Kernel Power Management Development 2010/2011

Rafael J. Wysocki

Faculty of Physics U. Warsaw / SUSE Labs

August 27, 2011

2.6.35 – present

- 5 major kernel releases, 1 pending.
- Multiple important changes.
- Lots of fixes.
- Substantial amount of new code.
- Increased number of developers involved.

2.6.35 – 2.6.36

- Workqueues freezing rework.
- Consolidation/rework of ACPI system suspend/resume.
- Continuation of ACPI GPEs handling rework related to runtime PM.
- Disabling ASPM for systems we aren't given control of via _OSC.
- ACPI procfs interface rework/removal.
- Rework of ACPI _OSC handling for PCI Express root complexes.
- Mechanism to prevent OOM from happening during hibernate memory preallocation.
- Assorted fixes.

2.6.36 – 2.6.37

- Empty generic runtime PM callbacks.
- Hibernate image compression for in-kernel hibernation.
- Default hibernate image size depends on RAM size (image size autotuning).
- Introduction of `struct wakeup_source` and wakeup events statistics.
- Runtime PM core rework and introduction of autosuspend.
- Introduction for OPP core code.
- PME status polling for legacy PCI devices.
- ACPI power resources reference counting fixes.
- Devices allowed to be removed during late suspend and early resume.
- Assorted fixes.

2.6.37 – 2.6.38

- Disabling of PCIe ASPM if BIOS asks us to (famous "*Phoronix regression*").
- Clearing of PCIe Root PME Status bits early during system resume.
- Synchronous runtime PM interface for interrupt handlers.
- Different list of devices used for each stage of device suspend.
- pm_generic_ operations prototype.
- Initial suspend trace point calls for perf.
- Rework of the ACPI NVS handling.
- Rework of the handling of ACPI power resources.
- Call ACPI _OSC once per root bridge.
- Assorted fixes.

2.6.38 – 2.6.39

- Use existing ACPI iomaps for NVS save/restore.
- Wakeup sysfs files are not created for devices that cannot wake up.
- `CONFIG_PM` depends on
  (`CONFIG_PM_SLEEP` || `CONFIG_PM_RUNTIME`).
- Preliminary support for device power domains.
- System-wide PM and runtime PM treat subsystems consistently.
- Introduction of `struct syscore_ops` for core subsystems PM.
- Removal of deprecated sysfs cpufreq file `sampling_rate_max` and per-cpu ondemand/conservative sysfs files.
- Report ASPM support to BIOS if not disabled from command line.
- Disabling of ASPM when `_OSC` control is not granted for PCIe.
- Backlight handling rework.
- Introduction of `CONFIG_HIBERNATE_CALLBACKS`.
- Assorted fixes.

2.6.39 – 3.0

- Power domain callbacks take precedence over subsystem ones.
- Subsystem data field added to `struct dev_pm_info`.
- Introduction of generic clock manipulation rountines for runtime PM.
- cpufreq re-creates sysfs directory and symlinks during CPU hotplug.
- cpufreq uses dynamic debug instead of custom infrastructure.
- Removal of sysdev suspend, resume and shutdown operations.
- Freezer uses SMP barriers (instead of generic memory barriers).
- Removal of `acpi_sleep=s4_nonvs`.
- Introduction of hibernate sysfs knob to control size of memory for drivers.
- Assorted fixes.

3.0 – present

- `struct dev_power_domain` renamed to `struct dev_pm_domain`.
- Preliminary support for generic I/O PM domains.
- Generic I/O PM domains used on SH7372.
- Introduction of generic "noirq" callback routines for subsystems.
- Race conditions between runtime PM and system sleep limited.
- cpufreq code reorganization.
- ACPI battery fixes and improvements.
- cpuidle doesn't depend on `pm_idle`.
- `pm_runtime_put_sync()` allowed to be called from interrupts off context.
- Assorted fixes.

3.2 material

- Multiple master domains allowed for generic PM domains.
- Per-device PM QoS.
- Freezer update.
- `might_sleep()` added to runtime PM helpers.
- New macro to test for runtime PM events.
- Storage keys in hibernation image on s390.
- Statistics debugfs file for suspend to RAM.
- Clock-related PM definitions and headers moved to separate file.
- Reference counting for `power.subsys_data`.

What's Next

- User space interface for PM QoS?
- Device/PM domain attributes to be used with PM QoS?
- Off-the-tree dependencies between devices?
- . . . ?