Subject: Re: [PATCH v6 01/10] Basic kernel memory functionality for the Memory Controller
Posted by Michal Hocko on Wed, 14 Dec 2011 16:38:44 GMT
View Forum Message <> Reply to Message

Sorry for jumping in that late but I was busy recently...

On Fri 25-11-11 15:38:07, Glauber Costa wrote:
> This patch lays down the foundation for the kernel memory component
> of the Memory Controller.
>
> As of today, I am only laying down the following files:
>
> * memory.independent_kmem_limit

Maybe has been already discussed but the name is rather awkward and it would deserve more clarification. It is independent in the way that it doesn't add up to the standard (user) allocations or it enables/disables accounting?

> * memory.kmem.limit_in_bytes (currently ignored)

What happens if we reach the limit? Are all kernel allocations considered or only selected caches? How do I find out which are those?

AFAIU you have implemented it for network buffers at this stage but I guess that dentries will follow...

> * memory.kmem.usage_in_bytes (always zero)
>
> Signed-off-by: Glauber Costa <glommer@parallels.com>
> Reviewed-by: Kirill A. Shutemov <kirill@shutemov.name>
> CC: Paul Menage <paul@paulmenage.org>
> CC: Greg Thelen <gthelen@google.com>
> ---
> Documentation/cgroups/memory.txt | 36 ++++++++++++-
> init/Kconfig | 14 +++++
> mm/memcontrol.c | 107 ++++++++++++++++++++++++++++++++++++--
> 3 files changed, 150 insertions(+), 7 deletions(-)
>
> diff --git a/Documentation/cgroups/memory.txt b/Documentation/cgroups/memory.txt
> index 06eb6d9..bf00cd2 100644
> --- a/Documentation/cgroups/memory.txt
> +++ b/Documentation/cgroups/memory.txt
> @@ -44,8 +44,9 @@ Features:
>   - oom-killer disable knob and oom-notifier
>   - Root cgroup has no limit controls.
>

> - Kernel memory and Hugepages are not under control yet. We just manage
> - pages on LRU. To add more controls, we have to take care of performance.
> + Hugepages is not under control yet. We just manage pages on LRU. To add more

Hugepages are not
Anyway this sounds outdated as we track both THP and hugetlb, right?

> + controls, we have to take care of performance. Kernel memory support is work
> + in progress, and the current version provides basically functionality.

s/basically/basic/

>
> Brief summary of control files.
>
> @@ -56,8 +57,11 @@ Brief summary of control files.
>   (See 5.5 for details)
>   memory.memsw.usage_in_bytes # show current res_counter usage for memory+Swap
>   (See 5.5 for details)
> + memory.kmem.usage_in_bytes # show current res_counter usage for kmem only.
> +   (See 2.7 for details)
>   memory.limit_in_bytes   # set/show limit of memory usage
>   memory.memsw.limit_in_bytes # set/show limit of memory+Swap usage
> + memory.kmem.limit_in_bytes # if allowed, set/show limit of kernel memory
>   memory.failcnt   # show the number of memory usage hits limits
>   memory.memsw.failcnt   # show the number of memory+Swap hits limits
>   memory.max_usage_in_bytes # show max memory usage recorded
> @@ -72,6 +76,9 @@ Brief summary of control files.
>   memory.oom_control   # set/show oom controls.
>   memory.numa_stat   # show the number of memory usage per numa node
>
> + memory.independent_kmem_limit # select whether or not kernel memory limits are
> +   independent of user limits

It is not clear to me what happens in enabled/disabled cases. Let's say
they are not independent. Do they form a single limit or it toggles kmem
charging on/off.

> +
> 1. History
>
> The memory controller has a long history. A request for comments for the memory
> @@ -255,6 +262,31 @@ When oom event notifier is registered, event will be delivered.
>   per-zone-per-cgroup LRU (cgroup's private LRU) is just guarded by
>   zone->lru_lock, it has no lock of its own.
>
> +2.7 Kernel Memory Extension (CONFIG_CGROUP_MEM_RES_CTLR_KMEM)
> +

> + With the Kernel memory extension, the Memory Controller is able to limit
> +the amount of kernel memory used by the system.

Per kmem cache? Or what is the granularity?

> Kernel memory is fundamentally
> +different than user memory, since it can't be swapped out, which makes it
> +possible to DoS the system by consuming too much of this precious resource.
> +Kernel memory limits are not imposed for the root cgroup.
> +
> +Memory limits as specified by the standard Memory Controller may or may not
> +take kernel memory into consideration. This is achieved through the file
> +memory.independent_kmem_limit. A Value different than 0 will allow for kernel
> +memory to be controlled separately.
> +
> +When kernel memory limits are not independent, the limit values set in
> +memory.kmem files are ignored.

This suggests that the independent_kmem_limit is toggle to
enable/disable accounting. Wouldn't kmem_limit_enabled (0/1 or on/off)
be more obvious in that case?

Also a description what happens when the limit is reached (in both
cases) would be helpful.

[...]
> @@ -343,9 +352,14 @@ enum charge_type {
> };
>
> /* for encoding cft->private value on file */
> -#define _MEM   (0)
> -#define _MEMSWAP (1)
> -#define _OOM_TYPE (2)
> +
> +enum mem_type {
> + _MEM = 0,
> + _MEMSWAP,
> + _OOM_TYPE,
> + _KMEM,
> +};

Probably a separate (cleanup) patch?

> +
> #define MEMFILE_PRIVATE(x, val) (((x) << 16) | (val))
> #define MEMFILE_TYPE(val) (((val) >> 16) & 0xffff)
> #define MEMFILE_ATTR(val) ((val) & 0xffff)
> @@ -3838,10 +3852,17 @@ static inline u64 mem_cgroup_usage(struct mem_cgroup *mem,

bool swap)
> u64 val;
>
> if (!mem_cgroup_is_root(mem)) {
> + val = 0;
> +#ifdef CONFIG_CGROUP_MEM_RES_CTLR_KMEM
> + if (!mem->kmem_independent_accounting)
> +   val = res_counter_read_u64(&mem->kmem, RES_USAGE);
> +#endif

OK so this suggests that independent accounting really means
kmem+user_usage.

>   if (!swap)
> -   return res_counter_read_u64(&mem->res, RES_USAGE);
> +   val += res_counter_read_u64(&mem->res, RES_USAGE);
>   else
> -   return res_counter_read_u64(&mem->memsw, RES_USAGE);
> +   val += res_counter_read_u64(&mem->memsw, RES_USAGE);
> +
> + return val;
> }
>
> val = mem_cgroup_recursive_stat(mem, MEM_CGROUP_STAT_CACHE);
[...]
--
Michal Hocko
SUSE Labs
SUSE LINUX s.r.o.
Lihovarska 1060/12
190 00 Praha 9
Czech Republic