NAME

dstat − versatile tool for generating system resource statistics

SYNOPSIS

dstat [−afv] [options..] [delay [count]]

DESCRIPTION

Dstat is a versatile replacement for vmstat, iostat and ifstat. Dstat overcomes some of the limitations and adds some extra features.

Dstat allows you to view all of your system resources instantly, you can eg. compare disk usage in combination with interrupts from your IDE controller, or compare the network bandwidth numbers directly with the disk throughput (in the same interval).

Dstat also cleverly gives you the most detailed information in columns and clearly indicates in what magnitude and unit the output is displayed. Less confusion, less mistakes, more efficient.

Dstat is unique in letting you aggregate block device throughput for a certain diskset or network bandwidth for a group of interfaces, ie. you can see the throughput for all the block devices that make up a single filesystem or storage system.

Dstat allows its data to be directly written to a CSV file to be imported and used by OpenOffice, Gnumeric or Excel to create graphs.

Note

Users of Sleuthkit might find Sleuthkit’s dstat being renamed to datastat to avoid a name conflict. See Debian bug #283709 for more information.

OPTIONS

−c, −−cpu
enable cpu stats (system, user, idle, wait, hardware interrupt, software interrupt)

−C 0,3,total
include cpu0, cpu3 and total (when using −c/−−cpu)

−d, −−disk
enable disk stats (read, write)

−D total,hda
include total and hda (when using −d/−−disk)

−g, −−page
enable page stats (page in, page out)

−i, −−int
enable interrupt stats

−I 5,10 include interrupt 5 and 10 (when using −i/−−int)

−l, −−load
enable load average stats (1 min, 5 mins, 15mins)

−m, −−mem
enable memory stats (used, buffers, cache, free)

−n, −−net
enable network stats (receive, send)

−N eth1,total
: include eth1 and total (when using −n/−−net)

−p, −−proc
: enable process stats (runnable, uninterruptible, new)

−r, −−io
: enable I/O request stats (read, write requests)

−s, −−swap
: enable swap stats (used, free)

−S swap1,total
: include swap1 and total (when using −s/−−swap)

−t, −−time
: enable time/date output

−T, −−epoch
: enable time counter (seconds since epoch)

−y, −−sys
: enable system stats (interrupts, context switches)

−−aio enable aio stats (asynchronous I/O)

−−fs, −−filesystem
: enable filesystem stats (open files, inodes)

−−ipc enable ipc stats (message queue, semaphores, shared memory)

−−lock enable file lock stats (posix, flock, read, write)

−−raw enable raw stats (raw sockets)

−−socket
: enable socket stats (total, tcp, udp, raw, ip−fragments)

−−tcp enable tcp stats (listen, established, syn, time_wait, close)

−−udp enable udp stats (listen, active)

−−unix enable unix stats (datagram, stream, listen, active)

−−vm enable vm stats (hard pagefaults, soft pagefaults, allocated, free)

−−plugin−name
: enable (external) plugins by plugin name, see **PLUGINS** for options

Possible internal stats are
: aio, cpu, cpu24, disk, disk24, disk24old, epoch, fs, int, int24, io, ipc, load, lock, mem, net, page, page24, proc, raw, socket, swap, swapold, sys, tcp, time, udp, unix, vm

−−list list the internal and external plugin names

−a, −−all
: equals −cdngy (default)

−f, −−full
: expand −C, −D, −I, −N and −S discovery lists

−v, −−vmstat
: equals −pmgdsc −D total

−−bits force bits for values expressed in bytes

−−float force float values on screen (mutual exclusive with **−−integer**)

−−integer
force integer values on screen (mutual exclusive with **−−float**)

−−bw, −−blackonwhite
change colors for white background terminal

−−nocolor
disable colors (implies **−−noupdate**)

−−noheaders
disable repetitive headers

−−noupdate
disable intermediate updates when delay > 1

−−output file
write CSV output to file

−−profile
show profiling statistics when exiting dstat

PLUGINS

While anyone can create their own dstat plugins (and contribute them) dstat ships with a number of plugins already that extend its capabilities greatly. Here is an overview of the plugins dstat ships with:

−−battery
battery in percentage (needs ACPI)

−−battery−remain
battery remaining in hours, minutes (needs ACPI)

−−cpufreq
CPU frequency in percentage (needs ACPI)

−−dbus number of dbus connections (needs python−dbus)

−−disk−tps
per disk transactions per second (tps) stats

−−disk−util
per disk utilization in percentage

−−dstat show dstat cputime consumption and latency

−−dstat−cpu
show dstat advanced cpu usage

−−dstat−ctxt
show dstat context switches

−−dstat−mem
show dstat advanced memory usage

−−fan fan speed (needs ACPI)

−−freespace
per filesystem disk usage

−−gpfs GPFS read/write I/O (needs mmpmon)

−−gpfs−ops
GPFS filesystem operations (needs mmpmon)

