Linux Performance Profiling and Monitoring

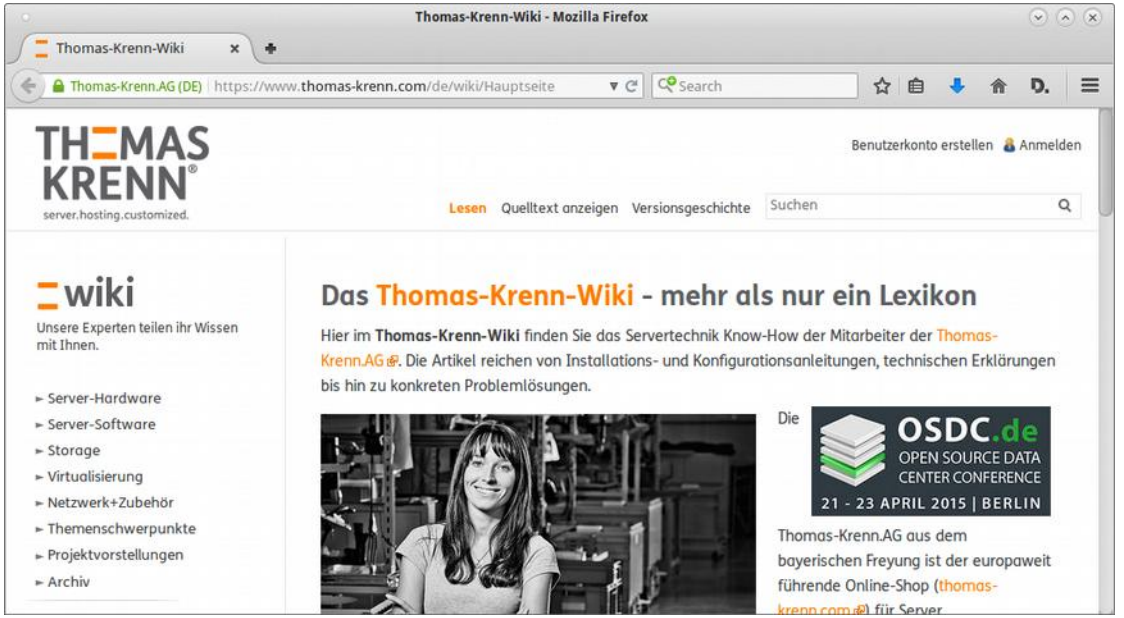
Georg Schönberger

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE
21 - 23 APRIL 2015 | BERLIN

THOMAS KRENN®
server.hosting.customized.

Thomas-Krenn.AG

- A server manufacturer in Bavaria
- Well visited knowledge base, Thomas-Krenn Wiki

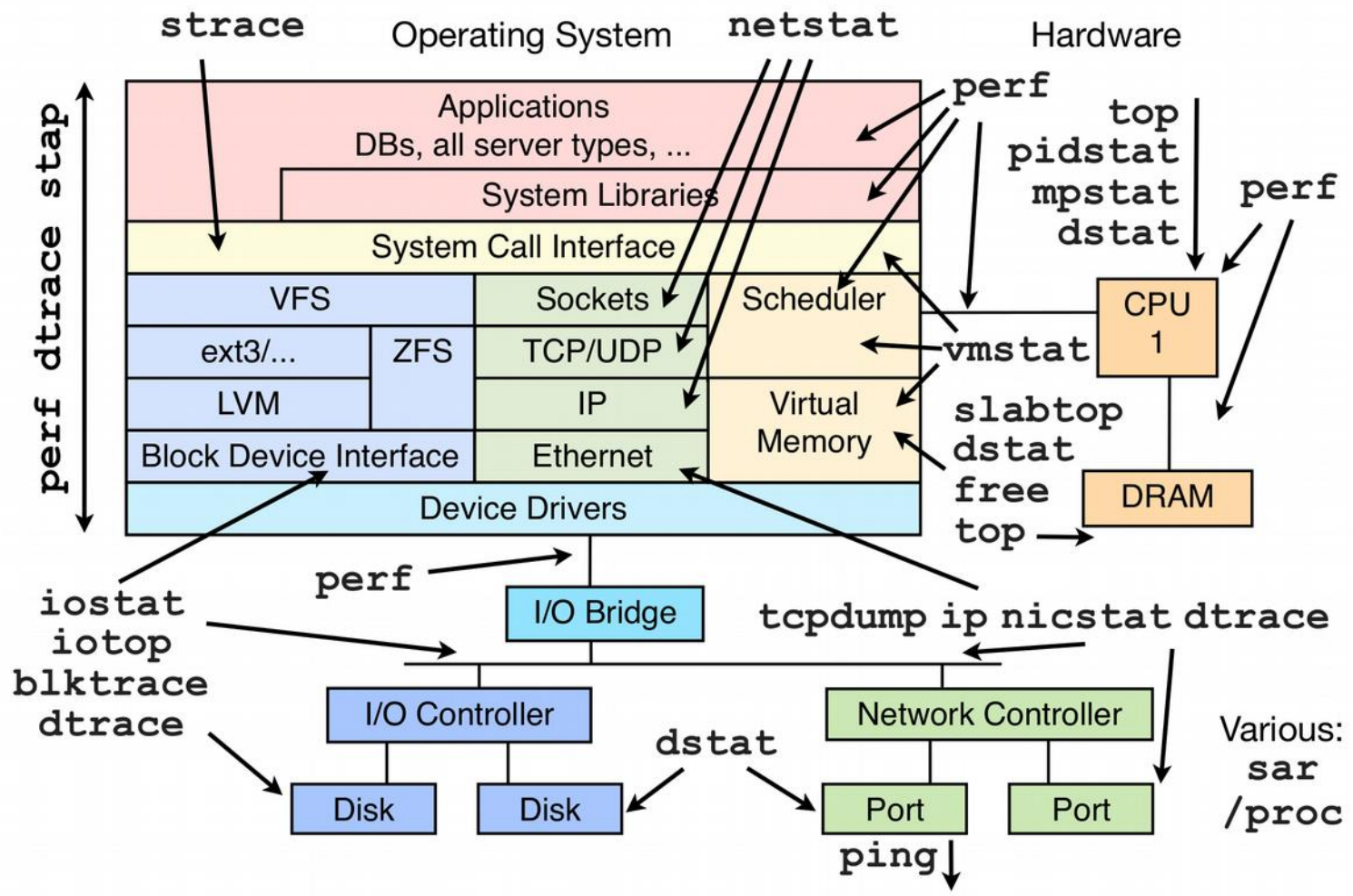


Agenda

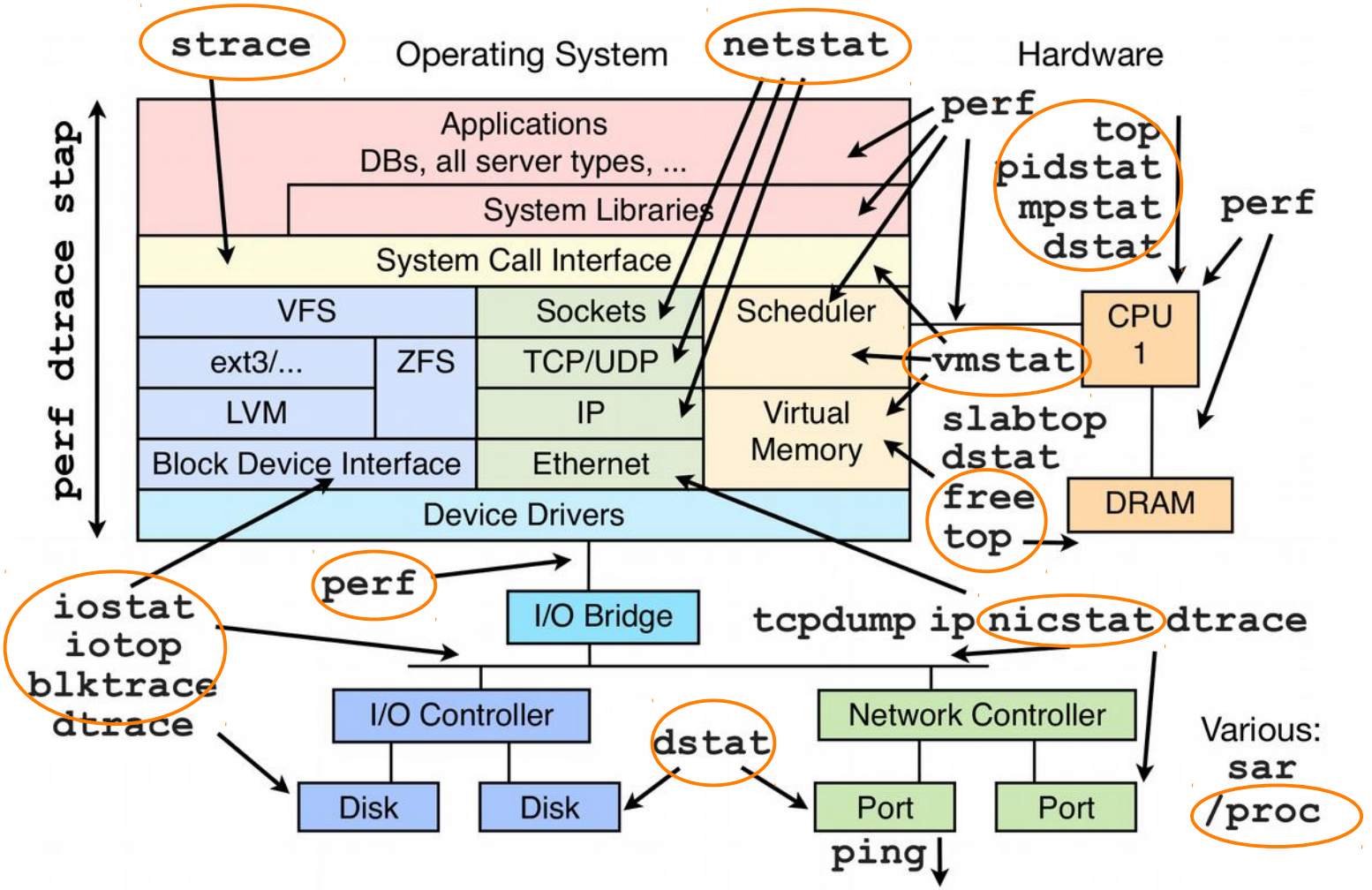
THOMAS KRENN®
server.hosting.customized.

- Collect Statistics
  - Sysstat Package
  - `dstat`
  - `nicstat`
  - `/proc` → raw counters
  - `sar` and `sadc`
- Watch online
  - `top`
  - `htop`
  - `iotop`
  - `iftop`
- Tracing
  - `perf_events`
  - `ftrace`
  - `perf-tools`
  - Flame graphs
- LXC
- MySQL
  - Slow Query Log
  - `innotop`

