

# Tema EP

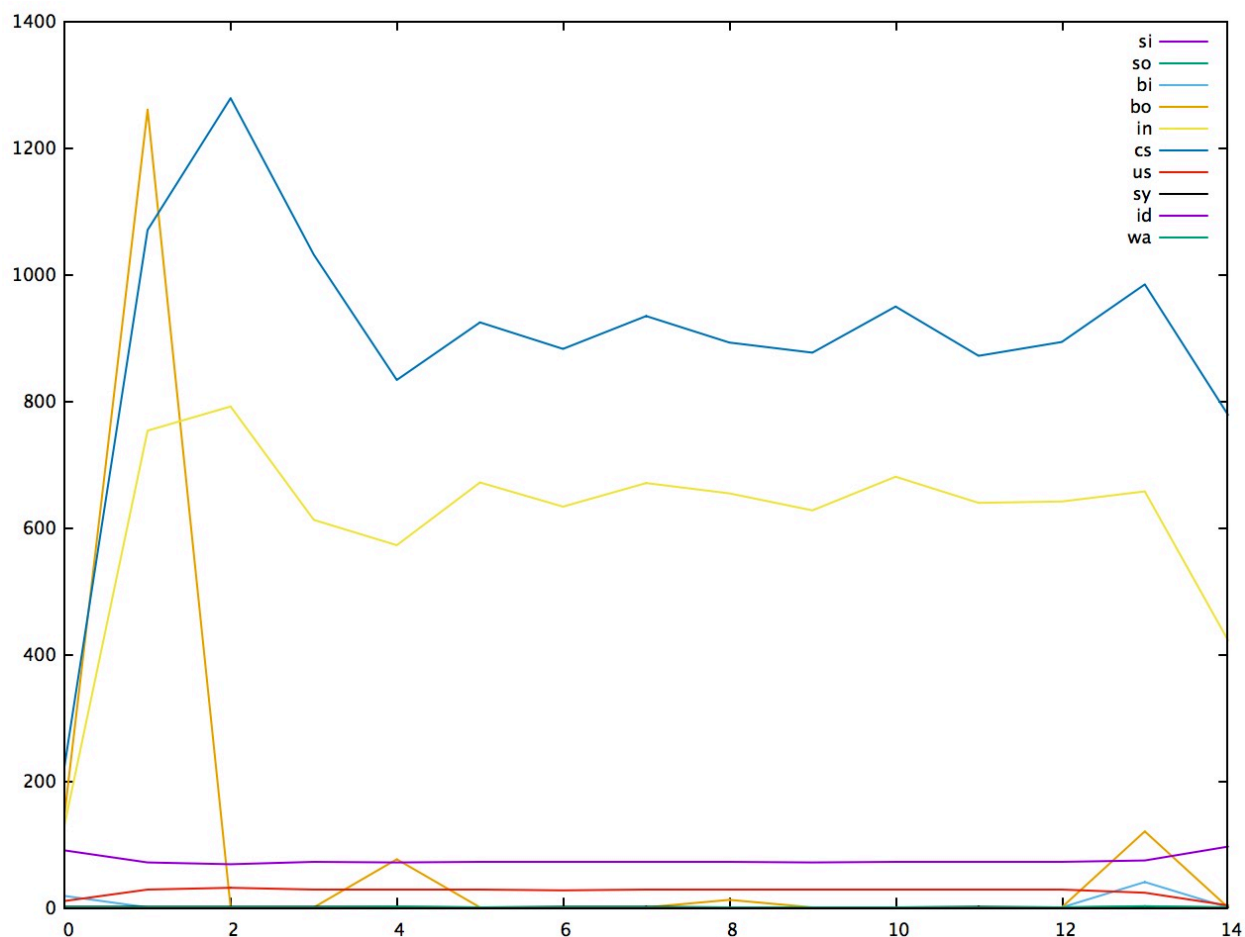
Serban Alexandru 341 C5

## VMSTAT

```
serban@serban-PC ~ $ vmstat
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs us sy id wa st
1  0     0 2827220 105748 704572    0    0    58    82   235  399 20  1 79  1  0
```

- Procs
  - r: The number of processes waiting for run time.
  - b: The number of processes in uninterruptible sleep.
- Memory
  - swpd: the amount of virtual memory used.
  - free: the amount of idle memory.
  - buff: the amount of memory used as buffers.
  - cache: the amount of memory used as cache.
  - inactive: the amount of inactive memory. (-a option)
  - active: the amount of active memory. (-a option)
- Swap (pentru analiza activitatii de swap)
  - si: Amount of memory swapped in from disk (/s).
  - so: Amount of memory swapped to disk (/s).
- IO
  - bi: Blocks received from a block device (blocks/s).
  - bo: Blocks sent to a block device (blocks/s).
- System (pentru analiza activitatii procesorului)
  - in: The number of interrupts per second, including the clock.
  - cs: The number of context switches per second.
- CPU
  - These are percentages of total CPU time.
  - us: Time spent running non-kernel code. (user time, including nice time)
  - sy: Time spent running kernel code. (system time)
  - id: Time spent idle. Prior to Linux 2.5.41, this includes IO-wait time.
  - wa: Time spent waiting for IO. Prior to Linux 2.5.41, included in idle.
  - st: Time stolen from a virtual machine. Prior to Linux 2.6.11, unknown.

