Linux Suspend/Resume…
…at the Speed of Light

Len Brown, Principal Engineer,
Intel Open Source Technology Center

19-Aug, 2015
LinuxCon North America/Linux Plumbers Conference
Seattle, WA

Acknowledgements

Todd Brandt – analyze_suspend.py maintainer
Rafael Wysocki – suspend/resume maintainer

Agenda

Concepts

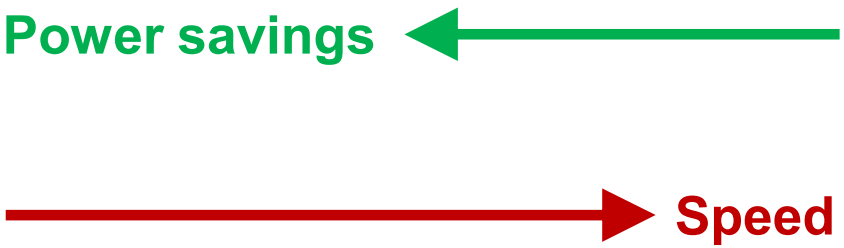
Tools

Results

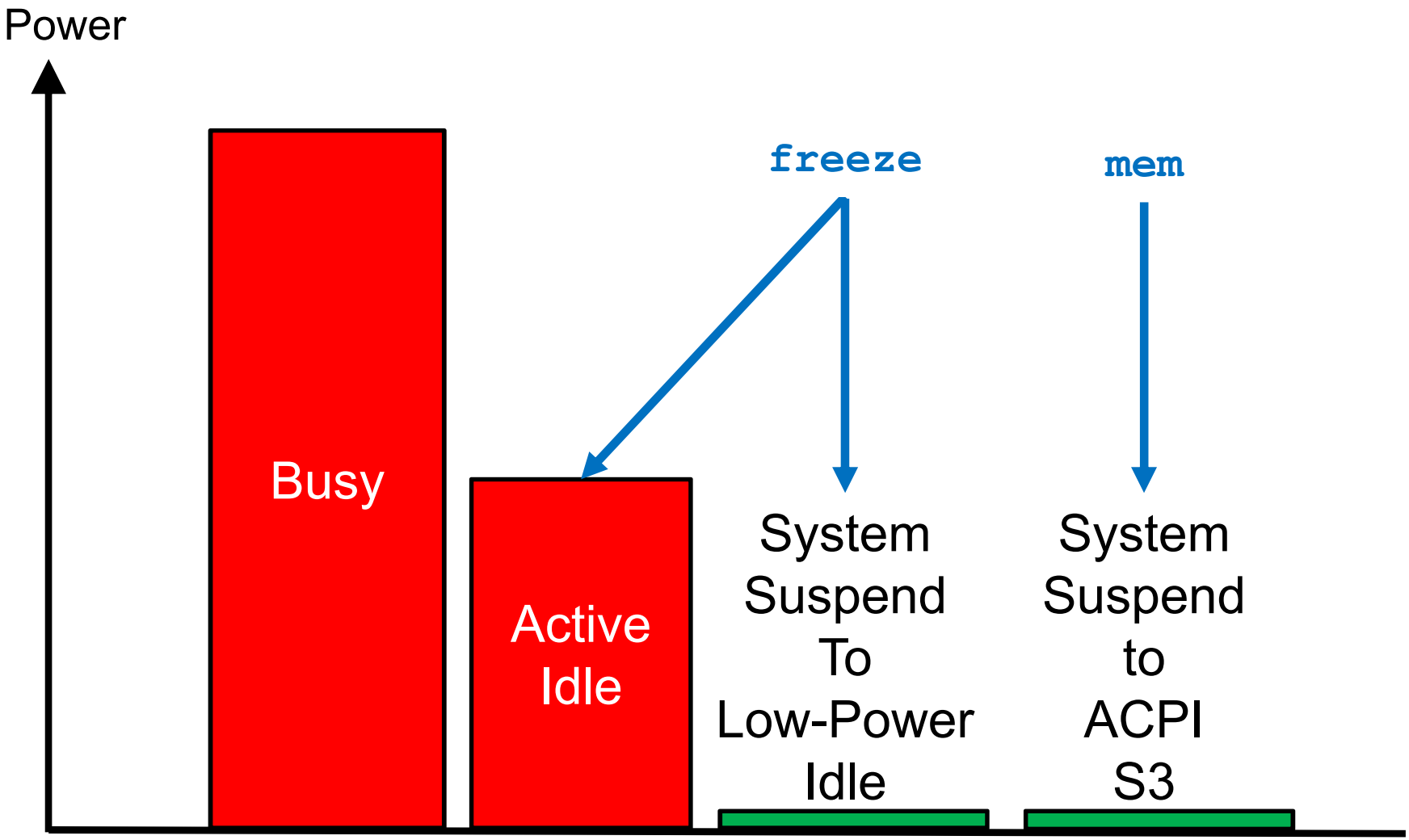
Future

Linux Suspend Types

```
$ cat /sys/power/state
disk mem standby freeze
```



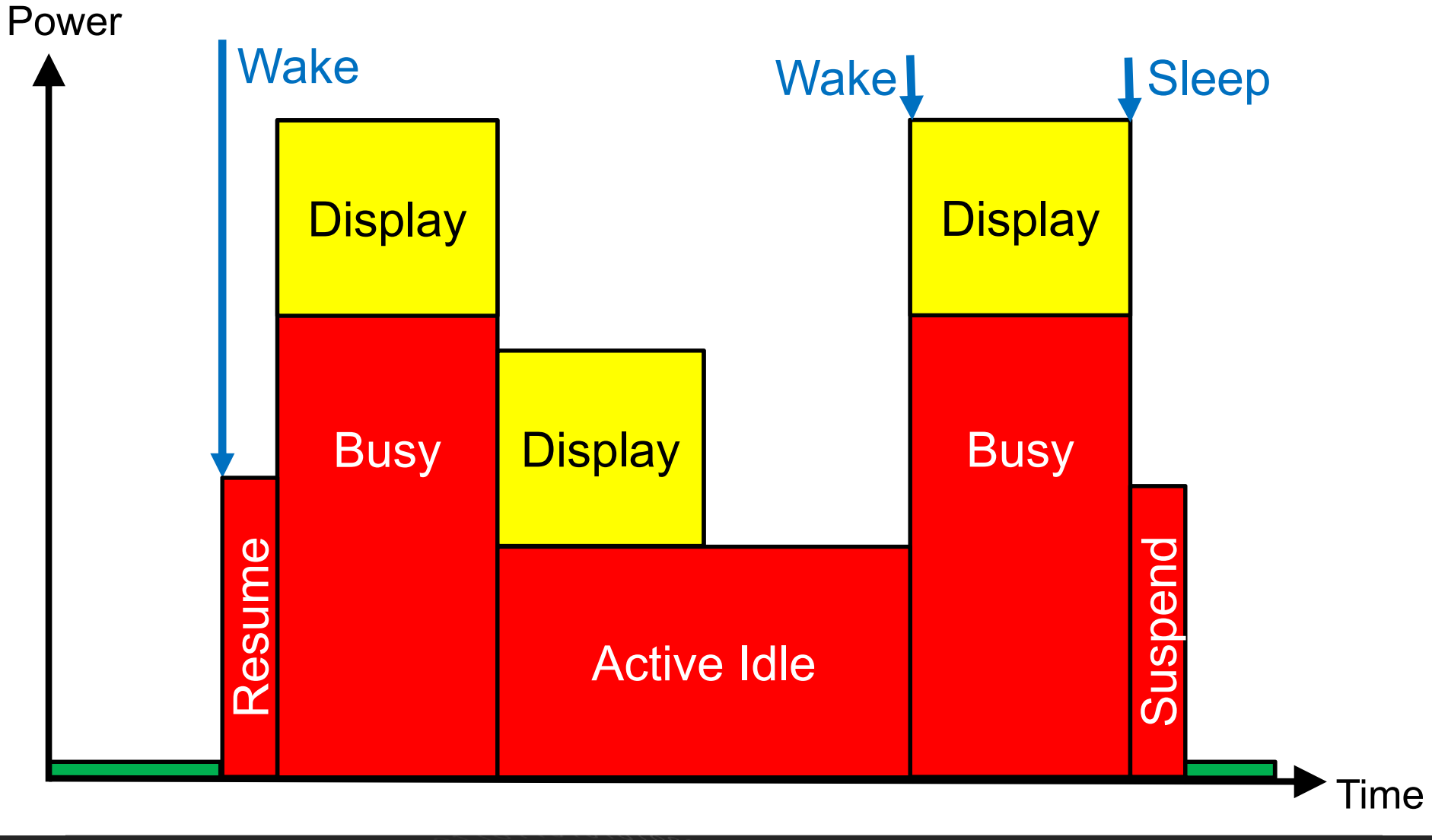
Saving Power with System Suspend



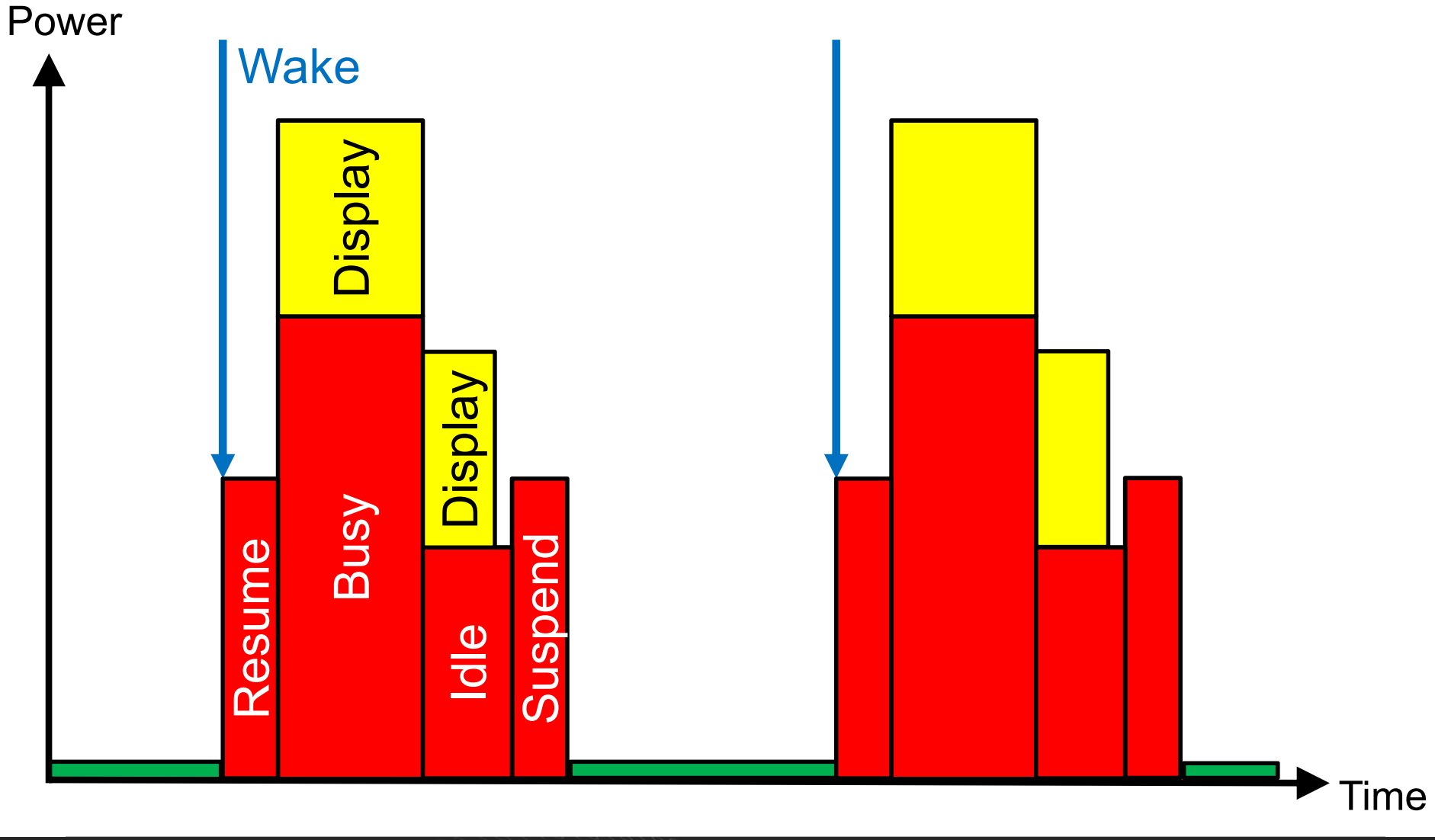
Suspend Trade-Offs

	Busy	Active Idle	Suspend to Low-Power Idle	Suspend to ACPI S3
Power				
Latency				
Applications				
Display				
Network				

Interactive Laptop Scenario



Interactive Handheld Scenario

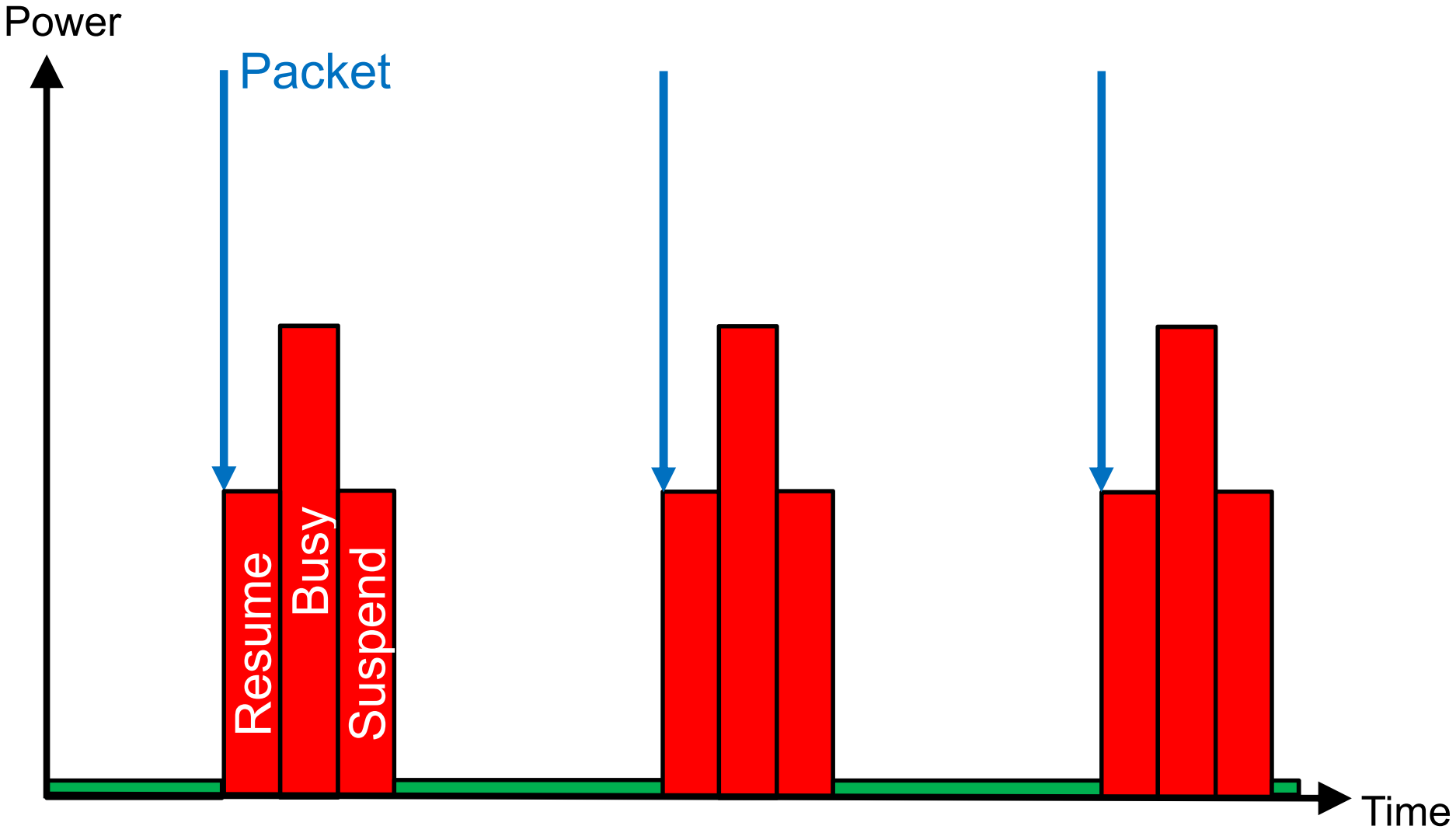


Laptop/Handheld Interactive Challenge

Resume “instantly”

Suspend “fast enough” to allow use of suspend, be energy efficient

Dark Resume Scenario



Dark Resume Challenge

Resume Latency = packet latency

Suspend + Resume time may exceed Active time

Suspended battery life depends directly on suspend & resume performance

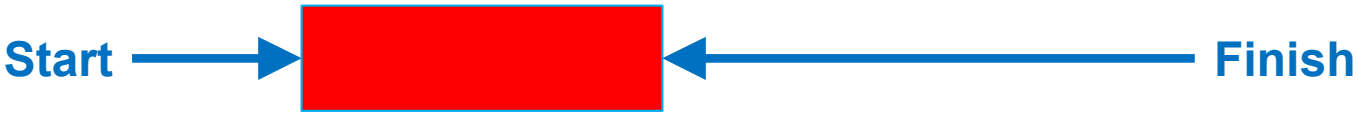
Going Faster

Going Faster (strategy 1)

Less work, or less waiting

Before:

After:

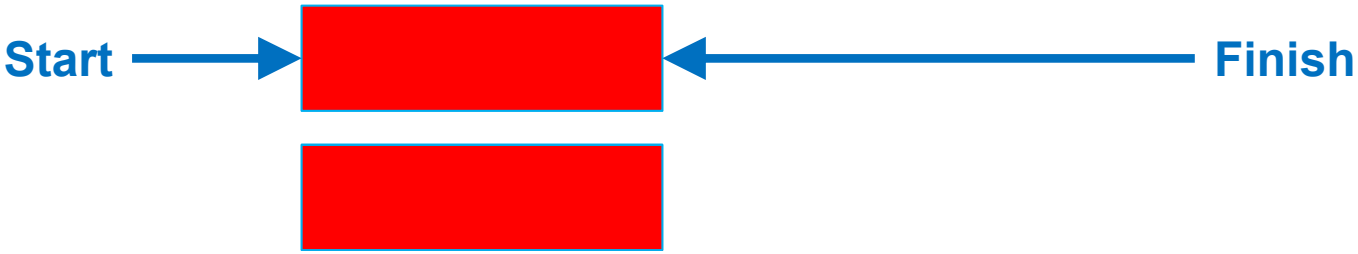


Going Faster (strategy 2)

Same work, in parallel, but still synchronous

Before:

After:



Going Faster (strategy 3)

Same work, asynchronous

Before:

After:

Going Faster (strategy 4)

Avoid work entirely

Before:

After:

Agenda

Concepts

Tools

Results

Future

Measuring Suspend Speed

Method 1: Use external measuring device

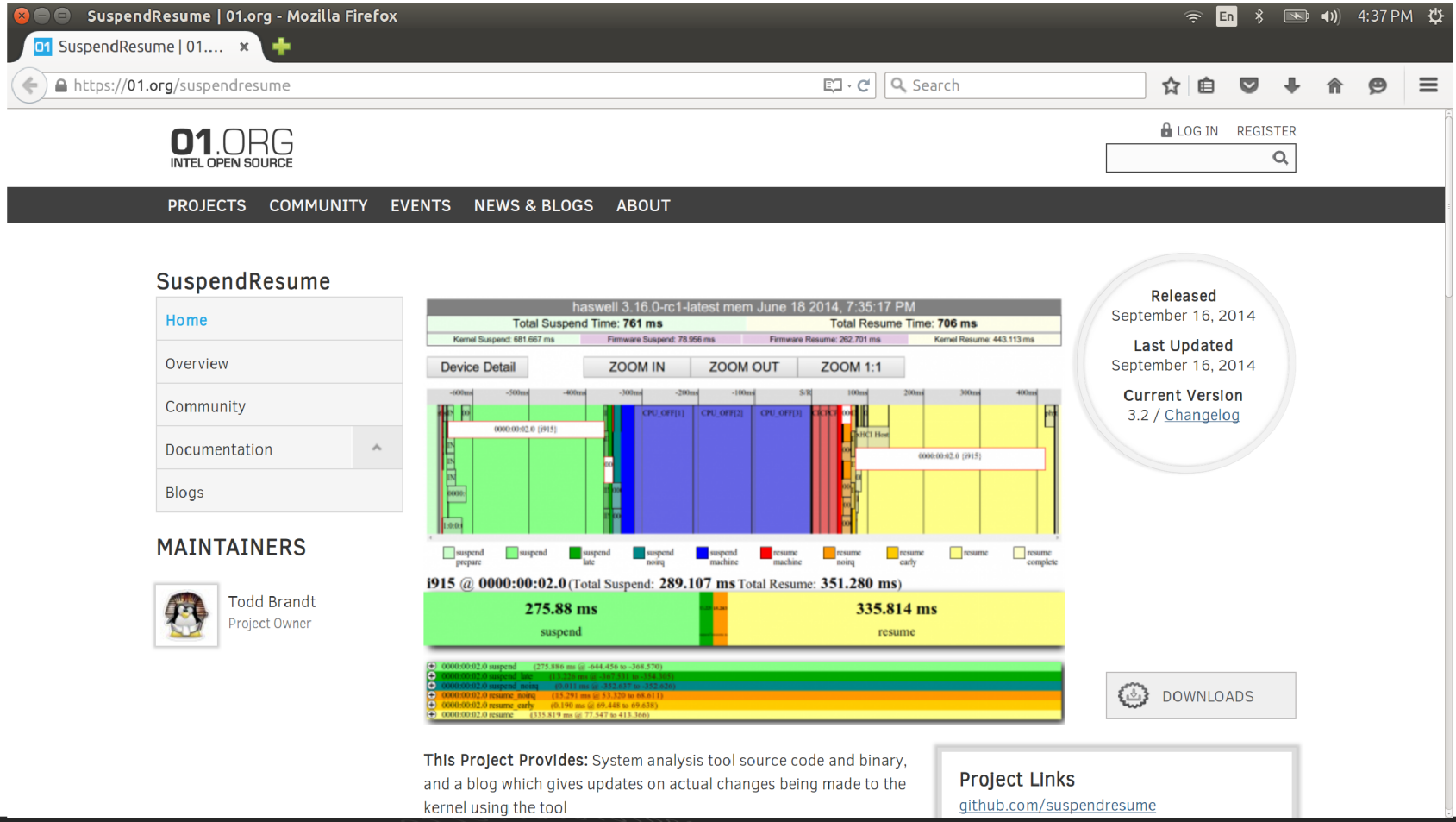
Measuring Suspend Speed

Method 2: Boot with “initcall_debug”, examine msgbuf

```
$ dmesg | grep call
…
[  661.392498] calling  phy0+ @ 2367, parent: 0000:07:00.0
[  661.417798] call phy0+ returned 0 after 24721 usecs
```

Measuring Suspend Speed

Method 3: Run analyze_suspend



Download and run analyze_suspend.py

```
$ git clone https://github.com/01org/suspendresume.git
$ cd suspendresume
$ sudo ./analyze_suspend.py
```

Generates output files in subdirectory: suspend-yymmdd-HHMMSS

HTML output:	<hostname>_<mode>.html
raw dmesg output:	<hostname>_<mode>_dmesg.txt
raw ftrace output:	<hostname>_<mode>_ftrace.txt

```
$ firefox suspend*/*.html
```

analyze_suspend.py -h

New script can re-analyze output of previous measurement

“initcall_debug” and dmesg used up through Linux 3.16, ftrace there-after

```
[general]
   -h          Print this help text
   -v          Print the current tool version
   -verbose    Print extra information during execution and analysis
   -status     Test to see if the system is enabled to run this tool
   -modes      List available suspend modes
   -m mode     Mode to initiate for suspend ['freeze', 'mem', 'disk'] (default: mem)
   -rtcwake t  Use rtcwake to autoresume after <t> seconds (default: disabled)
   -o subdir   Override the output subdirectory
   -addlogs    Add the dmesg and ftrace logs to the html output
 [advanced]
   -srgap      Add a visible gap in the timeline between sus/res (default: disabled)
   -f          Use ftrace to create device callgraphs (default: disabled)
   -filter "d1 d2 ..." Filter out all but this list of dev names
   -x2         Run two suspend/resumes back to back (default: disabled)
   -x2delay t  Minimum millisecond delay <t> between the two test runs (default: 0 ms)
   -postres t  Time after resume completion to wait for post-resume events (default: 0 S)
   -multi n d  Execute <n> consecutive tests at <d> seconds intervals. The outputs will
               be created in a new subdirectory with a summary page.
 [utilities]
   -fpdt       Print out the contents of the ACPI Firmware Performance Data Table
   -usbtopo    Print out the current USB topology with power info
   -usbauto    Enable autosuspend for all connected USB devices
 [re-analyze data from previous runs]
   -ftrace ftracefile  Create HTML output using ftrace input
   -dmesg dmesgfile    Create HTML output using dmesg (not needed for kernel >= 3.15)
   -summary directory  Create a summary of all test in this dir
```

Agenda

Concepts

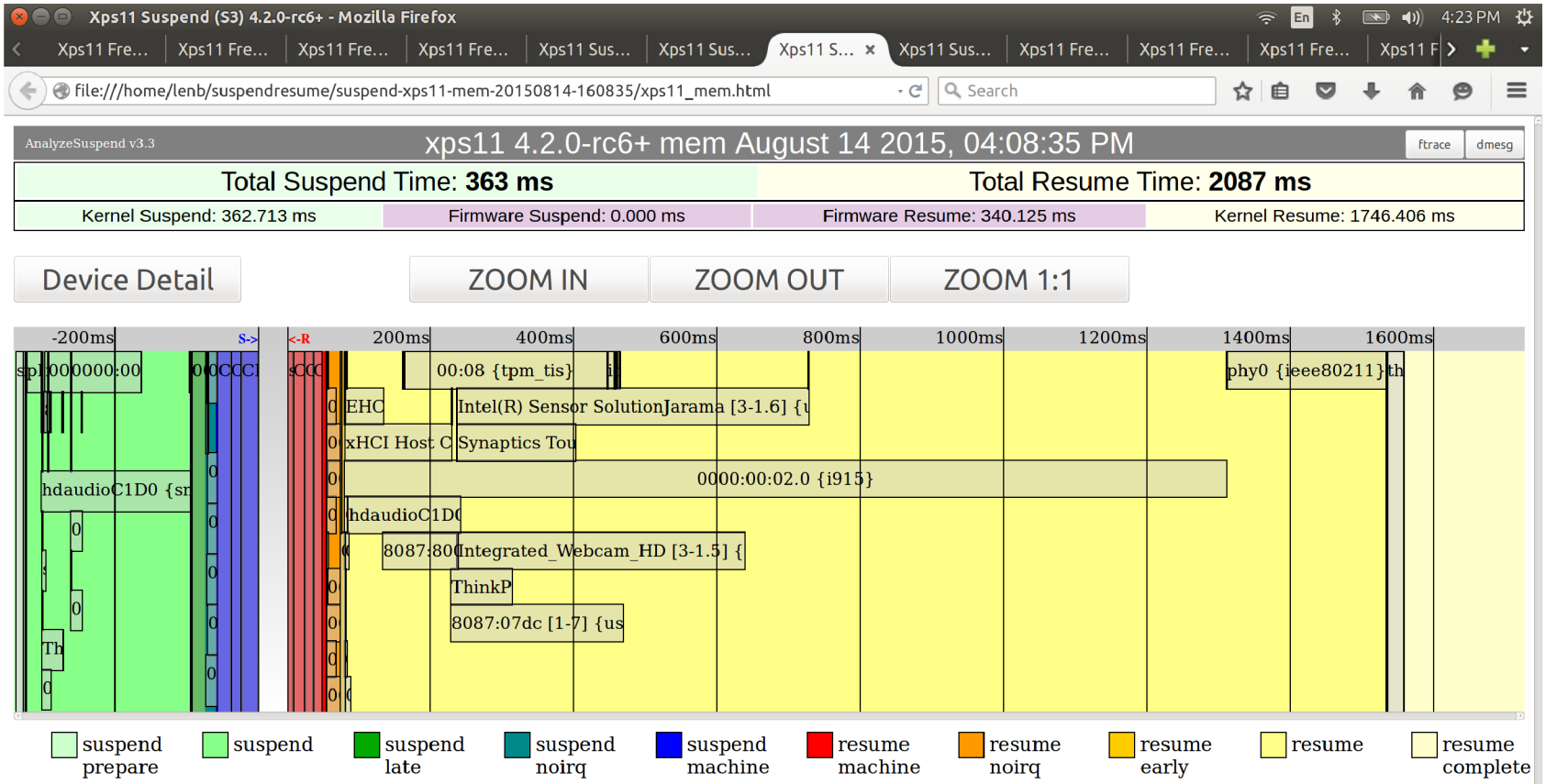
Tools

Results

Future

Suspend/Resume (mem)

ACPI S3: Firmware resume = 340ms
Display on: i915 resume > 1200ms



How to force Dark Suspend/Resume

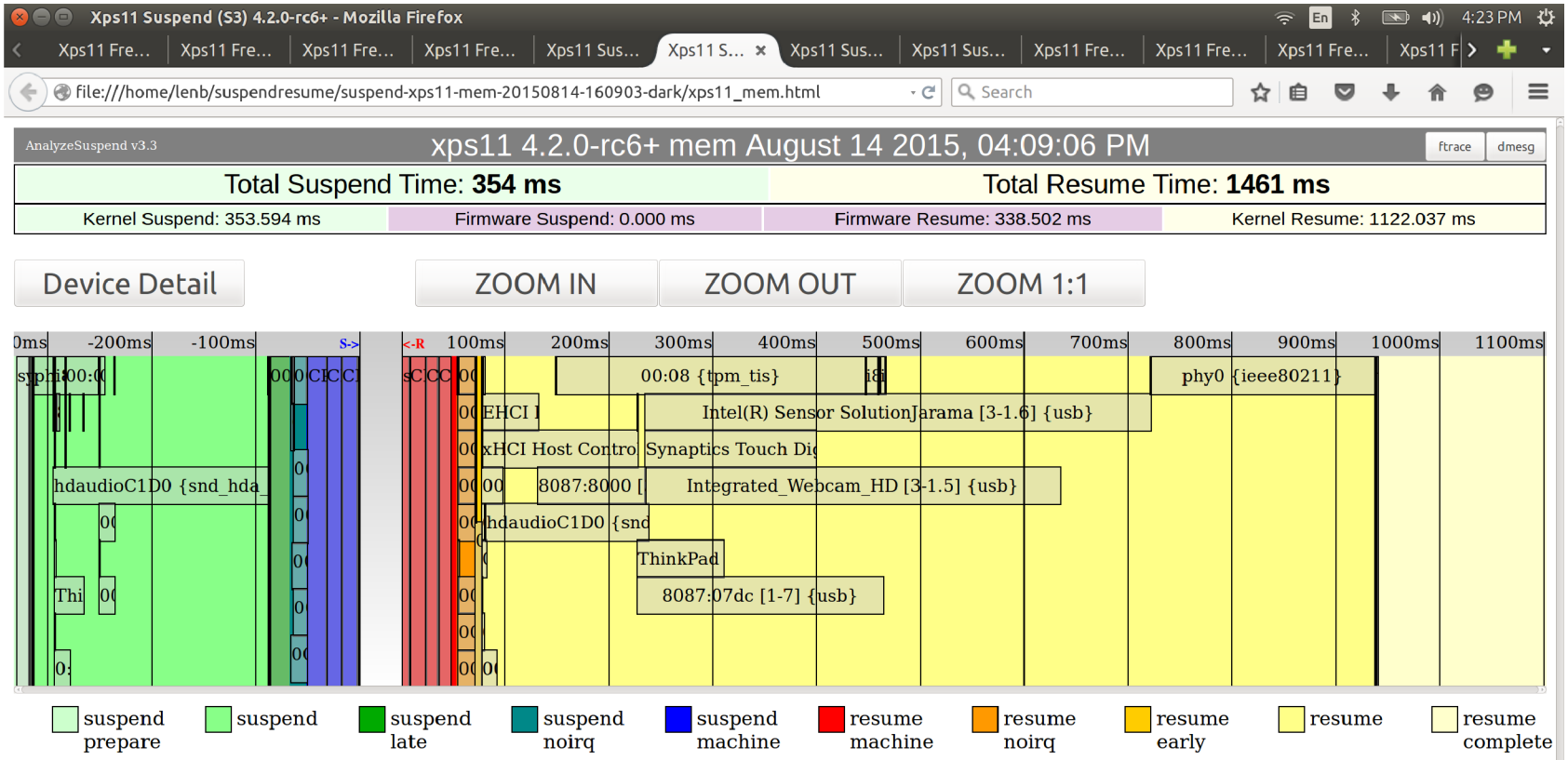
Run-time suspend display before system-suspend:

```
$ xset -display :0 dpms force off
$ sleep 2
$ sudo analyze_suspend.py
```

Display will not be resume upon system resume,
but reliability is platform dependent…

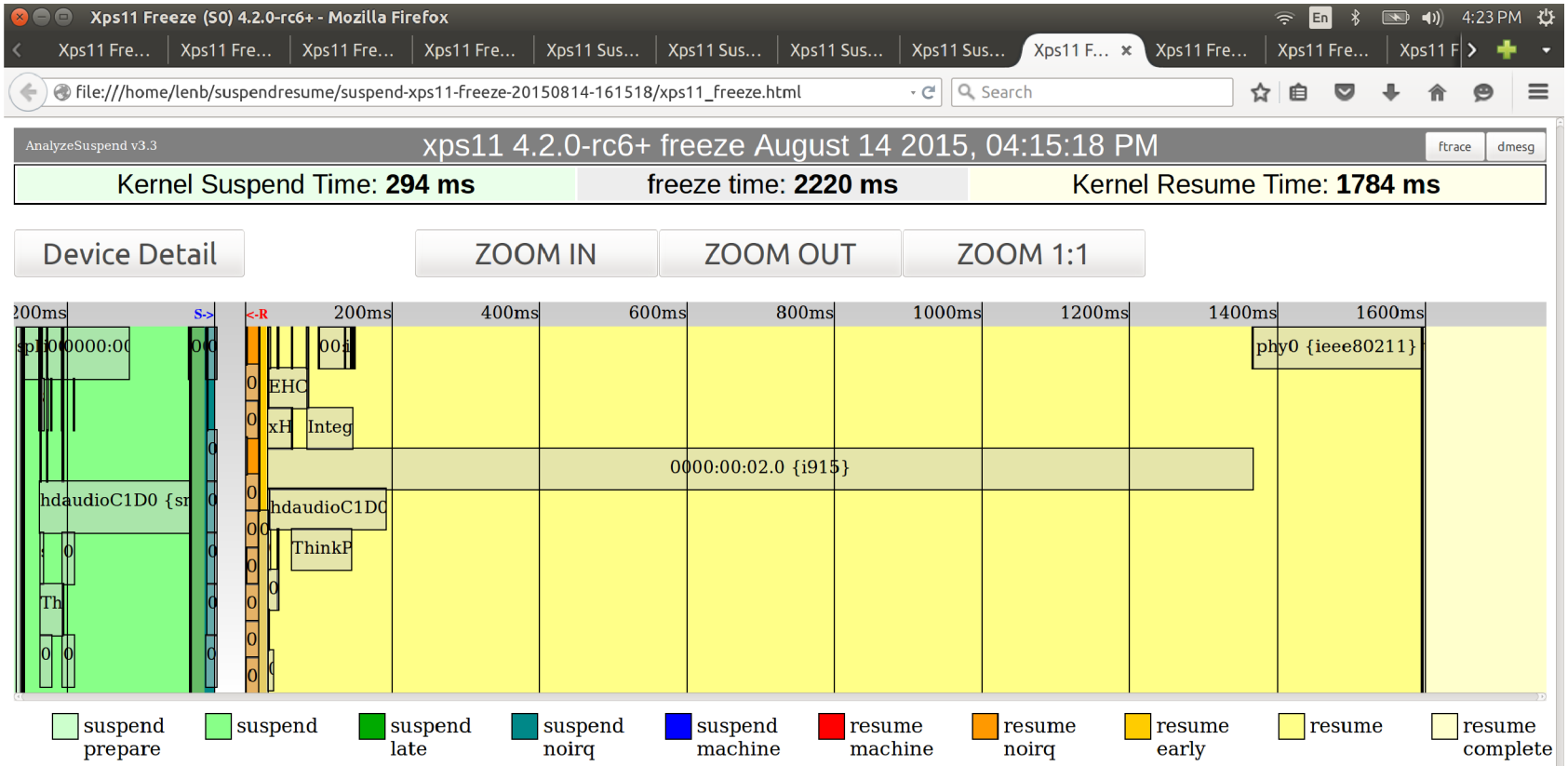
Dark Suspend/Resume (mem)

ACPI S3: Firmware resume = 340ms
Display OFF



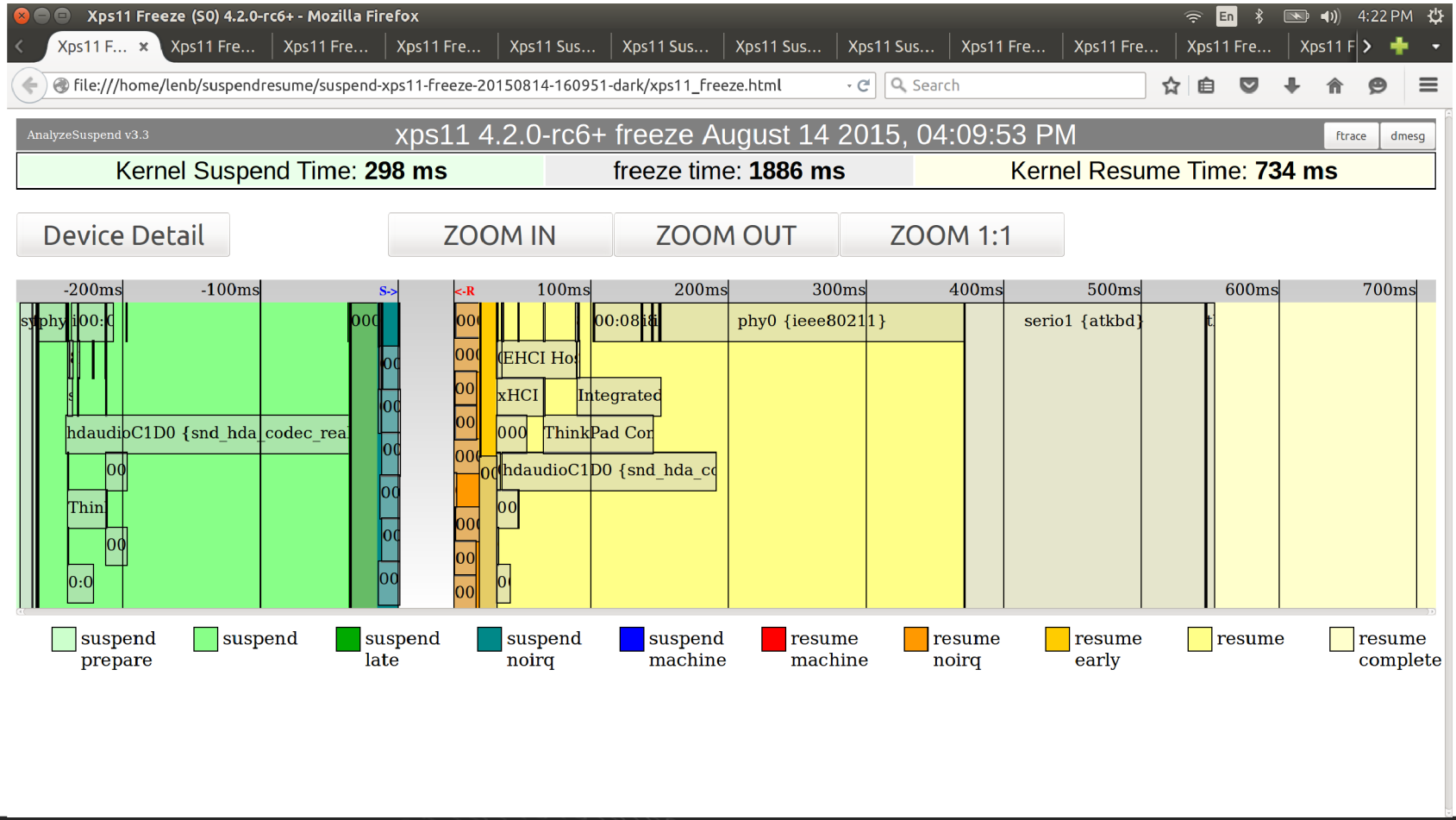
Suspend/Resume (freeze)

Firmware resume = 0
Display on: i915 resume > 1200ms



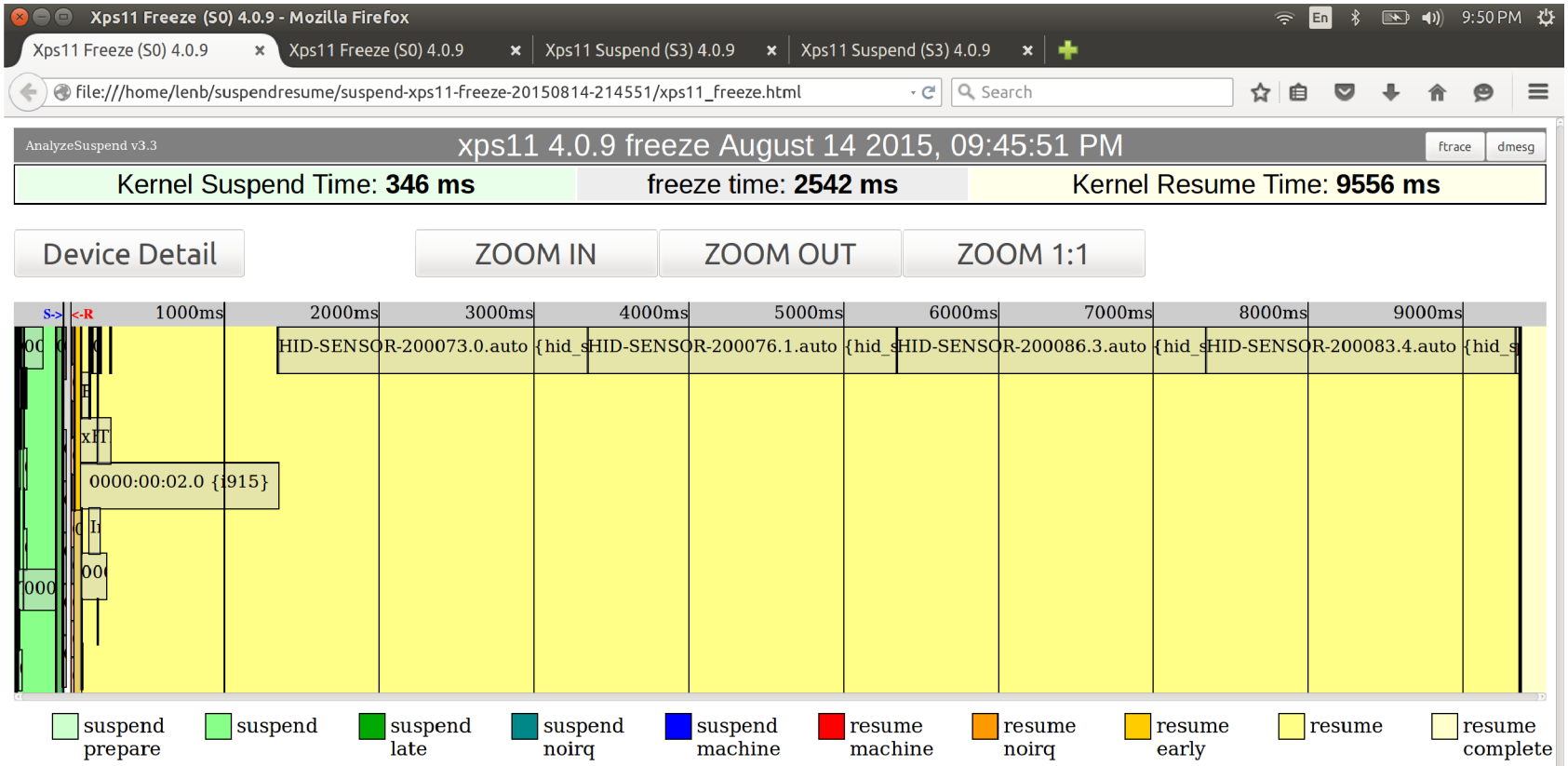
Dark Suspend/Resume (freeze)

Firmware resume = 0
Display OFF



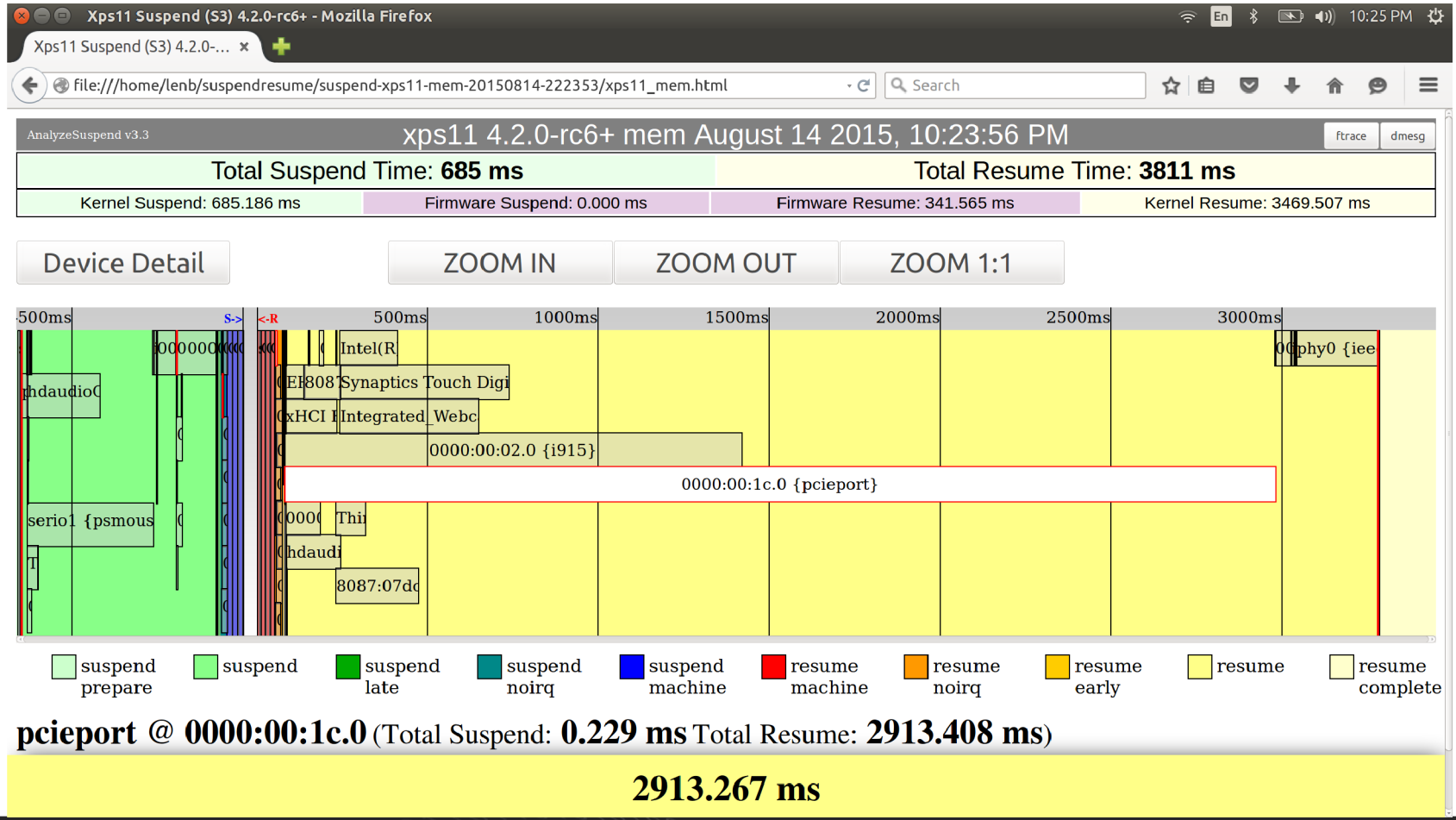
Things can go very wrong

Linux-4.0 sensors regression – no workaround
Fixed in Linux-4.2, Linux-4.1-stable. Not fixed in Linux-4.0-stable.



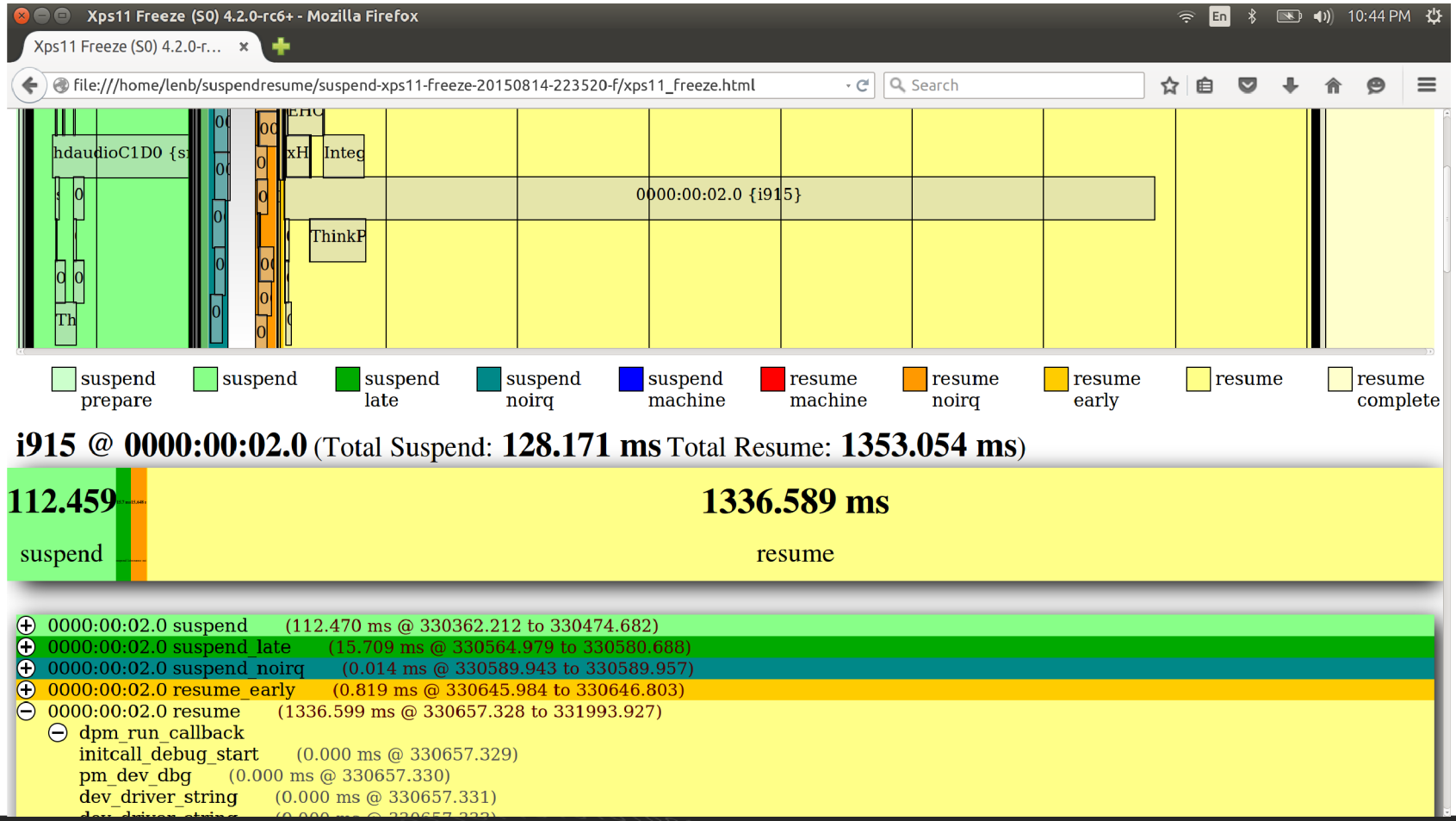
Things can go very wrong

pcieport resume 2900ms (https://bugzilla.kernel.org/show_bug.cgi?id=99751)
Workaround: boot with “pcie_ports=compat”



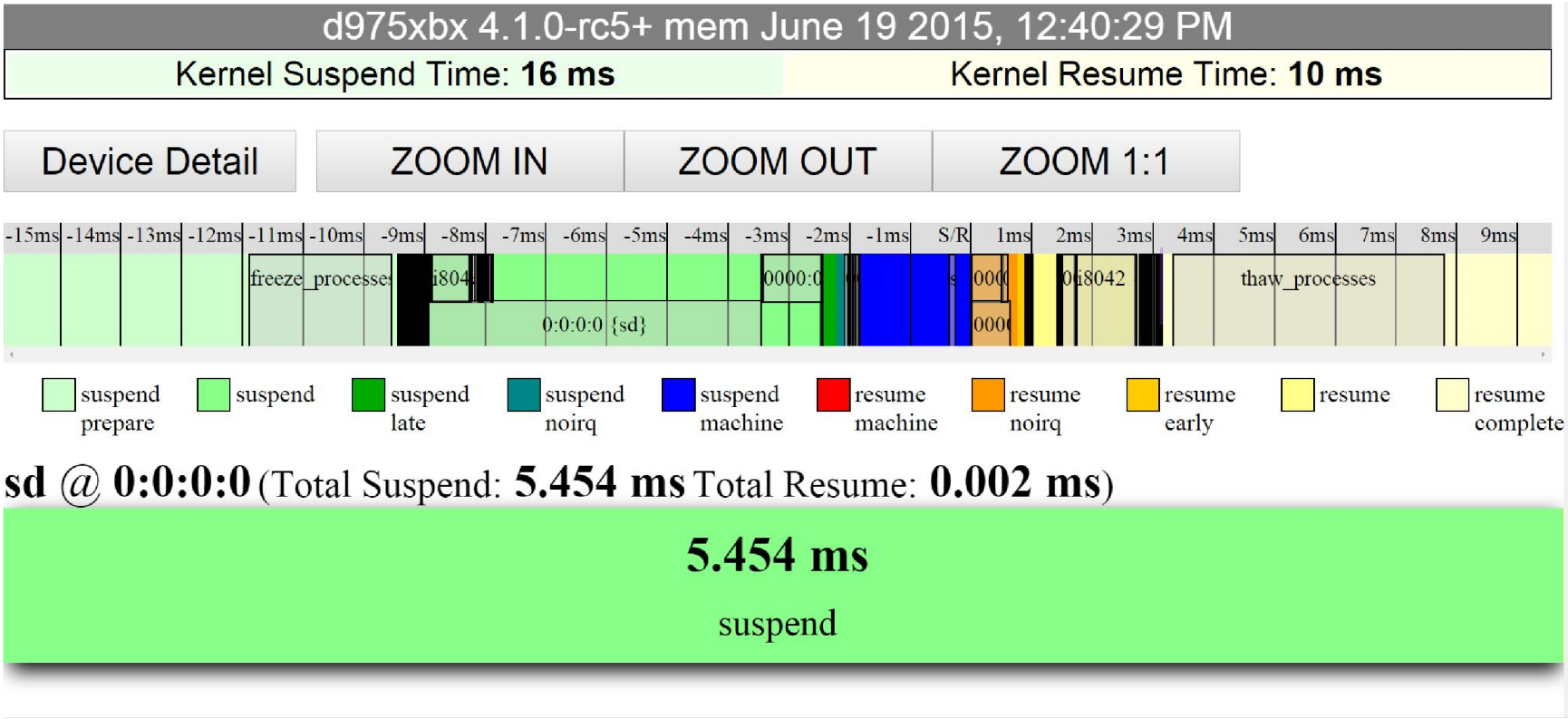
analyze_suspend -f

Captures full ftrace call graph, parses in HTML GUI
{HTML file size ~ 64MB}



regarding the “speed of light”

O(25ms) to suspend and wake on this stripped-down Core2 desktop
UP, no sync, no GFX, serial console, no network, no USB, SSD drive, yes ACPI, but not FPDT



Agenda

Concepts

Tools

Results

Future

What to do?

Run on more systems – help us!

- Prevent regressions
- Discover, report, fix more issues

Display, USB, Network, Audio

- When run-time suspended, keep it suspended
- When must resume, go asynchronous
- Optimize actual resume latency

Wireless network re-association speed


Q & A

Linux “freeze” History

Linux-3.18:

functional for 1st time, including wakeup

Linux-4.0:

freeze timers, improves deep idle-state residency

Key Patches

ATA drives can take multiple SECONDS to resume

This patch makes that ASYNCHRONOUS, not blocking the resume path to user-space

In Linux v3.15-rc1:

```
commit 200421a80f6e0a9e39d698944cc35cba103eb6ce
Author: Todd Brandt <todd.e.brandt@linux.intel.com>
Date:   Fri Mar 14 13:52:54 2014 -0700

libata: async resume
```

Key Patches

Fix race condition in resume_complete
(boot with “no_console_suspend” may workaround)

In Linux v4.2-rc1:

```
commit 32e8d689dc12e29fcb6ba9c65a33473d0cbdfec8
Author: Todd E Brandt <todd.e.brandt@linux.intel.com>
Date:   Thu May 28 12:55:53 2015 -0700

PM / sleep: trace_device_pm_callback coverage in dpm_prepare/complete
```

Key Patches

Fix race condition in resume_complete
(boot with “no_console_suspend” may workaround)

Linux 4.0 regression

Fixed in Linux v4.2, v4.1.4; NOT fixed in Linux 4.0-stable

```
4.2-rc3 commit 1e25aa9641e8f3fa39cd5e46b4afcafd7f12a44b
4.2-rc4 commit 88cc7b4eee1e7b9bca1a64dae5adaa044cf72312
4.1.4 commit be43d21df90d10f5f10252c114f5fb024b7ba5ae

Author: Srinivas Pandruvada <srinivas.pandruvada@linux.intel.com>
Date:   Mon Jun 1 16:36:27 2015 -0700

hid-sensor: Fix suspend/resume delay
```

https://bugzilla.kernel.org/show_bug.cgi?id=102891