
Subject: Re: [ckrm-tech] [PATCH 1/9] Containers (V9): Basic container framework
Posted by [Paul Jackson](#) on Wed, 02 May 2007 06:12:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Balbir wrote:

> Would it be possible to extract those test cases and integrate them
> with a testing framework like LTP? Do you have any regression test
> suite for cpusets that can be made available publicly so that
> any changes to cpusets can be validated?

There are essentially two sorts of cpuset regression tests of interest.

I have one such test, and the batch scheduler developers have various tests of their batch schedulers.

1) Testing batch schedulers against cpusets:

I doubt that the batch scheduler developers would be able to extract a cpuset test from their tests, or be able to share it if they did. Their tests tend to be large tests of batch schedulers, and only incidentally test cpusets -- if we break cpusets, in sometimes even subtle ways that they happen to depend on, we break them.

Sometimes there is no way to guess exactly what sorts of changes will break their code; we'll just have to schedule at least one run through one or more of them that rely heavily on cpusets before a change as big as rebasing cpusets on containers is reasonably safe. This test cycle won't be all that easy, so I'd wait until we are pretty close to what we think should be taken into the mainline kernel.

I suppose I will have to be the one co-ordinating this test, as I am the only one I know with a presence in both camps.

Once this test is done, from then forward, if we break them, we'll just have to deal with it as we do now, when the breakage shows up well down stream from the main kernel tree, at the point that a major batch scheduler release runs into a major distribution release containing the breakage. There is no practical way that I can see, as an ongoing basis, to continue testing for such breakage with every minor change to cpuset related code in the kernel. Any breakage found this way is dealt with by changes in user level code.

Once again, I have bcc'd one or more developers of batch schedulers, so they can see what nonsense I am spouting about them now ;).

2) Testing cpusets with a specific test.

There I can do better. Attached is the cpuset regression test I use. It requires at least 4 cpus and 2 memory nodes to do anything useful. It is copyright by SGI, released under GPL license.

This regression test is the primary cpuset test upon which I relied during the development of cpusets, and continue to rely. Except for one subtle race condition in the test itself, it has not changed in the last two to three years.

This test requires no user level code not found in an ordinary distro. It does require the taskset and numactl commands, for the purposes of testing certain interactions with them. It assumes that there are not other cpusets currently setup in the system that happen to conflict with the ones it creates.

See further comments within the test script itself.

--

I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.925.600.0401

File Attachments

1) [cpuset_test](#), downloaded 368 times
