

---

---

# 案例篇:內核線程 CPU 利用率太高, 我該怎麼辦?

— Raix Lai —  
2020/11/25

---

---

# 排查網路問題

- 丟包問題，從 Linux 網路收發著手，結合 TCP/IP 原理進行分析
- 另一常見問題，CPU 使用率升高
  - 在高併發場景下，內核的 ksoftirqd 的 CPU 使用率會比較高
  - 由網路收發的軟中斷造成的
- 分析 CPU 使用率高的問題，需要使用之前課程中教的 CPU 效能檢測工具

# 內核 Process

- 三個特殊 process, 也是 PID 號碼最小的 process
  - 0 號 process: idle process
    - 系統創建的第一號 process, 在初始化 1, 2 之後, 作為 CPU 空閒時啟動。
  - 1 號: init process
    - 通常是 systemd process, 在 user space 運行, 管理其他 user space 的 process
    - 是 user space 的 process 祖先
  - 2 號: kthreadd process
    - 在 kernel space 運行, 管理 kernel space 的 process
- 查找 kernel process, 需從 2 號開始, 查詢底下的子 process。

# 查找 kthreadd 底下的子 process

- `ps -f --ppid 2 -p 2`
- `ps -ef | grep "\[.*\]"`

```
raix852@ubuntu-18-01:~$ ps -f --ppid 2 -p 2
UID          PID    PPID  C  STIME TTY          TIME CMD
root         2       0  0  11:46 ?           00:00:00 [kthreadd]
root         3       2  0  11:46 ?           00:00:00 [rcu_gp]
root         4       2  0  11:46 ?           00:00:00 [rcu_par_gp]
root         5       2  0  11:46 ?           00:00:00 [kworker/0:0-cgr]
root         6       2  0  11:46 ?           00:00:00 [kworker/0:0H-kb]
root         7       2  0  11:46 ?           00:00:00 [kworker/0:1-cgr]
root         8       2  0  11:46 ?           00:00:00 [kworker/u4:0-ev]
root         9       2  0  11:46 ?           00:00:00 [mm_percpu_wq]
root        10      2  0  11:46 ?           00:00:00 [ksoftirqd/0]
```

# 常見的內核 Process

- ksoftirqd: 處理軟中斷的 kernel process
- kthreadd: 管理 kernel process
- kswapd0: 用於記憶體回收
- kworker: 用於執行 kernel 工作隊列, 分有綁定 CPU 與無綁定兩類。
- migration: 進行負載均衡時, 將 process 遷移到 idle CPU 上
- jbd2/sda1-8: 為文件系統提供日誌功能, 以保證數據完整性。
- pdflush: 將記憶體的髒頁寫入硬碟(3.10 已併入 kworker)

# DEMO 案例

[https://hackmd.io/\\_nkGEAvQMizFAKkU7q-2g](https://hackmd.io/_nkGEAvQMizFAKkU7q-2g)

