

# Cgroups as a Resource Management Technique

Vladimir Zapolskiy  
Nokia Corporation  
Keilalahdentie, Finland  
vz@mleia.com

## Abstract

Control Groups is a mechanism to manage process groups and distribute system resources among these groups. Its scalability and convenience of usage attracts wide attention to the framework, which is proven for its irreplaceable usefulness.

**Index Terms:** Resource management, Software architecture, Multitasking.

## I. RESOURCES

In despite of exponential rate growth in many areas of electronics and computer industry products, there is a permanent need of more and more advanced hardware, networking protocols, mass storage capacities and so forth. Retrospectively analysing it seems that the last mile in the race of improvements can't be approached at all and instant power isn't ever enough. However comprehensive exploitation of given engineering technologies allows to achieve better effectiveness almost freely.

Introduction of a term *resource* as an object of limited availability [1] gives a good start to a fruitful analysis within the framework of resource management and gives patterns of solutions for some classes of practical problems related to optimization and virtualization. Such diverse objects as network bandwidth, disc I/O, RAM and swap memory, CPU slices per task, RT scheduling capabilities, current consumption have a proximity of being a resource in general sense.

## II. CGROUPS

One of the most progressive system resource controller is cgroups [2], developed and naturally integrated into the Linux kernel. The basic idea of cgroups is to provide a mechanism to create, dynamically modify hierarchically compound sets of tasks, including both threads and processes, and assign QoS and limit values for these externally controlled groups of tasks.

As for today cgroups are successfully and widely used in the field, moreover the area of cgroups application stretches from high performance servers towards embedded devices, hopefully there is a strong tendency to cover both sides by Linux based solutions, and e.g. the innovative *systemd* substitution of System V *init* daemon compulsorily requires cgroups support in the kernel. One more example is sophisticated usage of cgroups in combination with process tracing by means of netlink based *proc connector* and user defined process policies in Nokia N900 and Nokia N9 products, such a technological mix allowed to achieve crucial level of performance and user experience having full blooded multitasking of applications.

Nowadays there is a large demand of cgroups mechanism extension into new areas, for instance control of system calls or power consumption per task, related tasks and discussions are open and accessible for participation.

## REFERENCES

- [1] Nikolay Igotti, "Resource Management and Need of Virtual Machines," *PDMI Computer Club lectures*, 2010.
- [2] Paul Menage, "Cgroups," *Linux kernel documentation*, 2008.