

# Know your tools: cgroups

Dariusz Puchalak

[Dariusz\\_Puchalak < at > ProbosIT.pl](mailto:Dariusz_Puchalak@ProbosIT.pl)



Platinum  
TRAINING  
PARTNER

Novell®

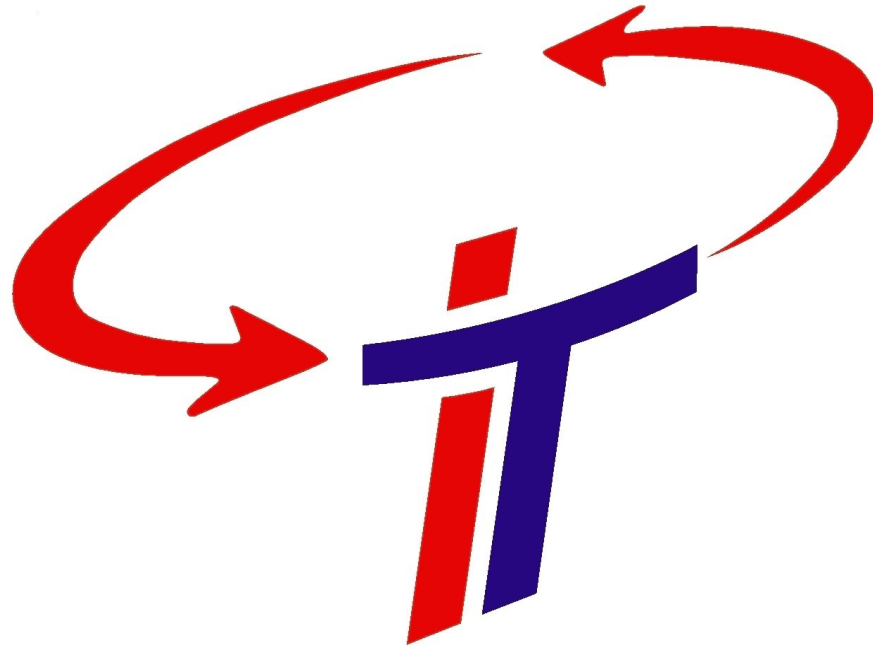
Compendium CE  
posiada status  
Novell Training  
Services Partner -  
Platinum level



redhat®  
CERTIFIED  
TRAINING  
PARTNER

Compendium  
CE posiada  
status Red Hat  
Certified  
Training Partner





**ProbosIT**

# Zarządzanie systemem IT

# Typowe podejście

Dodajemy:

- Dyski
- Procesory
- Pamięć

Czy można inaczej?

# /etc/security/limits.conf

```
#*          soft   core           0
#root       hard   core          100000
#*          hard   rss           10000
#@student   hard   nproc         20
#@faculty   soft   nproc         20
#@faculty   hard   nproc         50
#ftp        hard   nproc         0
#ftp        -     chroot        /ftp
#@student   -     maxlogins     4
```

# Control Groups

Control Groups provide a mechanism for aggregating/partitioning sets of tasks, and all their future children, into hierarchical groups with specialized behaviour.



# Features

- Resource limiting
- Prioritization
- Accounting
- Isolation
- Control

# What can be controlled

- CPU
- Memory
- Disk
- Network

# Tools

?

# Managing cgroups: tools

- mkdir
- echo
- mount

# CPU management

```
mkdir /dev/cpu
```

```
mount -t cgroup -ocpu cpu /dev/cpu
```

```
mkdir /dev/cpu/background
```

```
echo $PID > /dev/cpu/background/tasks
```

```
echo 1 > /dev/cpu/background/cpu.shares
```

# CPU management

```
mkdir /dev/cpu/admins
```

```
mkdir /dev/cpu/users
```

```
mkdir /dev/cpu/hackers
```

```
echo 400 > /dev/cpu/admins/cpu.shares
```

```
echo 100 > /dev/cpu/users/cpu.shares
```

```
echo 1 > /dev/cpu/hackers/cpu.shares
```

```
echo ... > /dev/cpu/.../tasks
```

