Subject: Re: [ckrm-tech] [PATCH 4/7] UBC: syscalls (user interface)
Posted by Andrew Morton on Fri, 18 Aug 2006 18:18:16 GMT
View Forum Message <> Reply to Message

On Fri, 18 Aug 2006 10:59:06 -0700
Rohit Seth <rohitseth@google.com> wrote:

> On Fri, 2006-08-18 at 09:42 -0700, Andrew Morton wrote:
> > On Fri, 18 Aug 2006 07:45:48 -0700
> > Dave Hansen <haveblue@us.ibm.com> wrote:
> >
> > > On Fri, 2006-08-18 at 12:08 +0400, Andrey Savochkin wrote:
> > > >
> > > > A) Have separate memory management for each container,
> > > >   with separate buddy allocator, lru lists, page replacement mechanism.
> > > >   That implies a considerable overhead, and the main challenge there
> > > >   is sharing of pages between these separate memory managers.
> > >
> > > Hold on here for just a sec...
> > >
> > > It is quite possible to do memory management aimed at one container
> > > while that container's memory still participates in the main VM.
> > >
> > > There is overhead here, as the LRU scanning mechanisms get less
> > > efficient, but I'd rather pay a penalty at LRU scanning time than divide
> > > up the VM, or coarsely start failing allocations.
> > >
> >
> > I have this mad idea that you can divide a 128GB machine up into 256 fake
> > NUMA nodes, then you use each "node" as a 512MB unit of memory allocation.
> > So that 4.5GB job would be placed within an exclusive cpuset which has nine
> > "mems" (what are these called?) and voila: the job has a hard 4.5GB limit,
> > no kernel changes needed.
> >
> Sounds like an interesting idea. Will have to depend on something like
> memory hot-plug to get the things move around...
>

mmm, hadn't thought that far ahead. One could manually resize such a
contained with sys_move_pages(). Or just sit and wait: normal page
allocation and reclaim activity would eventually resize the job to the new
set of mems.