Subject: Re: [PATCH v4 02/11] memcg: document cgroup dirty memory interfaces

Posted by Wu Fengguang on Fri, 29 Oct 2010 11:03:31 GMT

View Forum Message <> Reply to Message

Hi Greg,

On Fri, Oct 29, 2010 at 03:09:05PM +0800, Greg Thelen wrote:

> Document cgroup dirty memory interfaces and statistics.
>
> Signed-off-by: Andrea Righi <arighi@develer.com>
> Signed-off-by: Greg Thelen <gthelen@google.com>
> ---

> +Limiting dirty memory is like fixing the max amount of dirty (hard to reclaim)
> +page cache used by a cgroup. So, in case of multiple cgroup writers, they will
> +not be able to consume more than their designated share of dirty pages and will
> +be forced to perform write-out if they cross that limit.

It's more pertinent to say "will be throttled", as "perform write-out"
is some implementation behavior that will change soon.

> +- memory.dirty_limit_in_bytes: the amount of dirty memory (expressed in bytes)
> + in the cgroup at which a process generating dirty pages will start itself
> + writing out dirty data. Suffix (k, K, m, M, g, or G) can be used to indicate
> + that value is kilo, mega or gigabytes.

The suffix feature is handy, thanks! It makes sense to also add this
for the global interfaces, perhaps in a standalone patch.

> +A cgroup may contain more dirty memory than its dirty limit. This is possible
> +because of the principle that the first cgroup to touch a page is charged for
> +it. Subsequent page counting events (dirty, writeback, nfs_unstable) are also
> +counted to the originally charged cgroup.
> +
> +Example: If page is allocated by a cgroup A task, then the page is charged to
> +cgroup A. If the page is later dirtied by a task in cgroup B, then the cgroup A
> +dirty count will be incremented. If cgroup A is over its dirty limit but cgroup
> +B is not, then dirtying a cgroup A page from a cgroup B task may push cgroup A
> +over its dirty limit without throttling the dirtying cgroup B task.

It's good to document the above "misbehavior". But why not throttling
the dirtying cgroup B task? Is it simply not implemented or makes no
sense to do so at all?

Thanks,
Fengguang

_____

Containers mailing list
Containers@lists.linux-foundation.org
 https://lists.linux-foundation.org/mailman/listinfo/containe rs