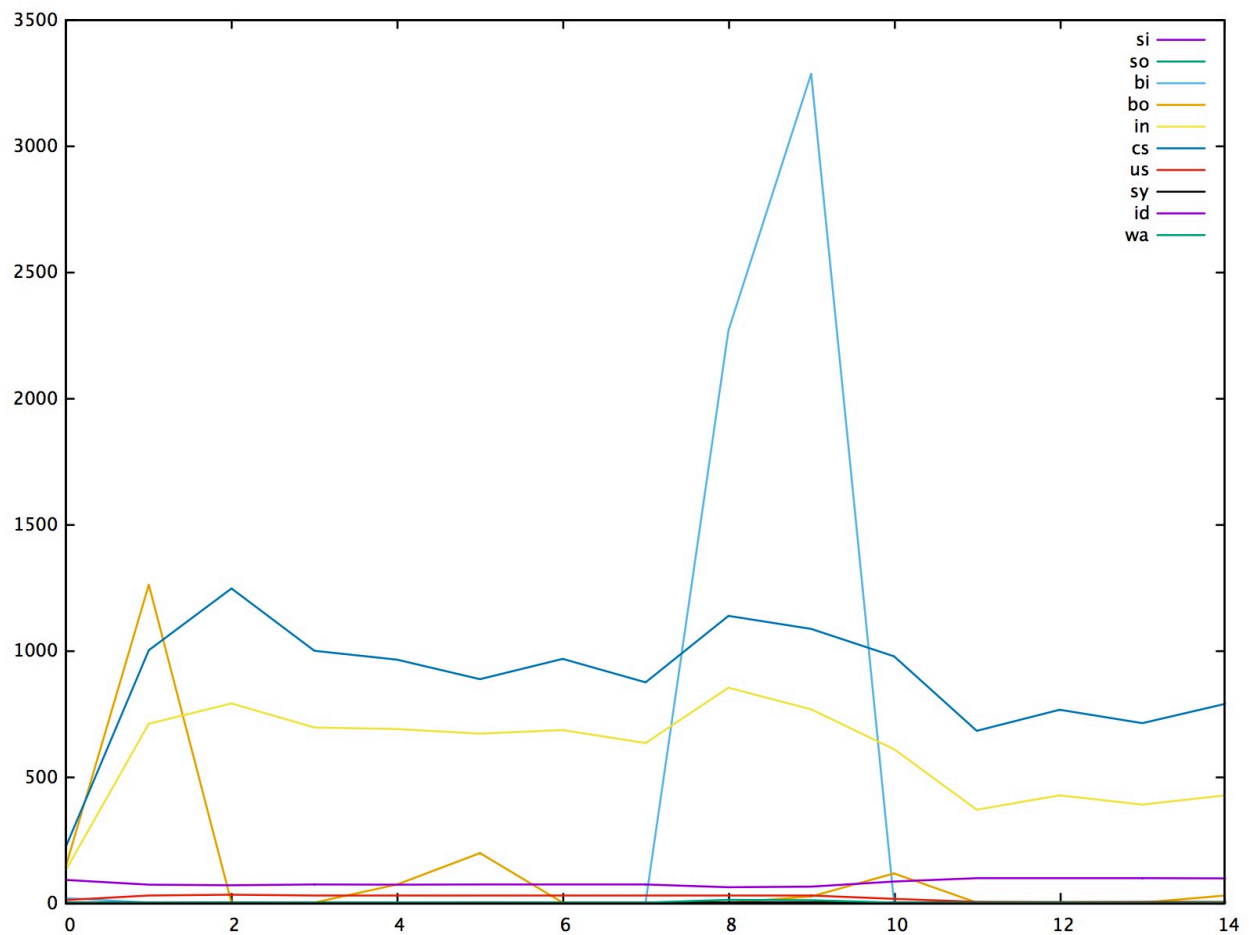
### VMSTAT : limita 128 MB



Putem observa ca :

- la inceput cand programul genereaza numerele putem observa ca sunt foarte multe schimbari de context
- bo este mare deoarece sunt foarte multe blocuri de date trimise catre hdd pentru a fi scrise
- in intreruperile au aceasi crestere ca si schimbarile de context

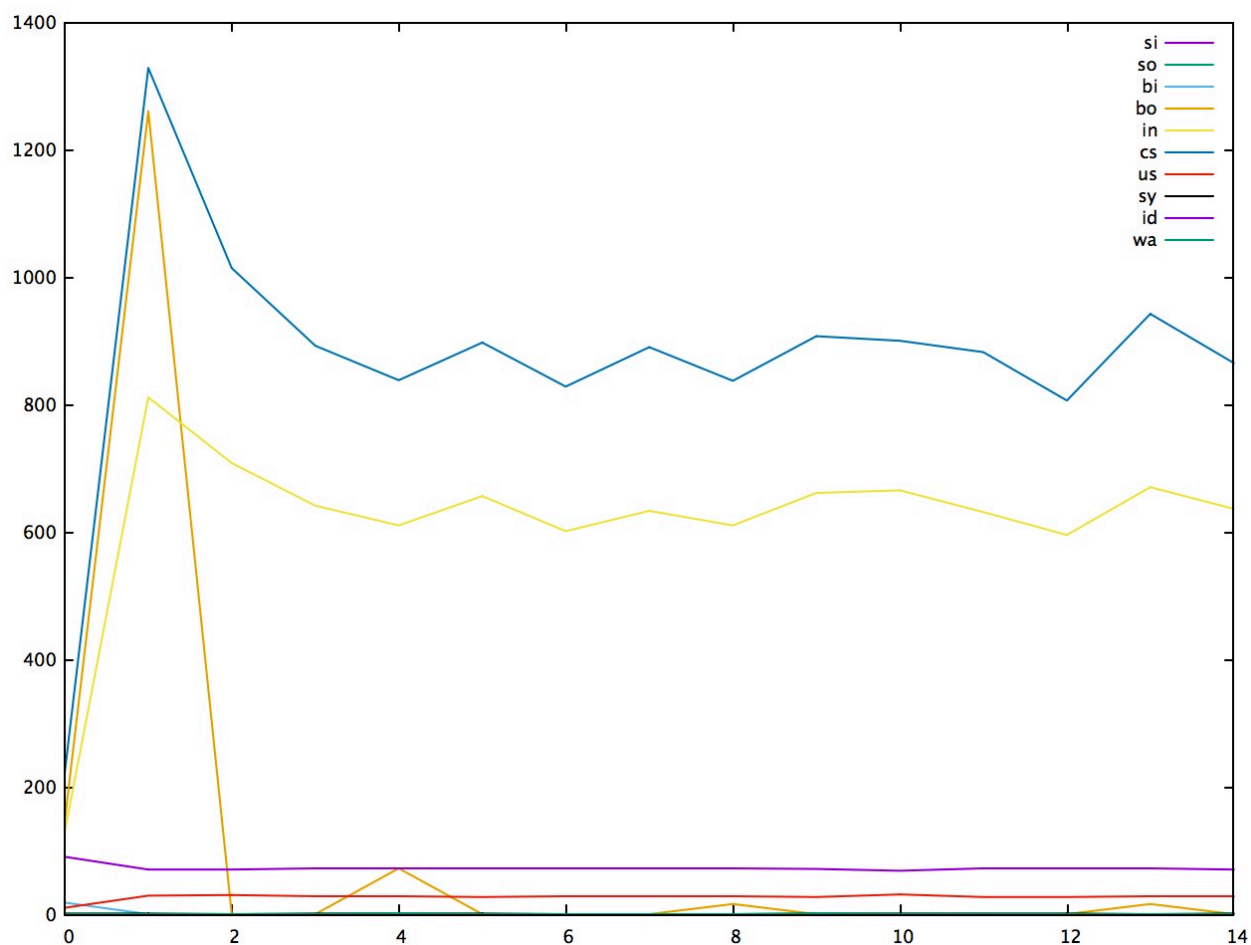
**VMSTAT : limita 256 MB**



Putem observa ca :

- in cazul limitari de 256 de MB se observa un numar mult mai mare de blocuri de intrare - bi
- bo, cs si in : au aceasi variatie ca si in cazul limitarii de 128 MB

