Power Management

Energy efficient software

Session 711

Ethan Bold
I/O Kit Team

Soren Spies
I/O Kit Team

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.
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

Display is on ..........................

Is awake .............................

Is asleep .............................
Keeps the Display Awake

kIOPMAssertionTypePreventUserIdleDisplaySleep

Display is on ..........................

Is awake ............................

Is asleep ............................
Keeps the Display Awake

\texttt{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

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

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

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

IOPMAssertionRelease(newAssertion);
```
#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

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

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

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

IOPMAssertionRelease(newAssertion);
```
#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

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

IOPMAssertionID newAssertion = kIOPMAssertionNULLID;

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

IOPMAssertionRelease(newAssertion);
```
Power Management Policy

App 0

App 1

App 2
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

```sh
% 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()
The diagram illustrates the state of various components (CPU, Network, Disk, Graphics, Audio) during different modes:

- **FullWake**:
  - Audio
  - Graphics
  - Disk
  - Network
  - CPU

- **DarkWake**:
  - Audio
  - Graphics
  - Disk
  - Network
  - CPU

- **Sleep**:
  - 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
- Audio
- Graphics
- Disk
- Network
- CPU

DarkWake
- Audio
- Graphics
- Disk
- Network
- CPU

Sleep
- Audio
- Graphics
- Disk
- Network
- CPU
Power Assertions
DarkWake
Debugging
## Sleep and Wake

% pmset -g log

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/16/12</td>
<td>10:57 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:11%)</td>
<td>62416 secs</td>
</tr>
<tr>
<td>5/16/12</td>
<td>10:58 PM</td>
<td>WakeRequests</td>
<td>Clients requested wake events: None</td>
<td></td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>DarkWake</td>
<td>DarkWake due to EHC1: Using AC (Charge:95%)</td>
<td>19 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:95%)</td>
<td>94 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:20 PM</td>
<td>Wake</td>
<td>Wake due to EC LID0: Using AC (Charge:95%)</td>
<td>2662 secs</td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Event</td>
<td>Description</td>
<td>Duration</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>----------------</td>
<td>-------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>5/16/12</td>
<td>10:57 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:11%)</td>
<td>62416 secs</td>
</tr>
<tr>
<td>5/16/12</td>
<td>10:58 PM</td>
<td>WakeRequests</td>
<td>Clients requested wake events: None</td>
<td></td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>DarkWake</td>
<td>DarkWake due to EHC1: Using AC (Charge:95%)</td>
<td>19 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:95%)</td>
<td>94 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:20 PM</td>
<td>Wake</td>
<td>Wake due to EC LID0: Using AC (Charge:95%)</td>
<td>2662 secs</td>
</tr>
</tbody>
</table>
## Sleep and Wake

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Details</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/16/12</td>
<td>10:57 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:11%)</td>
<td>62416 secs</td>
</tr>
<tr>
<td>5/16/12</td>
<td>10:58 PM</td>
<td>WakeRequests</td>
<td>Clients requested wake events: None</td>
<td></td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>DarkWake</td>
<td>DarkWake due to EHC1: Using AC (Charge:95%)</td>
<td>19 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:95%)</td>
<td>94 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:20 PM</td>
<td>Wake</td>
<td>Wake due to EC LID0: Using AC (Charge:95%)</td>
<td>2662 secs</td>
</tr>
</tbody>
</table>
# Sleep and Wake

% pmset -g log

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/16/12</td>
<td>10:57 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep: Using AC (Charge:11%)</td>
<td>62416 secs</td>
</tr>
<tr>
<td>5/16/12</td>
<td>10:58 PM</td>
<td>WakeRequests</td>
<td>Clients requested wake events: None</td>
<td></td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>DarkWake</td>
<td>DarkWake due to EHC1: Using AC (Charge:95%)</td>
<td>19 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:18 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep: Using AC (Charge:95%)</td>
<td>94 secs</td>
</tr>
<tr>
<td>5/17/12</td>
<td>4:20 PM</td>
<td>Wake</td>
<td>Wake due to EC LID0: Using AC (Charge:95%)</td>
<td>2662 secs</td>
</tr>
</tbody>
</table>
## Sleep and Wake

