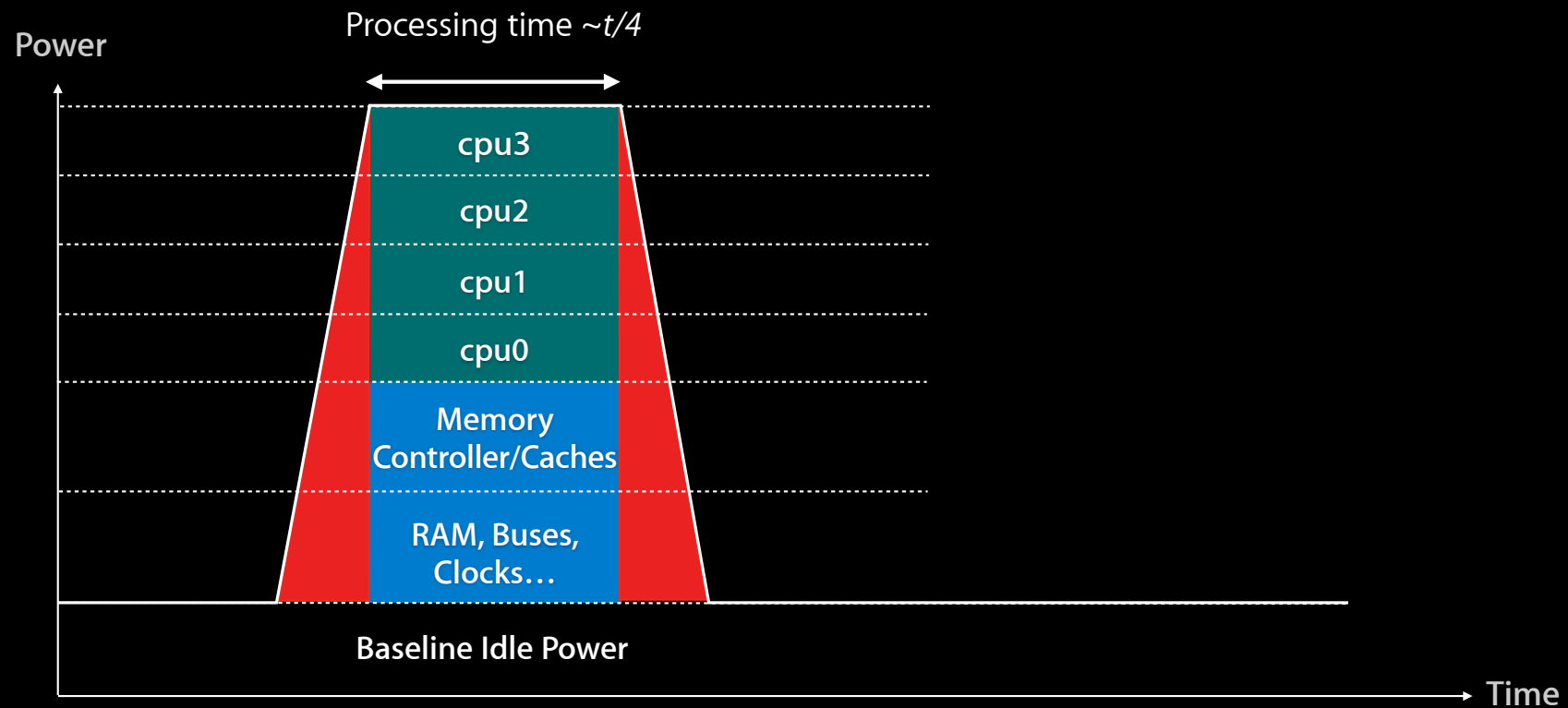
Extreme Efficiency

Processing on a single core



Extreme Efficiency

Processing on all cores



Rendering Frames

How many fps do you need?