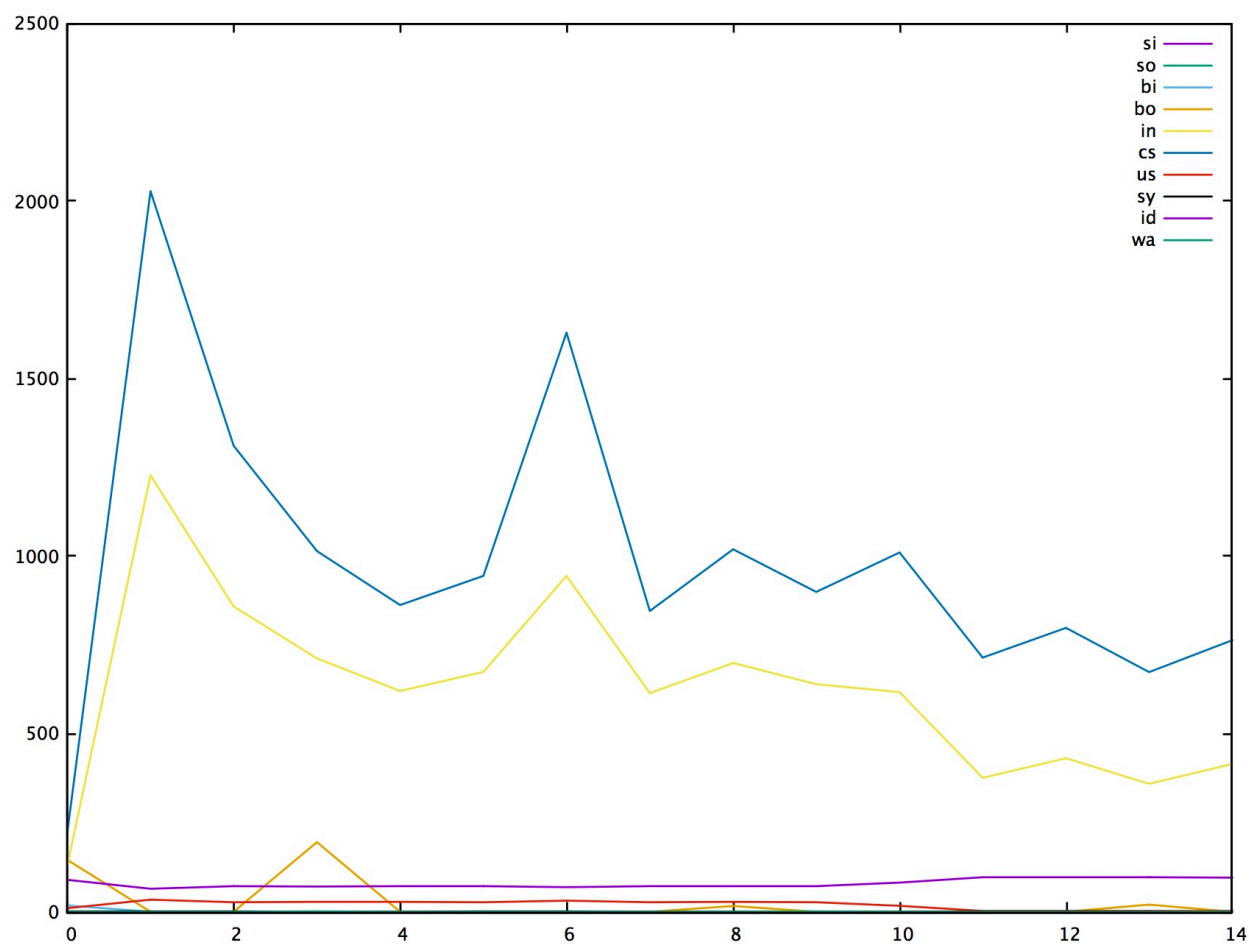
### VMSTAT : limita 512 MB



Putem observa ca :

- bo, cs si in au aceasi variatie ca in cazul limitarilor precedente
- si (Amount of memory swapped in from disk) este un pic mai mare la inceput cand se genereaza fisierul de numere
- nu se observa nicio crestere a bi in ciuda faptului ca avem 2 functii in cod care citesc de 2 ori fisierul generat la punctul 1

## VMSTAT : limita 1024 MB



Putem observa ca :

- cs si in isi pastreaza variatia
- bo este aproape inexistent in ciuda fatului in primele 2 secunde se genereaza fisierul cu numere

## MPSTAT

```
serban@serban-PC ~ $ mpstat
Linux 3.13.0-37-generic (serban-PC)      01/08/2017      _i686_      (4 CPU)

10:58:34 AM  CPU      %usr    %nice    %sys %iowait    %irq    %soft    %steal    %guest    %gnice   %idle
10:58:34 AM  all       19,96     0,06     1,05     0,50     0,00     0,03     0,00     0,00     0,00    78,39
```

**usr** : show the percentage of CPU utilization that occurred while executing at the user level (application)  
(pentru evaluarea timpului de calcul)

**nice** : show the percentage of CPU utilization that occurred while executing at the user level with nice priority

**sys** : show the percentage of CPU utilization that occurred while executing at the system level (kernel)

**iowait** : show the percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request  
(pentru examinarea timpului petrecut la citire/scriere date)