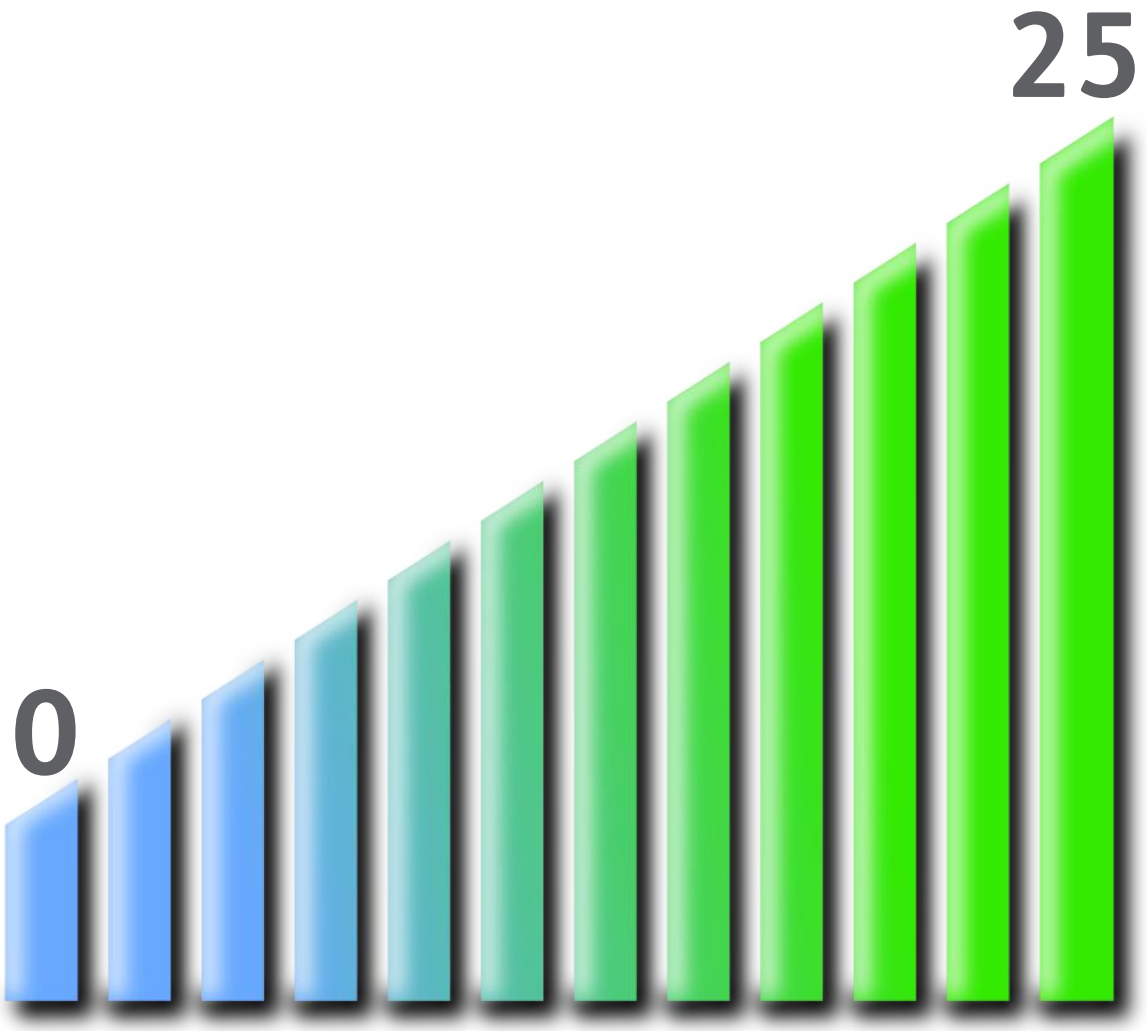
```
# find / -type f -name statistics
```



From Brendan Gregg, http://www.brendangregg.com/linuxperf.html



From Brendan Gregg, http://www.brendangregg.com/linuxperf.html


25
0

mpstat

THOMAS KRENN®
server.hosting.customized.

- Without Interval/Count → since system startup
- CPU usage per Core
  - Including Hyperthreading

```
# lscpu | grep -E 'core|socket'
Thread(s) per core:    2
Core(s) per socket:    2
```

- Check how well usage is balanced

```
# mpstat -P ALL
Linux 3.13.0-48-generic (X220) 2015-04-14      _x86_64_   (4 CPU)
14:28:21     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
14:28:21     all   11,59    0,09    3,62    0,03    0,00    0,04    0,00    0,00    0,00   84,64
14:28:21       0    6,45    0,05    1,87    0,04    0,00    0,07    0,00    0,00    0,00   91,53
14:28:21       1   16,44    0,11    5,56    0,01    0,00    0,00    0,00    0,00    0,00   77,89
14:28:21       2   17,15    0,14    5,55    0,03    0,00    0,05    0,00    0,00    0,00   77,08
14:28:21       3   16,27    0,11    4,89    0,01    0,00    0,02    0,00    0,00    0,00   78,70
```

mpstat

THOMAS KRENN®
server.hosting.customized.

```
# mpstat -P ALL 1 2
Linux 3.13.0-48-generic (X220) 2015-04-14      _x86_64_   (4 CPU)

15:24:44     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
15:24:45     all    5,21    0,00    7,12   17,81    0,00    0,27    0,00    0,00    0,00   69,59
15:24:45       0    1,43    0,00    1,43    0,00    0,00
15:24:45       1   11,88    0,00   23,76   64,36    0,00                                    0,00
15:24:45       2    4,12    0,00    1,03    0,00    0,00                                   94,85
15:24:45       3    3,03    0,00    1,01    0,00    0,00                                   95,96

15:24:45     CPU    %usr   %nice    %sys %iowait                                    %gnice   %idle
15:24:46     all    5,74    0,00    7,10   17,76    0,00                                    0,00   68,85
15:24:46       0    2,99    0,00    1,49    0,00    0,00    2,99    0,00    0,00    0,00   92,54
15:24:46       1   11,88    0,00   23,76   64,36    0,00    0,00    0,00    0,00    0,00    0,00
15:24:46       2    6,00    0,00    1,00    0,00    0,00    0,00    0,00    0,00    0,00   93,00
15:24:46       3    1,01    0,00    1,01    0,00    0,00    0,00    0,00    0,00    0,00   97,98
```

Core 1 is not idle and also deals with %iowait

vmstat

- High Level Statistics about
  - Virtual memory
  - Swap/Paging
  - I/O statistics
  - System interrupts and context switches
  - CPU statistics

```
# 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
 3  0    172 371856 137088 3125664    0    0     0 153060 7618 7059 17  9 56 17  0
 3  0    172 416596 137096 3125704    0    0     0 163420 8689 7419 11 10 61 17  0
 0  0    172 451716 137096 3089916    0    0     0     0  396 1848  3  1 96  0  0
 0  0    172 413916 137108 3118796    0    0     0    52  502 2218  9  2 90  0  0
 2  0    172 399756 137108 3118860    0    0 284884     0 14830 10941 10 13 66 12  0
 1  1    172 364948 137108 3118988    0    0 310792     0 16204 12738 20 13 53 14  0
```

vmstat

THOMAS KRENN®
server.hosting.customized.

Memory statistics

- `buff` Raw disk blocks like filesystem metadata
- `cache` Memory used for data information, pages with actual contents

```
$ 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    172 607760 182172 3313684    0    0   159   496  154  222 18  6 76  0  0
 0  0    172 607628 182172 3313684    0    0     0    52  387 2008  4  2 95  0  0
 0  0    172 607348 182172 3313684    0    0     0     0  397 2034  4  1 95  0  0
 0  0    172 606448 182172 3313684    0    0     0     0  378 1896  4  2 94  0  0
$ free
             total       used       free     shared    buffers     cached
Mem:       8056664    7450316     606348     491820     182172    3313684
-/+ buffers/cache:    3954460    4102204
Swap:      1048572        172    1048400
```

vmstat

THOMAS KRENN®
server.hosting.customized.

- Process related fields
  - r The number of runnable processes (running or waiting for run time)
    - If high → indicator for saturation
  - b The number of processes in uninterruptible sleep
    - Mostly waiting for I/O

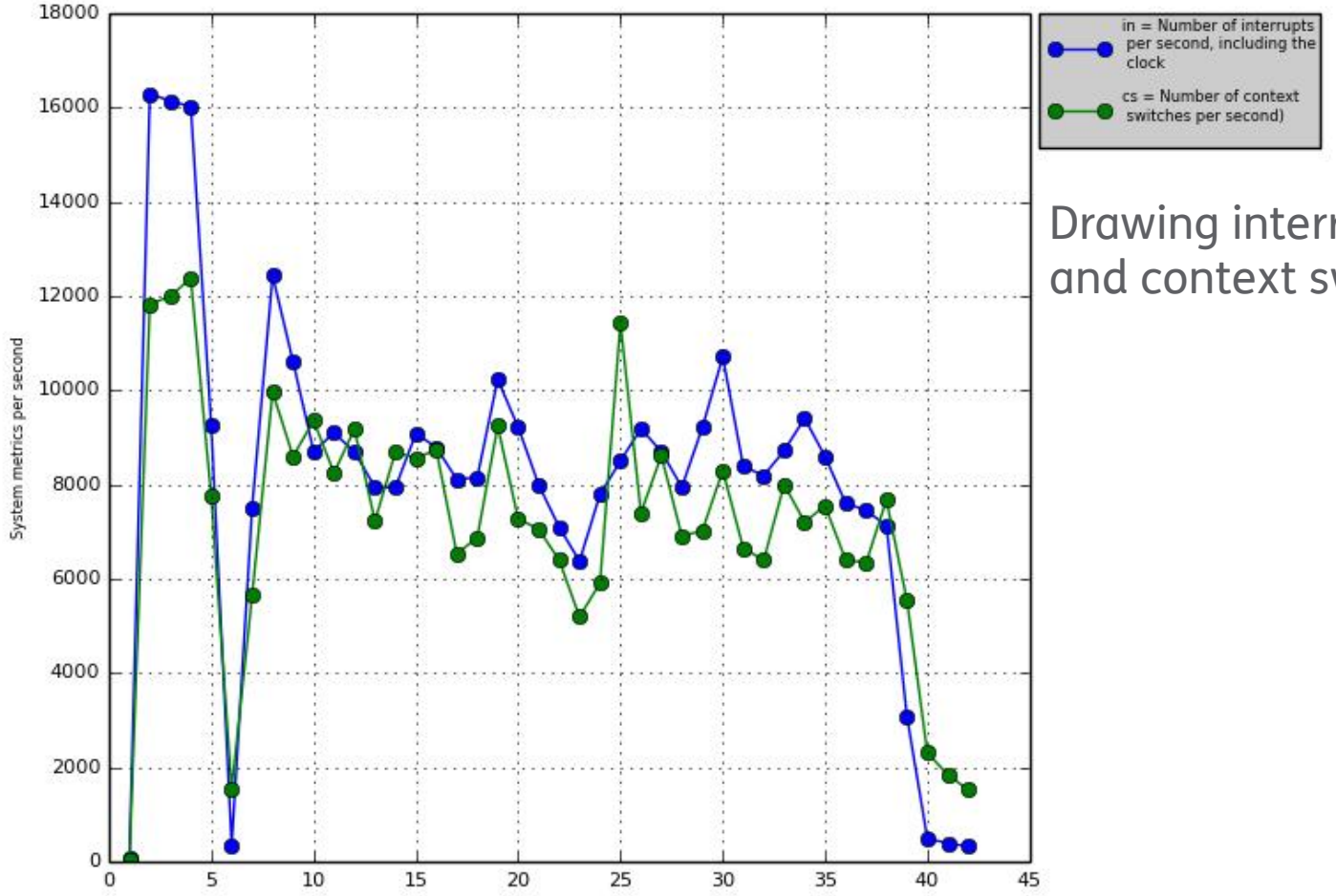
```
# vmstat 1
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd                                  bi    bo   in   cs us sy id wa st
[...]
 0  1                                  0     167524 9029 6955  6  6 70 18  0
 0  1    172                           0     0 138340 8133 6165  7  7 68 19  0
$ ps -eo ppid,pid,user,stat,pcpu,comm,wchan:32 | grep ext4
[...]
 7159  7161 root     Ds    3.2 fio            ext4_file_write
 7159  7162 root     Ds    3.2 fio            ext4_file_write
 7159  7164 root     Ds    3.2 fio            ext4_file_write
```

Processes doing I/O can bei in waiting state

Kernel function process is sleeping on

vmstat plots

THOMAS KRENN®
server.hosting.customized.



Drawing interrupts and context switches

https://clusterbuffer.wordpress.com/admin-tools/vmstat-plotter/

But we are not satisfied
with summaries and overviews...

What is PID 9059 doing?

pidstat

THOMAS KRENN®
server.hosting.customized.

- Report statistics for tasks being managed by kernel
- CPU bound → identify peak activity

```
$ top -b -n 1 -d 2 -o %CPU | head
[...]
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 9059 gschoenb  20   0   47532  21132   2444 R  96,9  0,3   0:02.14 python
    1 root      20   0   33880   3256   1500 S   0,0  0,0   0:02.35 init
$ pidstat -p 9059 -u 1 -l
Linux 3.13.0-48-generic (X220) 2015-04-15      _x86_64_   (4 CPU)

10:11:04      UID       PID    %usr %system  %guest    %CPU   CPU  Command
10:11:05     1000      9059  100,00    0,00    0,00  100,00     0  python ijk-matrix.py
-i matrix.in
10:11:06     1000      9059  100,00    0,00    0,00  100,00     0  python ijk-matrix.py
-i matrix.in
10:11:07     1000      9059  10                                 0  python ijk-matrix.py
-i matrix.in
```

Even check command line arguments!