- 24, 30, or 60 Hz
  - Use CVDisplayLink to sync w/display
- Sample runloops
- Avoid “open loop" on OpenGL
- Efficient memory use
  - Use CLIENT_STORAGE and TEXTURE_RANGE
- Dynamic frame rates
  - Fewer updates to background elements?
- QA1385: Driving OpenGL Rendering Loops

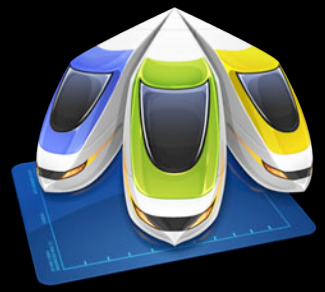
Extreme Efficiency

Determining parallelism with Instruments

Multicore

System Trace/Scheduling

Dispatch



Performance ~ Energy

Signs of energy leaks

- Swapping
- Memory stalls/cache misses
- Small disk I/Os (log spew)
- Excessive context switches
- Lock contention
- Lots of real work: Only one thread
- High-overhead communication
  - Small packets, buffers, etc.
- iOS memory warnings

Turn Off the Lights

- Peripherals are conservative
  - Power up quickly
  - Often wait for a period of inactivity
- Turn off when done

Turn Off the Lights

- Peripherals are conservative
  - Power up quickly
  - Often wait for a period of inactivity
- Turn off when done

Turn Off the Lights

- API to indicate “done”
  - Audio
  - Graphics/GPU
  - Camera
  - GPS
  - Networking: Wi-Fi and cellular
  - Bluetooth
- Assertions
  - Directly and indirectly on OS X
  - Indirectly on iOS (UIKitBackgroundTaskCompletion)

Turn Off the Lights

- API to indicate "done"
  - Audio
  - Graphics/GPU
  - Camera
  - GPS
  - Networking: Wi-Fi and cellular
  - Bluetooth
- Assertions
  - Directly and indirectly on OS X
  - Indirectly on iOS (UIKitBackgroundTaskCompletion)

Turn Off the Lights

CPU still idle?

- Exercise code
- Recheck idle

Turn Off the Lights

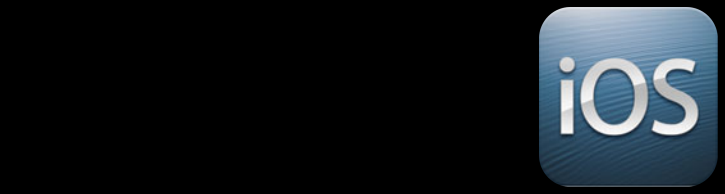
GPU Mux

- Developer QA#1734

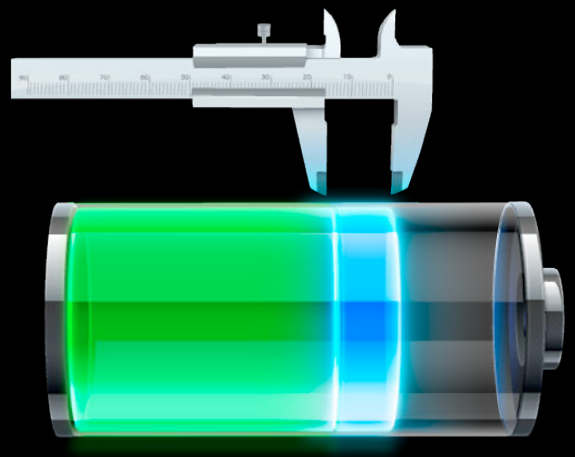
```
br cglBadApplicationNotMuxAwareLockDown
```

Turn Off the Lights

iOS Energy Diagnostics Instrument



- Energy usage
- CPU activity
- Network activity
- Display brightness
- Sleep/wake
- Bluetooth
- Wi-Fi
- GPS



More Information

Paul Danbold
Core OS Evangelist
danbold@apple.com

Documentation
OpenGL Programming Guide
http://developer.apple.com/

Downloads
Graphics Tools for Xcode
http://developer.apple.com/

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Learning Instruments	Presidio Wednesday 4:30PM
The Accelerate Framework	Russian Hill Thursday 10:15AM
iOS App Performance: Responsiveness	Presidio Thursday 11:30AM
iOS App Performance: Graphics and Animations	Presidio Thursday 3:15PM

Labs

Power Management Lab	Core OS Lab B Friday 10:15AM
OS X Performance Lab	Developer Tools Lab A Friday 9:00AM

Summary

Take the time to…

- Achieve absolute idle
- Do (real!) work efficiently
- Turn off the lights

- Remember: Energy = power x time


WWDC2012