Energy Aware Scheduling

Linux Plumbers Conference 2018

Dietmar Eggemann, Quentin Perret
Introduction

● Short history of Energy Aware Scheduling (EAS) patch-set
  ○ 2014/15: Patch-sets with active and idle energy costs data for CPUs and clusters
  ○ 2018: Patch-sets with active energy costs data for CPUs only and separate Energy Model (EM) framework

● Current v8 patch-set is ready for mainlining
  ○ EAS has been used for ARM big.LITTLE platforms in Android products over years
  ○ v8 patch-set will be part of the v4.19 version of Android Common Kernel
Possible future improvements

1. How to do task placement of new tasks?
2. How to handle overutilization with new tasks?
3. Should the EM deal with more than CPUs?
4. Where should we compute $P = CV^2f$?
1. How to do task placement of new tasks?

New

CPU 0
CPU 1
CPU 2
CPU 3
CPU 4
CPU 5
CPU 6
CPU 7

OPP 1
OPP 2
OPP 3

Capacity

CPU 0
CPU 1
CPU 2
CPU 3
CPU 4
CPU 5
CPU 6
CPU 7
1. How to do task placement of new tasks?

![Diagram showing task placement]

- **CPU 0**
- **CPU 1**
- **CPU 2**
- **CPU 3**
- **CPU 4**
- **CPU 5**
- **CPU 6**
- **CPU 7**

- **New**

- **find_idlest_cpu()**

- **OPP 1**
- **OPP 2**
- **OPP 3**

- **Capacity**

- **512**
1. How to do task placement of new tasks?

- Balancing options for new tasks?
  - Just use the current slow path (find_idlest_cpu())?
  - “Predict” the util_avg of new tasks as per post_init_entity_util_avg()?
  - Assume static initial util_avg (min_cap / 2 ? util_avg of parent?)
2. How to handle overutilization with new tasks

```c
static void enqueue_task_fair(struct rq *rq, struct task_struct *p, int flags)
{
    ...
    if (flags & ENQUEUE_WAKEUP)
        update_overutilized_status(rq);
    ...
}
```

- Wait for the PELT signal to ‘converge’?
- Initial util_avg value set to 0? Impact on frequency selection / initial EAS task placement?
3. Should the EM deal with more than CPUs?
4. Where should we compute $P = CV^2f$?

```c
get_power(cpu, Hz, mW) {
    dpc = get_from_dt(...);
    V = pm_opp_get_voltage(Hz);
    mW = dpc * V * V * Hz;
}
cpufreq_init() {
    _
    em_register_perf_domain(cpus,
        nr_opp, &get_power);
    _
}
```

`drivers/cpufreq/cpufreq-dt.c`

```c
cpu0 : {
    _
    dynamic-power-coefficient = ... ;
    _
}
```

`arch/arm64/boot/dts/xxx/platform.dts`
4. Where should we compute $P = CV^2f$?

```c
cpufreq_init() {
    ...
    em_register_perf_domain(cpus, nr_opps, &pm_opp_get_power);
    ...
}
```

drivers/cpufreq/scpi-cpufreq.c

```c
pm_opp_get_power(cpu, Hz, mW) {
    dpc = get_from_dt(...);
    V = pm_opp_get_voltage(Hz);
    mW = dpc * V * V * Hz;
}
```

drivers/pm_opp/of.c

```c
drivers/cpufreq/arm_big_little.c
```

```c
cpufreq_init() {
    ...
    em_register_perf_domain(cpus, nr_opps, &pm_opp_get_power);
    ...
}
```

drivers/cpufreq/cpufreq-dt.c

```c
drivers/cpufreq/arm_big_little.c
```

```c
cpufreq_init() {
    ...
    em_register_perf_domain(cpus, nr_opps, &pm_opp_get_power);
    ...
}
```

drivers/cpufreq/arm_big_little.c

```
arch/arm64/boot/dts/xxx/platform.dts
```