**irq** : show the percentage of time spent by the CPU or CPUs to service hardware interrupts

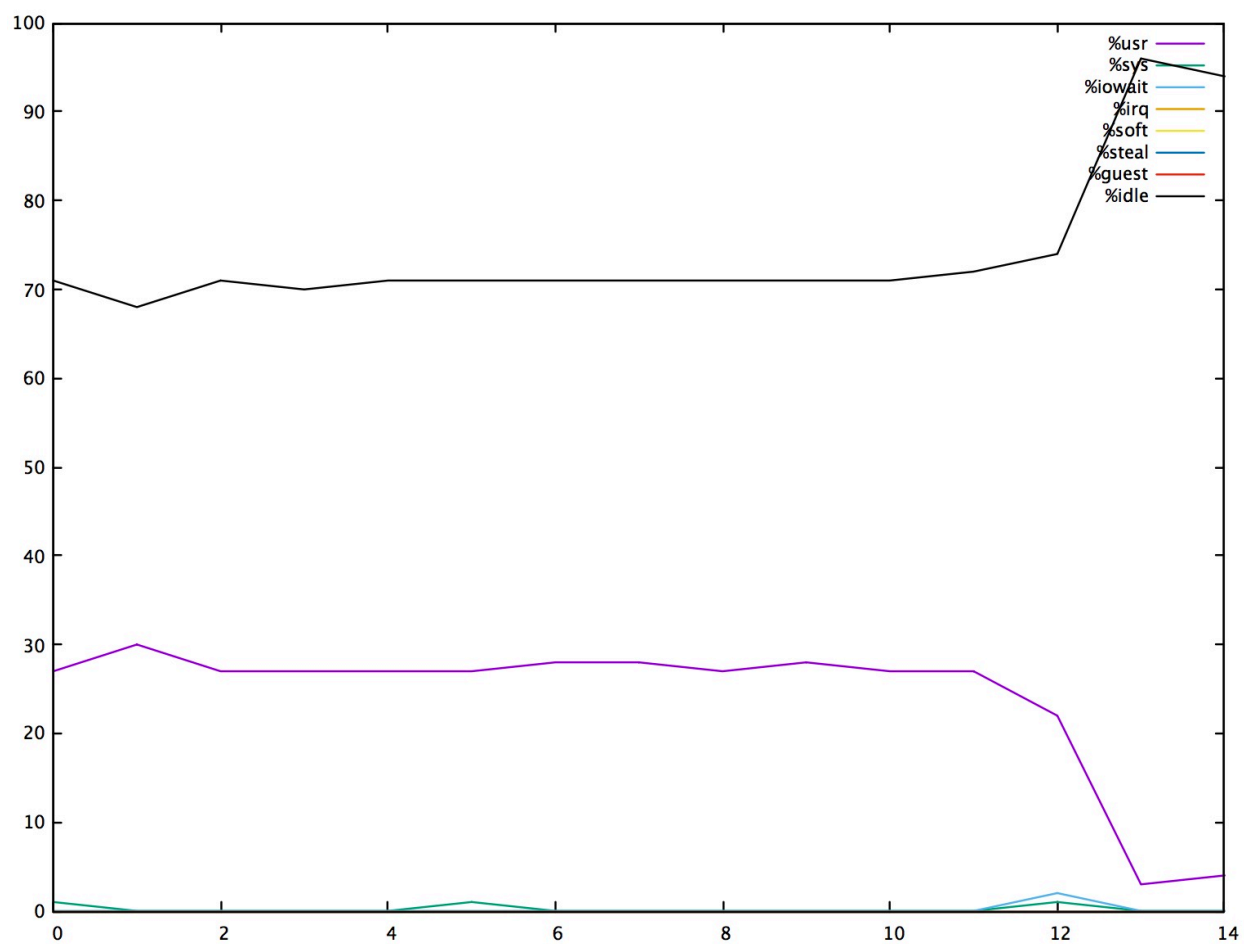
**soft** : show the percentage of time spent by the CPU or CPUs to service software interrupts

**steal** : show the percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor

**guest** : show the percentage of time spent by the CPU or CPUs to run a virtual processor

**idle** : show the percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request  
(pentru analiza perioadei de calcul)

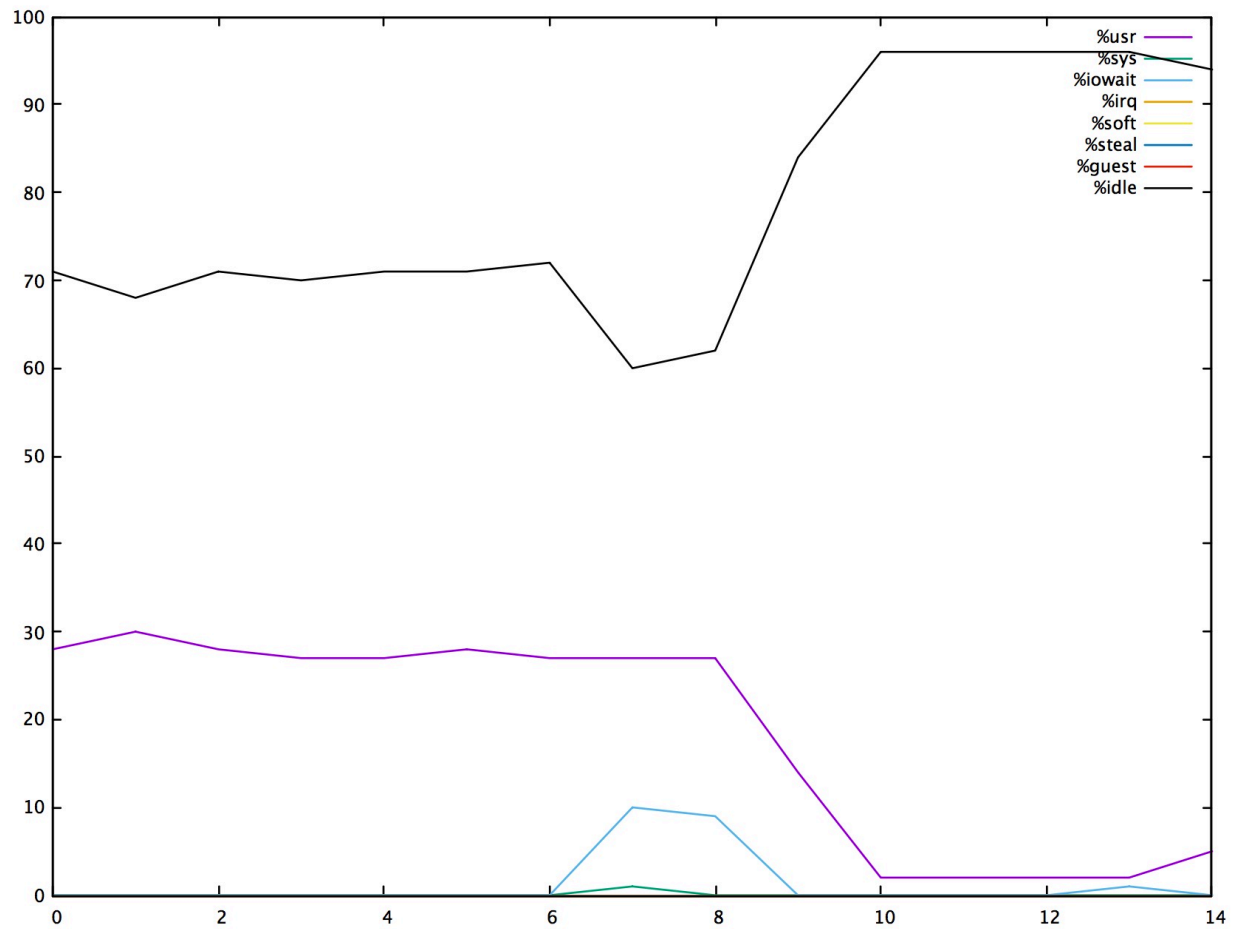
## MPSTAT : limita 128 MB



Putem observa ca :

- procesorul avand 4 core-uri si un singur core folosit 100% la un moment dat, media utilizarii de catre user este un pic peste 25% pentru ca si celelate core-uri au avut activitate pe parcursul rularii

