Subject: Re: BC: resource beancounters (v2)
Posted by Andrew Morton on Fri, 25 Aug 2006 17:50:47 GMT
View Forum Message <> Reply to Message

On Fri, 25 Aug 2006 20:30:26 +0400
Andrey Savochkin <saw@sw.ru> wrote:

> On Fri, Aug 25, 2006 at 07:30:03AM -0700, Andrew Morton wrote:
> >
> > D) Virtual scan of mm's in the over-limit container
> >
> > E) Modify existing physical scanner to be able to skip pages which
> >   belong to not-over-limit containers.
>
> I've actually tried (E), but it didn't work as I wished.
>
> It didn't handle well shared pages.
> Then, in my experiments such modified scanner was unable to regulate
> quality-of-service. When I ran 2 over-the-limit containers, they worked
> equally slow regardless of their limits and work set size.
> That is, I didn't observe a smooth transition "under limit, maximum
> performance" to "slightly over limit, a bit reduced performance" to
> "significantly over limit, poor performance". Neither did I see any fairness
> in how containers got penalized for exceeding their limits.
>
> My explanation of what I observed is that
> - since filesystem caches play a huge role in performance, page scanner will
>   be very limited in controlling container's performance if caches
>   stay shared between containers,
> - in the absence of decent disk I/O manager, stalls due to swapin/swapout
>   are more influenced by disk subsystem than by page scanner policy.
> So in fact modified page scanner provides control over memory usage only as
> "stay under limits or die", and doesn't show many advantages over (B) or (C).
> At the same time, skipping pages visibly penalizes "good citizens", not only
> in disk bandwidth but in CPU overhead as well.
>
> So I settled for (A)-(C) for now.
> But it certainly would be interesting to hear if someone else makes such
> experiments.
>

Makes sense. If one is looking for good machine partitioning then a shared
disk is obviously a great contention point. To address that we'd need to
be able to say "container A swaps to /dev/sda1 and container B swaps to
/dev/sdb1". But the swap system at present can't do that.