Dynamic Power Management

- What, in brief, DPM does.
- How it interacts with other aspects of low-power designs.
- How it relates to other power management technology for embedded Linux.
- Potential CELF activities related to power management.

Introducing DPM

- Software to set power states and communicate power-related information in embedded systems.
- Geared toward aggressive power conservation in battery-powered products.
- Runs on TI OMAPs, Intel XScale PXA27x...
- Originated from a collaboration between IBM Research and MontaVista Software.
- Maintained at dynamicpower.sourceforge.net.

What DPM Does

- Scales clocks, voltages, and other platform power parameters according to changes in system state.
- Hooks into scheduler and idle loop to trigger power state changes and modify idle behavior.
- Extends device and system PM in ways to be described in following slides.
- Provides interfaces for configuration and control.

DPM Clock and Voltage Scaling

- Can set different power operating points at each change in Linux scheduler state, for example:
  - Set a high-power operating point when running a task, set a low-power point while in the idle loop.
  - Different power points for different classes of tasks.
- Which operating point to set at each state is chosen by the system designer or higher-layer software.

How Clock and Voltage Scaling is Configured

- An *operating point* consists of named board-specific *power parameters* (“PLL=266”).
- A *policy* maps between system *operating states* (idle, running task) and chosen operating points.
- DPM provides code to set power parameters; operating points and policies are created and activated by system designer via userspace APIs.

How Applications Use DPM

- Each application runs under a power state.
  - State may specify more or less power relative to the default – policy defines the absolute levels.
- A power policy manager can assign DPM states to tasks, or a task can manage its own state.
- Most tasks are unaware of DPM, but interesting things have been done with DPM-aware apps.

Device Power Management Interactions

- Don't shut down clocks needed by active devices.
- Choose an operating point compatible with current device needs.
- Call drivers to scale device clocking when upstream clocks modified.
- Shut down devices incompatible with lower-power state in emergency low battery condition.

System State Interactions

- Power policy may modified in response to:
  - Kernel events such as changes in device state, low battery indication, "suspend" button press, timeouts.
  - Explicit instructions from applications.
  - Observed criteria such as changes in CPU load.
- DPM interfaces for entering sleep/standby modes in conjunction with features such as device constraints.

Power Policy Management

- Higher-layer, custom software employed to:
  - Configure DPM as needed for a particular system.
  - Make policy change decisions.
  - Perform custom actions at power state transitions.
- Additional software from silicon and OS vendors extend DPM for developer convenience, development tools, and improved power savings.

Relationships to Other PM Technology

- ACPI/hardware-based power state modification.
  - Don't step on each other, event handling overlap.
- Variable Scheduling Timeouts / Dynamic Tick.
  - Configure state during extended idle.
- Potential DPM core interactions?
  - Software Suspend / Hibernate-to-disk/flash.
  - Automatic device clock gating.
  - Size/cache locality improvements.

Common Design Challenges

- Frequent power state transitions, driven by a state machine – not a simple call to “set this state”.
- Distributing intelligence on allowed power states between kernel and userspace.
- Changes to device and platform PM support often requested.
- Unclear how to map unique designs to low-level DPM interfaces.

Potential Future Directions

- Merge with cpufreq.
- Generic PM subsystem support for platform-specific low-power states.
- PM subsystem support for userspace notification of kernel power events.
- D-Bus for userspace power event messages.
- Device clock tree management: shut down idle clock subtrees...

Want to Get Involved?

- DPM discussion/contributions at sourceforge.
- Embedded enhancements for device and platform PM at LKML and OSDL.
- Get active in the CELF PMWG: help discuss need and ideas, contribute code.
- Evaluate power savings of PM technology on real CE devices, let us know what works well on which platforms.