Subject: A consideration on memory controller.
Posted by KAMEZAWA Hiroyuki on Mon, 21 Jan 2008 08:07:59 GMT
View Forum Message <> Reply to Message

This mail is about memmory controller feature in my mind.
no patches, just a chitchat.

==
One of my purposes for contributing memory controller is to make applications
stable. That is, I'd like to reduce hiccups on the system and guarantee
applications some level of performance, throughput, latency, AMAP.

>From my experience in user support, one of causes of unexpected temporal
performance regression is File-I/O. iowait. In many case, customers say that
there are too many unused page caches, reduce it...
But delay is caused by caching is not correct.
That's just because there are too much dirty buffer and not enough free memory
for stable run, just the system is overloaded or parameter tuning was not
enough.

Shortage of free memory can delay system temporarlly. A big delay is caused by
write-back and a small reason is LRU-scan.
(A bit off-topic.
 For reducing amount of write-back, we can use dirty_ratio. But when we reduce
 dirty_ratio, syslogd hits it and delayed. This delays other applications
 which just doesn't issue I/O but call syslog(). Does anyone have good idea ?)

Kswapd reclaims freeable memory periodically for keeping free memory
to be some amount within min <-> low <-> high. But in emergency, an application
itself can reclaim memory by itself with calling try_to_free_pages().
This try_to_free_pages() scans LRU and reclaims some amount of memory and
delays an application which doesn't I/O just requesting memory.

If memory controller is used, we can limit maximum usage of memory per
applications. Workload can be isolated per cgroup.
This is good one progress. But maybe I need more features for my purpose....maybe.

One consideration is...
Now, memory controller can tamper LRU/relcaim handling but cannot do
free memory. For guaranteing amount of usable memory for an applicatons,
using VM is the best answer. But sometimes it can't be used.
I'm wondering whether we can add free-memory controller or not. It will
gather free memory for some cgroup with low <-> min <-> high + page-order setup
and work as buffer within cgroup <-> system workload.
But I'm not sure this idea is good or not ;)

BTW, I and YAMAMOTO-san is now considering followings for next series.

- back ground reclaim (Maybe it's better to wait for RvR's LRU set merge.)
- guarantee some amount of memory not to be reclaimed by global LRU.
- per cgroup swappiness.
- swap controller. (limit swap usage...maybe independet from memory
  controller.)

belows are no patch, no plan topics.
- limit amount of mlock.
- limit amount of hugepages.
- more parameters for page reclaim.
- balancing on NUMA (if we can find good algorythm...)
- dirty_ratio per cgroup.

- multi-level memory controller.

If you have feature-lists against memory controller, I'd like to see.

Note:
In last year, limit size of page-cache was posted but denied. It is said that
free memory is bad memory. Now, I never think anything just for limitig
page-cache will be accepted.

Thanks,
-Kame

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: A consideration on memory controller.
Posted by Balbir Singh on Mon, 21 Jan 2008 08:28:52 GMT
View Forum Message <> Reply to Message

* KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com> [2008-01-21 17:07:59]:

> This mail is about memmory controller feature in my mind.
> no patches, just a chitchat.
>
> ==
> One of my purposes for contributing memory controller is to make applications
> stable. That is, I'd like to reduce hiccups on the system and guarantee
> applications some level of performance, throughput, latency, AMAP.
>
> >From my experience in user support, one of causes of unexpected temporal

> performance regression is File-I/O. iowait. In many case, customers say that
> there are too many unused page caches, reduce it...
> But delay is caused by caching is not correct.
> That's just because there are too much dirty buffer and not enough free memory
> for stable run, just the system is overloaded or parameter tuning was not
> enough.
>
> Shortage of free memory can delay system temporarlly. A big delay is caused by
> write-back and a small reason is LRU-scan.
> (A bit off-topic.
> For reducing amount of write-back, we can use dirty_ratio. But when we reduce
> dirty_ratio, syslogd hits it and delayed. This delays other applications
> which just doesn't issue I/O but call syslog(). Does anyone have good idea ?)
>
> Kswapd reclaims freeable memory periodically for keeping free memory
> to be some amount within min <-> low <-> high. But in emergency, an application
> itself can reclaim memory by itself with calling try_to_free_pages().
> This try_to_free_pages() scans LRU and reclaims some amount of memory and
> delays an application which doesn't I/O just requesting memory.
>
> If memory controller is used, we can limit maximum usage of memory per
> applications. Workload can be isolated per cgroup.
> This is good one progress. But maybe I need more features for my purpose....maybe.
>
> One consideration is...
> Now, memory controller can tamper LRU/relcaim handling but cannot do
> free memory. For guaranteing amount of usable memory for an applicatons,
> using VM is the best answer.