# CPU management

```
moon:/# ls -l /dev/cpu/  
total 0  
drwxr-xr-x 2 root root 0 Jan 16 10:33 admins  
drwxr-xr-x 2 root root 0 Jan 16 10:26 background  
--w--w--w- 1 root root 0 Jan 14 17:20 cgroup.event_control  
-r--r--r-- 1 root root 0 Jan 14 17:20 cgroup.procs  
-rw-r--r-- 1 root root 0 Jan 14 17:20 cpu.rt_period_us  
-rw-r--r-- 1 root root 0 Jan 14 17:20 cpu.rt_runtime_us  
-rw-r--r-- 1 root root 0 Jan 14 17:20 cpu.shares  
drwxr-xr-x 2 root root 0 Jan 16 10:33 hackers  
-rw-r--r-- 1 root root 0 Jan 14 17:20 notify_on_release  
-rw-r--r-- 1 root root 0 Jan 14 17:20 release_agent  
drwxrwxr-x 3 root root 0 Jan 14 17:20 sysdefault  
-rw-r--r-- 1 root root 0 Jan 14 17:20 tasks  
drwxr-xr-x 2 root root 0 Jan 16 10:33 users
```

# CPU management

How many `cpu.shares` is 100%?



# CPU 100% shares is

```
cat /dev/cpu/*/cpu.shares | awk ' BEGIN  
  {S=0} { S=S+$1 } END {print S}'
```

# cpuset

```
mkdir /dev/cgroup
```

```
mount -t cgroup -ocpuset cpuset \  
/dev/cgroup
```

```
mkdir /dev/cgroup/produkcja
```

```
echo 1 > /dev/cgroup/produkcja/cpuset.cpu
```

# cpuset

```
moon:/# ls -l /dev/cgroup/produkcja/
```

```
total 0
```

```
--w--w--w- 1 root root 0 Jan 16 10:42 cgroup.event_control
-r--r--r-- 1 root root 0 Jan 16 10:42 cgroup.procs
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.cpu_exclusive
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.cpus
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.mem_exclusive
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.mem_hardwall
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.memory_migrate
-r--r--r-- 1 root root 0 Jan 16 10:42 cpuset.memory_pressure
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.memory_spread_page
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.memory_spread_slab
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.mems
-rw-r--r-- 1 root root 0 Jan 16 10:42 cpuset.sched_load_balance
-rw-r--r-- 1 root root 0 Jan 16 10:42
cpuset.sched_relax_domain_level
-rw-r--r-- 1 root root 0 Jan 16 10:42 notify_on_release
-rw-r--r-- 1 root root 0 Jan 16 10:42 tasks
```

# exclusive

If a cpuset is cpu or mem exclusive, no other cpuset, other than a direct ancestor or descendant, may share any of the same CPUs or Memory Nodes.

# Memory pressure

The `memory_pressure` of a cgroup provides a simple per-cgroup metric of the rate that the tasks in a cgroup are attempting to free up in use memory on the nodes of the cgroup to satisfy additional memory requests.

# memory

```
mkdir /dev/memory
```

```
mount -t cgroup none -o memory \  
/dev/memory/
```

```
moon:/# ls /dev/memory/  
cgroup.event_control  
cgroup.procs  
memory.failcnt  
memory.force_empty  
memory.limit_in_bytes  
memory.max_usage_in_bytes  
memory.memsw.failcnt  
memory.memsw.limit_in_bytes  
memory.memsw.max_usage_in_bytes  
memory.memsw.usage_in_bytes  
memory.move_charge_at_immigrate  
memory.oom_control  
memory.soft_limit_in_bytes  
memory.stat  
memory.swappiness  
memory.usage_in_bytes  
memory.use_hierarchy  
notify_on_release  
release_agent  
tasks
```

# Network

TC



# Disk

```
moon:/# mkdir /cgroup
```

```
moon:/# mount -t cgroup -o blkio none \  
/cgroup
```

```
moon:/# mkdir -p /cgroup/test1/ \  
/cgroup/test2
```

# Disk

```
echo 1000 > /cgroup/test1/blkio.weight
```

```
echo 500 > /cgroup/test2/blkio.weight
```

# blkio

```
Echo "8:16 1048576" > \  
/cgroup/blkio/blkio.read_bps_device
```

```
dd if=/mnt/common/zerofile of=/dev/null  
bs=4K count=1024
```

```
4194304 bytes (4.2 MB) copied, 4.0001 s,  
1.0 MB/s
```

# libpam-cgroups

cgcreate  
cgdelete  
cgset  
cgclassify  
cgexec

Cat /proc/\$PID/cgroup

# Wonder patch

The ~200 Line Linux Kernel Patch That  
Does Wonders

# Memory

Questions?