watchodg

系统stability检测工具

Agenda

- 1.总体流程
- 2.设备树中解析watchdog参数
- 3.设置定时器pet_timer
- 4.唤醒msm_watchdog线程
- 5.向其他CPU发送IPI中断
- 6.狗叫

watchdog总体流程

设备树中解析watchdog参数

msm_wdog_dt_to_pdata

```
wdog: qcom,wdt@17c10000{
        compatible = “qcom,msm-watchdog”;
        reg = <0x17c10000 0x1000>;    →wdog_data.phys_base = 0x17c10000, wdog_data.size = 0x1000
        reg-names = “wdt-base”;
        interrupts = <0 0 0>, <0 1 0>;    →bark irq  hwirq = 0x20, bite irq hwirq = 0x21(wdog_data中的bark_irq和bite_irq是虚拟中断号)
        qcom,bark-time = <11000>;      →wdog_data.bark_time = 11s
        qcom,pet-time = <9360>;        →wdog_data.pet_time = 9.36s
        qcom,ipi-ping;                 →wdog_data.do_ipi_ping = true
        qcom,wakeup-enable;            →wdog_data.wakeup_irq_enable = true
        qcom,scandump-size = <0x10100 0x10100 0x10100 0x10100
        0x18100 0x18100 0x18100 0x18100>;
};
```

设置pet_timer

init_watchdog_data

init_watchdog_data

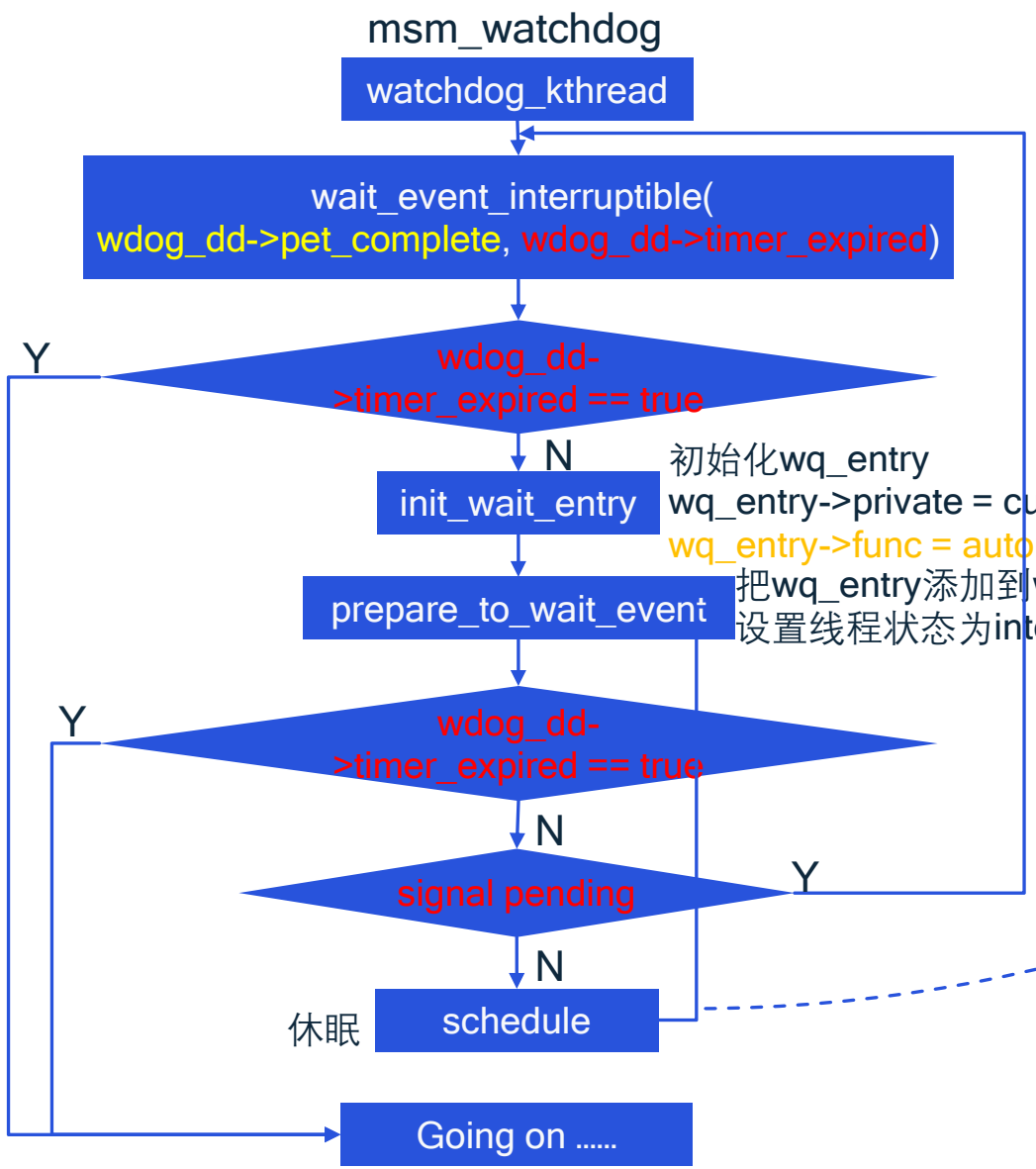
devm_request_irq — 注册bark irq中断处理函数wdog_bark_handler

set bark_time — write bark_time to register WDT0_BARK_TIME

set pet_timer — wdog_data->pet_timer.data = wdog_data
wdog_data->pet_timer.function = pet_task_wakeup
wdog_data->pet_timer.expires = jiffies+msecs_to_jiffies(wdog_data->pet_time)

first time to pet wdog — write 1 to register WDT0_RST

唤醒msm_watchdog线程



1.wachdog线程开始运行之后condition(timer_expired)为false，会创建wq_entry
2.把wq_entry添加到pet_complete等待队列，之后设置线程状态为interruptible
3.如果condition(pet_expired)为false，进入休眠

