# 1.Test write speed

|  |
| --- |
| * Test write speed: Read from /dev/zero and write a big file   *# sync; time dd if=/DATA1/test\_64G\_file of=/dev/null bs=4096 count=16777216; sync*  (Wait for the command to finish 🡪 End result will be total time used for writing the file) |

## sync - force transfer data from RAM to DISK:

|  |
| --- |
| Sync Syntax sync [--help] [--version] Why Use the Sync Command - To improve performance, a computer often keeps data in its memory rather than write it to disk because the RAM is much faster than accessing the hard disk.  - This is fine until there is a computer crash. When that happens the data that was held in memory is lost, or the file system is corrupted. The Sync command causes everything to be  written to disk so none of the data is lost. When to Use the Sync Command Usually, computers are shut down in an organized manner.  If the computer is going to be shut down or the processor stopped in an unusual manner, such as when debugging kernel code or in the event of a possible power outage, the Sync command forces an immediate transfer of the code in memory to disk. Because modern computers have potentially large [caches](https://www.lifewire.com/definition-of-cache-2483171), when you use the sync command, wait until all the LEDs that indicate activity stop blinking before turning off the power on the computer. |

## 1.2. dd - convert and copy a [file](https://www.lifewire.com/practical-examples-of-the-zip-command-2201158)

|  |
| --- |
| **dd** [OPTION]...DESCRIPTION Copy a [file](https://www.lifewire.com/examples-linux-unzip-command-2201157), converting and formatting according to the options.  if=Input FILE(read from FILE instead of stdin)  of=Output FILE (write to FILE instead of stdout)  bs=BYTES (read and write BYTES bytes at a time)  count=BLOCKS (copy only BLOCKS input blocks) |

## 1.3.time – measuring the running time of command

# 2. Open another session to PVR-VOD servers and monitor IO stat:

|  |
| --- |
| Open another session to PVR-VOD servers and monitor IO stat:  *# while true; do vmstat -n 1 1 | awk ‘{print $9}’ | sed -n ‘3p’ >> /home/vt\_admin/write\_speed.txt; sleep 1; done*  Press CTRL + C after the first command is finished. |

## 2.1. vmstat

|  |
| --- |
| *SYNOPSIS*  *vmstat [-a] [-n] [-t] [-S unit] [delay [ count]]*  *vmstat [-s] [-n] [-S unit]*  *vmstat [-m] [-n] [delay [ count]]*  *vmstat [-d] [-n] [delay [ count]]*  *vmstat [-p disk partition] [-n] [delay [ count]]*  *Example: vmstat -n 1 1*  - The -n switch causes the header to be displayed only once rather than periodically  - delay is the delay between updates in seconds. If no delay is specified, only one report is printed with the average values since boot.  - count is the number of updates. If no count is specified and delay is defined, count defaults to infinity. |

## 2.2.awk

|  |
| --- |
| [root@TESTBED-VOD-SVR ~]# vmstat -n 1 1  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 454136 779676 26788 30271044 0 0 13 335 0 0 1 2 97 0 0  [root@TESTBED-VOD-SVR ~]# vmstat -n 1 1 | awk '{print $9}'  bi  13  awk '{print $9}' => print the ninth column |

## 2.3.sed

|  |
| --- |
| [root@TESTBED-VOD-SVR ~]# vmstat -n 1 1 | awk '{print $9}' | sed -n '3p'  13 [p - print](http://www.grymoire.com/Unix/Sed.html" \l "toc-uh-9) 3p => print the third row  ' ' => recommend you do use quotes. If you have meta-characters in the command, quotes are necessary |

|  |
| --- |
| *while true; do vmstat -n 1 1 | awk ‘{print $9}’ | sed -n ‘3p’ >> /home/vt\_admin/write\_speed.txt; sleep 1; done*  => result append to text file, sleep 1s each time execute a new command |