Subject: [RFC][PATCH] UBC: user resource beancounters
Posted by dev on Wed, 16 Aug 2006 15:23:58 GMT
View Forum Message <> Reply to Message

The following patch set presents base of
User Resource Beancounters (UBC).
UBC allows to account and control consumption
of kernel resources used by group of processes.

The full UBC patch set allows to control:
- kernel memory. All the kernel objects allocatable
  on user demand should be accounted and limited
  for DoS protection.
  E.g. page tables, task structs, vmas etc.

- virtual memory pages. UBC allows to
  limit a container to some amount of memory and
  introduces 2-level OOM killer taking into account
  container's consumption.
  pages shared between containers are correctly
  charged as fractions (tunable).

- network buffers. These includes TCP/IP rcv/snd
  buffers, dgram snd buffers, unix, netlinks and
  other buffers.

- minor resources accounted/limited by number:
  tasks, files, flocks, ptys, siginfo, pinned dcache
  mem, sockets, iptentries (for containers with
  virtualized networking)

As the first step we want to propose for discussion
the most complicated parts of resource management:
kernel memory and virtual memory.
The patch set to be sent provides core for UBC and
management of kernel memory only. Virtual memory
management will be sent in a couple of days.

The patches in these series are:
diff-ubc-kconfig.patch:
    Adds kernel/ub/Kconfig file with UBC options and
    includes it into arch Kconfigs

diff-ubc-core.patch:
    Contains core functionality and interfaces of UBC:
    find/create beancounter, initialization,
    charge/uncharge of resource, core objects' declarations.

diff-ubc-task.patch:
Contains code responsible for setting UB on task,
it's inheriting and setting host context in interrupts.

Task contains three beancounters:
1. exec_ub - current context. all resources are charged
to this beancounter.
2. task_ub - beancounter to which task_struct is charged
itself.
3. fork_sub - beancounter which is inherited by
task's children on fork

diff-ubc-syscalls.patch:
Patch adds system calls for UB management:
1. sys_getluid - get current UB id
2. sys_setluid - changes exec_ and fork_ UBs on current
3. sys_setublimit - set limits for resources consumtions

diff-ubc-kmem-core.patch:
Introduces UB_KMEMSIZE resource which accounts kernel
objects allocated by task's request.

Objects are accounted via struct page and slab objects.
For the latter ones each slab contains a set of pointers
corresponding object is charged to.

Allocation charge rules:
1. Pages - if allocation is performed with __GFP_UBC flag - page
is charged to current's exec_ub.
2. Slabs - kmem_cache may be created with SLAB_UBC flag - in this
case each allocation is charged. Caches used by kmalloc are
created with SLAB_UBC | SLAB_UBC_NOCHARGE flags. In this case
only __GFP_UBC allocations are charged.

diff-ubc-kmem-charge.patch:
Adds SLAB_UBC and __GFP_UBC flags in appropriate places
to cause charging/limiting of specified resources.

diff-ubc-proc.patch:
Adds two proc entries user_beancounters and user_beancounters_sub
allowing to see current state (usage/limits/fails for each UB).
Implemented via seq files.

Patch set is applicable to 2.6.18-rc4-mm1

Thanks,
Kirill