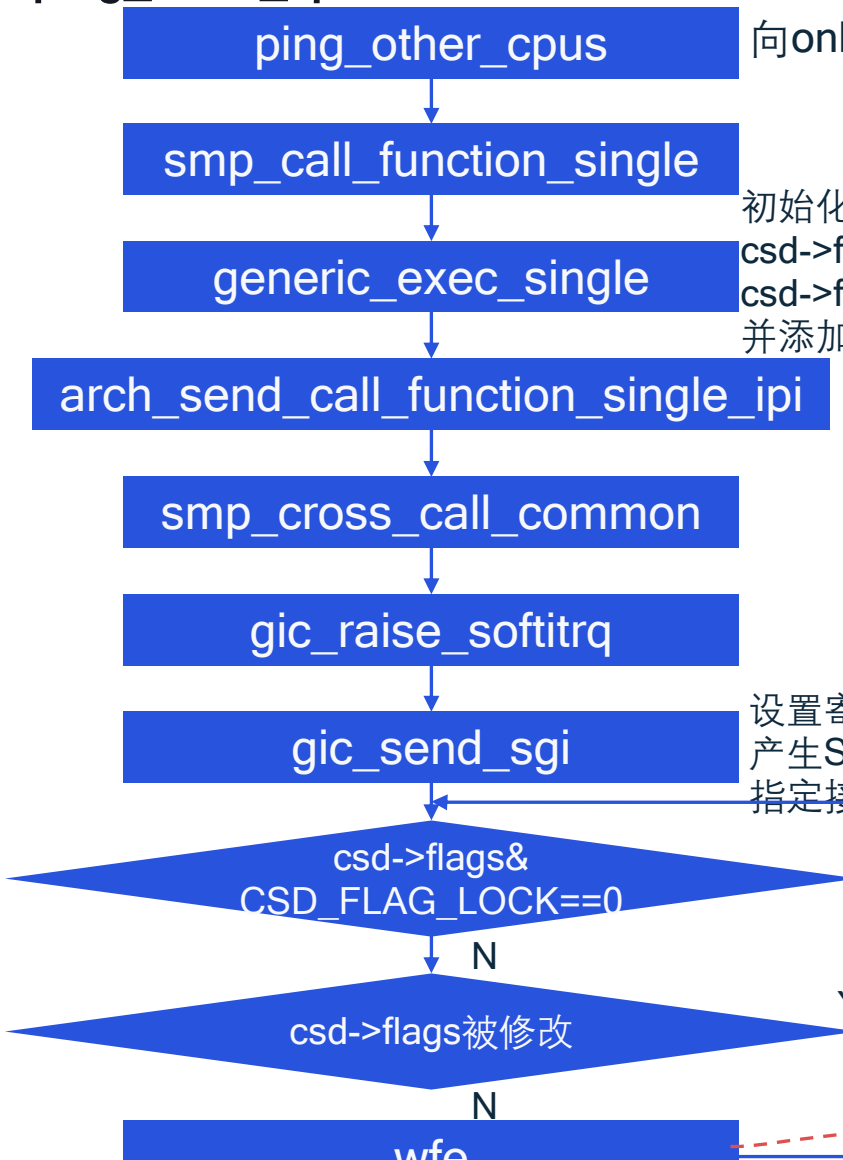
唤醒wdog_dd->pet_complete链表中的线程

4.pet_timer定时时间到，把timer_expired置为true
5.唤醒pet_complete等待队列上的线程–msm_watchdog

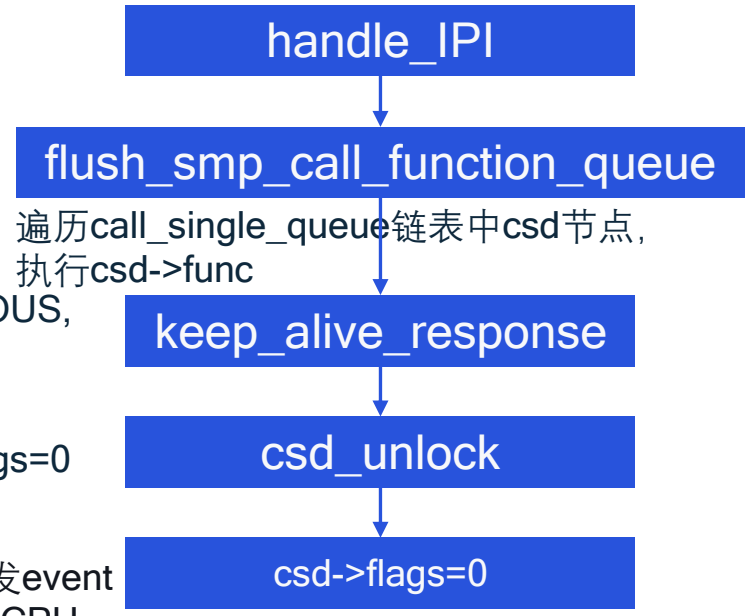
6.Watchdog线程判断condition(timer_expired)为true,跳出循环，继续往下执行（ping other cpu）

向其他cpu发送ipi中断

ping_other_cpus



target CPU接收到中断



stlr指令clear global monitor，同时会触发event
来唤醒调用wfe指令进入低功耗状态的CPU

cpu睡眠之前ldxr指令设置global monitor为
exclusive state

watchdog bark

wdog_bark_handler

向寄存器WDT_BITE_TIME中写入1，1/WDT_HZ时间之后触发bite中断