Command Line Tools
Overview

- Tools for monitoring system activity
  - Processor utilization, saturation and system load
  - Processor and process statistics
  - System memory activity
  - Syscalls
  - Stacks
  - Address spaces
**Terminology**

- **User time**
  - Time spent by the processor executing in user space
  - Instructions of the user program

- **System time**
  - Time spent by the processor executing in kernel context
  - Syscalls, dedicated kernel threads, interrupt handlers

- **Idle time**
  - Time spent by the processor executing other code than threads and interrupt handlers
Invariant

- \%usr – percentage of user time
- \%sys – percentage of system time
- \%idle – percentage of idle time

\%usr + \%sys + \%idle = 100
Terminology (3)

- **Utilization**
  - Percentage of time spent by doing useful work
  - \( \%\text{usr} + \%\text{sys} \)

- **Saturation**
  - How much more work is there than the processors can currently handle
  - Many possible measures
    - Commonly used: The length of the scheduling run queues + number of non-idle CPUs
Interpretation

- **Utilization**
  - Actually not a very good indicator of system health
  - 100 % utilization might be OK
    - But it can also signify a gradual decline in system performance
  - < 100 % utilization means wasted CPU cycles
    - But also a reserve for a sudden peak performance demand
  - *Nota bene*: Utilization is **measured** over a time interval
    - It is a **cumulative value** in each measurement interval
Interpretation (2)

- **Saturation**
  - Saturation threshold $T$
    - Optimal value where each CPU is utilized by exactly one thread ($T = \text{#CPUs}$)
  - Sustained saturation $> T$
    - Gradual performance degradation
    - Indicates potential ideal speedup if more processors were added to the system
  - Nota bene: Saturation is *averaged* over a time interval
    - Sampled number of threads running and in ready queues
    - Usually an *exponential moving average*
Observing load indicators

- **vmstat**
  - Cumulative and average values each time interval
  - **kthr:r** – length of the run queues for all CPUs
    - → *Non-saturated system* (saturation below threshold)
  - **cpu:us, cpu:sy, cpu:id** – utilization in percents
    - → *Mostly idle system* (utilization between 30 – 40 %)

```
# vmstat 5
kthr memory page disk faults cpu
r b w swap free re mf pi po fr de sr s1 s2 in sy cs us sy id
0 0 0 3540164 362860 0 3 0 0 0 0 1 2 -0 0 0 655 8803 819 1 28 70
0 0 0 3474752 298976 2 12 0 0 0 0 0 0 0 0 655 24510 835 4 38 58
0 0 0 3474652 298972 0 1 0 0 0 0 8 0 0 0 669 25881 902 4 38 57
```
Observing load indicators (2)

- **psrinfo**
  - Status of processors

- **uptime**
  - System uptime and standardized saturation
    - System load average over the last 1, 5 and 15 minutes
    - Exponentially dumped moving average of the number of running and runnable threads on all CPUs

```bash
# psrinfo
0   on-line   since 05/04/2014 12:21:06
1   on-line   since 05/04/2014 12:21:09
2   on-line   since 05/04/2014 12:21:09
3   on-line   since 05/04/2014 12:21:09

# uptime
2:10pm  up 6 days 1:48, 4 users, load average: 1.14, 1.18, 1.18
```
### Observing load indicators (3)