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/16/12 10:57 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:11%)</td>
<td>62416 secs</td>
</tr>
<tr>
<td>5/16/12 10:58 PM</td>
<td>WakeRequests</td>
<td>Clients requested wake events: None</td>
<td></td>
</tr>
<tr>
<td>5/17/12 4:18 PM</td>
<td>DarkWake</td>
<td>DarkWake due to EHC1: Using AC (Charge:95%)</td>
<td>19 secs</td>
</tr>
<tr>
<td>5/17/12 4:18 PM</td>
<td>Sleep</td>
<td>Clamshell Sleep Sleep: Using AC (Charge:95%)</td>
<td>94 secs</td>
</tr>
<tr>
<td>5/17/12 4:20 PM</td>
<td>Wake</td>
<td>Wake due to EC LID0: Using AC (Charge:95%)</td>
<td>2662 secs</td>
</tr>
</tbody>
</table>
% 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(AddressBookSource)
    Created PreventUserIdleSystemSleep "Address Book Source Sync"
6/6/12 8:52:16 AM PDT    Assertions        PID 5681(AddressBookSource)
    Released PreventUserIdleSystemSleep "Address Book Source Sync"
% 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"
% 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"
% 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(AddressBookSource)  
Created PreventUserIdleSystemSleep "Address Book Source Sync"

6/6/12 8:52:16 AM PDT  Assertions  PID 5681(AddressBookSource)  
Released PreventUserIdleSystemSleep "Address Book Source Sync"
% 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(AddressBookS sour)
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 x 2h = 200 Wh
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

Less power = More time
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

Less power = More time

Power

Time
Rationale

Battery life

• Make energy last!
• “0.1% CPU” can raise idle power 10%
  ▪ Costs a Mac 30–45 min of battery life

Less power = More time

Power

Time
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

Less power = More time
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

<table>
<thead>
<tr>
<th>Power</th>
<th>0 W</th>
<th>500 W</th>
<th>1000 W</th>
<th>1500 W</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00 AM</td>
<td>4:00 AM</td>
<td>8:00 AM</td>
<td>12:00 PM</td>
<td>4:00 PM</td>
</tr>
</tbody>
</table>

- Heating for 2 mins @ 1500W
- Constant clock power 3W
Microwave Energy

Energy = power x time

- Heating for 2 mins @ 1500W
- Constant clock power 3W
Microwave Energy
1500W x 6m + 3W x 1440m

Heating for 2mins @ 1500W

Constant clock power 3W

Energy

Time
12:00 AM 4:00 AM 8:00 AM 12:00 PM 4:00 PM 8:00 PM

Power

0 W 500 W 1000 W 1500 W

Energy

0 Wh 75 Wh 150 Wh 225 Wh
Microwave Energy
1500W x 6m + 3W x 1440m
Microwave Energy

1500W x 6m + 3W x 1440m

Power

Energy

500:1 Power Ratio

~2:1 Energy Ratio
Microwave Energy
1500W x 6m + 3W x 1440m

Power

Energy

500:1 Power Ratio

~2:1 Energy Ratio
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

```c
while(sleep(1)) {
    if (workAvailable) {
        launchWorkOnHelperThread();
        while(!workDone) {
            nanosleep(<100ms>);
        }
        workIsDone();
    }
}
```
Absolute Idle
CPU wakes expensive

\[ \text{if (workAvailable) \ldots} \quad \text{while(!workDone) \ldots} \]

Time

Power
Absolute Idle

CPU wakes expensive

if (workAvailable) …

while(!workDone) …

Useful work vastly outweighed by wake-up/idle costs

Power

cpu power

Time
Absolute Idle
CPU wakes expensive

if (workAvailable) …
while(!workDone) …

Useful work vastly outweighed by wake-up/idle costs

High frequencies inhibit platform idle

Memory Controller/Caches
RAM, Buses, Clocks…

Baseline Idle Power

Time

Power

cpu power
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
$ 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
...
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? ...]`
Eliminating Idle Energy Leaks

sc_usage(8) collates system calls

$ sudo sc_usage -e <target>

<table>
<thead>
<tr>
<th>TYPE</th>
<th>NUMBER</th>
<th>CPU_TIME</th>
<th>WAIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Idle</td>
<td></td>
<td>00:08.150(00:00.954)</td>
</tr>
<tr>
<td>System</td>
<td>Busy</td>
<td></td>
<td>00:01.889(00:00.055)</td>
</tr>
<tr>
<td>&lt;target&gt;</td>
<td>Usermode</td>
<td></td>
<td>00:00.000</td>
</tr>
<tr>
<td>mk_timer_arm</td>
<td>28(4)</td>
<td>00:00.000</td>
<td></td>
</tr>
<tr>
<td>psynch_cvwait</td>
<td>20(2)</td>
<td>00:00.000</td>
<td>00:18.229(00:02.044)</td>
</tr>
<tr>
<td>mach_msg_trap</td>
<td>19</td>
<td>00:00.000</td>
<td>00:05.066</td>
</tr>
<tr>
<td>kevent</td>
<td>12</td>
<td>00:00.000</td>
<td>00:05.994</td>
</tr>
<tr>
<td>gettimeofday</td>
<td>6(1)</td>
<td>00:00.000</td>
<td></td>
</tr>
<tr>
<td>workq_kernreturn</td>
<td>3</td>
<td>00:00.000</td>
<td>00:08.049</td>
</tr>
</tbody>
</table>
Eliminating Idle Energy Leaks

