

Analyzing cgroup behavior in case of SET

Even after the support of Set is added in docker, one important aspect is to verify the behavior of cgroups upon modification i.e. we need to check whether cgroups actually make changes in the resource allocation for that particular container. A new cgroup test was run to verify the set functionality and understand the behavior of cgroups.

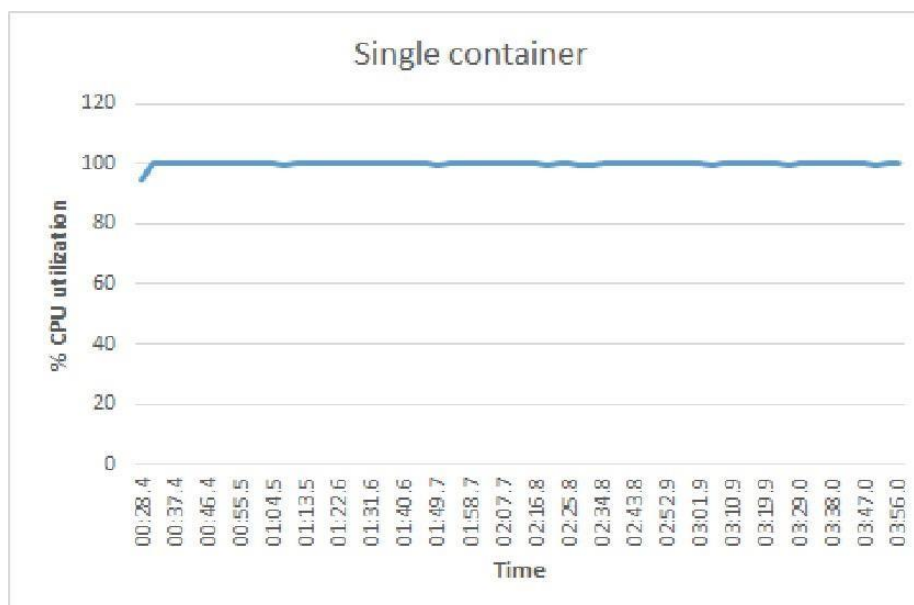
Test Details:

In the test, I replicate docker's interaction with cgroup using a shell script and a simple python daemon which throttles the CPU. The monitoring of behavior is also done using shell script which directly reads cgroup's VFS interface. Test creates two cgroups (acting as two containers - con1, con2) with equal CPU shares and attach process running python daemon to each of these cgroups. Then, we change the CPU share of second container and observe the behavior. In first run, the test directly writes to cgroup files associated with second container to change CPU share after 30 seconds.

Observations:

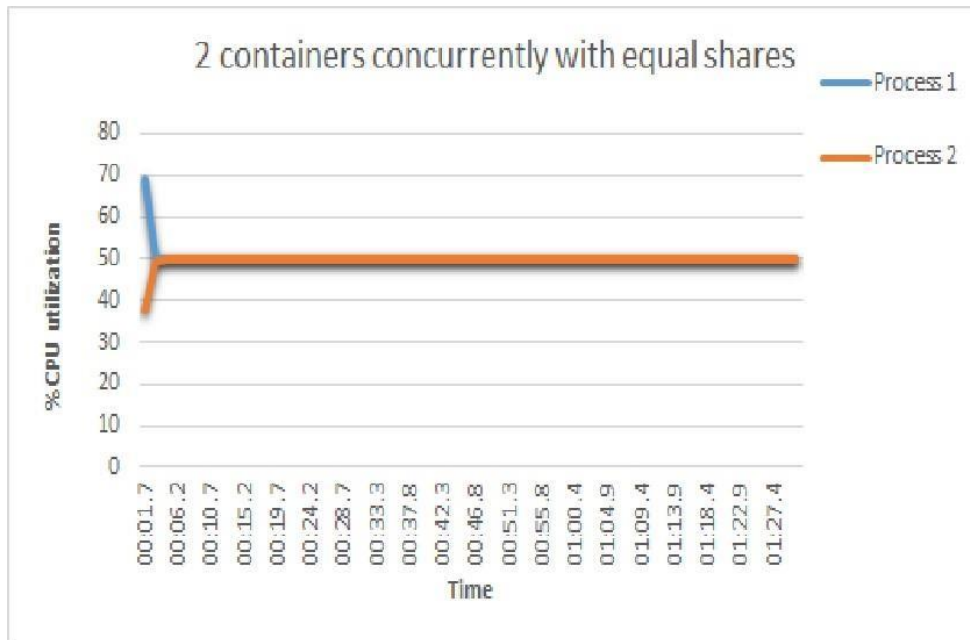
Part 1: When the containers are pinned to single processor

