

---

Subject: [PATCH v5 0/8] per-cgroup tcp buffer pressure settings

Posted by [Glauber Costa](#) on Tue, 04 Oct 2011 12:17:52 GMT

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

---

[[ v3: merge Kirill's suggestions, + a destroy-related bugfix ]]

[[ v4: Fix a bug with non-mounted cgroups + disallow task movement ]]

[[ v5: Compile bug with modular ipv6 + tcp files in bytes ]]

Kame, Kirill,

I am submitting this again merging most of your comments. I've decided to leave some of them out:

\* I am not using `res_counters` for `allocated_memory`. Besides being more expensive than what we need, to make it work in a nice way, we'd have to change the `!cgroup` code, including other protocols than `tcp`. Also,

\* I am not using `failcnt` and `max_usage_in_bytes` for it. I believe the value of those lies more in the allocation than in the pressure control. Besides, fail conditions lie mostly outside of the memory cgroup's control. (Actually, a `soft_limit` makes a lot of sense, and I do plan to introduce it in a follow up series)

If you agree with the above, and there are any other pressing issues, let me know and I will address them ASAP. Otherwise, let's discuss it. I'm always open.

All:

This patch introduces per-cgroup tcp buffers limitation. This allows sysadmins to specify a maximum amount of kernel memory that tcp connections can use at any point in time. TCP is the main interest in this work, but extending it to other protocols would be easy.

For this to work, I am hooking it into `memcg`, after the introduction of an extension for tracking and controlling objects in kernel memory. Since they are usually not found in page granularity, and are fundamentally different from userspace memory (not swappable, can't overcommit), they need their special place inside the Memory Controller.

Right now, the `kmem` extension is quite basic, and just lays down the basic infrastructure for the ongoing work.

Although it does not account kernel memory allocated - I preferred to keep this series simple and leave accounting to the slab allocations when they arrive.

What it does is to piggyback in the memory control mechanism already present in `/proc/sys/net/ipv4/tcp_mem`. There is a soft limit, and a hard limit, that will suppress allocation when reached. For each non-root cgroup, however,

the file `kmem.tcp_maxmem` will be used to cap those values.

The usage I have in mind here is containers. Each container will define its own values for soft and hard limits, but none of them will be possibly bigger than the value the box' `sysadmin` specified from the outside.

To test for any performance impacts of this patch, I used `netperf's` `TCP_RR` benchmark on `localhost`, so we can have both `recv` and `snd` in action. For this iteration, I am using the 1% confidence interval as suggested by Rick.

Command line used was `./src/netperf -t TCP_RR -H localhost -i 30,3 -l 99,1` and the results: (I haven't re-run this since nothing major changed from last version, nothing in core)

Without the patch

=====

Local /Remote	
Socket Size	Request Resp. Elapsed Trans.
Send Recv	Size Size Time Rate
bytes Bytes	bytes bytes secs. per sec
16384 87380	1 1 10.00 35356.22
16384 87380	

With the patch

=====

Local /Remote	
Socket Size	Request Resp. Elapsed Trans.
Send Recv	Size Size Time Rate
bytes Bytes	bytes bytes secs. per sec
16384 87380	1 1 10.00 35399.12
16384 87380	

The difference is less than 0.5 %

A simple test with a 1000 level nesting yields more or less the same difference:

1000 level nesting

=====

Local /Remote	
Socket Size	Request Resp. Elapsed Trans.

Send bytes	Recv Bytes	Size bytes	Size bytes	Time secs.	Rate per sec
16384	87380	1	1	10.00	35304.35
16384	87380				

Glauber Costa (8):

Basic kernel memory functionality for the Memory Controller socket: initial cgroup code.  
 foundations of per-cgroup memory pressure controlling.  
 per-cgroup tcp buffers control  
 per-netns ipv4 sysctl\_tcp\_mem  
 tcp buffer limitation: per-cgroup limit  
 Display current tcp memory allocation in kmem cgroup  
 Disable task moving when using kernel memory accounting

```
Documentation/cgroups/memory.txt | 38 +++++-
crypto/af_alg.c | 7 +-
include/linux/memcontrol.h | 56 +++++++
include/net/netns/ipv4.h | 1 +
include/net/sock.h | 127 ++++++++
include/net/tcp.h | 29 +++-
include/net/udp.h | 3 +-
include/trace/events/sock.h | 10 +-
init/Kconfig | 14 ++
mm/memcontrol.c | 371 ++++++++
net/core/sock.c | 104 ++++++
net/decnet/af_decnet.c | 21 +-
net/ipv4/proc.c | 7 +-
net/ipv4/sysctl_net_ipv4.c | 71 ++++++
net/ipv4/tcp.c | 58 +++++-
net/ipv4/tcp_input.c | 12 +-
net/ipv4/tcp_ipv4.c | 24 +-
net/ipv4/tcp_output.c | 2 +-
net/ipv4/tcp_timer.c | 2 +-
net/ipv4/udp.c | 20 +-
net/ipv6/tcp_ipv6.c | 20 +-
net/ipv6/udp.c | 4 +-
net/sctp/socket.c | 35 +++-
23 files changed, 905 insertions(+), 131 deletions(-)
```

--  
 1.7.6