21 | 套路篇

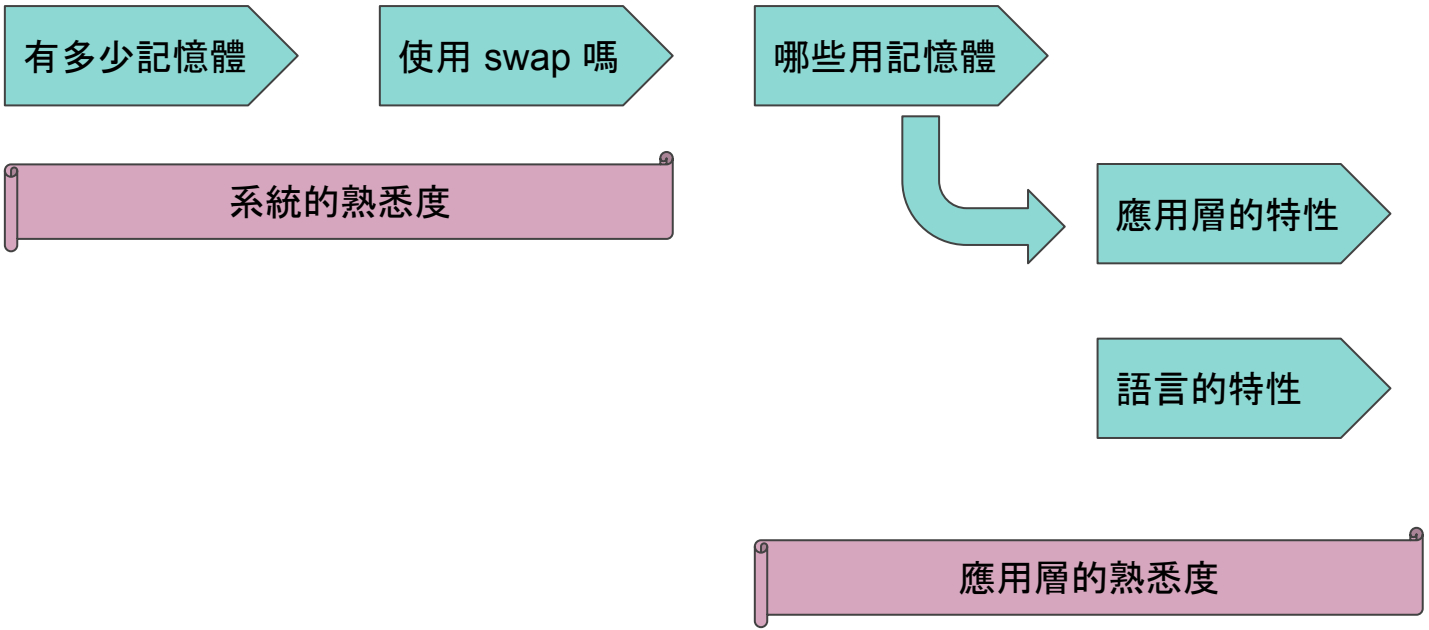
如何“快狠準”找到系统記憶體的問題？

如何"快狠準"找到系统記憶體的問題？

純個人感想:
不止系統要熟、工具熟練、反應快....
還要

經驗老到～

大方向



工具的解讀:使用率

TOP

```
top - 15:02:29 up 6 days,  8:21,  1 user,  load average: 0.02, 0.01, 0.00
Tasks: 131 total,   1 running,  83 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.3 sy,  0.0 ni, 99.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  3624464 total,  1380192 free,   577608 used,  1666664 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  2643000 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
31440 ts        20   0  743860  82740  12256 S   1.7  2.3  53:58.68 [TS_MAIN]
 1800 root      20   0  728464  34732  20668 S   0.7  1.0  16:49.74 docker-containe
23864 root      20   0   40528   3728   3116 R   0.3  0.1   0:00.02 top
31430 ts        20   0  281336  10636   7336 S   0.3  0.3   1:13.26 traffic_manager
    1 root      20   0   37816   5864   4036 S   0.0  0.2   0:12.01 systemd
```