## MPSTAT : limita 256 MB

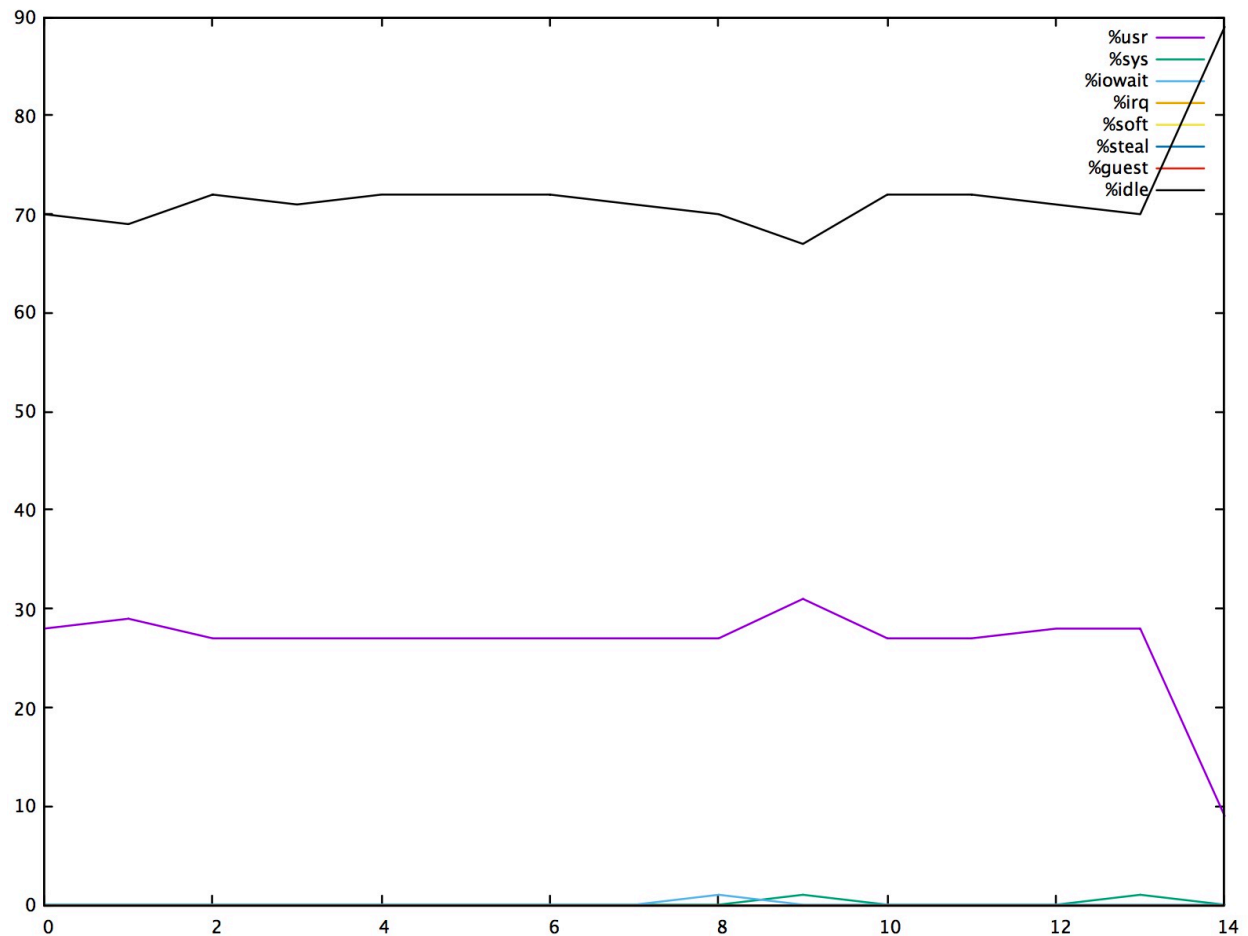


Putem observa ca :

- de asemenea observam activitatea userului de 25%
- timp de 2.5 secunde procesorul este in asteptare, pentru ca datele sunt citite din fisierul generat
- spre terminarea programului se observa o scadere in activitatea usr



