

Introduction au Power Management dans Linux

Implémentation, Utilisation et Benchmark

maxime.chevallier@smile.fr

21 mars 2017



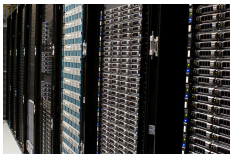
- 1 Enjeux
- 2 Power Management Dynamique
- 3 Endormissement

Enjeux

- N'utiliser que les ressources nécessaires
- Etre générique (ACPI, APM, SCPI)
- Respecter les contraintes utilisateur
- Rester transparent

Enjeux

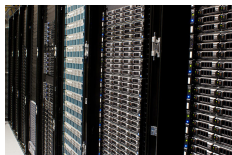
- N'utiliser que les ressources nécessaires
- Etre générique (ACPI, APM, SCPI)
- Respecter les contraintes utilisateur
- Rester transparent



x86, milliers d'unités

Enjeux

- N'utiliser que les ressources nécessaires
- Etre générique (ACPI, APM, SCPI)
- Respecter les contraintes utilisateur
- Rester transparent



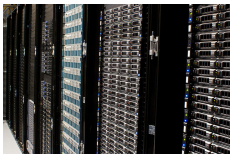
x86, milliers d'unités



x86, flexibilité

Enjeux

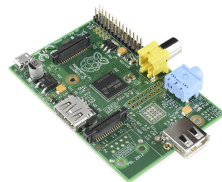
- N'utiliser que les ressources nécessaires
- Etre générique (ACPI, APM, SCPI)
- Respecter les contraintes utilisateur
- Rester transparent



x86, milliers d'unités



x86, flexibilité



ARM, autonomie

- 1 Enjeux
- 2 Power Management Dynamique
- 3 Endormissement

Minimiser la consommation d'un système actif

Compromis

- Ressources utilisées
- Ressources nécessaires
- Latences acceptables
- Consommation actuelle
- Température actuelle

PM core

- API pour drivers
- Interface sysfs
- PM dynamique
- Modes d'endormissement

Documentation : Documentation/power

Headers : include/linux/pm.h

Implémentation : kernel/power/

Etats

- active : Est capable d'I/O
- suspended : Pas d'I/O

Callbacks

```
runtime_suspend(dev)
runtime_resume(dev)
runtime_idle(dev)
```

Helpers

```
pm_runtime_*
pm_request_*
pm_schedule_*
pm_runtime_{get,put}*
pm_*_autosuspend
```

Indiquer au noyau les latence et débits à respecter

Paramètres globaux

- `cpu_dma_latency` (μ s)
- `memory_bandwidth` (mbps)
- `network_latency` (μ s)
- `network_throughput` (kbps)

Interface Userspace : `/dev/*` + `sysfs`

Que faire quand le CPU n'a rien à faire ?

- Choix du mode (governor)
 - `select()`
 - `reflect()`
- Implémentation (driver)

Que faire quand le CPU n'a rien à faire ?

- Choix du mode (governor)
 - `select()`
 - `reflect()`
- Implémentation (driver)

struct cpuidle_state

- `exit_latency`
- `power_usage`
- `target_residency`
- `int enter([...], int index)`

Que faire quand le CPU n'a rien à faire ?

- Choix du mode (governor)
 - `select()`
 - `reflect()`
- Implémentation (driver)

struct cpuidle_state

- `exit_latency`
- `power_usage`
- `target_residency`
- `int enter([...], int index)`

`powertop, /sys/devices/system/cpu/cpuX/cpuidle`

Driver intel_idle

i7 4702MQ

name	latency	residency	utilisation
C0	-	-	1.5%
POLL	0	0	0.2%
C1-HSW	2	2	0.6%
C1E-HSW	10	20	0.2%
C3-HSW	33	100	0.0%
C6-HSW	133	400	0.0%
C7s-HSW	166	500	97.4%

Driver ACPI processor_idle

i5 6500

name	latency	residency
C0	-	-
POLL	0	0
C1	1	2
C2	151	302
C3	256	512

Dynamic **V**oltage and **F**requency **S**caling

Dynamic Voltage and Frequency Scaling

cpufreq

/sys/devices/system/cpu/cpuX/cpufreq/

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :
 - performance

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :
 - performance
 - powersave

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :
 - performance
 - powersave
 - userspace

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :
 - performance
 - powersave
 - userspace
 - ondemand

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :
 - performance
 - powersave
 - userspace
 - ondemand
 - conservative

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :
 - performance
 - powersave
 - userspace
 - ondemand
 - conservative
 - schedutil (linux 4.6)