工具的解讀:應用程序～記憶體使用狀況

pidstat -r

```
root@afu16-1:~# pidstat -r
Linux 4.15.0-1078-gcp (afu16-1)         07/15/2020      _x86_64_        (4 CPU)

02:57:50 PM   UID       PID  minflt/s  majflt/s      VSZ      RSS   %MEM  Command
02:57:50 PM     0         1      0.13      0.00    37816     5864   0.16  systemd
02:57:50 PM     0       413      0.02      0.00    40976     7308   0.20  systemd-journal
02:57:50 PM     0       452      0.00      0.00   102968     1568   0.04  lvmetad
02:57:50 PM     0       483      0.12      0.00    42608     3956   0.11  systemd-udevd
02:57:50 PM     0      1154      0.00      0.00    16124     2948   0.08  dhclient
```

工具的解讀:使用率

Free

```
ubuntu@aws:~$ free -m
               total        used        free      shared  buff/cache   available
Mem:            7624         398        5381          10        1844        6918
Swap:              0           0           0
```

工具的解讀:虛擬記憶體

vmstat

Virtual Meomory Statistics

```
ubuntu@aws:~$ vmstat 1

procs -----------memory---------- -swap- -----io---- -system-- ------cpu-----
 r  b  swpd   free         buff       cache       si   so    bi    bo    in     cs  us  sy  id  wa st

 0  0     0   5504944   166908   1722512    0    0     3    56    35    63   0  0 100   0  0

 0  0     0   5504936   166908   1722548    0    0     0   168   165   300   0  0 100   0  0

 0  0     0   5504936   166908   1722548    0    0     0   200   136   269   0  0 100   0  0
```

工具的解讀:記憶體緩存與洩漏

關於記憶體緩存效率
cachestat
cachetop

關於記憶體洩漏:
memleak
valgrind

```
14:16:11 Buffers MB: 233 / Cached MB: 1150 / Sort: HITS / Order: ascending
PID      UID      CMD              HITS     MISSES   DIRTIES  READ_HIT%  WRITE_HIT%
  23015 root     cachetop            4          0          0     100.0%        0.0%
    351 root     jbd2/sda1-8         4          0          0     100.0%        0.0%
  31433 root     traffic_manager    61         40         20      40.6%       19.8%
```

關於系統記憶體的設定

sysctl：系統基礎設定

swap：　swapon -s / free
　　swapoff -a && swapon -a
　　etc/fstab / vm.swappiness=0

drop_cache：釋放緩存
　　sh -c "sync; echo [1|2|3] > /proc/sys/vm/drop_caches"

