CLOUD COMPUTING APPLICATIONS

Containers: cgroups

Prof. Reza Farivar

Cgroups

- Control Groups
- Linux kernel feature which limits, isolates and measures resource usage of a group of processes
  - Since Linux Kernel 2.6.24
- Resources quotas for memory, CPU, network and IO
- Create a control group and assign resource limits on it:
  - e.g. 3GB of memory limit and 70% of CPU
- Add a process id to the group
- Process resource usage will be throttled
  - The application may exceed the limits in normal scenarios
  - it will be throttled back to pre set limits in case system is facing resource crunch

Cgroup Controllers (v2)

Controller	Brief Description
cpu	Cgroups can be guaranteed a minimum number of "CPU shares" when a system is busy. This does not limit a cgroup's CPU usage if the CPUs are not busy.
cpuset	Provides accounting for CPU usage by groups of processes
freezer	can suspend and restore (resume) all processes in a cgroup. Freezing a cgroup */A* also causes its children, for example, processes in */A/B*, to be frozen.
hugetlb	This supports limiting the use of huge pages by cgroups
io	The *io* cgroup controls and limits access to specified block devices by applying IO control in the form of throttling and upper limits against leaf nodes and intermediate nodes in the storage hierarchy. Two policies are available. The first is a proportional- weight time-based division of disk implemented with CFQ. This is in effect for leaf nodes using CFQ. The second is a throttling policy which specifies upper I/O rate limits on a device.
memory	The memory controller supports reporting and limiting of process memory, kernel memory, and swap used by cgroups.
Perf_event	This controller allows *perf* monitoring of the set of processes grouped in a cgroup.
pids	This controller permits limiting the number of process that may be created in a cgroup (and its descendants).
rdma	The RDMA controller permits limiting the use of RDMA/IB- specific resources per cgroup.

Detailed Documentation at: https://www.kernel.org/doc/Documentation/admin-guide/cgroup-v2.rst

Cgroup example

- Controllers mounted in the cgroups file system
  - /cgroup directory
    - `/sys/fs/cgroup/memory`
    - `/sys/fs/cgroup/cpu`
- Making a control group
  - `/cgroup/memory/mytestcgroup`
- Setting limits
  - `echo 2097152 > /sys/fs/cgroup/memory/mytestcgroup/memory.limit_in_bytes`
  - `echo 2097152 > /sys/fs/cgroup/memory/mytestcgroup/memory.memsw.limit_in_bytes`
    - Set both memory AND swap space limit to 2 MB
- Running a process
  - `cgexec -g memory:mytestcgroup ./<binary_name>`

Cgroup Scheduling

- When we think of containers as lightweight VMs, it is natural to think of resources in terms of discrete resources such as number of processors.
- However, the Linux kernel schedules processes dynamically, just as the hypervisor schedules requests onto discrete hardware.
- the cpu subsystem schedules CPU access to each cgroup using either the Completely Fair Scheduler (CFS)—the default on Linux and Docker—or the Real-Time Scheduler (RT).
- Scheduling cgroups in CFS requires us to think in terms of time slices instead of processor counts.
- CPU shares provide tasks in a cgroup with a relative amount of CPU time, providing an opportunity for the tasks to run.
- The file `cpu.shares` defines the number of shares allocated to the cgroup.