Subject: Re: [ckrm-tech] [PATCH 4/7] UBC: syscalls (user interface)
Posted by Paul Jackson on Fri, 18 Aug 2006 18:56:24 GMT
View Forum Message <> Reply to Message

Chandra wrote:
> In order to minimize this effect, resource controllers should be
> providing both minimum and maximum amount of resources available for a
> resource group.

No - not "should be." Rather "could also be."

The fair sharing model (such as in CKRM) that strives for maximum utilization of resources respecting priorities and min/max limits is (I suppose) quite useful for certain workloads and customers.

The hardwall NUMA placement model (such as in cpusets) that strives for maximum processor and memory isolation between jobs, preferring to leave allocated resources unused rather than trying to share them, is also quite useful for some. Customers with 256 thread, one or two day long run time, -very- tightly coupled huge OpenMP Fortran jobs that need to complete within a few percent of the same time, every runtime, demand it.

Don't presume that fair sharing -should- always be preferred to hardwall NUMA placement.

Just not so.

Besides -- what benefit would CKRM gain from Andrew's latest brainstorm? Doesn't CKRM already have whatever means it needs to define and share pools of memory?

--

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] [PATCH 4/7] UBC: syscalls (user interface)
Posted by Chris Friesen on Fri, 18 Aug 2006 19:16:55 GMT
View Forum Message <> Reply to Message

Paul Jackson wrote:

> The fair sharing model (such as in CKRM) that strives for maximum
> utilization of resources respecting priorities and min/max limits is
> (I suppose) quite useful for certain workloads and customers.
>

> The hardwall NUMA placement model (such as in cpusets) that strives
> for maximum processor and memory isolation between jobs, preferring
> to leave allocated resources unused rather than trying to share them,
> is also quite useful for some. Customers with 256 thread, one or
> two day long run time, -very- tightly coupled huge OpenMP Fortran
> jobs that need to complete within a few percent of the same time,
> every runtime, demand it.

Hypothetically, if you can guarantee that those threads get a specified
amount of time, but may possibly get *more* cpu time and thus finish
faster, what's the problem?

Chris

Subject: Re: [ckrm-tech] [PATCH 4/7] UBC: syscalls (user interface)
Posted by Chandra Seetharaman on Fri, 18 Aug 2006 19:48:24 GMT
View Forum Message <> Reply to Message

On Fri, 2006-08-18 at 11:56 -0700, Paul Jackson wrote:
> Chandra wrote:
> > In order to minimize this effect, resource controllers should be
> > providing both minimum and maximum amount of resources available for a
> > resource group.
>
> No - not "should be." Rather "could also be."
>
> The fair sharing model (such as in CKRM) that strives for maximum
> utilization of resources respecting priorities and min/max limits is
> (I suppose) quite useful for certain workloads and customers.
>
> The hardwall NUMA placement model (such as in cpusets) that strives
> for maximum processor and memory isolation between jobs, preferring
> to leave allocated resources unused rather than trying to share them,
> is also quite useful for some. Customers with 256 thread, one or
> two day long run time, -very- tightly coupled huge OpenMP Fortran
> jobs that need to complete within a few percent of the same time,
> every runtime, demand it.
>
> Don't presume that fair sharing -should- always be preferred to
> hardwall NUMA placement.

Not at all. During our prior discussions on the topic we have concluded
that resource management (CKRM\b\b\b\bResource Groups, UBC) and resource
isolation (cpuset) are totally orthogonal and have their own values from
the customer point of view. I still stand by that :).

>

> Just not so.
>
> Besides -- what benefit would CKRM gain from Andrew's latest
> brainstorm? Doesn't CKRM already have whatever means it needs to
> define and share pools of memory?

Yes, we do. That implementation is deemed complex, intrusive and it also increases maintenance burden. So, the object here (in this thread) is to get a simple memory controller that does resource management (not resource isolation).

>
--

---------- ----------
Chandra Seetharaman | Be careful what you choose....
- sekharan@us.ibm.com | .....you may get it.
---------- ----------