Subject: Re: [PATCH v6 04/10] Account tcp memory as kernel memory
Posted by KAMEZAWA Hiroyuki on Mon, 28 Nov 2011 03:14:18 GMT
View Forum Message <> Reply to Message

Some nitpicks.

On Fri, 25 Nov 2011 15:38:10 -0200
Glauber Costa <glommer@parallels.com> wrote:

> Now that we account and control tcp memory buffers memory for pressure
> controlling purposes, display this information as part of the normal memcg
> files and other usages.
>

> +extern struct mem_cgroup *mem_cgroup_from_cont(struct cgroup *cont);
> +extern struct mem_cgroup *parent_mem_cgroup(struct mem_cgroup *mem);
> +
> static inline
> int mm_match_cgroup(const struct mm_struct *mm, const struct mem_cgroup *cgroup)
> {
> diff --git a/include/net/sock.h b/include/net/sock.h
> index d802761..da38de2 100644
> --- a/include/net/sock.h
> +++ b/include/net/sock.h
> @@ -65,6 +65,9 @@
> #include <net/dst.h>
> #include <net/checksum.h>
>
> +int sockets_populate(struct cgroup *cgrp, struct cgroup_subsys *ss);
> +void sockets_destroy(struct cgroup *cgrp, struct cgroup_subsys *ss);
> +
> /*

Hmm, what is this 'populate' function for ?
mem_cgroup_sockets_init() ?

>   * This structure really needs to be cleaned up.
>   * Most of it is for TCP, and not used by any of
> diff --git a/include/net/tcp_memcg.h b/include/net/tcp_memcg.h
> new file mode 100644
> index 0000000..5f5e158
> --- /dev/null
> +++ b/include/net/tcp_memcg.h
> @@ -0,0 +1,17 @@
> +#ifndef _TCP_MEMCG_H
> +#define _TCP_MEMCG_H

> +
> +struct tcp_memcontrol {
> + struct cg_proto cg_proto;
> + /* per-cgroup tcp memory pressure knobs */
> + struct res_counter tcp_memory_allocated;
> + struct percpu_counter tcp_sockets_allocated;
> + /* those two are read-mostly, leave them at the end */
> + long tcp_prot_mem[3];
> + int tcp_memory_pressure;
> +};
> +
> +struct cg_proto *tcp_proto_cgroup(struct mem_cgroup *memcg);
> +int tcp_init_cgroup(struct cgroup *cgrp, struct cgroup_subsys *ss);
> +void tcp_destroy_cgroup(struct cgroup *cgrp, struct cgroup_subsys *ss);
> +#endif /* _TCP_MEMCG_H */
> diff --git a/mm/memcontrol.c b/mm/memcontrol.c
> index 5f29194..2df5d3c 100644
> --- a/mm/memcontrol.c
> +++ b/mm/memcontrol.c
> @@ -49,6 +49,8 @@
> #include <linux/cpu.h>
> #include <linux/oom.h>
> #include "internal.h"
> +#include <net/sock.h>
> +#include <net/tcp_memcg.h>

ok, tcp_memcg.h ... some other men may like tcp_memcontrol.h..

>
> #include <asm/uaccess.h>
>
<snip>

> static int alloc_mem_cgroup_per_zone_info(struct mem_cgroup *mem, int node)
> @@ -4954,7 +4983,7 @@ static void mem_cgroup_put(struct mem_cgroup *mem)
> /*
>   * Returns the parent mem_cgroup in memcgroup hierarchy with hierarchy enabled.
>   */
> -static struct mem_cgroup *parent_mem_cgroup(struct mem_cgroup *mem)
> +struct mem_cgroup *parent_mem_cgroup(struct mem_cgroup *mem)
> {
>   if (!mem->res.parent)
>   return NULL;
> @@ -5037,6 +5066,7 @@ mem_cgroup_create(struct cgroup_subsys *ss, struct cgroup *cont)
>   res_counter_init(&mem->res, &parent->res);
>   res_counter_init(&mem->memsw, &parent->memsw);
>   res_counter_init(&mem->kmem, &parent->kmem);
> +

unnecessary blank line.

>   /*
>   * We increment refcnt of the parent to ensure that we can
>   * safely access it on res_counter_charge/uncharge.
> @@ -5053,6 +5083,7 @@ mem_cgroup_create(struct cgroup_subsys *ss, struct cgroup *cont)
>   mem->last_scanned_node = MAX_NUMNODES;
>   INIT_LIST_HEAD(&mem->oom_notify);
>
> +
ditto.

<snip>

Reviewed-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>