Subject: Re: [PATCH v7 04/10] tcp memory pressure controls
Posted by KAMEZAWA Hiroyuki on Wed, 30 Nov 2011 01:49:43 GMT
View Forum Message <> Reply to Message

On Tue, 29 Nov 2011 21:56:55 -0200
Glauber Costa <glommer@parallels.com> wrote:

> This patch introduces memory pressure controls for the tcp
> protocol. It uses the generic socket memory pressure code
> introduced in earlier patches, and fills in the
> necessary data in cg_proto struct.
>
>
> Signed-off-by: Glauber Costa <glommer@parallels.com>
> CC: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujtisu.com>
> CC: Eric W. Biederman <ebiederm@xmission.com>

some comments.

> ---
> Documentation/cgroups/memory.txt | 2 +
> include/linux/memcontrol.h | 3 ++
> include/net/sock.h | 2 +
> include/net/tcp_memcontrol.h | 17 +++++++++
> mm/memcontrol.c | 36 ++++++++++++++++++--
> net/core/sock.c | 42 ++++++++++++++++++++----
> net/ipv4/Makefile | 1 +
> net/ipv4/tcp_ipv4.c | 8 ++++-
> net/ipv4/tcp_memcontrol.c | 73 ++++++++++++++++++++++++++++++++++++++
> net/ipv6/tcp_ipv6.c | 4 ++
> 10 files changed, 181 insertions(+), 7 deletions(-)
> create mode 100644 include/net/tcp_memcontrol.h
> create mode 100644 net/ipv4/tcp_memcontrol.c
>
> diff --git a/Documentation/cgroups/memory.txt b/Documentation/cgroups/memory.txt
> index 3cf9d96..1e43da4 100644
> --- a/Documentation/cgroups/memory.txt
> +++ b/Documentation/cgroups/memory.txt
> @@ -299,6 +299,8 @@ and set kmem extension config option carefully.
> thresholds. The Memory Controller allows them to be controlled individually
> per cgroup, instead of globally.
>
> +* tcp memory pressure: sockets memory pressure for the tcp protocol.
> +
> 3. User Interface
>

> 0. Configuration
> diff --git a/include/linux/memcontrol.h b/include/linux/memcontrol.h
> index 60964c3..fa2482a 100644
> --- a/include/linux/memcontrol.h
> +++ b/include/linux/memcontrol.h
> @@ -85,6 +85,9 @@ extern struct mem_cgroup *try_get_mem_cgroup_from_page(struct page
*page);
> extern struct mem_cgroup *mem_cgroup_from_task(struct task_struct *p);
> extern struct mem_cgroup *try_get_mem_cgroup_from_mm(struct mm_struct *mm);
>
> +extern struct mem_cgroup *mem_cgroup_from_cont(struct cgroup *cont);
> +extern struct mem_cgroup *parent_mem_cgroup(struct mem_cgroup *mem);
> +

use 'memcg' please.

> -static struct mem_cgroup *mem_cgroup_from_cont(struct cgroup *cont)
> +struct mem_cgroup *mem_cgroup_from_cont(struct cgroup *cont)
> {
>   return container_of(cgroup_subsys_state(cont,
>   mem_cgroup_subsys_id), struct mem_cgroup,
> @@ -4717,14 +4732,27 @@ static int register_kmem_files(struct cgroup *cont, struct
cgroup_subsys *ss)
>
>   ret = cgroup_add_files(cont, ss, kmem_cgroup_files,
>   ARRAY_SIZE(kmem_cgroup_files));
> +
> + if (!ret)
> + ret = mem_cgroup_sockets_init(cont, ss);
>   return ret;
> };

You does initizalication here. The reason what I think is
1. 'proto_list' is not available at createion of root cgroup and
    you need to delay set up until mounting.

If so, please add comment or find another way.
This seems not very clean to me.

> +static DEFINE_RWLOCK(proto_list_lock);
> +static LIST_HEAD(proto_list);
> +
> +#ifdef CONFIG_CGROUP_MEM_RES_CTLR_KMEM
> +int mem_cgroup_sockets_init(struct cgroup *cgrp, struct cgroup_subsys *ss)
> +{

```
> + struct proto *proto;
> + int ret = 0;
> +
> + read_lock(&proto_list_lock);
> + list_for_each_entry(proto, &proto_list, node) {
> +  if (proto->init_cgroup)
> +   ret = proto->init_cgroup(cgrp, ss);
> +   if (ret)
> +    goto out;
> + }
```

seems indent is bad or {} is missing.

```
> +EXPORT_SYMBOL(memcg_tcp_enter_memory_pressure);
> +
> +int tcp_init_cgroup(struct cgroup *cgrp, struct cgroup_subsys *ss)
> +{
> + /*
> +  * The root cgroup does not use res_counters, but rather,
> +  * rely on the data already collected by the network
> +  * subsystem
> +  */
> + struct res_counter *res_parent = NULL;
> + struct cg_proto *cg_proto;
> + struct tcp_memcontrol *tcp;
> + struct mem_cgroup *memcg = mem_cgroup_from_cont(cgrp);
> + struct mem_cgroup *parent = parent_mem_cgroup(memcg);
> +
> + cg_proto = tcp_prot.proto_cgroup(memcg);
> + if (!cg_proto)
> +  return 0;
> +
> + tcp = tcp_from_cgproto(cg_proto);
> + cg_proto->parent = tcp_prot.proto_cgroup(parent);
> +
> + tcp->tcp_prot_mem[0] = sysctl_tcp_mem[0];
> + tcp->tcp_prot_mem[1] = sysctl_tcp_mem[1];
> + tcp->tcp_prot_mem[2] = sysctl_tcp_mem[2];
> + tcp->tcp_memory_pressure = 0;
```

Question:

Is this value will be updated when an admin chages sysctl ?

I guess, this value is set at system init script or some which may
happen later than mounting cgroup.
I don't like to write a guideline 'please set sysctl val before

mounting cgroup'

Thanks,
-Kame