```
Processes: 284 total, 3 running, 5 stuck, 276 sleeping, 1340 threads
15:50:13
Load Avg: 1.52, 1.47, 1.48  CPU usage: 2.62% user, 3.10% sys, 94.27% idle   SharedLibs: 1196K resident, 0B data,
0B linkedit.
MemRegions: 90613 total, 2434M resident, 65M private, 640M shared. PhysMem: 8119M used (1295M wired), 72M
unused.
VM: 725G vsize, 1026M framework vsize, 8785612(0) swapins, 9281977(0) swapouts. Networks: packets: 26752134/33G
in, 15542577/3924M out.
Disks: 2508267/95G read, 3057143/126G written.

PID    COMMAND      %CPU  TIME     #TH   #WQ   #PORT MEM    PURG   CMPRS  PGRP  PPID  STATE    BOOSTS
%CPU_ME %CPU_OTHRS UID  FAULTS
26631  a.out        0.0   00:00.00 5     0     14    368K   0B     0B     26631 25842 sleeping *0[1]
0.00000 0.00000    501  344
26627  Python       0.0   00:01.64 32    0     53    35M    0B     0B     26627 26617 sleeping *0[1]
0.00000 0.00000    501  33469
26617  Vim          0.0   00:01.12 31    0     71    22M    0B     0B     26617 5829  sleeping *0[5]
0.00000 0.00000    501  13083
26595  QuickLookSat 0.0   00:00.36 2     0     41    11M    0B     0B     26595 1     sleeping  0[0]
0.00000 0.00000    501
```

- if you need more performances you can
    - decrease granularity of locks (example implementing a forward linked list with double pointers to head and tail)
    - lock free algorithms (more error prone though)