`/sys/devices/system/cpu/cpuX/cpufreq/`

Dynamic Voltage and Frequency Scaling

cpufreq

- Implémentation hardware (policy)
- Implémentation software (governor) :
 - performance
 - powersave
 - userspace
 - ondemand
 - conservative
 - schedutil (linux 4.6)

`/sys/devices/system/cpu/cpuX/cpufreq/`

devfreq

Similaire pour les devices non-CPU

Tuples (Fréquence, Tension) pour un périphérique

```
operating-points = <
/* kHz uV */
792000 1100000
396000 950000
198000 850000
>;
```

Actions en fonction de la température

Thermal zone

Actions en fonction de la température

Thermal zone

- Température (trip_point)

Actions en fonction de la température

Thermal zone

- Température (trip_point)
- Politique :

Actions en fonction de la température

Thermal zone

- Température (trip_point)
- Politique :
 - step_wise

Actions en fonction de la température

Thermal zone

- Température (trip_point)
- Politique :
 - step_wise
 - fair_share

Actions en fonction de la température

Thermal zone

- Température (trip_point)
- Politique :
 - step_wise
 - fair_share
 - userspace

Actions en fonction de la température

Thermal zone

- Température (trip_point)
- Politique :
 - step_wise
 - fair_share
 - userspace
- Cooling device

Actions en fonction de la température

Thermal zone

- Température (trip_point)
- Politique :
 - step_wise
 - fair_share
 - userspace
- Cooling device

Cooling device

- Hardware : Ventilateur
- Software : cpufreq

Actions en fonction de la température

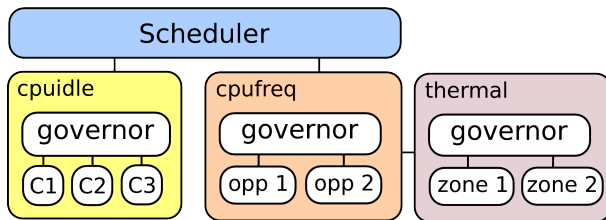
Thermal zone

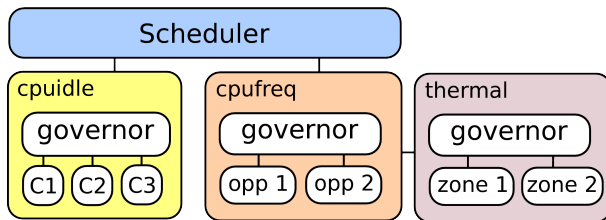
- Température (trip_point)
- Politique :
 - step_wise
 - fair_share
 - userspace
- Cooling device

Cooling device

- Hardware : Ventilateur
- Software : cpufreq

/sys/class/thermal/





A venir

- Unifier cpuidle et cpufreq
- Energy Aware Scheduler

- 1 Enjeux
- 2 Power Management Dynamique
- 3 Endormissement

struct dev_pm_ops

Ensemble de callbacks :

prepare()	freeze()
complete()	thaw()
suspend()	poweroff()
resume()	restore()

struct dev_pm_ops

Ensemble de callbacks :

<code>prepare()</code>	<code>freeze()</code>
<code>complete()</code>	<code>thaw()</code>
<code>suspend()</code>	<code>poweroff()</code>
<code>resume()</code>	<code>restore()</code>

Wakeup

- `enable_irq_wake()`
- `disable_irq_wake()`

ACPI State : S1

```
freeze > /sys/power/state
```

Suspend to Idle

- Entièrement software
- Freeze userspace
- Périphériques lowpower
- Toujours supporté

Réveil en quelques millisecondes

ACPI State : S2

`standby > /sys/power/state`

Standby

- Suspend to Idle +
- Coupure des coeurs non-boot
- Coupure de certains composants bas niveau
- Support dépendant de la plateforme

Réveil en quelques millisecondes

ACPI State : S3

```
mem > /sys/power/state
```

Suspend to RAM

- Standby +
- Périphériques en lowpower
- CPU en lowpower
- RAM en auto-rafraichissement
- Support dépendant de la plateforme

Réveil en quelques centaines de millisecondes

ACPI State : S4

```
disk > /sys/power/state
```

Suspend to disk

- Image mémoire persistée
- Système en lowpower, voire éteint

parametres

```
/sys/power/disk
```

- platform
- shutdown
- reboot
- suspend

Device Tree

- wakeup-source (Générique)
- gpio-key,wakeup
- enable-sdio-wakeup
- linux,wakeup
- etc. (Anciens bindings)

Device Tree

- wakeup-source (Générique)
- gpio-key,wakeup
- enable-sdio-wakeup
- linux,wakeup
- etc. (Anciens bindings)

sysfs

- enabled > /sys/devices/.../power/wakeup
- disabled > /sys/devices/.../power/wakeup

Consultation : /sys/kernel/debug/wakeup_sources

Merci