Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction
Posted by Paul Jackson on Wed, 20 Sep 2006 23:37:22 GMT
View Forum Message <> Reply to Message

Paul M., responding to Alan:
> > I'm also not clear how you handle shared pages correctly under the fake
> > node system, can you perhaps explain that further how this works for say
> > a single apache/php/glibc shared page set across 5000 containers each a
> > web site.
>
> If you can associate files with containers, you can have a "shared
> libraries" container that the libraries/binaries for apache/php/glibc
> are associated with - all pages from those files are then accounted to
> the shared container.

The way you "associate" a file with a cpuset is to have some task in that cpuset open that file and touch its pages -- where that task does so before any other would be user of the file. Then so long as those pages have any users or aren't reclaimed, they stay in memory or swap, free for anyone to reference (free so far as cpusets cares, which is not in the slightest.)

Such pre-touching of files is common occurrence on the HPC (High Perf Comp.) apps that run on the big honkin NUMA iron where cpusets were born. I'm guessing that someone hosting 5000 web servers would rather not deal with that particular hastle.

--

I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.925.600.0401

Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction
Posted by Paul Menage on Wed, 20 Sep 2006 23:53:30 GMT
View Forum Message <> Reply to Message

On 9/20/06, Paul Jackson <pj@sgi.com> wrote:
>
> The way you "associate" a file with a cpuset is to have some task in
> that cpuset open that file and touch its pages -- where that task does
> so before any other would be user of the file.

An alternative would be a way of binding files (or directory hierarchies) to a particular set of memory nodes. Then you wouldn't need to pre-fault the data. Extended attributes might be one way of doing it.

>
> Such pre-touching of files is common occurrence on the HPC (High Perf
> Comp.) apps that run on the big honkin NUMA iron where cpusets were
> born. I'm guessing that someone hosting 5000 web servers would rather
> not deal with that particular hastle.

I'm looking at it from the perspective of job control systems that
need to have a good idea what big datasets the jobs running under them
are touching/sharing.

Paul