This is a hard question? In the past it has been suggested that we use
hard limits to implement guarantees. Once we have the kernel memory
controller, guarantees might be easier to implement (we need account
for non-reclaimable resources)

But sometimes it can't be used.
> I'm wondering whether we can add free-memory controller or not. It will
> gather free memory for some cgroup with low <-> min <-> high + page-order setup
> and work as buffer within cgroup <-> system workload.
> But I'm not sure this idea is good or not ;)
>

I think it might be good to explore it more. The other idea is to
limit a soft-limit, such that memory is only reclaimed when there is
memory pressure.

>
> BTW, I and YAMAMOTO-san is now considering followings for next series.

>

Yes, we should consider some of these patches when the memory
controller makes it to mainline.

> - back ground reclaim (Maybe it's better to wait for RvR's LRU set merge.)
> - guarantee some amount of memory not to be reclaimed by global LRU.
> - per cgroup swappiness.
> - swap controller. (limit swap usage...maybe independet from memory
> controller.)
>
> belows are no patch, no plan topics.
> - limit amount of mlock.
> - limit amount of hugepages.
> - more parameters for page reclaim.
> - balancing on NUMA (if we can find good algorythm...)
> - dirty_ratio per cgroup.
>
> - multi-level memory controller.
>
We might also need to consider the following

1. Implementation of shares
2. Implementation of virtual memory limit
> If you have feature-lists against memory controller, I'd like to see.
>
>
> Note:
> In last year, limit size of page-cache was posted but denied. It is said that
> free memory is bad memory. Now, I never think anything just for limitig
> page-cache will be accepted.
>

This topic needs more discussion, we have some form of page-cache
control built into the memory controller.

> Thanks,
> -Kame
>

--
Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

_____
Containers mailing list
Containers@lists.linux-foundation.org

Subject: Re: A consideration on memory controller.
Posted by KAMEZAWA Hiroyuki on Mon, 21 Jan 2008 09:19:20 GMT
View Forum Message <> Reply to Message

On Mon, 21 Jan 2008 13:58:52 +0530
Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

> > If memory controller is used, we can limit maximum usage of memory per
> > applications. Workload can be isolated per cgroup.
> > This is good one progress. But maybe I need more features for my purpose....maybe.
> >
> > One consideration is...
> > Now, memory controller can tamper LRU/relcaim handling but cannot do
> > free memory. For guaranteing amount of usable memory for an applicatons,
> > using VM is the best answer.
>
> This is a hard question? In the past it has been suggested that we use
> hard limits to implement guarantees. Once we have the kernel memory
> controller, guarantees might be easier to implement (we need account
> for non-reclaimable resources)
>
yes, I'm looking forward to see the kernel memory controller.
But maybe guarantee amount of *immediately usable* memory (like mempool)
for cgroup is not the same issue as to guarantee free-cache for kernel
memory.

>
> But sometimes it can't be used.
> > I'm wondering whether we can add free-memory controller or not. It will
> > gather free memory for some cgroup with low <-> min <-> high + page-order setup
> > and work as buffer within cgroup <-> system workload.
> > But I'm not sure this idea is good or not ;)
> >
>
> I think it might be good to explore it more. The other idea is to
> limit a soft-limit, such that memory is only reclaimed when there is
> memory pressure.
>
thanks, I'll dig more.

> > - back ground reclaim (Maybe it's better to wait for RvR's LRU set merge.)
> > - guarantee some amount of memory not to be reclaimed by global LRU.
> > - per cgroup swappiness.
> > - swap controller. (limit swap usage...maybe independet from memory

> > controller.)
> >
> > belows are no patch, no plan topics.
> > - limit amount of mlock.
> > - limit amount of hugepages.
> > - more parameters for page reclaim.
> > - balancing on NUMA (if we can find good algorythm...)
> > - dirty_ratio per cgroup.
> >
> > - multi-level memory controller.
> >
> We might also need to consider the following
>
> 1. Implementation of shares
> 2. Implementation of virtual memory limit
limiting virtual memory like vm.overcommit_memory ?

> > If you have feature-lists against memory controller, I'd like to see.
> >
> >
> > Note:
> > In last year, limit size of page-cache was posted but denied. It is said that
> > free memory is bad memory. Now, I never think anything just for limitig
> > page-cache will be accepted.
> >
>
> This topic needs more discussion, we have some form of page-cache
> control built into the memory controller.
>
Hmm. ok. I'looking forward to see.

