Power Capping Framework

The power capping framework provides a consistent interface between the kernel and the user space that allows power capping drivers to expose the settings to user space in a uniform way.

Terminology

The framework exposes power capping devices to user space via sysfs in the form of a tree of objects. The objects at the root level of the tree represent 'control types', which correspond to different methods of power capping. For example, the intel-rapl control type represents the Intel "Running Average Power Limit" (RAPL) technology, whereas the 'idle-injection' control type corresponds to the use of idle injection for controlling power.

Power zones represent different parts of the system, which can be controlled and monitored using the power capping method determined by the control type the given zone belongs to. They each contain attributes for monitoring power, as well as controls represented in the form of power constraints. If the parts of the system represented by different power zones are hierarchical (that is, one bigger part consists of multiple smaller parts that each have their own power controls), those power zones may also be organized in a hierarchy with one parent power zone containing multiple subzones and so on to reflect the power control topology of the system. In that case, it is possible to apply power capping to a set of devices together using the parent power zone and if more fine grained control is required, it can be applied through the subzones.

Example sysfs interface tree:

```
/sys/devices/virtual/powercap
â””â”€â”€intel-rapl
    â”œâ”€â”€intel-rapl:0
    â”‚Â Â  â”œâ”€â”€constraint_0_name
    â”‚Â Â  â”œâ”€â”€constraint_0_power_limit_uw
    â”‚Â Â  â”œâ”€â”€constraint_0_time_window_us
    â”‚Â Â  â”œâ”€â”€constraint_1_name
    â”‚Â Â  â”œâ”€â”€constraint_1_power_limit_uw
    â”‚Â Â  â”œâ”€â”€constraint_1_time_window_us
    â”‚Â Â  â”œâ”€â”€device -> ../../intel-rapl
    â”‚Â Â  â”œâ”€â”€energy_uj
    â”‚Â Â  â”œâ”€â”€intel-rapl:0:0
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_name
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_power_limit_uw
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_time_window_us
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_name
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_power_limit_uw
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_time_window_us
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€device -> ../../intel-rapl:0
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€energy_uj
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€max_energy_range_uj
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€name
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€enabled
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€power
    â”‚Â Â  â”‚Â Â  â”‚Â Â  â”œâ”€â”€async
    â”‚Â Â  â”‚Â Â  â”‚Â Â  []
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€subsystem -> ../../../../../../class/power_cap
    â”‚Â Â  â”‚Â Â  â””â”€â”€uevent
    â”‚Â Â  â”œâ”€â”€intel-rapl:0:1
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_name
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_power_limit_uw
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_time_window_us
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_name
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_power_limit_uw
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_time_window_us
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€device -> ../../intel-rapl:0
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€energy_uj
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€max_energy_range_uj
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€name
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€enabled
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€power
    â”‚Â Â  â”‚Â Â  â”‚Â Â  â”œâ”€â”€async
    â”‚Â Â  â”‚Â Â  â”‚Â Â  []
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€subsystem -> ../../../../../../class/power_cap
    â”‚Â Â  â”‚Â Â  â””â”€â”€uevent
    â”‚Â Â  â”œâ”€â”€max_energy_range_uj
    â”‚Â Â  â”œâ”€â”€max_power_range_uw
    â”‚Â Â  â”œâ”€â”€name
    â”‚Â Â  â”œâ”€â”€enabled
    â”‚Â Â  â”œâ”€â”€power
    â”‚Â Â  â”‚Â Â  â”œâ”€â”€async
    â”‚Â Â  â”‚Â Â  []
    â”‚Â Â  â”œâ”€â”€subsystem -> ../../../../../class/power_cap
```

```
â”‚Â Â  â”œâ”€â”€enabled
â”‚Â Â  â”œâ”€â”€uevent
â”œâ”€â”€intel-rapl:1
â”‚Â Â  â”œâ”€â”€constraint_0_name
â”‚Â Â  â”œâ”€â”€constraint_0_power_limit_uw
â”‚Â Â  â”œâ”€â”€constraint_0_time_window_us
â”‚Â Â  â”œâ”€â”€constraint_1_name
â”‚Â Â  â”œâ”€â”€constraint_1_power_limit_uw
â”‚Â Â  â”œâ”€â”€constraint_1_time_window_us
â”‚Â Â  â”œâ”€â”€device -> ../../intel-rapl
â”‚Â Â  â”œâ”€â”€energy_uj
â”‚Â Â  â”œâ”€â”€intel-rapl:1:0
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_name
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_power_limit_uw
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_time_window_us
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_name
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_power_limit_uw
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_time_window_us
â”‚Â Â  â”‚Â Â  â”œâ”€â”€device -> ../../intel-rapl:1
â”‚Â Â  â”‚Â Â  â”œâ”€â”€energy_uj
â”‚Â Â  â”‚Â Â  â”œâ”€â”€max_energy_range_uj
â”‚Â Â  â”‚Â Â  â”œâ”€â”€name
â”‚Â Â  â”‚Â Â  â”œâ”€â”€enabled
â”‚Â Â  â”‚Â Â  â”œâ”€â”€power
â”‚Â Â  â”‚Â Â  â”‚Â Â  â”œâ”€â”€async
â”‚Â Â  â”‚Â Â  â”‚Â Â  []
â”‚Â Â  â”‚Â Â  â”œâ”€â”€subsystem -> ../../../../../../class/power_cap
â”‚Â Â  â”‚Â Â  â””â”€â”€uevent
â”‚Â Â  â”œâ”€â”€intel-rapl:1:1
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_name
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_power_limit_uw
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_0_time_window_us
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_name
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_power_limit_uw
â”‚Â Â  â”‚Â Â  â”œâ”€â”€constraint_1_time_window_us
â”‚Â Â  â”‚Â Â  â”œâ”€â”€device -> ../../intel-rapl:1
â”‚Â Â  â”‚Â Â  â”œâ”€â”€energy_uj
â”‚Â Â  â”‚Â Â  â”œâ”€â”€max_energy_range_uj
â”‚Â Â  â”‚Â Â  â”œâ”€â”€name
â”‚Â Â  â”‚Â Â  â”œâ”€â”€enabled
â”‚Â Â  â”‚Â Â  â”œâ”€â”€power
â”‚Â Â  â”‚Â Â  â”‚Â Â  â”œâ”€â”€async
â”‚Â Â  â”‚Â Â  â”‚Â Â  []
â”‚Â Â  â”‚Â Â  â”œâ”€â”€subsystem -> ../../../../../../class/power_cap
â”‚Â Â  â”‚Â Â  â””â”€â”€uevent
â”‚Â Â  â”œâ”€â”€max_energy_range_uj
â”‚Â Â  â”œâ”€â”€max_power_range_uw
â”‚Â Â  â”œâ”€â”€name
â”‚Â Â  â”œâ”€â”€enabled
â”‚Â Â  â”œâ”€â”€power
â”‚Â Â  â”‚Â Â  â”œâ”€â”€async
â”‚Â Â  â”‚Â Â  []
â”‚Â Â  â”œâ”€â”€subsystem -> ../../../../../class/power_cap
â”‚Â Â  â”œâ”€â”€uevent
â”œâ”€â”€power
â”‚Â Â  â”œâ”€â”€async
â”‚Â Â  []
â”œâ”€â”€subsystem -> ../../../../class/power_cap
â”œâ”€â”€enabled
â””â”€â”€uevent
```

