Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by Rohit Seth on Wed, 06 Sep 2006 00:17:58 GMT
View Forum Message <> Reply to Message

On Tue, 2006-09-05 at 10:46 -0700, Dave Hansen wrote:
> On Tue, 2006-09-05 at 19:02 +0400, Kirill Korotaev wrote:
> > > Core Resource Beancounters (BC) + kernel/user memory control.
> > >
> > > BC allows to account and control consumption
> > > of kernel resources used by group of processes.
> >
> > Hi Kirill,
> >
> > I've honestly lost track of these discussions along the way, so I hope
> > you don't mind summarizing a bit.
> >
> > Do these patches help with accounting for anything other than memory?
> > Will we need new user/kernel interfaces for cpu, i/o bandwidth, etc...?
> >
> > Have you given any thought to the possibility that a task might need to
> > move between accounting contexts? That has certainly been a
> > "requirement" pushed on to CKRM for a long time, and the need goes
> > something like this:
> >
> > 1. A system runs a web server, which services several virtual domains
> > 2. that web server receives a request for foo.com
> > 3. the web server switches into foo.com's accounting context
> > 4. the web server reads things from disk, allocates some memory, and
> > makes a database request.
> > 5. the database receives the request, and switches into foo.com's
> > accounting context, and charges foo.com for its resource use
> > etc...
> >

I'm wondering why not have different processes to serve different domains on the same physical server...particularly when they have different database to work on. Is the amount of memory that you save by having a single copy that much useful that you are even okay to serialize the whole operation (What would happen, while the request for foo.com is getting worked on, there is another request for foo_bar.com...does it need to wait for foo.com request to get done before it can be served).

> So, the goal is to run _one_ copy of an application on a system, but
> account for its resources in a much more fine-grained way than at the
> application level.
>

What is that fine grained way. If not process based then can it be associated with file system location?

-rohit