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

pidstat

THOMAS KRENN®
server.hosting.customized.

_ I/O bound → device report

```
# mpstat -P ALL 1
10:25:31     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:25:32     all   14,88    0,00    9,40   13,84    0,00    1,04    0,00    0,00    0,00   60,84
10:25:32       0   22,45    0,00    1,02    0,00    0,00    0,00    0,00    0,00    0,00   76,53
10:25:32       1   13,73    0,00   34,31   51,96    0,00    0,[illegible]                    0,00
10:25:32       2   17,86    0,00    0,00    0,00    [illegible]    3,[illegible]                   78,57
10:25:32       3    6,12    0,00    0,00    0,00    0,00    [illegible]                   93,88
# pidstat -d 1
Linux 3.13.0-48-generic (X220) 2015-04-15      _x86_64_   (4 CPU)

10:26:35      UID       PID   kB_rd/s   kB_wr/s kB_ccwr/s  Command
10:26:36        0      9208      0,00   2303,85      0,00  fio
10:26:36        0      9209      0,00   2996,15      0,00  fio
10:26:36        0      9210      0,00   2023,08      0,00  fio
10:26:36        0      9211      0,00   1284,62      0,00  fio
```

Which process is causing %iowait?

Device report reveals command and I/O

pidstat

THOMAS KRENN®
server.hosting.customized.

- How much memory is PID 8461 using?
  - Major faults require I/O operations, good indicator you need more RAM!

```
# pidstat -r -p 8461 1 3
Linux 3.13.0-49-generic (X220) 2015-04-21    _x86_64_ (4 CPU)

10:09:06      UID       PID  minflt/s  majflt/s     VSZ    RSS   %MEM  Command
10:09:07     1000      8461      8,00      0,00 2018384 786688  9,76  firefox
10:09:08     1000      8461     11,00      0,00 2018384 786688  9,76  firefox
10:09:09     1000      8461     23,00      0,00 2018448 786892  9,77  firefox
Average:     1000      8461     14,00      0,00 2018405 786756  9,77  firefox
```

Minor and major page faults

Current used share of physical memory

iostat

- I/O subsystem statistics
- CPU or device utilization report
- Without argument → summary since boot
  - Skip that with `-y` option

```
# iostat
Linux 3.13.0-48-generic (X220) 2015-04-15      _x86_64_   (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          16,16    0,09    4,79    0,46    0,00   78,50

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda              83,80        41,64       531,43   22375057  285581196
```

iostat

- CPU util report → %iowait
- Not really reliable → %iowait is some kind of %idle time

```
# taskset 1 fio -rw=randwrite [...] &
# iostat -y -c 1 3
[…]
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          17,32    0,00    6,56   13,65    0,00   62,47
# taskset 1 sh -c "while true; do true; done" &
# iostat -y -c 1 3
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          35,59    0,00    7,02    0,00    0,00   57,39
```

http://www.percona.com/blog/2014/06/03/trust-vmstat-iowait-numbers/

iostat

THOMAS KRENN®
server.hosting.customized.

- Extended device util report → %util
  - man iostat → ... *for devices serving requests in parallel, such as RAID arrays and modern SSDs, this number does not reflect their performance limits.*
- In theory
  - 94,4% util 23032 IOPS
  - 99,6% util 24300 IOPS

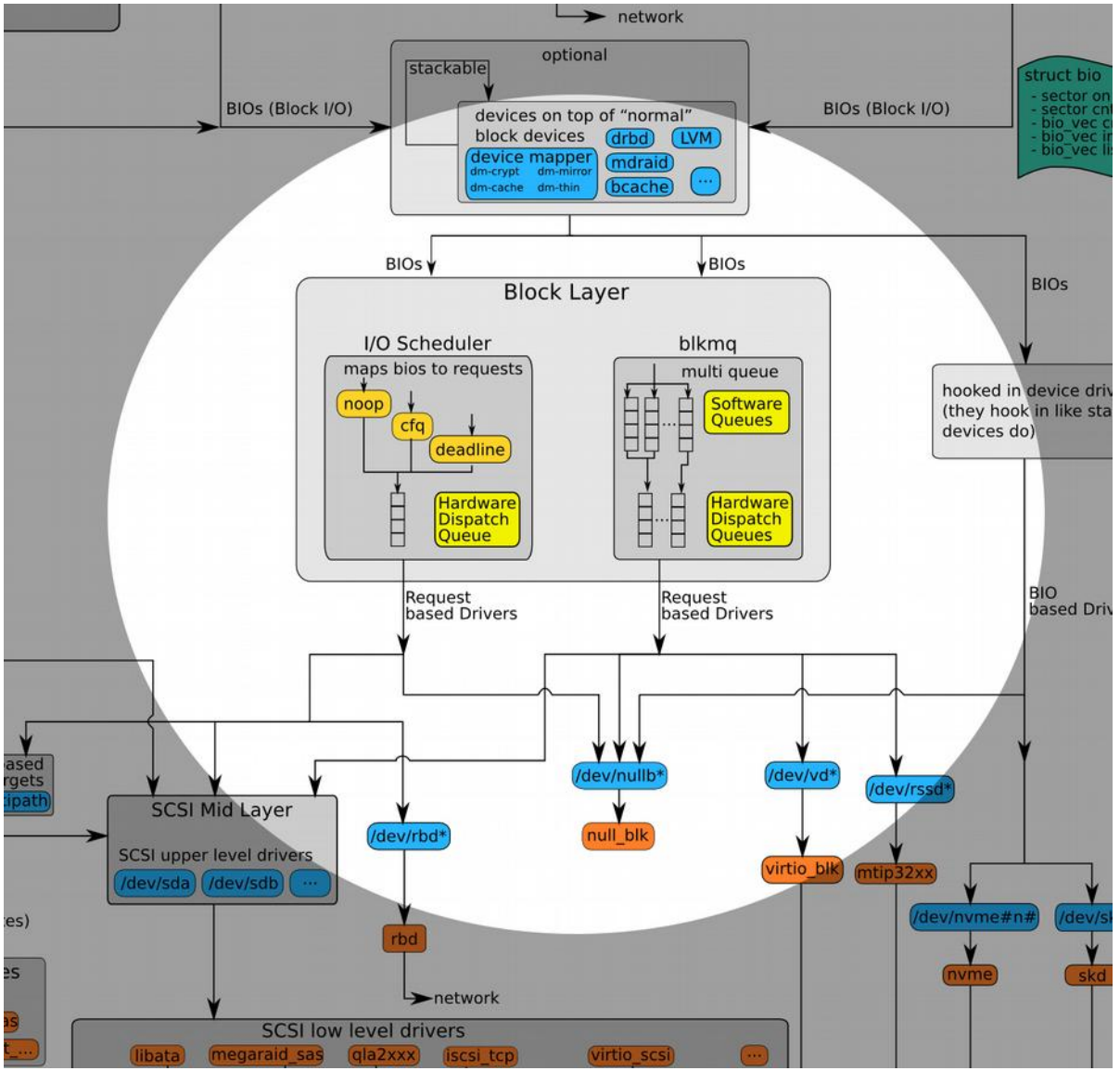
iostat

```
# iostat -y -d -x 1 3
Linux 3.13.0-48-generic (X220) 2015-04-15      _x86_64_   (4 CPU)

Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await
r_await w_await  svctm  %util
sda               0,00     2,00    0,00 23032,00     0,00 92136,00     8,00     2,90    0,13
0,00    0,13   0,04  94,40
# iostat -y -d -x 1 3
Linux 3.13.0-48-generic (X220) 2015-04-15      _x[illegible]64_   (4 CPU)

Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/[illegible]    wkB/s avgrq-sz avgqu-sz   await
r_await w_await  svctm  %util
sda               0,00  2917,00    0,00 43175,00     0,00 184[illegible]0,00     8,55   135,75    3,15
0,00    3,15   0,02  99,60
```

Only 5% `util` increase,
but IOPS nearly doubled!



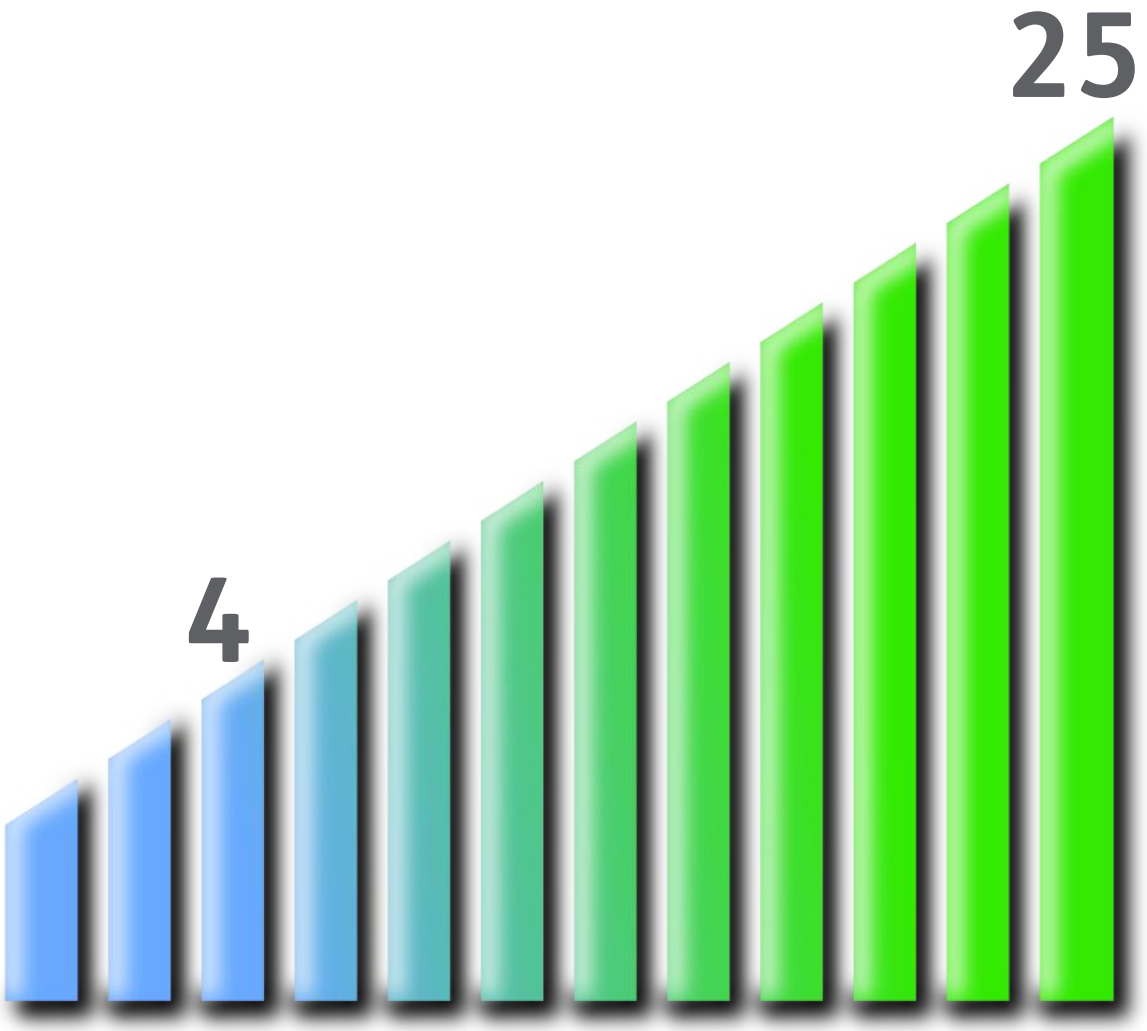
https://www.thomas-krenn.com/de/wiki/Linux_Storage_Stack_Diagramm

iostat

THOMAS KRENN®
server.hosting.customized.

- `avgqu-sz` Avg. queue length of requests issued
  - `(delta[time_in_queue] / interval) / 1000.0`
    - `time_in_queue` Requets waiting for device, effected by `in_flight`
- `await` Avg. time requests being served
  - `delta[read_ticks + write_ticks] / delta[read_IOs + write_Ios]`
  - `ticks` also effected by `in_flight`
- Therefore serving more requests while `await` is not increasing, is a good performance indicator

- Monitoring IO Performance using iostat and pt-diskstats
- Block layer statistics


25
4

dstat

THOMAS KRENN®
server.hosting.customized.