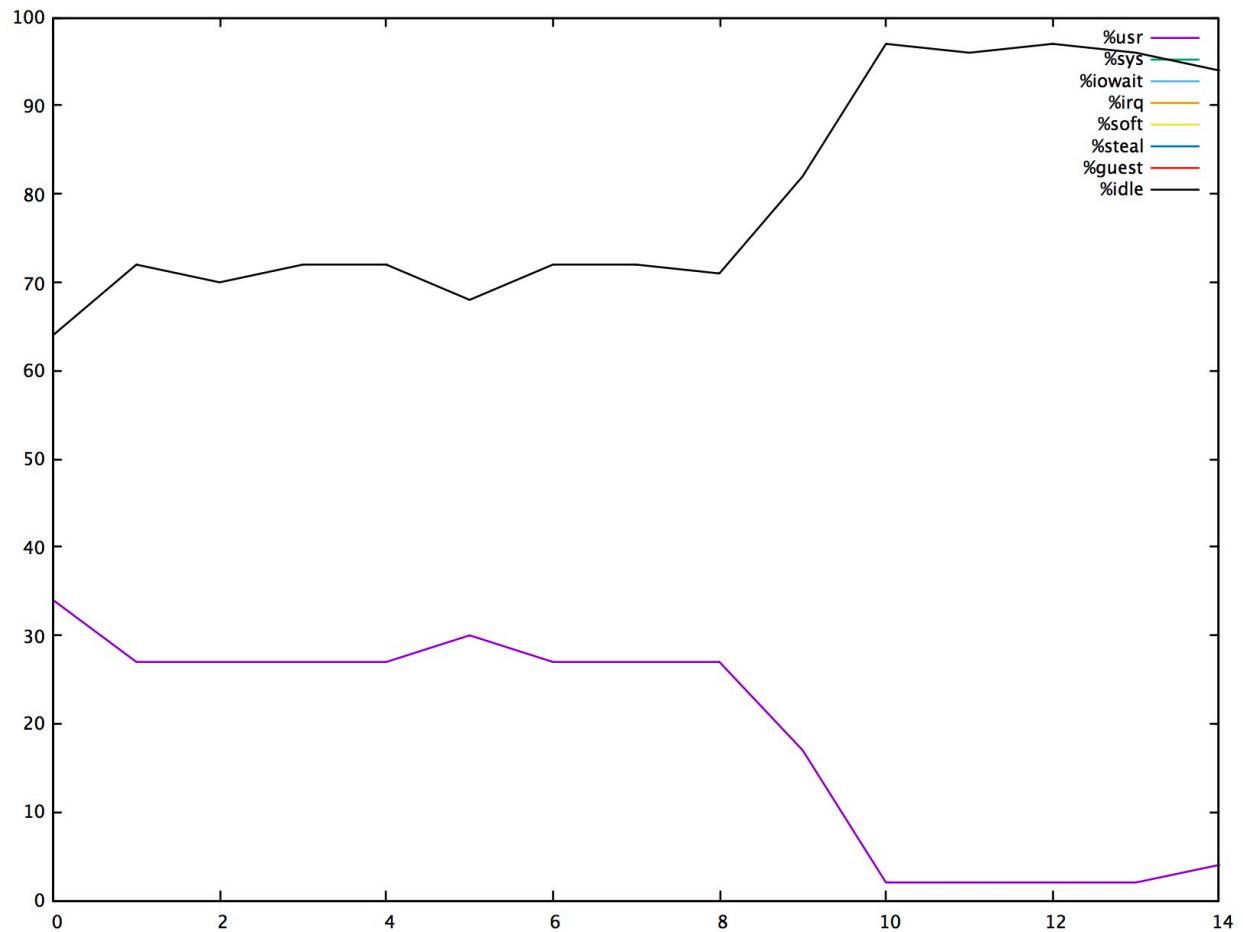
### MPSTAT : limita 512 MB



Putem observa ca :

- in aceasi perioada de timp ca si in cazul limitarii de 256 MB putem observa faptul ca procesorul asteapta citirea datelor din fisierul generat
- se poate observa acest lucru si spre finalul programului cand sunt citite numerele pentru calculul abaterii standard

### MPSTAT : limita 1024 MB



Putem observa ca :

- spre deosebire de celelalte limitari, in cazul limitarii de 1024 MB se poate observa o activitate de peste 30% in prima secunda a rularii (atunci cand sunt generate numerele)
- calculul abaterii standard implica multe operatii si se poate observa un spike de activitate intre secunda 4 si 6, acesta activitate poate fi observata si in cazul limitarii de 512 MB, dar mai tarziu

## Iostat

```
serban@serban-PC ~ $ iostat -d
Linux 3.13.0-37-generic (serban-PC)      01/08/2017      _i686_  (4 CPU)

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                 0,24         4,81         6,93     677195     976660
```

### rkB/s

The number of kilobytes read from the device per second.

### wkB/s

The number of kilobytes written to the device per second.  
(pentru a observa cand sunt datele citite/scrise din hdd)

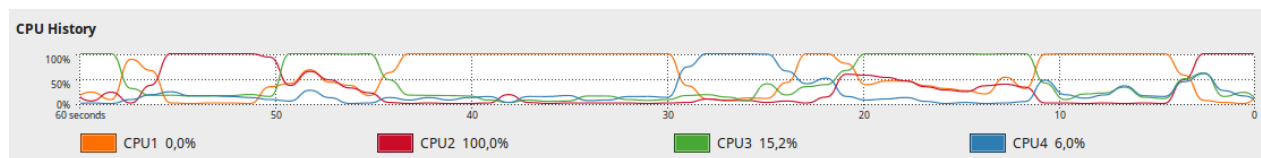
### kB\_read

The total number of kilobytes read.

### kB\_wrtn

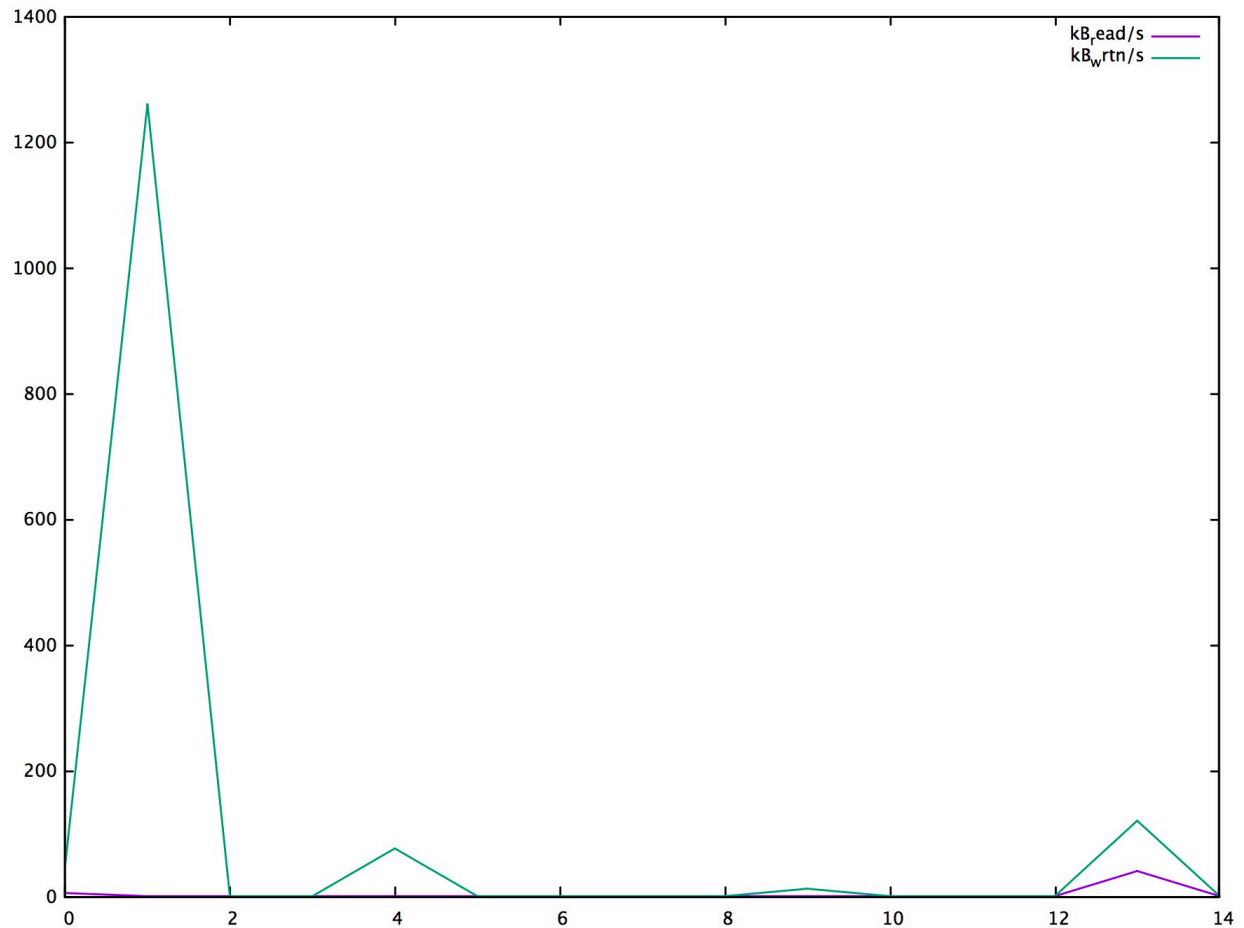
The total number of kilobytes written.