sc_usage(8) collates system calls

$ sudo sc_usage -e <target>

<table>
<thead>
<tr>
<th>TYPE</th>
<th>NUMBER</th>
<th>CPU_TIME</th>
<th>WAIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Idle</td>
<td></td>
<td>00:08.150(00:00.954)</td>
</tr>
<tr>
<td>System</td>
<td>Busy</td>
<td></td>
<td>00:01.889(00:00.055)</td>
</tr>
<tr>
<td>&lt;target&gt;</td>
<td>Usermode</td>
<td></td>
<td>00:00.000</td>
</tr>
<tr>
<td>mk_timer_arm</td>
<td>28(4)</td>
<td>00:00.000</td>
<td></td>
</tr>
<tr>
<td>psynch_cvwait</td>
<td>20(2)</td>
<td>00:00.000</td>
<td>00:18.229(00:02.044)</td>
</tr>
<tr>
<td>mach_msg_trap</td>
<td>19</td>
<td>00:00.000</td>
<td>00:05.066</td>
</tr>
<tr>
<td>kevent</td>
<td>12</td>
<td>00:00.000</td>
<td>00:05.994</td>
</tr>
<tr>
<td>gettimeofday</td>
<td>6(1)</td>
<td>00:00.000</td>
<td></td>
</tr>
<tr>
<td>workq_kernreturn</td>
<td>3</td>
<td>00:00.000</td>
<td>00:08.049</td>
</tr>
</tbody>
</table>
Eliminating Idle Energy Leaks

sc_usage(8) collates system calls

$ sudo sc_usage -e <target>

<table>
<thead>
<tr>
<th>TYPE</th>
<th>NUMBER</th>
<th>CPU_TIME</th>
<th>WAIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Idle</td>
<td></td>
<td>00:08.150</td>
<td>(00:00.954)</td>
</tr>
<tr>
<td>System Busy</td>
<td></td>
<td>00:01.889</td>
<td>(00:00.055)</td>
</tr>
<tr>
<td>&lt;target&gt; Usermode</td>
<td>00:00.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mk_timer_arm</td>
<td>28(4)</td>
<td>00:00.000</td>
<td></td>
</tr>
<tr>
<td>psynch_cvwait</td>
<td>20(2)</td>
<td>00:00.000</td>
<td>00:18.229(00:02.044)</td>
</tr>
<tr>
<td>mach_msg_trap</td>
<td>19</td>
<td>00:00.000</td>
<td>00:05.066</td>
</tr>
<tr>
<td>kevent</td>
<td>12</td>
<td>00:00.000</td>
<td>00:05.994</td>
</tr>
<tr>
<td>gettimeofday</td>
<td>6(1)</td>
<td>00:00.000</td>
<td></td>
</tr>
<tr>
<td>workq_kernreturn</td>
<td>3</td>
<td>00:00.000</td>
<td>00:08.049</td>
</tr>
</tbody>
</table>
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'
Extreme Efficiency

- Faster completion \( \sim \) 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

```cpp
#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

Numerical Recipes in C
Extreme Efficiency
Processing on a single core

Power @ Battery

Processing time $t$

cpu0

Memory Controller/Caches

RAM, Buses, Clocks…

Baseline Idle Power

Time
Extreme Efficiency
Processing on all cores

- Baseline Idle Power
- Processing time $\sim t/4$
- Memory Controller/Caches
- RAM, Buses, Clocks…

Power vs. Time Graph
Rendering Frames

How many fps do you need?

- 24, 30, or 60 Hz
  - Use CVDisplayLink to sync with 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

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Instruments</td>
<td>Presidio</td>
<td>Wednesday 4:30PM</td>
</tr>
<tr>
<td>The Accelerate Framework</td>
<td>Russian Hill</td>
<td>Thursday 10:15AM</td>
</tr>
<tr>
<td>iOS App Performance: Responsiveness</td>
<td>Presidio</td>
<td>Thursday 11:30AM</td>
</tr>
<tr>
<td>iOS App Performance: Graphics and Animations</td>
<td>Presidio</td>
<td>Thursday 3:15PM</td>
</tr>
</tbody>
</table>
## Labs

<table>
<thead>
<tr>
<th>Lab</th>
<th>Time</th>
</tr>
</thead>
</table>
| 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