- Combines several classic tools
  - Prints metrics and uses colors
  - Has a plugin concept

```
root@X220:~
root@X220: ~ 135x10
root@X220:~# dstat -vtin
Terminal width too small, trimming output.
---procs--- ------memory-usage----- ---paging-- -dsk/total- ---system-- ----total-cpu-usage---- ----system---- ----interrupts--->
run blk new| used  buff  cach  free|  in   out | read  writ| int   csw |usr sys idl wai hiq siq|     time     |  43   44   45 >
  0   0 0.2|1855M 287M 1776M 3949M|   0     0 |  23k  184k| 155   808 | 20   5  74   0   0   0|22-04 08:00:44|   0   15    0 >
1.0   0   0|1855M 287M 1776M 3949M|   0     0 |   0     0 | 315  1392 |  3   1  96   0   0   0|22-04 08:00:45|   0   32    0 >
1.0   0   0|1855M 287M 1776M 3949M|   0     0 |   0     0 | 403  1965 |  4   2  94   0   0   0|22-04 08:00:46|   1   33    0 >
  0   0   0|1855M 287M 1776M 3949M|   0     0 |   0     0 | 412  1868 |  6   2  92   0   0   0|22-04 08:00:47|   0   37    0 >
  0   0   0|1855M 287M 1768M 3957M|   0     0 |   0     0 | 313  1540 |  3   1  96   0   0   0|22-04 08:00:48|   1   42    0 >
1.0   0   0|1855M 287M 1768M 3957M|   0     0 |   0     0 | 492  1688 |  3   2  96   0   0   0|22-04 08:00:49|   0   69    0 >^C
```

```
root@X220:~
root@X220: ~ 36x10
root@X220:~# dstat --top-mem
--most-expensive-
  memory process
firefox      701M
firefox      701M
firefox      701M
firefox      701M
firefox      701M
firefox      699M
```

nicstat

THOMAS KRENN®
server.hosting.customized.

- Print network device statistics
  - %Util depends on speed and duplex mode
  - Sat also takes errors into account

```
# nicstat -l
Int       Loopback   Mbit/s Duplex State
vboxnet0        No        0   unkn    up
eth0            No     1000   full    up
lo             Yes        -   unkn    up
wlan0           No        0   unkn    up
# nicstat -i eth0 1 5
    Time      Int   rKB/s   wKB/s   rPk/s   wPk/s    rAvs    wAvs %Util    Sat
14:52:21     eth0    3.08    0.36    3.13    2.48  1007.6   149.4  0.00   0.05
14:52:22     eth0   19.89    1.23   16.98   17.97  1199.6   70.00  0.02   0.00
14:52:23     eth0   21.42    1.09   21.99   16.00   997.1   70.00  0.02   0.00
# nicstat -i eth0 -t 1 2
14:57:36    InKB   OutKB   InSeg  OutSeg Reset  AttF %ReTX InConn OutCon Drops
TCP         0.00    0.00    2.88    2.51  0.02  0.00 0.000   0.00   0.04  0.00
14:57:37    InKB   OutKB   InSeg  OutSeg Reset  AttF %ReTX InConn OutCon Drops
TCP         0.00    0.00    0.00    0.00  0.00  0.00 0.000   0.00   0.00  0.00
```

Check if your network is saturated, Drops can be an indicator!

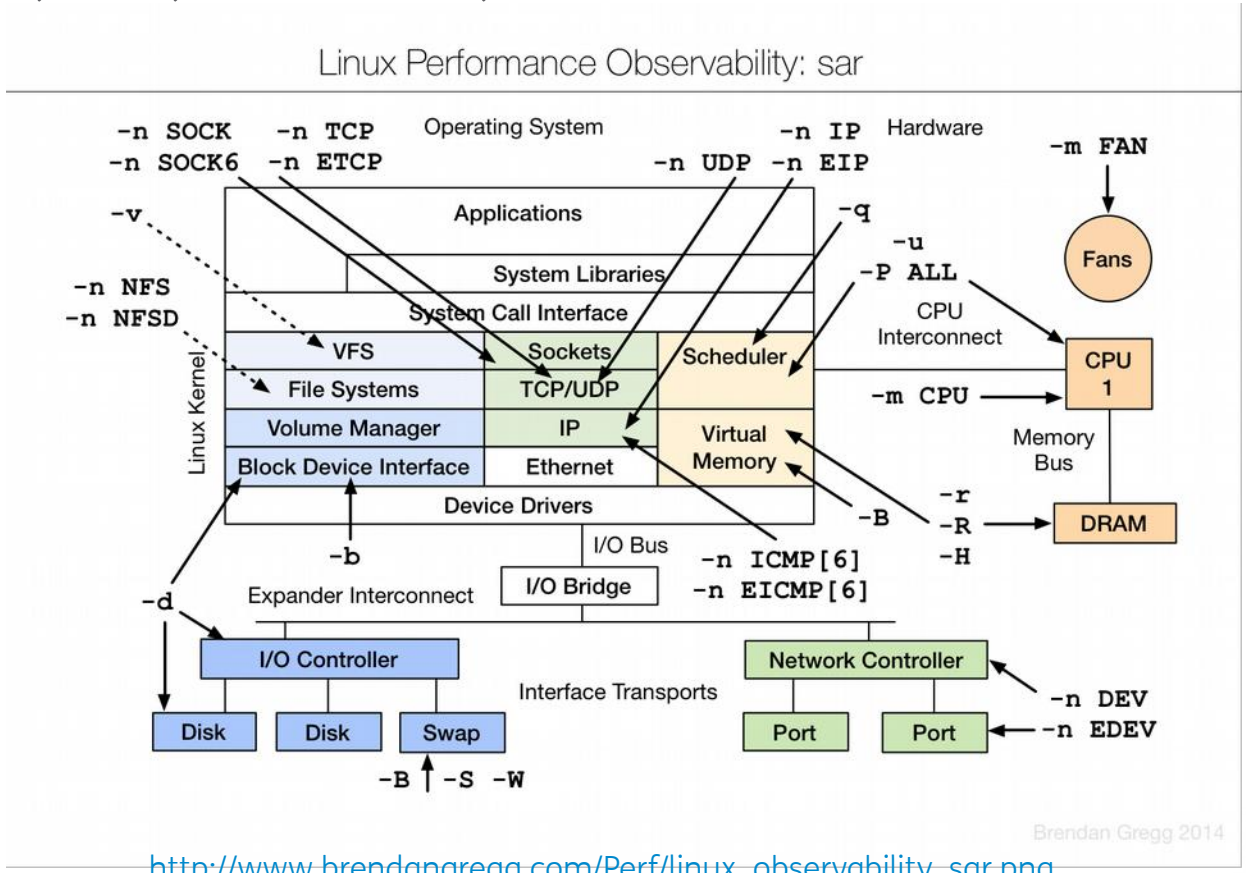
OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

Do you have a basic chronicle of your system's data?

Yes	No

sar

- It's easy with system activity reporter
  - sar, sadc, sa1 and sa2, sadf

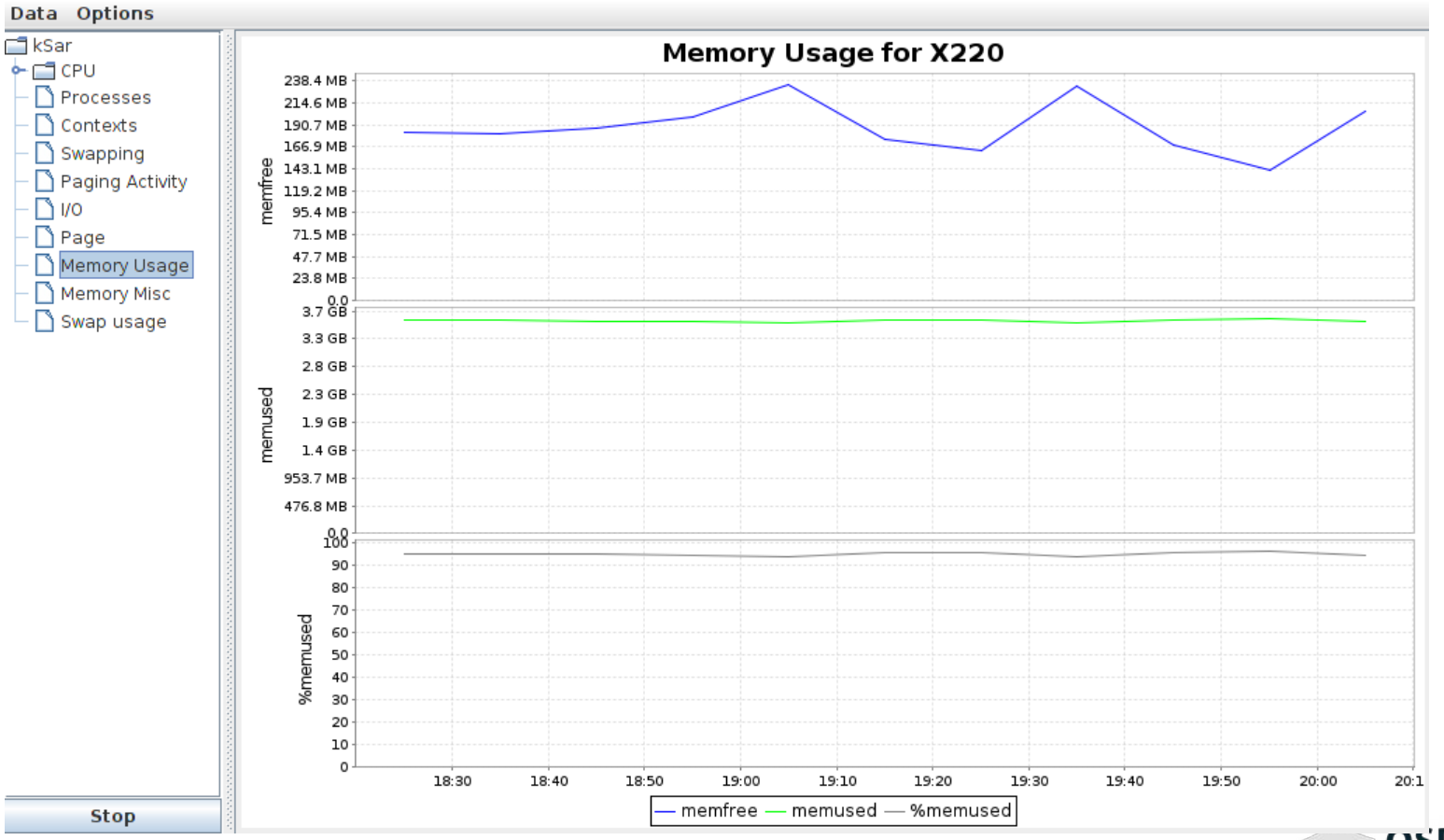


http://www.brendangregg.com/Perf/linux_observability_sar.png

ksar

Mitigates character encoding and number format problems

```
_ LC_ALL=POSIX sar -A -f sa10 > ksar.out.txt
```



The /proc filesystem

THOMAS KRENN®
server.hosting.customized.

- /proc knows everything

```
$ sudo procinfo -n 2
Memory:        Total        Used        Free     Buffers
RAM:         8056664     7769252      287412      161476
Swap:        1048572         172     1048400

Bootup: Thu Apr  9 07:16:14 2015   Load average: 0.69 0.47 0.47 1/600 1711

user  :      05:45:48.75  11.7%  page in :          4844721
nice  :      00:02:35.09   0.1%  page out:         22309056
system:      01:48:13.52   3.6%  page act:          3638373
IOwait:      00:00:48.03   0.0%  page dea:           799382
hw irq:      00:00:00.19   0.0%  page flt:        112500382
sw irq:      00:01:18.48   0.0%  swap in :                0
idle  :   1d 17:48:14.29  84.5%  swap out:               44
uptime:   5d 06:59:42.28         context :        247128528
[...]
```

Overview

THOMAS KRENN®
server.hosting.customized.

