Subject: Re: [RFC][-mm] Memory controller hierarchy support (v1) Posted by KAMEZAWA Hiroyuki on Mon, 21 Apr 2008 00:41:43 GMT

View Forum Message <> Reply to Message

On Sat, 19 Apr 2008 14:04:00 +0530 Balbir Singh <balbir@linux.vnet.ibm.com> wrote: > YAMAMOTO Takashi wrote: >>> - spin lock irgsave(&counter->lock, flags); >>> - ret = res counter charge locked(counter, val); >>> - spin unlock irgrestore(&counter->lock, flags); >>> + *limit exceeded at = NULL; >>> + local_irq_save(flags); >>> + for (c = counter; c != NULL; c = c->parent) { >>> + spin_lock(&c->lock); >>> + ret = res_counter_charge_locked(c, val); >>> + spin unlock(&c->lock); $>>> + if (ret < 0) {$ >>> + *limit exceeded at = c; >>> + goto unroll; >>>+ } > >> + } >>> + local_irq_restore(flags); > >> + return 0; > >> + >>> +unroll: >>> + for (unroll_c = counter; unroll_c != c; unroll_c = unroll_c->parent) { >>> + spin lock(&unroll c->lock); >>> + res counter uncharge locked(unroll c, val); >>> + spin_unlock(&unroll_c->lock); > >> + } >>> + local_irq_restore(flags); >>> return ret; >>> } > > > > i wonder how much performance impacts this involves. >> it increases the number of atomic ops per charge/uncharge and > > makes the common case (success) of every charge/uncharge in a system > > touch a global (ie. root cgroup's) cachelines. > > > Yes, it does. I'll run some tests to see what the overhead looks like. The > multi-hierarchy feature is very useful though and one of the TODOs is to make > the feature user selectable (possibly at run-time) I think multilevel cgroup is useful but this routines handling of hierarchy seems never good. An easy idea to aginst this is making a child borrow some

amount of charge from its parent for reducing checks. If you go this way, please show possibility to reducing overhead in your plan.

BTW, do you have ideas of attributes for children<->parent other than 'limit'? For example, 'priority' between childlen.

Thanks,

-Kame

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