Am observat urmatorul compartament al procesorului



Fiind un procesor quadcore cele 4 procesoare erau folosite pe rand la intensitate maxima.

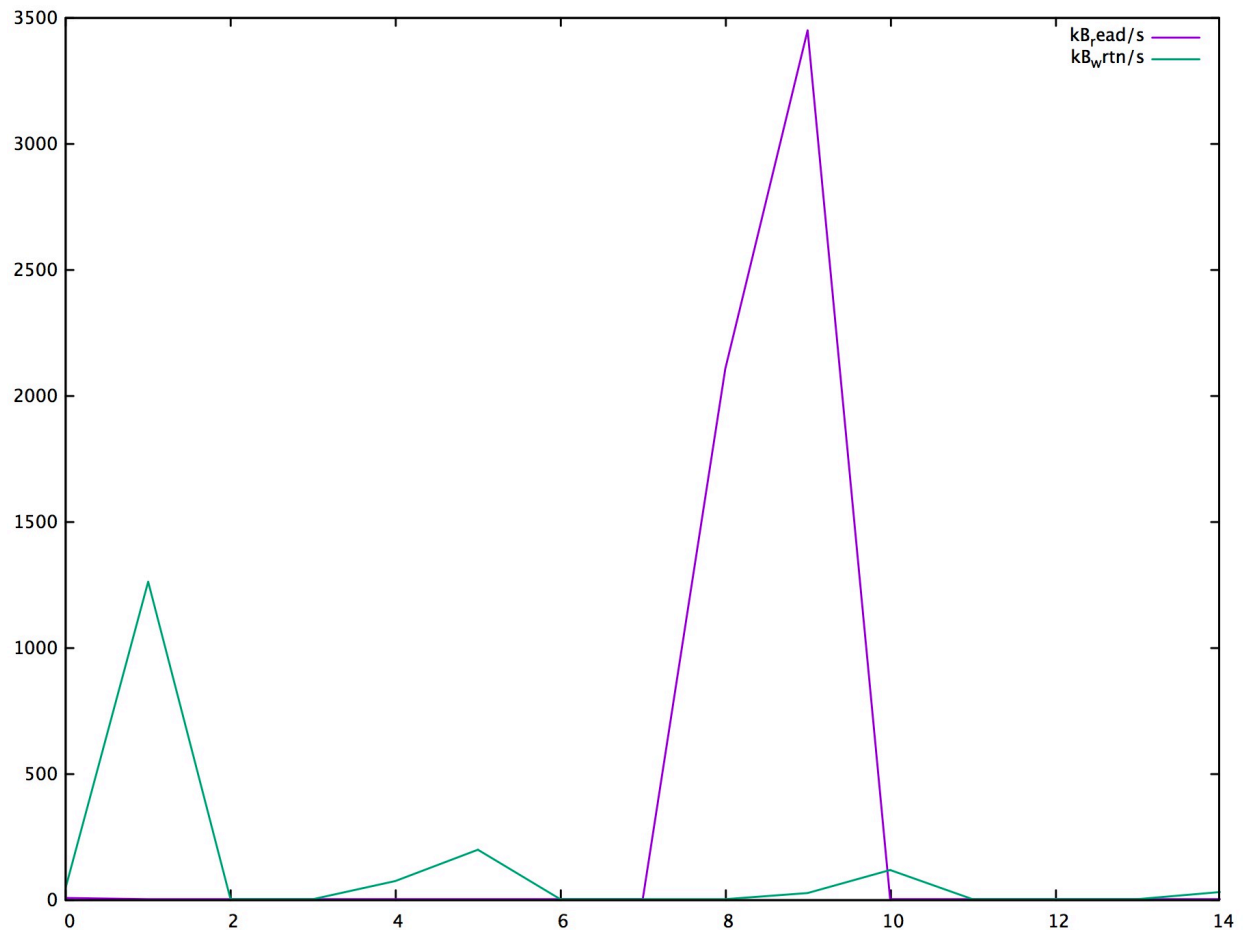
## IOSTAT : limita 128 MB



Putem observa ca :

- in primele 2 secunde a fost creat fisierul cu numere
- celelalte scrieri au fost realizate cand a fost atinsa limita memoriei virtuale de aceea fiecare scriere este insotita si de o citire