- Collect Statistics
  - Sysstat Package
  - `dstat`
  - `nicstat`
  - `/proc` → raw counters
  - `sar` and `sadc`
- Watch online
  - `top`
  - `htop`
  - `iotop`
  - `iftop`
- Tracing
  - `perf_events`
  - `ftrace`
  - `perf-tools`
  - Flame graphs
  - `strace`
- LXC
- MySQL
  - Slow Query Log
  - `innotop`

top

- System summary at beginning
- Per process metrics afterwards
  - Default sorted by CPU usage

1, 5 and 15 min load average

```
$ top -b -n 1| head -15
top - 15:33:50 up 3 days, 19:02,  3 users,  load average: 0.13, 0.51, 0.59
Tasks: 668 total,   1 running, 667 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.5%us,  0.3%sy,  0.1%ni, 98.1%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  132009356k total, 23457172k used, 108552184k free,  1600120k buffers
Swap:  3904444k total,        0k used,  3904444k free, 12682188k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
29276 root      20   0  6928 3488  668 S   19  0.0  22:55.72 ossec-syscheckd
 1193 gschoenb  20   0 17728 1740  936 R    4  0.0   0:00.02 top
11257 root      20   0 22640 2636 1840 S    4  0.0  70:38.88 openvpn
19907 www-data  20   0  197m  61m  52m S    4  0.0   0:06.18 apache2
  775 root      20   0     0    0    0 S    2  0.0   8:03.13 md3_raid10
 3712 root      39  19     0    0    0 S    2  0.0  22:45.85 kipmi0
12807 root      -3   0     0    0    0 S    2  0.0   6:20.30 drbd2_asender
 18653 root      20   0     0    0    0 S    2  0.0  12:40.19 drbd1_receiver
```

top

THOMAS KRENN®
server.hosting.customized.

Memory usage

- `VIRT` The total size of virtual memory for the process
  - Also including e.g. not already mapped `heap` or `swap`
- `RES` How many blocks are really allocated and mapped to address space → resident
  - Also includes file-backed memory (like shared libraries, `mmap`)
  - Can be used concurrently by processes
- `SHR` is shared or file-backed memory
- `RES - SHR = anon mem` (`malloc`)

```
$ cat /proc/17692/statm
1115764 611908 16932 26 0 848936 0
```

- https://www.linux.com/learn/tutorials/42048-uncover-the-meaning-of-tops-statistics
- http://www.linuxdevcenter.com/pub/a/linux/2006/11/30/linux-out-of-memory.html

top

- Can consume resources on it's own
- Toggle `f` and select fields, e.g. `SWAP`
- `-u` let's you see processes from a user
- Toggle `k` to kill a PID
- Toggle `r` to renice a PID
- But
  - `top` can miss short living processes
  - high `%CPU` → so what?
    - Keep an eye on the tracing part

htop

THOMAS KRENN®
server.hosting.customized.

- „Super advanced“ `top`
  - Uses colors, views can be customized

```
root@X220: ~
root@X220: ~ 100x28

  1  [||||                                  7.1%]     Tasks: 158, 247 thr; 1 running
  2  [||                                    2.6%]     Load average: 0.29 0.25 0.30
  3  [||                                    3.9%]     Uptime: 17:34:43
  4  [|||                                   4.5%]
  Mem:7867M used:1913M buffers:287M cache:1781M
  Swp:1023M used:0K

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 1333 root       20   0  576M  178M  153M S  5.9  2.3  8:42.76 /usr/bin/X -core :0 -seat seat0 -auth
 2605 gschoenb    9 -11  426M  7312  4896 S  4.6  0.1  6:50.54 /usr/bin/pulseaudio --start --log-tar
10567 gschoenb   20   0 1813M  709M 54868 S  4.6  9.0  4:59.68 /usr/lib/firefox/firefox
 2666 gschoenb   -6 -11  426M  7312  4896 S  2.6  0.1  3:48.93 /usr/bin/pulseaudio --start --log-tar
10800 gschoenb   20   0 1813M  709M 54868 S  2.0  9.0  0:24.41 /usr/lib/firefox/firefox
11763 root       20   0 33164  2280  1456 R  1.3  0.0  0:00.19 htop
 2994 gschoenb   20   0  537M  8112  5972 S  0.7  0.1  0:39.64 conky
  611 avahi      20   0 32348  1796  1368 S  0.7  0.0  0:03.53 avahi-daemon: running [X220.local]
 2412 gschoenb   20   0 40240  2392   932 S  0.7  0.0  0:05.74 dbus-daemon --fork --session --addres
 2675 gschoenb   20   0  844M 18904 12836 S  0.7  0.2  0:02.35 nm-applet
 2696 gschoenb   20   0  721M 20652 11840 S  0.7  0.3  0:17.89 /usr/lib/x86_64-linux-gnu/xfce4/panel
 6183 gschoenb   20   0 1249M  202M 73092 S  0.7  2.6  2:42.92 /usr/lib/libreoffice/program/soffice.
11542 gschoenb   20   0 1166M 46516 22024 S  0.0  0.6  0:08.37 /usr/bin/python /usr/bin/terminator
 3946 gschoenb   20   0 1178M 28360 16988 S  0.0  0.4  1:20.54 linphone
10292 root       20   0  139M 15672  4188 S  0.0  0.2  0:02.10 /opt/teamviewer/tv_bin/teamviewerd -f
10625 gschoenb   21   1 1813M  709M 54868 S  0.0  9.0  0:00.22 /usr/lib/firefox/firefox
 1252 root       20   0  4364   696   520 S  0.0  0.0  0:16.16 acpid -c /etc/acpi/events -s /var/run
 2570 gschoenb   20   0 1349M 47532 24260 S  0.0  0.6  0:20.69 pidgin
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

iotop

- Simple top like I/O monitor
- Which process is causing I/O
  - Filtering specific PID is possible

Show writes, reads and command in realtime

```
# iotop -o -b
Total DISK READ :       0.00 B/s | Total DISK WRITE :      63.94 M/s
Actual DISK READ:       0.00 B/s | Actual DISK WRITE:      63.90 M/s
  TID  PRIO  USER     DISK READ  DISK WRITE  SWAPIN      IO    COMMAND
19153 be/4 root        0.00 B/s   63.89 M/s  0.00 % 75.44 % fio --rw=randwrite --name=test
--filename=test.fio --size=300M --direct=1 --bs=4k
17715 be/4 gschoenb    0.00 B/s   46.18 K/s  0.00 %  0.00 % firefox [mozStorage #1]
# iotop -o -b
Total DISK READ :      69.02 M/s | Total DISK WRITE :      65.92 K/s
Actual DISK READ:      69.02 M/s | Actual DISK WRITE:     345.12 K/s
  TID  PRIO  USER     DISK READ  DISK WRITE  SWAPIN      IO    COMMAND
19176 be/4 root       69.02 M/s    0.00 B/s  0.00 % 88.28 % fio --rw=read --name=test
--filename=test.fio --size=300M --direct=1 --bs=8k
```

Bandwidth live usage

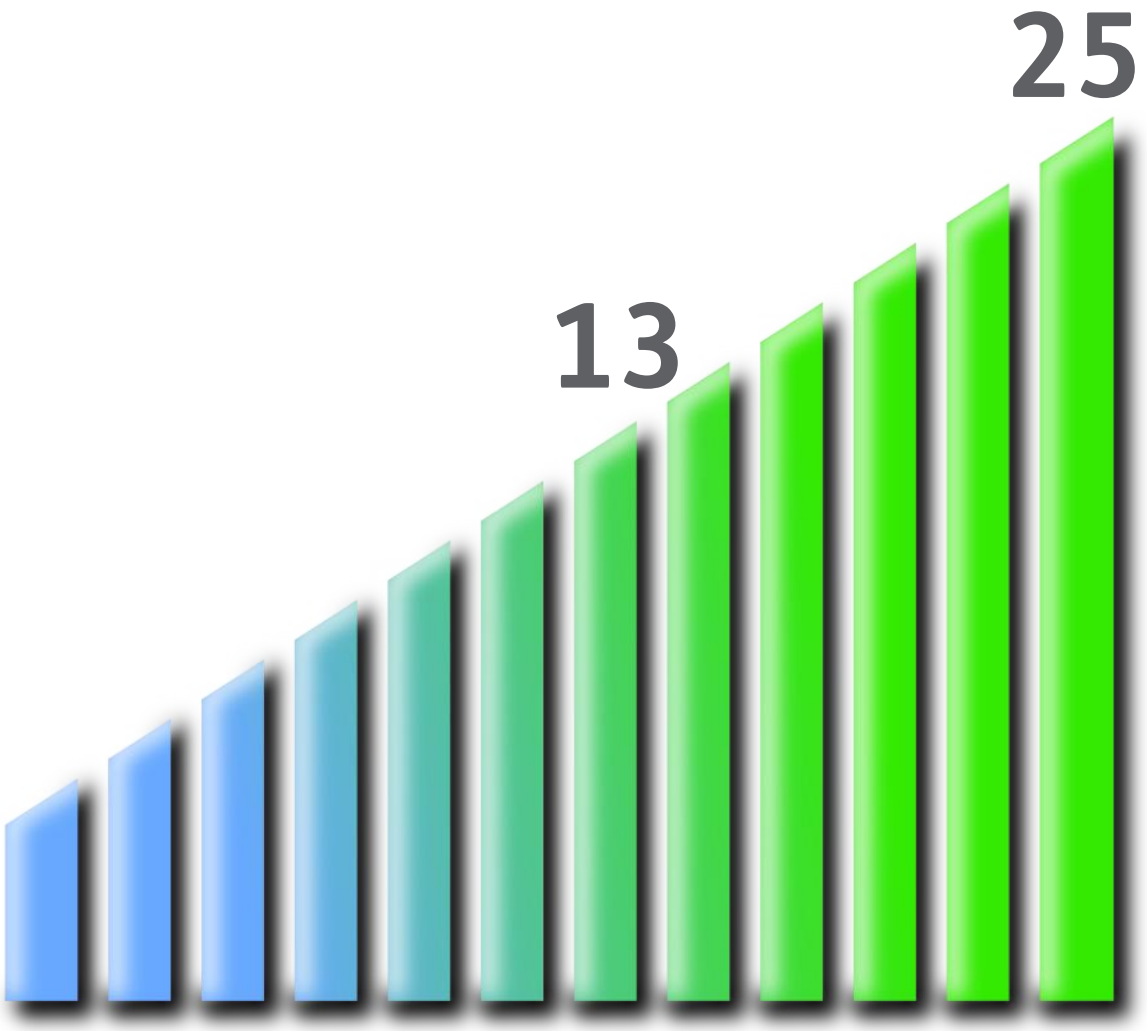
THOMAS KRENN®
server.hosting.customized.

- iftop
  - Per interface usage
- nethogs
  - Per proces

```
NetHogs version 0.8.0

  PID USER     PROGRAM                                    DEV        SENT      RECEIVED
17692 gschoenb /usr/lib/firefox/firefox                   eth0       0.162       0.194 KB/sec
16585 root     /usr/bin/ssh                               eth0       0.000       0.000 KB/sec
16611 gschoenb evolution                                  eth0       0.000       0.000 KB/sec
?     root     unknown TCP                                           0.000       0.000 KB/sec

  TOTAL                                                               0.162       0.194 KB/sec
```


25
13

Overview

THOMAS KRENN®
server.hosting.customized.

- Collect Statistics
  - Sysstat Package
  - `dstat`
  - `nicstat`
  - `/proc` → raw counters
  - `sar` and `sadc`
- Watch online
  - `top`
  - `htop`
  - `iotop`
  - `iftop`
- Tracing
  - `perf_events`
  - `ftrace`
  - `perf-tools`
  - Flame graphs
- LXC
- MySQL
  - Slow Query Log
  - `innotop`

whereis tracing

Profiling

THOMAS KRENN®
server.hosting.customized.

- Create profile about usage characteristics
  - Count specific samples/events
  - Count objects
- Next slides focus on system profiling
  - `ftrace`
  - `perf_events` and `perf`
- Collecting statistics about tracepoints
  - Lines of kernel code with defined event

ftrace

THOMAS KRENN®
server.hosting.customized.

- Part of the Linux kernel since 2.6.27 (2008)
- What is going on inside the kernel
- Common task is to trace events
- With `ftrace` configured, only `debugfs` is required

```
# cat /proc/sys/kernel/ftrace_enabled
1
# mount | grep debug
none on /sys/kernel/debug type debugfs (rw)
/sys/kernel/debug/tracing# cat available_tracers
blk mmiotrace function_graph wakeup_rt wakeup function nop
```

ftrace

THOMAS KRENN®
server.hosting.customized.

- Interact with files in `sys`
  - Easier with `trace-cmd`