關於一篇：微調處理能力

應用層的熟悉度

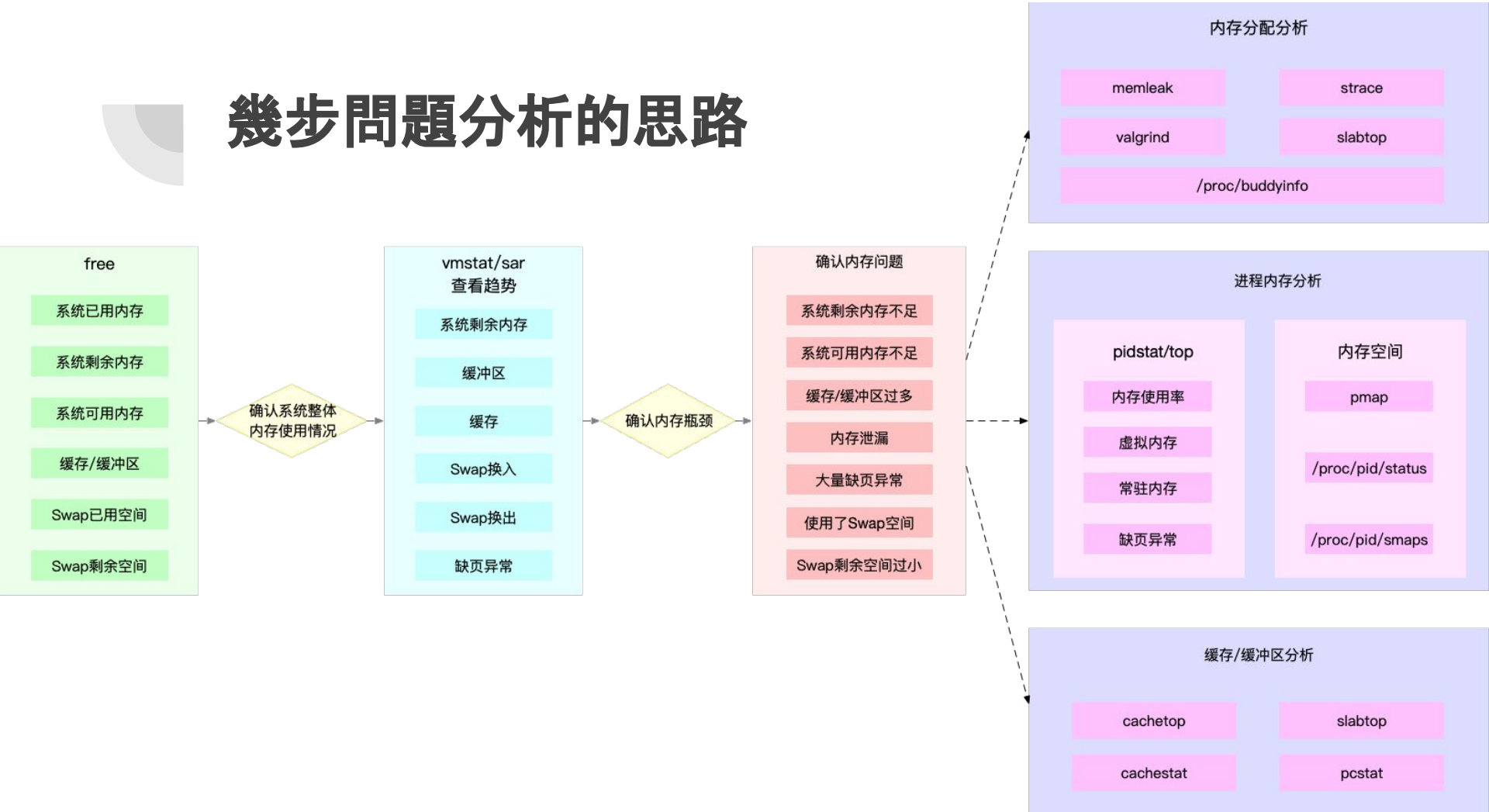
GO 語言：CPU 資源足夠下，記憶體使用少。

JAVA 語言：記憶體怪獸，需要熟悉 JVM。

NoSQL層：很多基於記憶體IO的應用服務。

DB層：Buffer_pool 觀念不能少。

幾步問題分析的思路


free
系统已用内存
系统剩余内存
系统可用内存
缓存/缓冲区
Swap已用空间
Swap剩余空间
确认系统整体
内存使用情况
vmstat/sar
查看趋势
系统剩余内存
缓冲区
缓存
Swap换入
Swap换出
缺页异常
确认内存瓶颈
确认内存问题
系统剩余内存不足
系统可用内存不足
缓存/缓冲区过多
内存泄漏
大量缺页异常
使用了Swap空间
Swap剩余空间过小
内存分配分析
memleak
strace
valgrind
slabtop
/proc/buddyinfo
进程内存分析
pidstat/top
内存使用率
虚拟内存
常驻内存
缺页异常
内存空间
pmap
/proc/pid/status
/proc/pid/smaps
缓存/缓冲区分析
cachetop
slabtop
cachestat
pcstat

常見的優化思路

1. 最好禁止 Swap。如果必須開啟 Swap，降低 swappiness 的值，減少內存回收時 Swap 的使用傾向。
2. 減少內存的動態分配。比如，可以使用內存池、大頁（HugePage）等。
3. 盡量使用緩存和緩衝區來訪問數據。比如，可以使用堆(heap)棧(stack)明確聲明內存空間，來存儲需要緩存的數據；或者用Redis 這類的外部緩存組件，優化數據的訪問。
4. 使用 cgroups 等方式限制進程的內存使用情況。這樣，可以確保系統內存不會被異常進程耗盡。
5. 通過 /proc/pid/oom_adj，調整核心應用的oom_score。這樣，可以保證即使內存緊張，核心應用也不會被 OOM 殺死。

總結

多看、多學、多好奇

經驗是累積出來的。