−−helloworld
Hello world example dstat plugin

−−innodb−buffer
show innodb buffer stats

−−innodb−io
show innodb I/O stats

−−innodb−ops
show innodb operations counters

−−lustre
show lustre I/O throughput

−−memcache−hits
show the number of hits and misses from memcache

−−mysql5−cmds
show the MySQL5 command stats

−−mysql5−conn
show the MySQL5 connection stats

−−mysql5−io
show the MySQL5 I/O stats

−−mysql5−keys
show the MySQL5 keys stats

−−mysql−io
show the MySQL I/O stats

−−mysql−keys
show the MySQL keys stats

−−net−packets
show the number of packets received and transmitted

−−nfs3 show NFS v3 client operations

−−nfs3−ops
show extended NFS v3 client operations

−−nfsd3
show NFS v3 server operations

−−nfsd3−ops
show extended NFS v3 server operations

−−ntp show NTP time from an NTP server

−−postfix
show postfix queue sizes (needs postfix)

−−power
show power usage

−−proc−count
show total number of processes

−−qmail
show qmail queue sizes (needs qmail)

−−rpc show RPC client calls stats

−−rpcd show RPC server calls stats

−−sendmail
show sendmail queue size (needs sendmail)

−−snooze
show number of ticks per second

−−squid
show squid usage statistics

−−test show test plugin output

−−thermal
system temperature sensors

−−top−bio
show most expensive block I/O process

−−top−bio−adv
show most expensive block I/O process (incl. pid and other stats)

−−top−childwait
show process waiting for child the most

−−top−cpu
show most expensive CPU process

−−top−cpu−adv
show most expensive CPU process (incl. pid and other stats)

−−top−cputime
show process using the most CPU time (in ms)

−−top−cputime−avg
show process with the highest average timeslice (in ms)

−−top−int
show most frequent interrupt

−−top−io
show most expensive I/O process

−−top−io−adv
show most expensive I/O process (incl. pid and other stats)

−−top−latency
show process with highest total latency (in ms)

−−top−latency−avg
show process with the highest average latency (in ms)

−−top−mem
show process using the most memory

−−top−oom
show process that will be killed by OOM the first

−−utmp
show number of utmp connections (needs python−utmp)

−−vmk−hba
show VMware ESX kernel vmhba stats

−−vmk−int
show VMware ESX kernel interrupt stats

−−vmk−nic
show VMware ESX kernel port stats

−−vm−memctl
show ballooning status inside VMware guests

−−vz−cpu
show CPU usage per OpenVZ guest

−−vz−io
show I/O usage per OpenVZ guest

−−vz−ubc
show OpenVZ user beancounters

−−wifi wireless link quality and signal to noise ratio

ARGUMENTS

delay is the delay in seconds between each update

count is the number of updates to display before exiting

The default delay is 1 and count is unspecified (unlimited)

INTERMEDIATE UPDATES

When invoking dstat with a **delay** greater than 1 and without the **−−noupdate** option, it will show intermediate updates, ie. the first time a 1 sec average, the second update a 2 second average, etc. until the delay has been reached.

So in case you specified a delay of 10, **the 9 intermediate updates are NOT snapshots**, they are averages over the time that passed since the last final update. The end result is that you get a 10 second average on a new line, just like with vmstat.

EXAMPLES

Using dstat to relate disk−throughput with network−usage (eth0), total CPU−usage and system counters:

```
dstat −dnyc −N eth0 −C total −f 5
```

Checking dstat’s behaviour and the system impact of dstat:

```
dstat −taf −−debug
```

Using the time plugin together with cpu, net, disk, system, load, proc and top_cpu plugins:

```
dstat −tcndylp −−top−cpu
```

this is identical to

```
dstat −−time −−cpu −−net −−disk −−sys −−load −−proc −−top−cpu
```

Using dstat to relate cpu stats with interrupts per device:

```
dstat −tcyif
```

BUGS

Since it is practically impossible to test dstat on every possible permutation of kernel, python or distribution version, I need your help and your feedback to fix the remaining problems. If you have improvements or bugreports, please send them to: [1]*dag@wieers.com*

Note

Please see the TODO file for known bugs and future plans.

FILES

Paths that may contain external dstat_*.py plugins:

~/.dstat/
(path of binary)/plugins/
/usr/share/dstat/
/usr/local/share/dstat/

SEE ALSO

Performance tools

ifstat(1), iftop(8), iostat(1), mpstat(1), netstat(1), nfsstat(1), nstat, vmstat(1), xosview(1)

Debugging tools

htop(1), lslk(1), lsof(8), top(1)

Process tracing

ltrace(1), pmap(1), ps(1), pstack(1), strace(1)

Binary debugging

ldd(1), file(1), nm(1), objdump(1), readelf(1)

Memory usage tools

free(1), memusage, memusagestat, slabtop(1)

Accounting tools

dump−acct, dump−utmp, sa(8)

Hardware debugging tools

dmidecode, ifinfo(1), lsdev(1), lshal(1), lshw(1), lsmod(8), lspci(8), lsusb(8), smartctl(8), x86info(1)

Application debugging

mailstats(8), qshape(1)

Xorg related tools

xdpyinfo(1), xrestop(1)

Other useful info

collectl(1), proc(5), procinfo(8)

AUTHOR

Written by Dag Wieers [1]*dag@wieers.com*

Homepage at [2]*http://dag.wieers.com/home−made/dstat/*

This manpage was initially written by Andrew Pollock [3]*apollock@debian.org* for the Debian GNU/Linux system.

REFERENCES

1. dag@wieers.com
   mailto:dag@wieers.com

2. http://dag.wieers.com/home−made/dstat/
   http://dag.wieers.com/home−made/dstat/

3. apollock@debian.org
   mailto:apollock@debian.org