```
#!/bin/bash

DEBUGFS=`grep debugfs /proc/mounts | awk '{ print $2; }'`

echo $$ > $DEBUGFS/tracing/set_ftrace_pid
echo function > $DEBUGFS/tracing/current_tracer
echo 1 > $DEBUGFS/tracing/tracing_on
exec $*
echo 0 > $DEBUGFS/tracing/tracing_on
```

perf_events and perf

THOMAS KRENN®
server.hosting.customized.

- Used to be called performance counters for Linux
- A lot of updates for kernel 4.1
  - https://lkml.org/lkml/2015/4/14/264
- CPU performance counters, tracepoints, kprobes and uprobes
- Per package with `linux-tools-common`

```
# which perf
/usr/bin/perf
# dpkg -S /usr/bin/perf
linux-tools-common: /usr/bin/perf
```

perf list

THOMAS KRENN®
server.hosting.customized.

- perf list
  - Shows supported events

This also includes static tracepoints

```
# perf list | wc -l
1779
# perf list | grep Hardware
  cpu-cycles OR cycles                                    [Hardware event]
  instructions                                            [Hardware event]
  cache-references                                        [Hardware event]
  cache-misses                                            [Hardware event]
  branch-instructions OR branches                         [Hardware event]
  branch-misses                                           [Hardware event]
  bus-cycles                                              [Hardware event]
  stalled-cycles-frontend OR idle-cycles-frontend         [Hardware event]
  stalled-cycles-backend OR idle-cycles-backend           [Hardware event]
  ref-cycles                                              [Hardware event]
  L1-dcache-loads                                         [Hardware cache event]
  L1-dcache-load-misses                                   [Hardware cache event]
  L1-dcache-stores                                        [Hardware cache event]
  L1-dcache-store-misses                                  [Hardware cache event]
```

Raw CPU counters

THOMAS KRENN®
server.hosting.customized.

- Each CPU has it's own raw counters
  - They should be documented by the hardware manufacturer
    - https://download.01.org/perfmon/
- `libpfm4` is a nice way to find raw masks

```
# perf list | grep rNNN
  rNNN                                          [Raw hardware event descriptor]
# git clone git://perfmon2.git.sourceforge.net/gitroot/perfmon2/libpfm4
# cd libpfm4
# make
# cd examples/
# ./showevtinfo | grep LLC | grep MISSES
Name      : LLC_MISSES
[...]
# ./check_events LLC_MISSES | grep Codes
Codes           : 0x53412e
# perf stat -e r53412e sleep 5
```

Now we collect last level cache misses with the raw mask

Tracepoints

THOMAS KRENN®
server.hosting.customized.

- `perf` also has trace functionalities

```
# perf list | grep -i trace | wc -l
1716
```

  - Filesystem
  - Block layer
  - Syscalls

```
# perf stat -e 'syscalls:sys_enter_mmap' ./helloWorld.out
Hello world!

 Performance counter stats for './helloWorld.out':

                 8 syscalls:sys_enter_mmap

       0,000556961 seconds time elapsed
```

perf stat

THOMAS KRENN®
server.hosting.customized.

_ Get a counter summary

Easy to compare performance of different algorithms

```
# perf stat python numpy-matrix.py -i matrix.in

 Performance counter stats for 'python numpy-matrix.py -i matrix.in':

        576,104221 task-clock (msec)         #    0,930 CPUs utilized
               319 context-switches          #    0,554 K/sec
                 4 cpu-migrations            #    0,007 K/sec
             9.738 page-faults               #    0,017 M/sec
     1.743.664.199 cycles                    #    3,027 GHz                      [82,63%]
       831.364.029 stalled-cycles-frontend   #   47,68% frontend cycles idle     [83,75%]
       458.760.523 stalled-cycles-backend    #   26,31% backend  cycles idle     [67,26%]
     2.793.953.303 instructions              #    1,60  insns per cycle
                                             #    0,30  stalled cycles per insn [84,28%]
       573.342.473 branches                  #  995,206 M/sec                    [83,78%]
         3.586.249 branch-misses             #    0,63% of all branches         [82,70%]

       0,619482128 seconds time elapsed
```

perf record

THOMAS KRENN®
server.hosting.customized.

- Record samples to a file
  - Can be post-processed with `perf report`
  - `-a` records on all CPUs
  - `-g` records call graphs
    - Install debug symbols

```
# perf record -a -g sleep 5
[ perf record: Woken up 4 times to write data ]
[ perf record: Captured and wrote 2.157 MB perf.data (~94254 samples) ]
```

Nice way to record what's currently running on all CPUs

perf report

THOMAS KRENN®
server.hosting.customized.

- Displays profile of a record
  - Can be sorted and or filtered
  - Shows all samples

```
# perf report -i perf.data.dd --stdio --showcpuutilization --sort comm,dso
[...]
# Overhead       sys       usr  Command      Shared Object
# ........  ........  ........  .......  .................
    95.00%    95.00%     0.00%       dd  [kernel.kallsyms]
                |
                |--33.22%-- _aesni_enc1
                |           __ablk_encrypt
                |           ablk_encrypt
                |           crypt_scatterlist
                |           crypt_extent
                |           ecryptfs_encrypt_page
                |           ecryptfs_write_end
                |           generic_file_buffered_write
                |           __generic_file_aio_write
                |           generic_file_aio_write
                |           do_sync_write
                |           vfs_write
                |           sys_write
                |           system_call_fastpath
                |           __GI___libc_write
                |           0x415f65643d524550
                |--9.11%-- _cond_resched
                |           |
                |           |--57.94%-- ext4_dirty_inode
                |           |           __mark_inode_dirty
                |           |           generic_write_end
                |           |           ext4_da_write_end
                |           |           generic_file_buffered_write
```

Command and shared object

Traced method

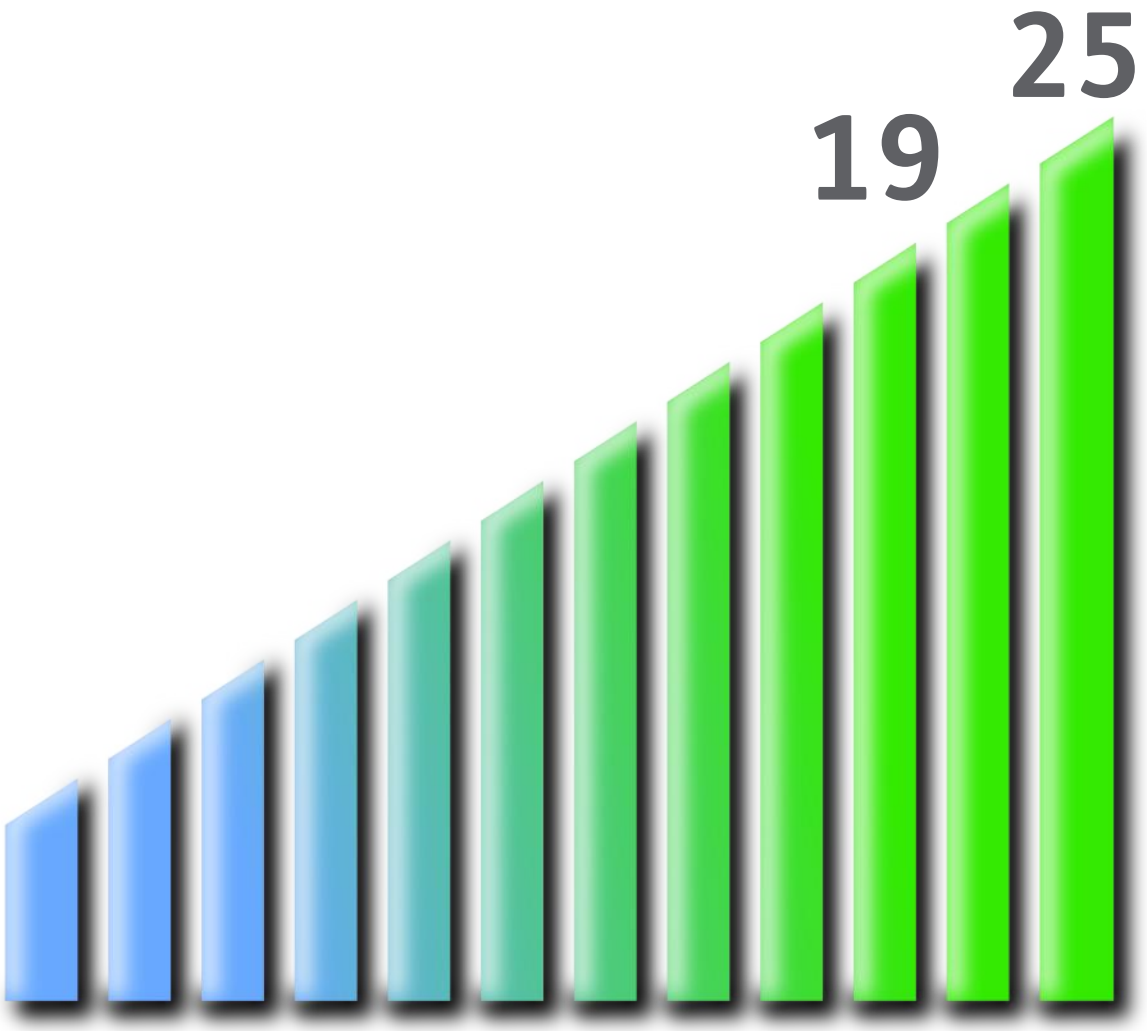
dd writes data

perf-tools

THOMAS KRENN®
server.hosting.customized.

- By Brendan Gregg
  - https://github.com/brendangregg/perf-tools
  - Mostly quick hacks, read Warnings!
- Using `perf_events` and `ftrace`
- Good examples what can be done with `perf` and `ftrace`
  - `iosnoop` Shows I/O access for commands, including latency
  - `cachestat` Linux page cache hit/miss statistics
  - `functrace` Count kernel functions matching wildcards

Nice, this are simple bash scripts!

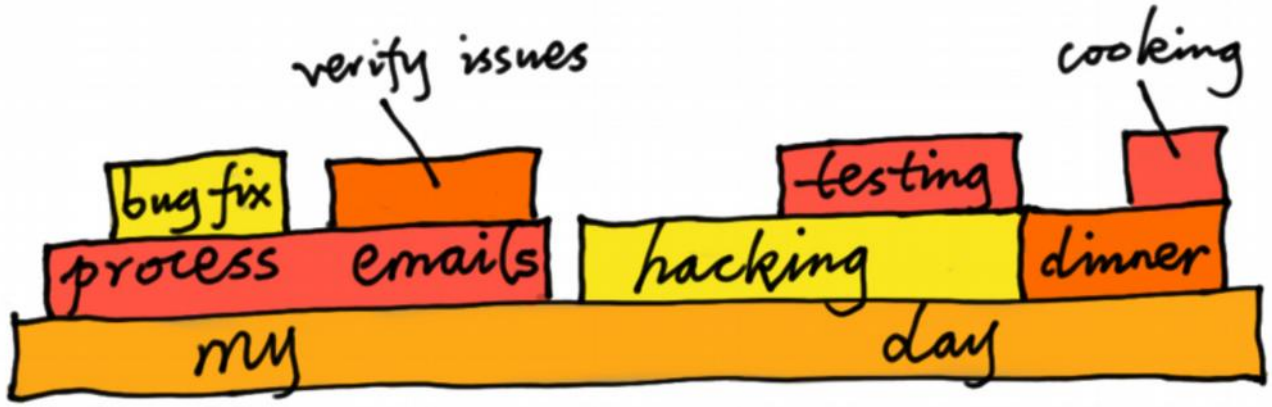

19
25

