

Monitoring System Performance

Performance Agents

By Lukas Steinegger

Motivation

- Is my system up and running?
- It is running but is it available from the outside?
- But is it going to fail?
- Do I need more hardware?
- Is the service responsive?
- Is there too much latency?
- Is my system doing what it should, do users get what is promised?

Performance Agents

How do I get this Information?

Small programs running (in the background) of the host machine collecting data

- what to collect (some metrics):
 - cpu load: per core/socket/board/...
 - memory: bandwidth/load?
 - storage: capacity
 - networking: bandwidth
 - in total / per process / ...
- collect data at central point, e.g. database

Implementations and Tools

- Telegraf [<https://www.influxdata.com/time-series-platform/telegraf/>]
- Prometheus [<https://prometheus.io/>]
- LIKWID [<https://hpc.fau.de/research/tools/likwid/>]
- cc-collector [<https://github.com/ClusterCockpit/cc-metric-collector>]
- Performance Co-Pilot [<https://pcp.io/>]
- Profit-HPC [<https://profit-hpc.de/>]
- ...
- My Tool [<https://gitlab.gwdg.de/lukas.steinegger/performance-agent>]

The */proc* Directory

- A central place to find system information
- It's a virtual directory provided by the kernel
- Content of files is generated on read
- under unix(-like) OSs

We'll take a closer look at:

- `/proc/stat`
- `/proc/meminfo`
- `/proc/<pid>/stat`

cat /proc/stat

```
Lukas@buttercup:~$ cat /proc/stat
cpu 220610 1192 84980 1353290 7036 0 2835 0 0 0
cpu0 57228 329 21019 337477 1724 0 63 0 0 0
cpu1 51890 230 23752 337950 1696 0 692 0 0 0
cpu2 56258 236 21551 337345 1807 0 71 0 0 0
cpu3 55232 395 18657 340515 1807 0 2008 0 0 0
intr 16567543 0 9590 0 0 0 0 0 0 71548 0 0 188 0
```

cpu | cpuN: list of values

- (1) user
 - ...
 - (3) system
 - (4) idle
 - (5) iowait
 - (6) irq
 - ...
 - unit: often 1/100 of second
- > sysconf(_SC_CLK_TCK)

intr: #of interrupts serviced

processes: #total spawned processes

procs_running: #currently running procs.

procs_blocked: #procs. block -> waiting for IO

... many more

which values are shown is system dependent!

cat /proc/meminfo

collection of memory usage
information of the system

MemTotal: RAM available in total
MemFree: non-occupied RAM (LowFree + HighFree)
MemAvailable: estimation of free RAM used for new process
...
HighMem: user-space (accessible RAM)
LowMem: user-space + kernel (accessible RAM)
...
Dirty: memory waiting to be written to disk
WriteBack: memory that is currently written back to disk
...
Mapped: memory used via mmap()

```
MemTotal:      7898720 kB
MemFree:       543120 kB
MemAvailable:  624044 kB
Buffers:       2616 kB
Cached:        878404 kB
SwapCached:    16304 kB
Active:        764860 kB
Inactive:      2596564 kB
Active(anon):  504968 kB
Inactive(anon): 2492872 kB
Active(file):  259892 kB
Inactive(file): 103692 kB
Unevictable:   93500 kB
Mlocked:       44 kB
SwapTotal:     2097148 kB
SwapFree:      1555964 kB
Dirty:         744 kB
Writeback:     0 kB
AnonPages:    2561828 kB
Mapped:        354120 kB
Shmem:         519572 kB
KReclaimable:  83412 kB
Slab:          926536 kB
SReclaimable:  83412 kB
```

cat /proc/<pid>/stat

```
Lukas@buttercup:~$ cat /proc/1/stat
1 (systemd) S 0 1 1 0 -1 4194560 47360 796817 1550 39496 152 179 1815 3294 20 0
1 0 15 173076480 2862 18446744073709551615 1 1 0 0 0 0 671173123 4096 1260 0 0 0
17 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

- (1) pid
- (2) program name
- (3) state
 - **R**unning
 - **S**leeping
 - **D** Waiting
 - **Z**ombie
 - **T** Stopped
 - **X** Dead
 - ...
- (4) ppid: pid of the parent process
- ...
- (14) utime: time spent in user mode
- (15) stime: time spent in kernel mode
- ...
- (20) num_threads
- (22) starttime
- ...


```
cat /proc/<pid>/*
```

```
cd /proc/self -> /proc/<pid>
```

```
cd /proc/<pid>/cwd
```

```
cat /proc/<pid>/cmdline
```

```
cat /proc/<pid>/status
```

```
cat /proc/<pid>/environ | tr '\000' '\n'
```

```
cat /proc/<pid>/io
```

