Subject: Re: [PATCH 0/4] swapcgroup(v2)
Posted by KAMEZAWA Hiroyuki on Fri, 23 May 2008 05:19:37 GMT
View Forum Message <> Reply to Message

On Fri, 23 May 2008 10:21:04 +0530
Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

> KOSAKI Motohiro wrote:
> >> One option is to limit the virtual address space usage of the cgroup to ensure
> >> that swap usage of a cgroup will *not* exceed the specified limit. Along with a
> >> good swap controller, it should provide good control over the cgroup's memory usage.
> > >
> > > unfortunately, it doesn't works in real world.
> > > IMHO you said as old good age.
> > >
> > > because, Some JavaVM consume crazy large virtual address space.
> > > it often consume >10x than phycal memory consumption.
> > >
> >
> > Have you seen any real world example of this?
I have no objection to that virual-address-space limitation can work well on
well-controlled-system. But there are more complicated systems in chaos.

One example I know was that a team for the system tried to count all vm space
for setting vm.overcommit_memory to be proper value. The just found they can't
do it on a server with tens of applications after a month.

One of difficult problem is that a system administrator can't assume the total
size of virtual address space of proprietary applications/library.
An application designer can estimate "the virutal address usage of an application
is between XXM to XXXXM. but admin can't esitmate the total.

In above case, the most problematic user of virual adddress space was pthreads.
Default stack size of pthreads on ia64 was 10M bytes (--; And almost all application
doesn't answer how small they can set its stack size to. It's crazy to set this value
per applications. Then, "stack" of 2000 threads requires 20G bytes of virtual
address space on 12G system ;) They failed to use overcommit.

Thanks,
-Kame