```
# view flamegraph
```

Flamegraph

THOMAS KRENN®
server.hosting.customized.

- Visualization how resources are distributed among code

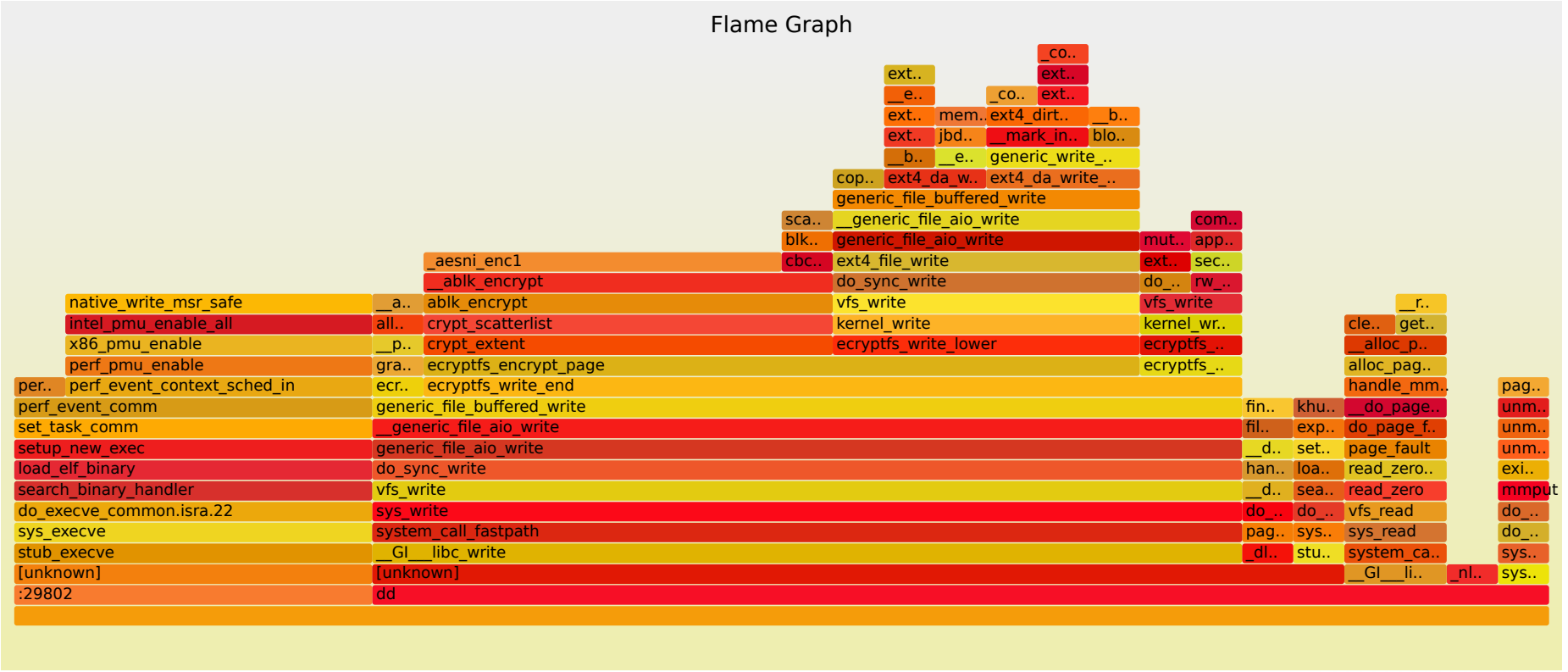


Powered by @agentzh, http://agentzh.org/misc/slides/yapc-na-2013-flame-graphs.pdf

Flamegraph

THOMAS KRENN®
server.hosting.customized.

```
# perf record -g dd if=/dev/zero of=test.data count=1 bs=1M
# mv perf.data perf.data.dd
# perf script -i perf.data.dd | ./FlameGraph/stackcollapse-perf.pl > out.dd.folded
# ./FlameGraph/flamegraph.pl out.dd.folded > out.perf.dd.svg
```



OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

```
# tail -n 2 /special/lxc
```

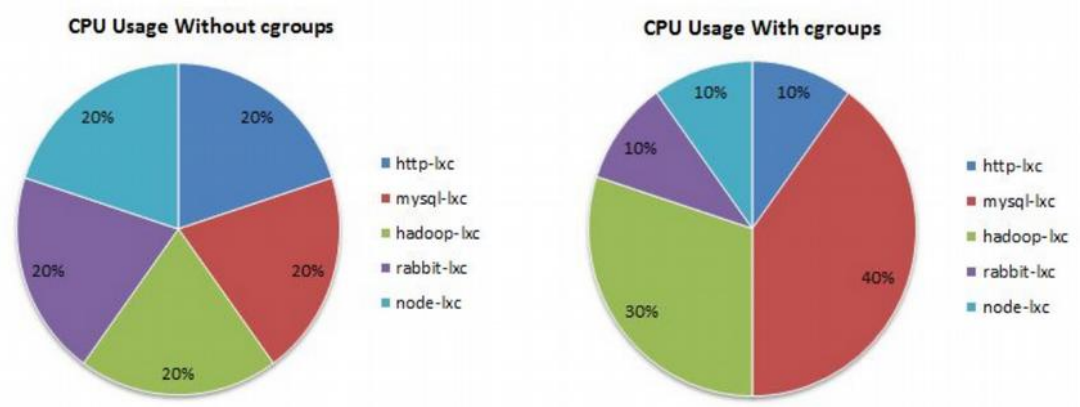
Linux Container

THOMAS KRENN®
server.hosting.customized.

- Lightweight „virtual machines“ using features provided by a modern Linux kernel
  - `cgroups` Aggregate or partition tasks and their children to hierarchical groups to isolate resources
  - `namespaces` Wrap a resource in an abstraction so that it appears to processes they have their own isolated resource
- Each container shares the kernel running on the host
  - Some may refer to it as „native performance“

Linux Container

- cgroups are divided into subsystems, e.g.
  - cpusets
  - blkio
  - memory



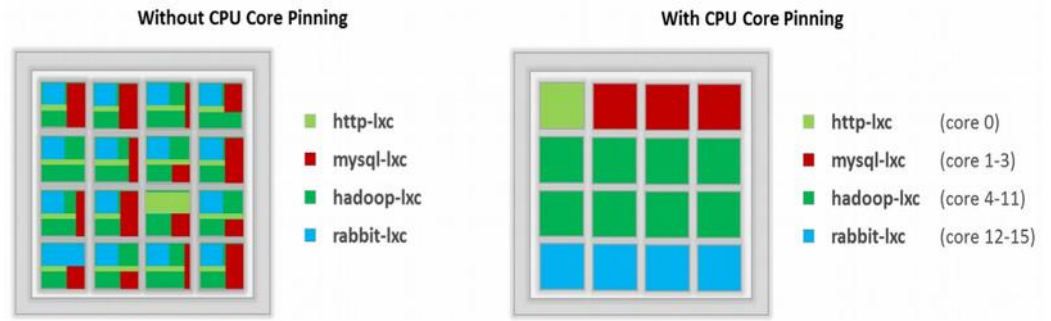


Image from Boden Russel, http://de.slideshare.net/BodenRussell/realizing-linux-containerslxc

Linux Container

THOMAS KRENN®
server.hosting.customized.

- cgroup created per container per subsystem

```
# lxc-ls --fancy
NAME     STATE    IPV4        IPV6  GROUPS  AUTOSTART
----------------------------------------------------
ubuntu1  RUNNING  10.0.3.119  -     -       NO
# lxc-info -n ubuntu1
Name:           ubuntu1
State:          RUNNING
PID:            7548
IP:             10.0.3.119
CPU use:        1.80 seconds
BlkIO use:      22.68 MiB
Memory use:     30.85 MiB
KMem use:       0 bytes
Link:           vethC8TJUT
 TX bytes:      3.33 KiB
 RX bytes:      3.49 KiB
 Total bytes:   6.82 KiB
```

Linux Container

THOMAS KRENN®
server.hosting.customized.

- `lxc-info` takes `cgroups` into account

Value	Origin
CPU use	`lxc-cgroup -n ubuntu1 cpuacct.usage`
BlkIO use	`lxc-cgroup -n ubuntu1 blkio.throttle.io_service_bytes`
Memory use	`lxc-cgroup -n ubuntu1 memory.usage_in_bytes`
KMem use	`lxc-cgroup -n ubuntu1 memory.kmem.usage_in_bytes`
Link	`cat /sys/class/net/veth0EP3QM/statistics/*_bytes`

- `cgroups` providing further info
  - `memory.stat`
  - `memory.failcnt`
  - `cpuset.cpus`

Linux Container

THOMAS KRENN®
server.hosting.customized.

- `lxc-top` monitors container overall usage

```
# lxc-top
Container          CPU      CPU      CPU      BlkIO       Mem
Name              Used      Sys     User      Total      Used
ubuntu1           1.94     1.11     0.84   32.22 MB  14.71 MB
ubuntu2           1.43     0.88     0.79   10.61 MB  17.88 MB
TOTAL 2 of 2      3.36     1.99     1.63   42.83 MB  32.59 MB
```

- Traditional tools without `lxcfs` do not work!

```
root@host # lxc-cgroup -n ubuntu1 memory.limit_in_bytes
33554432
root@container # free -h
             total       used       free     shared    buffers     cached
Mem:          489M       202M       287M       488K        26M       115M
```

```
# sh -c "while true; do mysql; done"
```

MySQL

THOMAS KRENN®
server.hosting.customized.

- Percona provides a lot of good tools
- First step, generate a summery

```
# pt-mysql-summary
# Percona Toolkit MySQL Summary Report ######################
# Instances ##################################################
  Port  Data Directory             Nice OOM Socket
  ===== ========================== ==== === ======
        /var/lib/mysql             0    0   /var/lib/mysql/mysql.sock
# MySQL Executable ###########################################
       Path to executable | /usr/sbin/mysqld
              Has symbols | Yes
# Report On Port 3306 ########################################
                     User | root@localhost
                     Time | 2015-04-14 07:49:09 (CEST)
                 Hostname | mysql1
                Databases | 15
                  Datadir | /var/lib/mysql/
[...]
```

MySQL

THOMAS KRENN®
server.hosting.customized.

- Extended status prints also counters
  - Can be monitored with `pmp-check-mysql-status` Plugin

```
# mysqladmin ext | wc -l
345
# mysqladmin ext | grep Threads_running
| Threads_running                                | 3
# mysqladmin ext | grep Innodb_buffer_pool_pages_free
| Innodb_buffer_pool_pages_free                  | 12298
```

- Slow Query log
  - Queries exceeding a specific runtime
  - OFF by default, runtime and log file must be defined
  - Query Cache is ignored

MySQL

THOMAS KRENN®
server.hosting.customized.

- Easy way to log all queries → `long_query_time 0`
- `pt-query-digest`
  - Process slow log and generate report

```
# pt-query-digest mysql-slow.log
[...]
# Attribute          total     min     max     avg     95%  stddev  median
# ============     ======= ======= ======= ======= ======= ======= =======
# Exec time        184984s      9s    419s     51s    151s     45s     42s
# Lock time            15s       0      3s     4ms       0    71ms       0
# Rows sent        500.05M       0   2.65M 139.79k   1.69M 491.22k    3.89
# Rows examine       3.23G       0 234.22M 923.81k   2.49M   5.53M 440.37k
# Query size       128.45M       6   2.75M  35.91k  68.96k 136.30k  10.29k
# Profile
# Rank Query ID           Response time    Calls R/Call    V/M   Item
# ==== ================== ================ ===== ======== ===== ==========
#    1 0x7A8EB8C13A4A8435 29885.0000 16.2%   305  97.9836 22.08 SELECT
#    2 0xA45C5FB6D066119B 26077.0000 14.1%   369  70.6694 22.49 SELECT
#    3 0x67A347A2812914DF 13737.0000  7.4%   397  34.6020 14.53 SELECT
#    4 0xD7A9797E81785092 11855.0000  6.4%   121  97.9752 22.05 SELECT
```

MySQL – innotop

THOMAS KRENN®
server.hosting.customized.

- Live analysis of SQL queries
  - Sort by execution time
  - Not only for innodb

```
[RO] InnoDB Txns (? for help) localhost, 81s, InnoDB 4s :-), 127.50 QPS, 31/0/0 con/run/coc thds, 5.