1) Running single container:



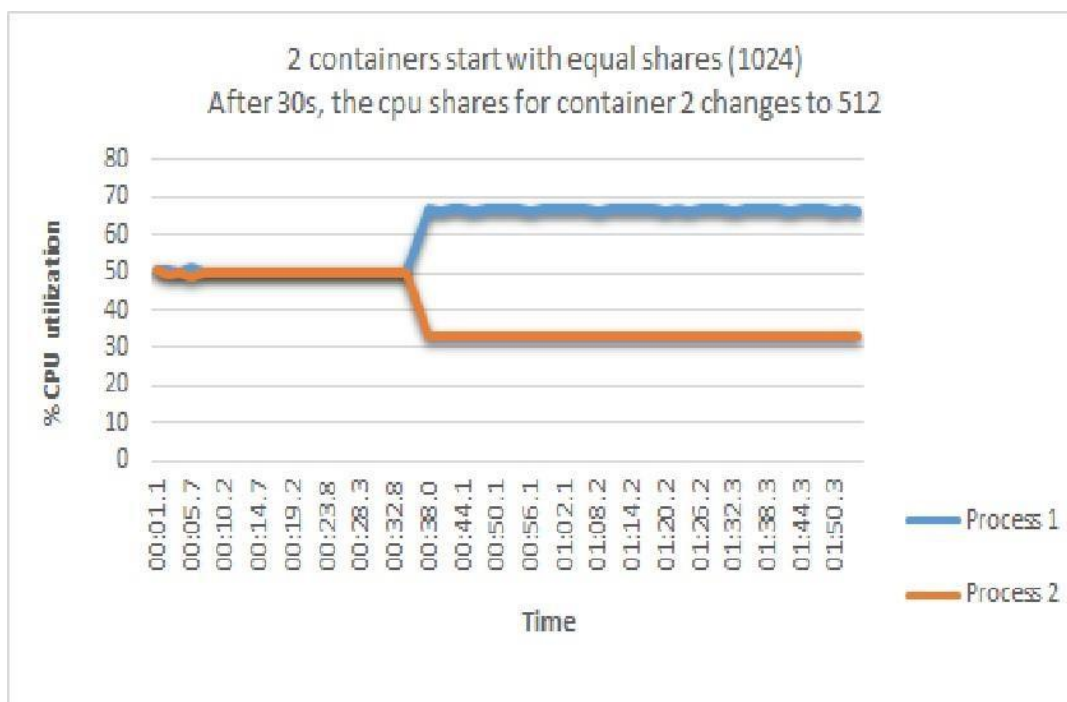
Result: When only one container runs, it utilizes 100% CPU as expected

2) Running 2 containers on single processor with equal shares:



Result: When 2 containers with equal share are run on single processor, each container utilizes 50% CPU which is in conformance with the expected behavior.

3) Running 2 containers on single processor with equal shares (1024) and change cpushares of second container to 512 after 30 seconds.



Result: When 2 containers are run with equal shares i.e. 1024, they utilize 50% CPU each till 30 seconds. Now, when cpushares of container 2 are changed to 512, the CPU utilization of Container 2 drops to 33% and the CPU utilization of container 1 rises to 67%. This behavior is in conformance with the expected behavior.