The above example illustrates a case in which the Intel RAPL technology, available in IntelÂ® IA-64 and IA-32 Processor Architectures, is used. There is one control type called intel-rapl which contains two power zones, intel-rapl:0 and intel-rapl:1, representing CPU packages. Each of these power zones contains two subzones, intel-rapl:j:0 and intel-rapl:j:1 (j = 0, 1), representing the "core" and the "uncore" parts of the given CPU package, respectively. All of the zones and subzones contain energy monitoring attributes (energy_uj, max_energy_range_uj) and constraint attributes (constraint_*) allowing controls to be applied (the constraints in the 'package' power zones apply to the whole CPU packages and the subzone constraints only apply to the respective parts of the given package individually). Since Intel RAPL doesn't provide instantaneous power value, there is no power_uw attribute.

In addition to that, each power zone contains a name attribute, allowing the part of the system represented by that zone to be identified. For example:

```
cat /sys/class/power_cap/intel-rapl/intel-rapl:0/name
```

package-0

Depending on different power zones, the Intel RAPL technology allows one or multiple constraints like short term, long term and peak power, with different time windows to be applied to each power zone. All the zones contain attributes representing the constraint names, power limits and the sizes of the time windows. Note that time window is not applicable to peak power. Here, constraint_j_* attributes correspond to the jth constraint (j = 0,1,2).

For example:

```
constraint_0_name
constraint_0_power_limit_uw
constraint_0_time_window_us
constraint_1_name
constraint_1_power_limit_uw
constraint_1_time_window_us
constraint_2_name
constraint_2_power_limit_uw
constraint_2_time_window_us
```

Power Zone Attributes

Monitoring attributes

energy_uj (rw)
    Current energy counter in micro joules. Write "0" to reset. If the counter can not be reset, then this attribute is read only.
max_energy_range_uj (ro)
    Range of the above energy counter in micro-joules.
power_uw (ro)
    Current power in micro watts.
max_power_range_uw (ro)
    Range of the above power value in micro-watts.
name (ro)
    Name of this power zone.

It is possible that some domains have both power ranges and energy counter ranges; however, only one is mandatory.

Constraints

constraint_X_power_limit_uw (rw)
    Power limit in micro watts, which should be applicable for the time window specified by "constraint_X_time_window_us".
constraint_X_time_window_us (rw)
    Time window in micro seconds.
constraint_X_name (ro)
    An optional name of the constraint
constraint_X_max_power_uw(ro)
    Maximum allowed power in micro watts.
constraint_X_min_power_uw(ro)
    Minimum allowed power in micro watts.
constraint_X_max_time_window_us(ro)
    Maximum allowed time window in micro seconds.
constraint_X_min_time_window_us(ro)
    Minimum allowed time window in micro seconds.

Except power_limit_uw and time_window_us other fields are optional.

Common zone and control type attributes

enabled (rw): Enable/Disable controls at zone level or for all zones using a control type.

Power Cap Client Driver Interface

The API summary:

Call powercap_register_control_type() to register control type object. Call powercap_register_zone() to register a power zone (under a given control type), either as a top-level power zone or as a subzone of another power zone registered earlier. The number of constraints in a power zone and the corresponding callbacks have to be defined prior to calling powercap_register_zone() to register that zone.

To Free a power zone call powercap_unregister_zone(). To free a control type object call powercap_unregister_control_type(). Detailed API can be generated using kernel-doc on include/linux/powercap.h.