History  Versions  Undo  Dirty Buf  Used Bufs  Txns  MaxTxnTime  LStrcts
     21         0        0.10%     99.98%    29       01:03

ID   User   Host      Txn Status  Time   Undo  Query Text
  7  plmce  localhost ACTIVE      01:03     0  SELECT * FROM imdb.movie_info WHERE movie_id = 3224
 14  plmce  localhost ACTIVE      01:01     0  SELECT * FROM imdb.movie_info WHERE movie_id = 706
 21  plmce  localhost ACTIVE      00:59     0  SELECT * FROM imdb.movie_info WHERE movie_id = 787
 26  plmce  localhost ACTIVE      00:58     0  SELECT * FROM imdb.person_info WHERE person_id = 140
 28  plmce  localhost ACTIVE      00:57     0  SELECT * FROM imdb.cast_info WHERE movie_id = 3264 a
 30  plmce  localhost ACTIVE      00:57     0  SELECT cast_info.* FROM imdb.cast_info INNER JOIN im
 31  plmce  localhost ACTIVE      00:56     0  SELECT * FROM imdb.person_info WHERE person_id = 433
 33  plmce  localhost ACTIVE      00:56     0  SELECT * FROM imdb.person_info WHERE person_id = 428
 37  plmce  localhost ACTIVE      00:55     0  SELECT * FROM imdb.movie_info WHERE movie_id = 2630
 38  plmce  localhost ACTIVE      00:55     0  SELECT cast_info.* FROM imdb.cast_info INNER JOIN im
 42  plmce  localhost ACTIVE      00:54     0  SELECT * FROM imdb.cast_info WHERE movie_id = 551 an
 47  plmce  localhost ACTIVE      00:53     0  SELECT * FROM imdb.person_info WHERE person_id = 929
 59  plmce  localhost ACTIVE      00:50     0  SELECT * FROM imdb.movie_info WHERE movie_id = 1930
 60  plmce  localhost ACTIVE      00:49     0  SELECT * FROM imdb.cast_info WHERE movie_id = 4793 a
 61  plmce  localhost ACTIVE      00:49     0  SELECT * FROM imdb.movie_info WHERE movie_id = 3718
 69  plmce  localhost ACTIVE      00:47     0  SELECT cast_info.* FROM imdb.cast_info INNER JOIN im
 84  plmce  localhost ACTIVE      00:41     0  SELECT * FROM imdb.movie_info WHERE movie_id = 891
 92  plmce  localhost ACTIVE      00:38     0  SELECT * FROM imdb.movie_info WHERE movie_id = 947
 96  plmce  localhost ACTIVE      00:36     0  SELECT * FROM imdb.person_info WHERE person_id = 206
101  plmce  localhost ACTIVE      00:35     0  SELECT * FROM imdb.person_info WHERE person_id = 174
106  plmce  localhost ACTIVE      00:33     0  SELECT * FROM imdb.cast_info WHERE movie_id = 3578 a
109  plmce  localhost ACTIVE      00:32     0  SELECT * FROM imdb.movie_info WHERE movie_id = 3924
114  plmce  localhost ACTIVE      00:30     0  SELECT * FROM imdb.cast_info WHERE movie_id = 3679 a
117  plmce  localhost ACTIVE      00:29     0  SELECT * FROM imdb.cast_info WHERE movie_id = 4863 a
```

Thanks for your attention!

_ gschoenberger@thomas-krenn.com
_ @devtux_at

THOMAS KRENN®
server.hosting.customized.

Backup slides

iostat

- CPU util report → %iowait
- With iostat, avg. over CPUs
  - Take mpstat into account

```
# iostat -y -c 1 3
Linux 3.13.0-48-generic (X220) 2015-04-15      _x86_64_   (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          11,41    0,00    9,28   12,20    0,00   67,11
# mpstat -P ALL 1
Linux 3.13.0-48-generic (X220) 2015-04-15      _x86_64_   (4 CPU)

12:40:47     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:40:48     all   10,08    0,00    7,36   16,08    0,00    0,82    0,00    0,00    0,00   65,67
12:40:48       0   13,68    0,00   25,26   61,05    0,00    0,00    0,00    0,00    0,00    0,00
12:40:48       1   18,81    0,00    0,99    0,00    0,00    0,00    0,00    0,00    0,00   80,20
12:40:48       2    1,43    0,00    1,43    0,00    0,00    4,29    0,00    0,00    0,00   92,86
12:40:48       3    4,04    0,00    1,01    0,00    0,00    0,00    0,00    0,00    0,00   94,95
```

The /proc filesystem

THOMAS KRENN®
server.hosting.customized.

- /proc knows everything
- Obviously ps uses /proc

```
$ cat /proc/self/status | head
Name:cat
State:	R (running)
Pid:	13871
Uid:	1000	1000	1000	1000
Gid:	1000	1000	1000	1000
```

- Or values from free

```
$ cat /proc/meminfo | grep Swap
SwapCached:            0 kB
SwapTotal:       1048572 kB
SwapFree:        1048400 kB
$ cat /proc/meminfo | grep -E 'Buffer|Cache'
Buffers:          183988 kB
Cached:          3365960 kB
SwapCached:            0 kB
```

varnishstat and top

THOMAS KRENN®
server.hosting.customized.

_ Good example for application based statistics

```
# varnishstat -1 | wc -l
311
# # varnishstat -1 | grep -E 'cache|nuked'
MAIN.cache_hit                 64616         1.58 Cache hits
MAIN.cache_hitpass              5775         0.14 Cache hits for pass
MAIN.cache_miss                38611         0.95 Cache misses
MAIN.n_lru_nuked                   0           .   Number of LRU nuked objects
# varnishtop -i ReqUR -1
   558.00 ReqURL /de/wikiDE/api.php
   384.00 ReqURL /favicon.ico
   321.00 ReqURL /en/wikiEN/api.php
   282.00 ReqURL /res/font/FSMeWeb/bold/fs_me_web-bold.woff
   282.00 ReqURL /res/font/FSMeWeb/regular/fs_me_web-regular.woff
   280.00 ReqURL /de/wikiDE/skins/tkskin/images/thomas-krenn-logo-grey.png
   278.00 ReqURL /de/wikiDE/skins/tkskin/images/TK_Logo_200x90.png
```

blktrace

THOMAS KRENN®
server.hosting.customized.

- blktrace
  - Captures I/O traces
  - Trace is stored in a binary format
- blkparse
  - Reads traces recorded by blktrace
- btt
  - blktrace timeline
  - Post-processing tool

```
/var/log# head -n 1 syslog.1
# blktrace -d /dev/sda (run parallel)
# blkparse -i sda.blktrace.0[...]
```

Device | CPU | Sequence # | Time stamp | PID | Action | RWBS | Start block + # of blocks | Process

```
  8,0    0        1     0.000000000   610  A   WS 228404656 + 8 <- (8,3) 226854320
  8,0    0        2     0.000001180   610  Q   WS 228404656 + 8 [kworker/0:3]
  8,0    0        3     0.000006522   610  G   WS 228404656
                                              I   WS 228404656
                                              D   WS 228404656
                                              A   WS 228404664
                                              Q   WS 228404664
                                              G   WS 228404664
                                              I   WS 228404664 + 8 [kworker/0:3]
                                              D   WS 228404664 + 8 [kworker/0:3]
  8,0    3        1     0.431446519  5789  A    R 425590688 + 32 <- (8,3) 424040352
  8,0    3        2     0.431447779  5789  Q    R 425590688 + 32 [head]
  8,0    3        3     0.431452713  5789  G    R 425590688 + 32 [head]
  8,0    3        4     0.431454407  5789  P    N [head]
  8,0    3        5     0.431456930  5789  I    R 425590688 + 32 [head]
  8,0    3        6     0.431458376  5789  U    N [head] 1
  8,0    3        7     0.431461366  5789  D    R 425590688 + 32 [head]
```

Actions

A	IO was remapped to a different device
Q	IO handled by request queue code
G	Get request
P	Plug request
I	IO inserted onto request queue
U	Unplug request
D	IO issued to driver

RWBS

R	read
W	write
D	block discard
B	barrier operation
S	synchronous operation

```
CPU0 (sda):
 Reads Queued:          0,        0KiB  Writes Queued:          7,       28KiB
 Read Dispatches:       0,        0KiB  Write Dispatches:       7,       28KiB
 Reads Requeued:        0              Writes Requeued:        0
 Reads Completed:       0,        0KiB  Writes Completed:       5,       20KiB
 Read Merges:           0,              Write Merges:           0,        0KiB
 Read depth:            1              Write depth:            7
 IO unplugs:            0              Timer unplugs:          0
CPU1 (sda):
 Reads Queued:          0,        0KiB  Writes Queued:          2,        4KiB
 Read Dispatches:       0,        0KiB  Write Dispatches:       1,        4KiB
 Reads Requeued:        0              Writes Requeued:        0
 Reads Completed:       1,       16KiB  Writes Completed:       5,       12KiB
 Read Merges:           0,        0KiB  Write Merges:           0,        0KiB
 Read depth:            1              Write depth:            7
 IO unplugs:            0              Timer unplugs:          0
CPU3 (sda):
 Reads Queued:          1,       16KiB  Writes Queued:          0,        0KiB
 Read Dispatches:       1,       16KiB  Write Dispatches:       0,        0KiB
 Reads Requeued:        0              Writes Requeued:        0
 Reads Completed:       0,        0KiB  Writes Completed:       0,        0KiB
 Read Merges:           0,        0KiB  Write Merges:           0,        0KiB
 Read depth:            7              Write depth:            7
 IO unplugs:            0              Timer unplugs:          0

Total (sda):
 Reads Queued:          1,       16KiB  Writes Queued:          9,       32KiB
 Read Dispatches:       1,       16KiB  Write Dispatches:       8,       32KiB
 Reads Requeued:        0              Writes Requeued:        0
 Reads Completed:       1,       16KiB  Writes Completed:      10,       32KiB
 Read Merges:           0,        0KiB  Write Merges:           0,        0KiB
 IO unplugs:            1              Timer unplugs:          0

Throughput (R/W): 37KiB/s / 74KiB/s
Events (sda): 62 entries
Skips: 0 forward (0 -   0.0%)
```

per CPU details

writes submitted on this CPU

writes completed on this CPU

per device details

avg. throughput

```
# btt -i sda.blktrace.0
==================== All Devices ====================

            ALL           MIN           AVG           MAX           N
--------------- ------------- ------------- ------------- -----------

Q2Q               0.000016944   0.000022114   0.000042534           6
Q2G               0.000000694   0.000001430   0.000005342           7
G2I               0.000000314   0.000000725   0.000002793           7
I2D               0.000000375   0.000000906   0.000003652           7
D2C               0.000992471   0.001018423   0.001048992           5
Q2C               0.000993887   0.001022085   0.001060779           5
[...]
```

D2C Driver and device time – the average time from when the actual IO was issued to the driver until is completed (completion trace) back to the block IO layer.

Q2C Measures the times for the complete life cycle of IOs during the run.

MySQL pt-query-digest

THOMAS KRENN®
server.hosting.customized.

```
# Query 1: 0.00 QPS, 0.01x concurrency, ID 0x67A347A2812914DF at byte 61989638
# This item is included in the report because it matches --limit.
# Scores: V/M = 172.18
# Time range: 2012-05-23 00:00:26 to 2015-04-17 00:10:33
# Attribute    pct   total     min     max     avg     95%  stddev  median
# ============ === ======= ======= ======= ======= ======= ======= =======
# Count         12   11267
# Exec time     29 462888s      3s   2629s     41s    130s     84s     17s
# Lock time      0   531ms       0   599us    47us    93us    28us    38us
# Rows sent     99  82.85G     306  63.71M   7.53M  46.53M  13.84M 915.49k
# Rows examine  38  82.85G     306  63.71M   7.53M  46.53M  13.84M 915.49k
# Query size     0 683.69k      47      79   62.14   72.65    7.79   56.92
# String:
# Databases    XXXXX (10159/90%)... 1 more
# Hosts        localhost (9524/84%), XXXXX (1743/15%)
# Users        XXXXX (9738/86%), XXXXX (1529/13%)
# Query_time distribution
# 100us
#   1ms
#  10ms
# 100ms
#    1s  #################
#  10s+  ##################################################################
```

MySQL

THOMAS KRENN®
server.hosting.customized.

- Performance Schema
  - Per default on in 5.6, older profiling commands are deprecated with 5.6.7
  - A structured way in SQL to get timing information
- Runtime and query execution statistics
- Sys scchema (ps_helper) provides a more comfortable way

```
> select * from schema_table_statistics
where table_schema='sbtest' limit 1 \G
*************************** 1. row ***************************
table_schema: sbtest
table_name: sbtest
rows_fetched: 158764154
[...]
```