- **`sar`**
  - **Custom saturation**
    - **runq-sz** – run queue size average during the time interval
    - **%runocc** – run queue occupancy
      - Percentage of time when the runq-sz was non-zero

```bash
# sar -q 1 5
SunOS zulu.ms.mff.cuni.cz 5.11 11.2 i86pc 05/04/2014

<table>
<thead>
<tr>
<th>Time</th>
<th>runq-sz</th>
<th>%runocc</th>
<th>swpq-sz</th>
<th>%swpocc</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:51:26</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:51:31</td>
<td>1.0</td>
<td>40</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>15:51:36</td>
<td>1.5</td>
<td>40</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>15:51:41</td>
<td>1.3</td>
<td>60</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>15:51:46</td>
<td>1.0</td>
<td>40</td>
<td>0.0</td>
<td>0</td>
</tr>
</tbody>
</table>

Average 1.2 36 0.0 0
Sampling considerations

- **Standard tools**
  - Usually use the default system scheduling frequency
    - 100 Hz on Solaris
    - Possibility to miss activity that starts and completes between two sampling ticks

- **Microstate accounting**
  - Use of high resolution timers for accounting for finer state changes
### Observing processors

- **mpstat**
  - Per-processor statistics
    - Page faults, interrupts, context switches, mutex and rwlock events, syscalls, user, system and idle times

```bash
# mpstat 1
CPU  minf  mjf  xcal  intr  ithr  csw  icsw  migr  smtx  srw  syscl  usr  sys  wt  idl
   0  152   0    7   416   202  138   10   13    7    3  2566    1  50   0  48
   1  145   0    7   220     1  209   10   17    8    3  2514    2  29   0  69
   2  203   0   10    19    3  271   10   15    8    4  1780    1  18   0  81
   3  157   0    8   15    0  239    8   16    8    3  2214    1  18   0  81
```
Observing processes and threads

- **prstat**

  - Microstates per thread
    - **LAT** – latency
      - How long the thread had to wait for a CPU
      - Possible speed-up estimate

```
# prstat
PID  USERNAME  SIZE   RSS  STATE  PRI  NICE  TIME    CPU  PROCESS/NLWP
1497 root  1118M  1098M  cpu3  10   0 147:01:03  25% VirtualBox/14
11012 root  6828K  1972K  cpu2   0   0  0:12:56  7.0% bash/1
18465 root  6828K  1972K  sleep  0   0  0:13:08  7.0% bash/1
2901 root  6824K  1968K  cpu0   0   0  0:28:22  6.9% bash/1
1587 root  1100M  1081M  sleep  59   0  27:18:05  5.0% VirtualBox/14
11372 root  6948K  3492K  cpu1  59   0  0:00:00  0.0% prstat/1
  549 root  37M   19M   sleep  59   0  0:07:14  0.0% Xorg/1
```

Total: 149 processes, 352 lwps, load averages: 5.15, 5.18, 5.08

```
# prstat -mL
PID  USERNAME  USR  SYS  TRP  TFL  DFL  LCK  SLP  LAT  VCX  ICX  SCL  SIG  PROCESS/LWPID
1497 root  1.2  98  0.0  0.0  0.0  0.0  0.0  0.5  218  269  14K  0  VirtualBox/6
18465 root  16  12  0.1  0.0  0.0  0.0  0.0  55  17  1K  1K  15K  130  bash/1
1587 root  6.2  21  0.0  0.0  0.0  0.0  0.0  72  1.0  457  98  42K  0  VirtualBox/6
```
Observing system memory

- **vmstat**
  - System memory statistics
  - **sr** – scan rate
    - Speed (in pages/second) the system is scanning memory to reclaim pages
    - Indicates *memory pressure*

# vmstat -p 1

<table>
<thead>
<tr>
<th>memory</th>
<th>page</th>
<th>executable</th>
<th>anonymous</th>
<th>filesystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>swap</td>
<td>free re mf fr de sr epi epo epf api apo apf fpi fpo fpf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3538820</td>
<td>361752 130 1517 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3498864</td>
<td>323956 11 58 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3498760</td>
<td>323884 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3498760</td>
<td>323884 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Observing system memory (2)

- **vmstat**
  - System memory statistics
  - File system and executable paging
    - Necessary
  - Anonymous paging
    - Indicates *memory shortage*

