Control Group Subsystems in RHEL7

Finnbarr P. Murphy

(fpm@fpmurphy.com)

Control groups (*cgroups*) are a Linux kernel feature that enables you to allocate resources — such as CPU time, system memory, disk I/O, network bandwidth, etc. — among hierarchically ordered groups of processes running on a system. Initially developed by Google engineers Paul Menage and Rohit Seth in 2006 under the name “process containers”, it was merged into kernel version 2.6.24 and extensively enhanced since then. RHEL6 was the first Red Hat distribution to support *cgroups*.

Cgroups provide system administrators with fine-grained control over allocating, prioritizing, denying, managing, and monitoring system resources. A *cgroup* is a collection of processes that are bound by the same criteria. These groups are typically hierarchical, where each group inherits limits from its parent group.

The problem with the traditional use of *cgroups* is summarized by the following except from a Red hat guide:

> Control Groups provide a way to hierarchically group and label processes, and to apply resource limits to them. Traditionally, all processes received similar amount of system resources that administrator could modulate with the process niceness value. With this approach, applications that involved a large number of processes got more resources than applications with few processes, regardless of the relative importance of these applications.

In RHEL6, administrators had to build custom *cgroup* hierarchies to meet their application needs. In RHEL7, it is no longer necessary to build custom cgroups as resource management settings have moved from the process level to the application level via binding the system of *cgroup* hierarchies with the *systemd* unit tree. By default, *systemd* automatically creates a hierarchy of slices, scopes and services to provide a unified structure for the *cgroup* tree.

A resource controller, also called *cgroup subsystem*, represents a single resource, such as CPU time or memory. The Linux kernel provides a range of resource controllers which can be seen by cat’ing */proc/cgroups*

```
# cat /proc/cgroups
#subsys_name    hierarchy    num_cgroups    enabled
cpuset    2    1    1
cpu    3    1    1
cpuacct    3    1    1
memory    4    1    1
devices    5    1    1
freezer    6    1    1
net_cls    7    1    1
blkio    8    1    1
perf_event    9    1    1
hugetlb    10    1    1
```


A quick explanation of each of the above *cgroup* subsystems:

- *cpuset*: Assigns individual CPUs (on a multicore system) and memory nodes to tasks in a cgroup.
- *cpu*: Uses the scheduler to provide cgroup tasks access to the CPU.
- *cpuacct*: Automatic reports on CPU resources used by tasks in a cgroup.
- *memory*: Sets limits on memory use by tasks in a cgroup, and generates automatic reports on memory resources used by those tasks.
- *devices*: Allows or denies access to devices by tasks in a cgroup.
- *freezer*: Suspends or resumes tasks in a cgroup.
- *net_cls*: Tags network packets with a class identifier (classid) to enable the Linux traffic controller to identify packets originating from a particular cgroup task.
- *blkio*: Sets limits on input/output access to and from block devices.
- *perf_event*: Permits monitoring cgroups with the *perf* tool.
- *hugetlb*: Enables large virtual memory pages and the enforcing of resource limits on these pages.

You can also use the *lsusbsys* utility to view the control group subsystems.:

```
# lssubsys
cpuset
cpu,cpuacct
memory
devices
freezer
net_cls
blkio
perf_event
hugetlb

# lssubsys -im
cpuset /sys/fs/cgroup/cpuset
cpu,cpuacct /sys/fs/cgroup/cpu,cpuacct
memory /sys/fs/cgroup/memory
devices /sys/fs/cgroup/devices
freezer /sys/fs/cgroup/freezer
net_cls /sys/fs/cgroup/net_cls
blkio /sys/fs/cgroup/blkio
perf_event /sys/fs/cgroup/perf_event
hugetlb /sys/fs/cgroup/hugetlb

# mount | grep cgroup
tmpfs on /sys/fs/cgroup type tmpfs (rw,nosuid,nodev,noexec,seclabel,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,releas
e_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct,
cpu)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/net_cls type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_even
t)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
```

If you install the *kernal-doc* RPM, you will find documentation on each of the above *cgroup* subsystems under */usr/share/doc/kernel-doc-/Documentation/cgroups/*.