Regards,
-Kame

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: A consideration on memory controller.
Posted by Balbir Singh on Mon, 21 Jan 2008 09:50:36 GMT
View Forum Message <> Reply to Message

* KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com> [2008-01-21 18:19:20]:

> On Mon, 21 Jan 2008 13:58:52 +0530

> Balbir Singh <balbir@linux.vnet.ibm.com> wrote:
>
> > > If memory controller is used, we can limit maximum usage of memory per
> > > applications. Workload can be isolated per cgroup.
> > > This is good one progress. But maybe I need more features for my purpose....maybe.
> > >
> > > One consideration is...
> > > Now, memory controller can tamper LRU/relcaim handling but cannot do
> > > free memory. For guaranteing amount of usable memory for an applicatons,
> > > using VM is the best answer.
> >
> > This is a hard question? In the past it has been suggested that we use
> > hard limits to implement guarantees. Once we have the kernel memory
> > controller, guarantees might be easier to implement (we need account
> > for non-reclaimable resources)
> >
> yes, I'm looking forward to see the kernel memory controller.
> But maybe guarantee amount of *immediately usable* memory (like mempool)
> for cgroup is not the same issue as to guarantee free-cache for kernel
> memory.
>
>
> >
> > But sometimes it can't be used.
> > > I'm wondering whether we can add free-memory controller or not. It will
> > > gather free memory for some cgroup with low <-> min <-> high + page-order setup
> > > and work as buffer within cgroup <-> system workload.
> > > But I'm not sure this idea is good or not ;)
> > >
> >
> > I think it might be good to explore it more. The other idea is to
> > limit a soft-limit, such that memory is only reclaimed when there is
> > memory pressure.
> >
> thanks, I'll dig more.
>
> > > - back ground reclaim (Maybe it's better to wait for RvR's LRU set merge.)
> > > - guarantee some amount of memory not to be reclaimed by global LRU.
> > > - per cgroup swappiness.
> > > - swap controller. (limit swap usage...maybe independet from memory
> > > controller.)
> > >
> > > belows are no patch, no plan topics.
> > > - limit amount of mlock.
> > > - limit amount of hugepages.
> > > - more parameters for page reclaim.
> > > - balancing on NUMA (if we can find good algorythm...)
> > > - dirty_ratio per cgroup.

> > >
> > > - multi-level memory controller.
> > >
> > We might also need to consider the following
> >
> > 1. Implementation of shares
> > 2. Implementation of virtual memory limit
> limiting virtual memory like vm.overcommit_memory ?
>

Sort of, yes. The main idea is to limit paging rate and swap usage of
the control group.

>
> > > If you have feature-lists against memory controller, I'd like to see.
> > >
> > >
> > > Note:
> > > In last year, limit size of page-cache was posted but denied. It is said that
> > > free memory is bad memory. Now, I never think anything just for limitig
> > > page-cache will be accepted.
> > >
> >
> > This topic needs more discussion, we have some form of page-cache
> > control built into the memory controller.
> >
> Hmm. ok. I'looking forward to see.
>

Could you elaborate on what sort of page-cache control you need, is it
global page-cache control?

> Regards,
> -Kame
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

--
Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

_____
Containers mailing list
Containers@lists.linux-foundation.org

Subject: Re: A consideration on memory controller.
Posted by KAMEZAWA Hiroyuki on Mon, 21 Jan 2008 10:22:23 GMT
View Forum Message <> Reply to Message

On Mon, 21 Jan 2008 15:20:36 +0530
Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

> > > This topic needs more discussion, we have some form of page-cache
> > > control built into the memory controller.
> > >
> > Hmm. ok. I'looking forward to see.
> >
>
> Could you elaborate on what sort of page-cache control you need, is it
> global page-cache control?
>
What I mentioned to in my mail was global page cache.

But, my purpose is isolate workload between applications and guarantee
stable performance/latency and memory controller's limiting page usage
feature will do half of work.
So, per-cgroup page-cache control is (maybe) good enough for making applications
stable by keeping room for immediate use of anon and by avoiding unnecessary swapout.

but above can be archieved by
 - reserve/guarantee amount of free memory by some technique, bacground
   kthread, throttling,
 - do good design of swappiness.
 - etc...

So, the word "limiting page cache" itself is not important. Above will
limit amount of page-cache as a result.
Ways to reach my goal is not one, maybe. I should find the best one :)

Thanks,
-Kame

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers