

Integration of PM-runtime with System-wide Power Management

Rafael J. Wysocki

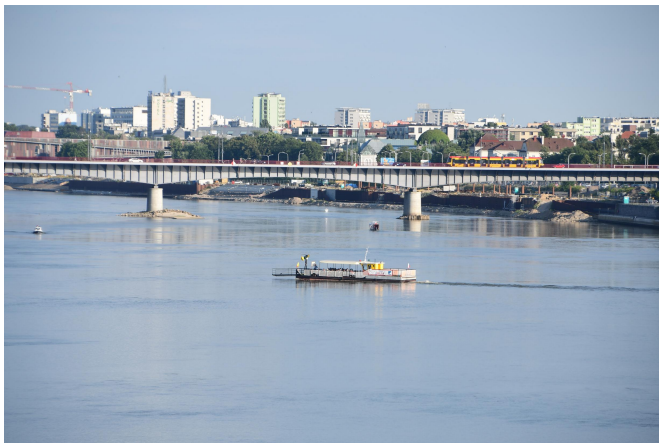
Intel Linux Systems Engineering

September 10, 2019

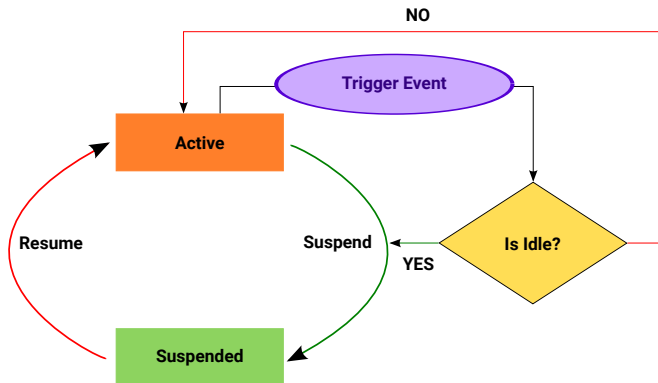
It Is All About Energy-efficiency



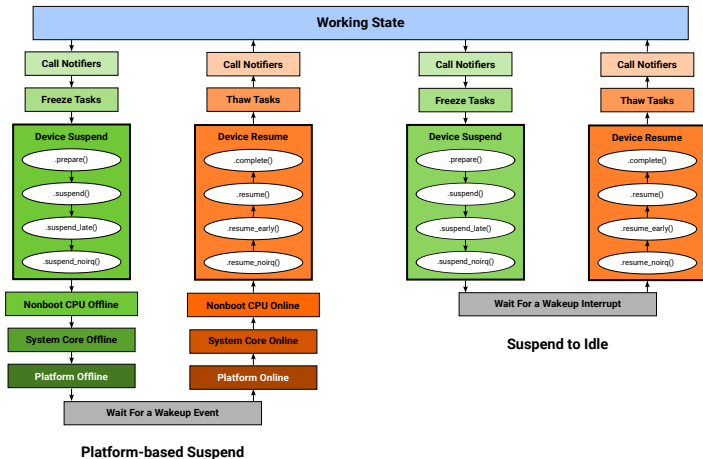
Two Different Ways To Get There



High-level View to PM-runtime



System Suspend Control Flows



Transparent Handling of Suspended Devices

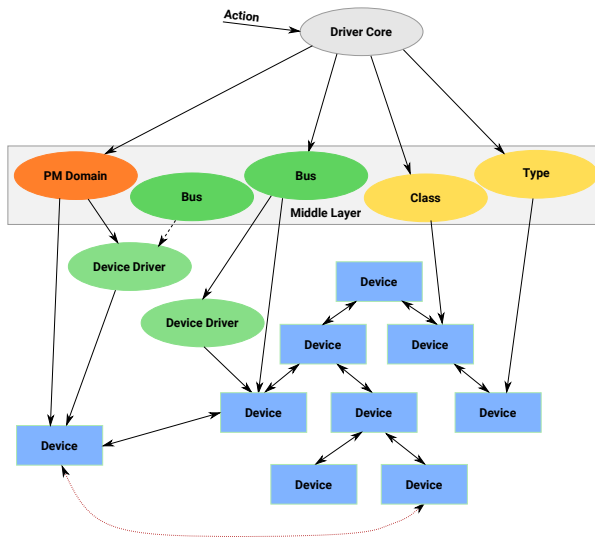


PM-runtime Callbacks Re-Use



Intel
OpenSource
TECHNOLOGY CENTER

Driver Core and Power Management



Devices May Need To Be Reconfigured



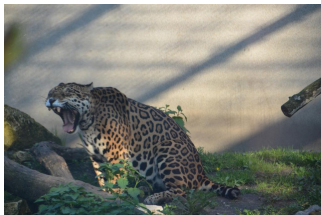
Differences Related To Wakeup

PM-runtime

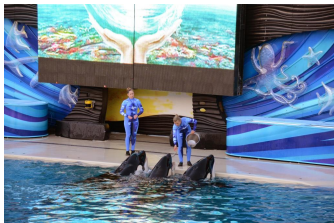
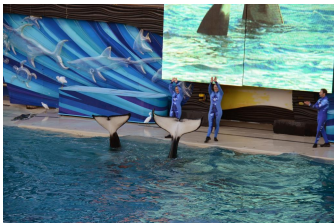
Device wakeup always **enabled**.

System-wide PM

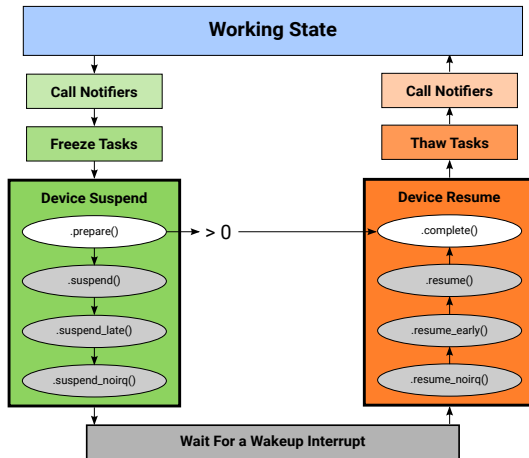
`/sys/devices/.../power/wakeup`: **enabled** or **disabled**.



Intermittent Interactions Between Devices

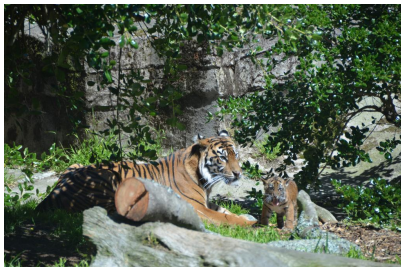


Direct-complete Optimization Idea

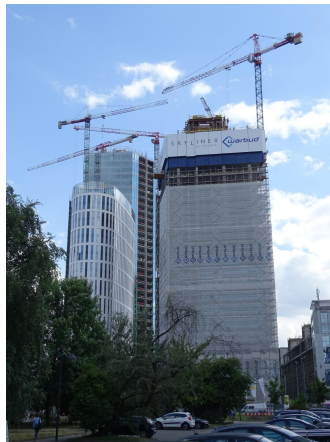


Limitations Of Direct-complete

- 1 Devices subject to intermittent interactions cannot do it.
- 2 Cannot be done if the children do not do it too.



Wrappers Around PM-runtime Callbacks



Limitations Of Callback Wrappers

- 1 Invoke middle-layer PM-runtime callbacks.
- 2 Disable PM-runtime.
- 3 Questionable approach to leaving devices in suspend.



Driver Flags For System-wide Power Management

- DPM_FLAG_NEVER_SKIP
- DPM_FLAG_SMART_PREPARE
- DPM_FLAG_SMART_SUSPEND
- DPM_FLAG_LEAVE_SUSPENDED



Coverage Gaps

- ① Runtime resume resulting from intermittent interactions not covered.
- ② DPM_FLAG_SMART_SUSPEND (generally) required for callback re-use.



PM-runtime And System-wide PM: Observations

Observation 1

Devices cannot be runtime-suspended during system-wide suspend/resume.

Observation 2

Two cases for runtime resume during system-wide PM transitions:

- Called from system-wide PM callbacks for the same device.
- Called from somewhere else.

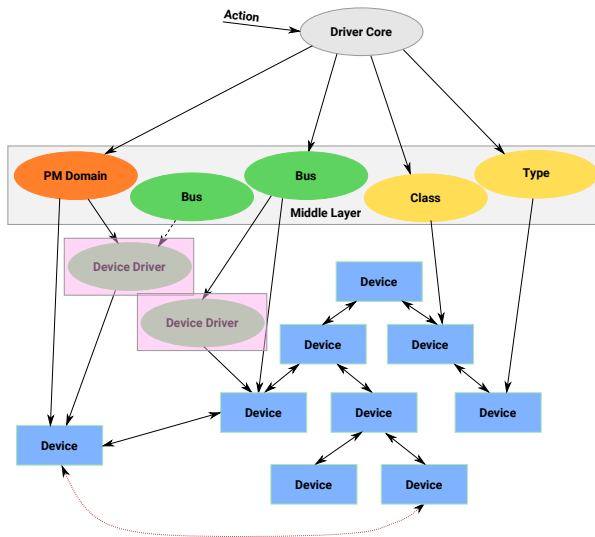


Runtime Resume During System-wide Suspend

- ① If called from system-wide PM callbacks for the same device, run it.
- ② Otherwise:
 - If called after some system-wide PM code has run, block it.
 - Else (always) defer suspending the target device to the “noirq” phase.



Generally, Callback Wrappers Do Not Help



Plan: Eliminate The Reason For Using Callback Wrappers

Concern addressed by them

Prevent the same callback from running twice in a row back to back.

There are cases in which they are not necessary already

- PM-aware bus types (PCI, USB).
- ACPI PM domain.

What can be done elsewhere

- 1 Make all PM-aware middle-layer code honor PM-runtime status.
- 2 For pass-through middle layers, make the PM core do that.

Comments, Questions, Concerns?



References



Rafael J. Wysocki, *Power Management Challenges in Linux* (https://www.linuxplumbersconf.org/2017/ocw//system/presentations/4652/original/linux_pm_challenges.pdf).



Rafael J. Wysocki, *PM Infrastructure in the Linux Kernel – Current Status and Future* (https://events.linuxfoundation.org/sites/events/files/slides/kernel_PM_infra_0.pdf).



Greg Kroah-Hartman, *Linux Driver Model* (<https://www.youtube.com/watch?v=AdPxeGHIZ74>).

Disclaimer

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

© Intel Corporation