# Linux 性能優化實戰

## CH 26 | 案例篇：如何找出狂打日誌的“內鬼”？

Bill Chang

2020/07/30

# Outline

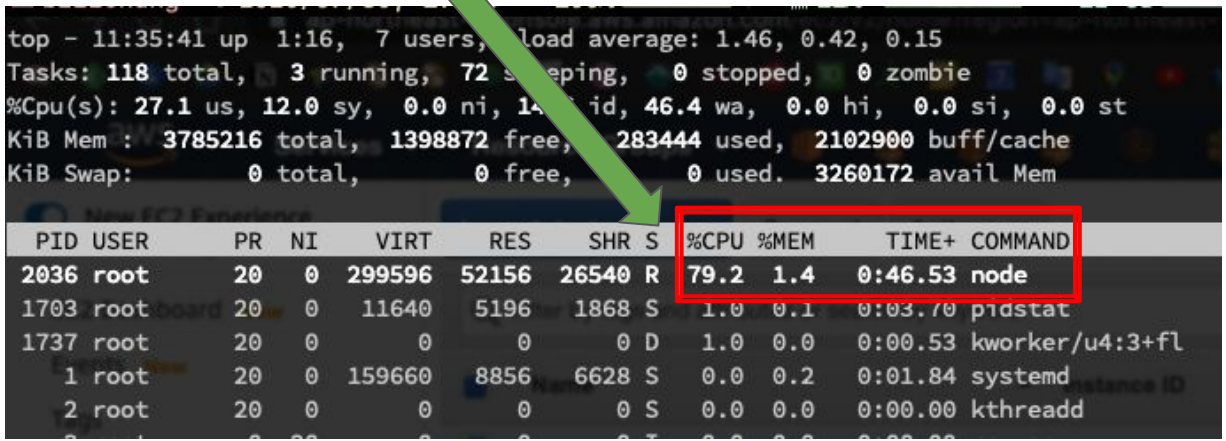
- 排除問題步驟
- 動態調整 log level
- 思考題

# 排除問題步驟

1. 找出是否有 process 佔用較多計算資源
2. 找 I/O 相關數值
3. 定位 process ID，使用找出目標檔案與大致寫入內容

# 排除問題步驟 - 找出是否有 process 佔用較多計算資源

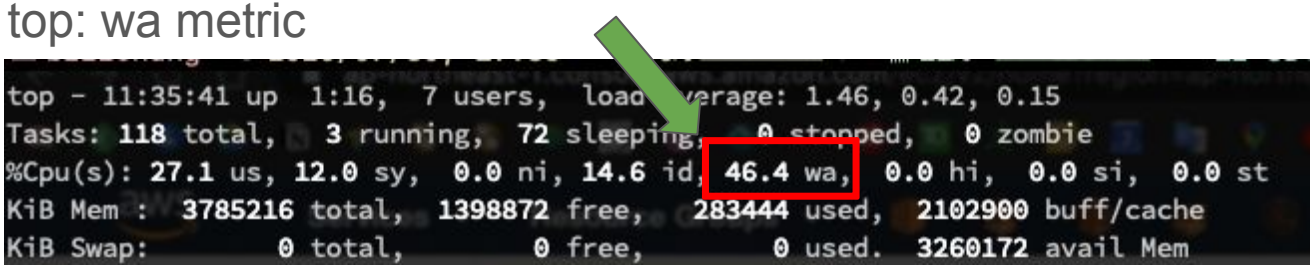
- top
- htop
- ...




wa, IO-wait : time waiting for I/O completion

# 排除問題步驟 - 找 I/O 相關數值

- top: wa metric




wa, IO-wait : time waiting for I/O completion

- pidstat
  - -d: Report I/O statistics (kernels 2.6.20 and later only)
- iostat
  - -d: Display the device utilization report
  - -x:Display extended statistics.
  - %util 在不同硬體環境下的意義

# 排除問題步驟 - 定位 process ID，使用找出目標檔案與大致寫入內容

- strace
  - -p (pid): Attach to the process with the process ID pid and begin tracing.
  - strace -p <pid> 後，觀察 write/stat 等 syscall
- lsof (List opened files)
  - -p:
    - This option excludes or selects the listing of files for the processes…
    - …PID numbers that begin with '^' (negation) represent exclusions.
  - FD 欄位
    - **w** for write access
    - **u** for read and write access
  - TYPE 欄位
    - …or ''REG'' for a regular file;
  - SIZE, SIZE/OFF, or OFFSET 欄位，顯示 SIZE 或者 OFFSET

# 動態調整 log level

- SIGUSR1 & SIGUSR2:
  - The **SIGUSR1** and **SIGUSR2** signals are set aside for you to use any way you want. They’re useful for simple interprocess communication, if you write a signal handler for them in the program that receives the signal.
  - https://www.gnu.org/software/libc/manual/html_node/Miscellaneous-Signals.html

# 動態調整 log level - Lab

- SIGUSR1: 提升 log level (log 量減少)
- SIGUSR2: 降低 log level (log 量增加, I/O 增加)
- 兩個 Signal 都用來調 level 太浪費，也可以只用一個循環

# 思考题

- “那么，今天的问题就是，这些内存到底是被 Buffer 还是 Cache 占用了呢？有没有什么方法来确认你的分析结果呢？”
- → **watch -d -n 1 cat /proc/meminfo**