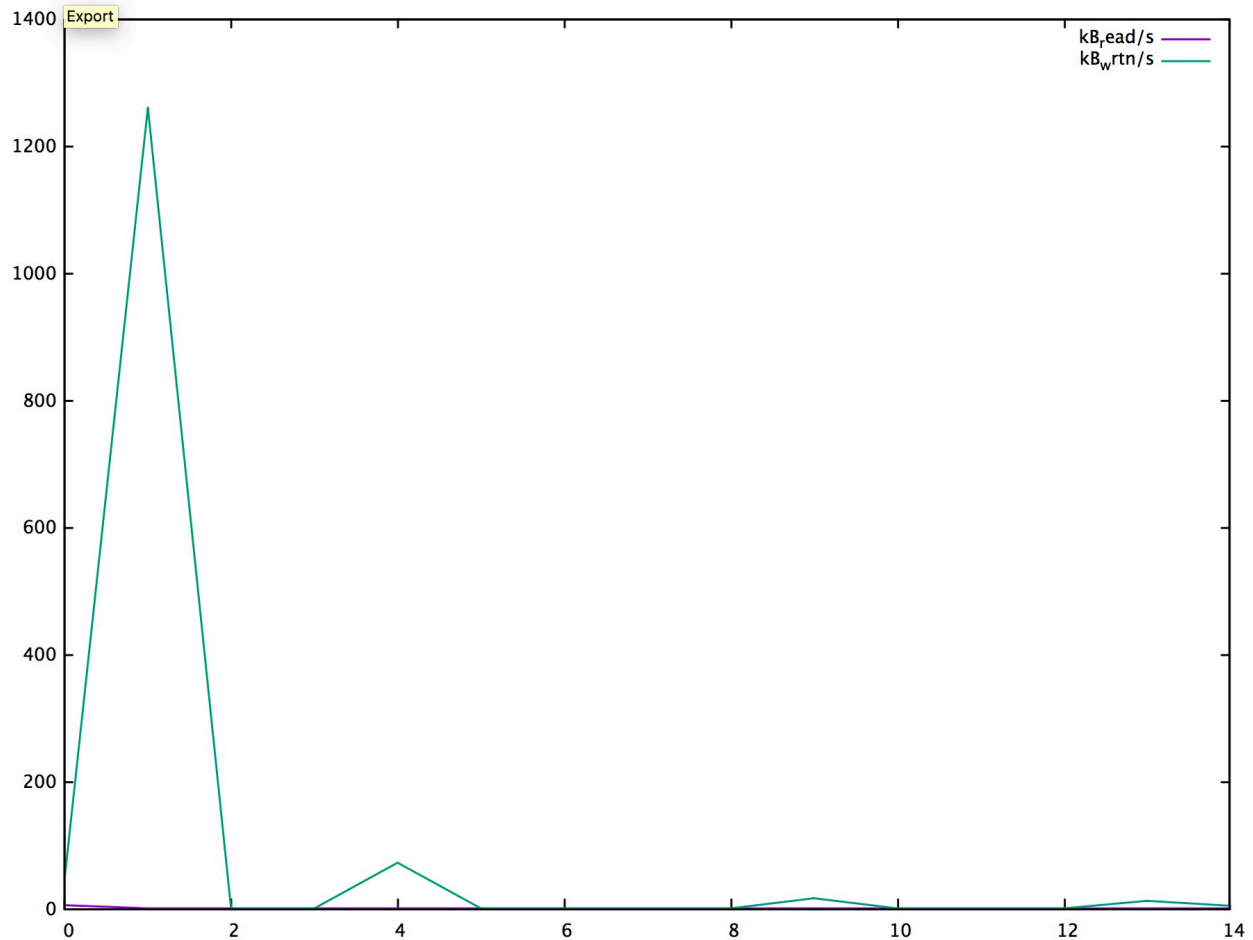
### IOSTAT : limita 256 MB



Putem observa ca :

- in cazul limitarii de 256 se poate observa ca datele nu au fost pastrate in memorie dupa generare si cand a fost calculata abaterea stadard acestea au fost recitite complet din fisier (8-10)

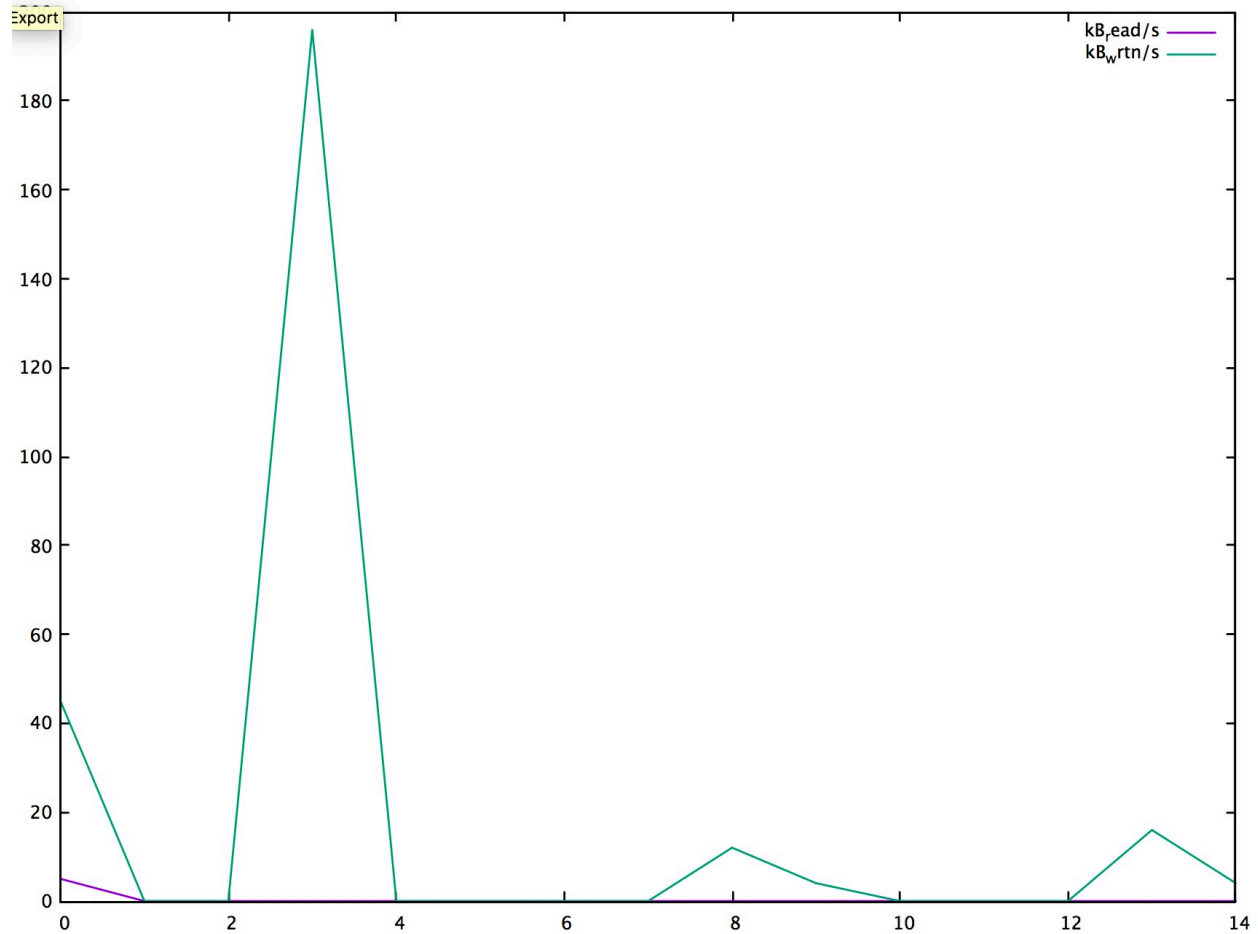
## IOSTAT : limita 512 MB



Putem observa ca :

- in acest caz se observa crearea fisierului in primele 2 secunde si asemanator ca si in cazul limitarii de 128, 3 swap-uri intre hdd si memoria virtuala

## IOSTAT : limita 1024 MB



Putem observa ca :

- asemenea, se observa scrierea numerelor in fisier, de data aceasta este realizata mai tarziu
- avem doar 2 swap-uri datorita limitei mai mari de memorie

## Concluzii :

- am observat comportamentul de RAM Bottleneck (swapping)
- am observat diferenta dintre CPU User Load si CPU System Load
- in cazul limitarii de 256 am analizat un caz de High IO Read Load
- in prima parte a programului am putut observa scenariul de High IO Write Load
- analizand rezultatele obtinute in cadrul limitarii de 256 in cadrul graficului pentru mpstat am vazut cum astepta procesorul in idle ca datele sa fie citite din hdd (CPU Waiting for IO)

## Bibliografie :

- [https://www.thomas-krenn.com/en/wiki/Linux\\_Performance\\_Measurements\\_using\\_vmstat](https://www.thomas-krenn.com/en/wiki/Linux_Performance_Measurements_using_vmstat)
- pagini de manual vmstat mpstat iostat