
Subject: Re: [RFC] slub: show dead memcg caches in a separate file

Posted by [Glauber Costa](#) on Tue, 08 May 2012 03:30:16 GMT

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

On 05/07/2012 07:04 PM, Suleiman Souhlal wrote:

> On Thu, May 3, 2012 at 11:47 AM, Glauber Costa<glommer@parallels.com> wrote:

>> One of the very few things that still unsettles me in the kmem

>> controller for memcg, is how badly we mess up with the

>> /proc/slabinfo file.

>>

>> It is alright to have the cgroup caches listed in slabinfo, but once

>> they die, I think they should be removed right away. A box full

>> of containers that come and go will rapidly turn that file into

>> a supreme mess. However, we currently leave them there so we can

>> determine where our used memory currently is.

>>

>> This patch attempts to clean this up by creating a separate proc file

>> only to handle the dead slabs. Among other advantages, we need a lot

>> less information in a dead cache: only its current size in memory

>> matters to us.

>>

>> So besides avoiding pollution of the slabinfo files, we can access

>> dead cache information itself in a cleaner way.

>>

>> I implemented this as a proof of concept while finishing up

>> my last round for submission. But I am sending this separately

>> to collect opinions from all of you. I can either implement

>> a version of this for the slab, or follow any other route.

>

> I don't really understand why the "dead" slabs are considered as

> polluting slabinfo.

>

> They still have objects in them, and I think that hiding them would

> not be the right thing to do (even if they are available in a separate

> file): They will incorrectly not be seen by programs like slabtop.

>

Well, technically speaking, they aren't consider. I consider. The difference is subtle, but boils down to if no one else consider this a problem... there is no problem.

Now let me expand on the subject of why I do consider this unneeded information (needed, just not here)

Consider a hosting box with ~100 caches. Let us say that a container touches 50 of them, we still have 50 caches per container. Objects in those caches, may take a long time to go away. Let's say, in 40 of those caches.

The number of entries in /proc/slabinfo is not proportional to the number of active containers: It becomes proportional to the number of containers that *ever* existed on the machine - even if those numbers drop with time, they still can drop slowly.

In use cases where containers come and go frequently, before a shrinker can be called to wipe some of them out, we are easily in the 1000s of lines in /proc/slabinfo. It becomes too much information, and it usually makes it hard to find the one you are looking for.

But there is another aspect: those dead caches have one thing in common, which is the fact that no new objects will ever be allocated on them. You can't tune them, or do anything with them. I believe it is misleading to include them in slabinfo.

The fact that the caches change names - to append "dead" may also break tools, if that is what you are concerned about.

For all the above, I think a better semantics for slabinfo is to include the active caches, and leave the dead ones somewhere else.