```bash
# vmstat -p 1
```

<table>
<thead>
<tr>
<th>memory</th>
<th>page</th>
<th>executable</th>
<th>anonymous</th>
<th>filesystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>swap 3538820</td>
<td>free 361752</td>
<td>re 130</td>
<td>mf 1517</td>
<td>0 0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fr 0</td>
<td>de 0</td>
<td>sr 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>epi 0</td>
<td>epo 0</td>
<td>epf 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>api 0</td>
<td>apo 0</td>
<td>apf 0 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fpi 0</td>
<td>fpo 0</td>
<td>fpf 0 0 0</td>
</tr>
</tbody>
</table>

```
Observing system memory (3)

- **vmstat -S**
  - System swap statistics
  - **kthr:w** – number of threads that were swapped-out
    - Indicates a **severe memory shortage**

```
# vmstat -S 1
  kthr memory page disk faults cpu
    r b w swap free si so pi po fr de sr s1 s2 -- --  in sy cs us sy id
  0 0 7 2801904 38124 0 0 4 15099 15369 0 873355 283 0 0 0 15734 26722 45493 1 42 57
```
Observing system memory (4)

- **prstat -mL**
  - Processor statistics with microaccounting
  - TFL – percentage of time the thread has spent processing instruction page faults
  - DFL – percentage of time the thread has spent processing data page faults

```
# prstat -mL
   PID  USERNAME USR SYS TRP  TFL  DFL  LCK  SLP  LAT  VCX  ICX  SCL  SIG  PROCESS/LWPID
 1497  root   1.2  98  0.0  0.0  0.0  0.0  0.0  0.5  218  269  14K   0  VirtualBox/6
 18465 root   16  12  0.1  0.0  0.0  0.0  55  17   1K   1K  15K  130   bash/1
 1587  root   6.2  21  0.0  0.0  0.0  0.0  72  1.0  457  98  42K   0  VirtualBox/6
```
Tracing syscalls

- **truss**
  - Similar to `strace` in Linux

```bash
# truss echo 'Hello world!'
execve("/usr/bin/echo", 0x08047CF4, 0x08047D00) argc = 2
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_ANON,
     -1, 0) = 0xFEFB0000
resolvepath("/usr/lib/ld.so.1", "/lib/ld.so.1", 1023) = 12
resolvepath("/usr/bin/echo", "/usr/bin/echo", 1023) = 13
sysconfig(_CONFIG_PAGESIZE) = 4096
xstat(2, "/usr/bin/echo", 0x080479B8) = 0
open("/var/ld/ld.config", O_RDONLY) Err#2 ENOENT
... fstat64(1, 0x080479E0) = 0
write(1, "Hello world!", 13) = 13
_exit(0)
```
Displaying thread stack

- **pstack**

```bash
# pstack 1587/10
1587: /opt/VirtualBox/amd64/VirtualBox --comment centos --startvm cc4605e0-a
----------------- lwp# 10 / thread# 10 ---------------------
ffffffffd7fff0a234a sigtimedwait (ffffffffd7ffbbcc2eb0, fffffff7ffbbcc2c40, 0)
ffffffffd7fffb8bd84 sigwaitinfo () + c
ffffffffd7ffedf95fe _Z13rttimerThreadP11RTTHREADINTPv () + 38e
ffffffffd7ffedd3b3c rtThreadMain () + 2c
ffffffffd7ffedf8d2b _Z18rtThreadNativeMainPv () + 7b
ffffffffd7ff099de5 _thrp_setup () + 8d
ffffffffd7fff0a0a0 _lwp_start ()
```
Displaying address space

- **pmap**

```
# pmap 8394
8394:  less /etc/passwd
08045000   12K  rw---    [ stack ]
08050000   108K r-x--  /usr/bin/less
0807A000   24K rwx--  /usr/bin/less
08080000   32K rwx--    [ heap ]
FEA00000  2416K r-x--  /usr/lib/locale/en_US.UTF-8/en_US.UTF-8.so.3
FEC6B000    4K rwx--  /usr/lib/locale/en_US.UTF-8/en_US.UTF-8.so.3
FED9E000    4K rwxs-   [ anon ]
...
FEFFD000   4K rwx--  /lib/ld.so.1
total      4420K
```