Subject: Re: [PATCH 1/2] rcfs core patch
Posted by dev on Tue, 13 Mar 2007 08:28:06 GMT
View Forum Message <> Reply to Message

```
>>>well, Linux-VServer is "working", "secure", "flexible"
>>>_and_ non-intrusive ... it is quite natural that less
>>>won't work for me ... and regarding patches, there
>>>will be a 2.2 release soon, with all the patches ...
>
>
>>ok. please check your dcache and slab accounting then
>>(analyzed according to patch-2.6.20.1-vs2.3.0.11.diff):
>
>
> development branch, good choice for new features
> and code which is currently tested ...
```

you know better than I that stable branch doesn't differ much,
especially in securiy (because it lacks these controls at all).

BTW, killing arbitrary task in case of RSS limit hit
doesn't look acceptable resource management approach, does it?

```
>>Both are full of races and problems. Some of them:
>>1. Slabs allocated from interrupt context are charged to
>>   current context.
>>   So charged values contain arbitrary mess, since during
>>   interrupts context can be arbitrary.
>
>
>>2. Due to (1) I guess you do not make any limiting of slabs.
>>   So there are number of ways how to consume a lot of kernel
>>   memory from inside container and
>>   OOM killer will kill arbitrary tasks in case of
>>   memory-shortage after that.
>>   Don't think it is secure... real DoS.
>
>
>>3. Dcache accounting simply doesn't work, since
>>   charges/uncharges are done on current context (sic!!!),
>>   which is arbitrary. i.e. lookup can be done in VE context,
>>   while dcache shrink can be done from another context.
>>   So the whole problem with dcache DoS is not solved at
>>   all, it is just hard to trigger.
>
>
>>4. Dcache accounting is racy, since your checks look like:
>>   if (atomic_read(de->d_count))
>>     charge();
```

>>   which obviously races with other dput()'s/lookups.
>
>
>>5. Dcache accounting can be hit if someone does `find /`
>>   inside container.
>>   After that it is impossible to open something new,
>>   since all the dentries for directories in dcache will
>>   have d_count > 0 (due it's children).
>>   It is a BUG.
>
>
>>6. Counters can be non-zero on container stop due to all
>>   of the above.
>
>
> looks like for the the first time you are actually
> looking at the code, or at least providing feedback
> and/or suggestions for improvements (well, not many
> of them, but hey, nobody is perfect :)
It's a pity, but it took me only 5 minutes of looking into the code,
so "not perfect" is a wrong word here, sorry.

>>There are more and more points which arise when such a
>>non-intrusive accounting is concerned.
>
>
> never claimed that Linux-VServer code is perfect,
> (the Linux accounting isn't perfect either in many
> ways) and Linux-VServer is constantly improving
> (see my other email) ... but IIRC, we are _not_
> discussing Linux-VServer code at all, we are talking
> about a superior solution, which combines the best
> of both worlds ...
Forget about Vserver and OpenVZ. It is not a war.
We are looking for something working, new and robust.
I'm just trying you to show that non-intrusive and pretty small
accounting/limiting code like in Vserver
simply doesn't work. The problem of resource controls is much more complicated.
So non-intrusiveness is a very weird argument from you (and the only).

>>I'm really suprised, that you don't see them
>>or try to behave as you don't see them :/
>
>
> all I'm saying is that there is no point in achieving
> perfect accounting and limits (and everything else)
> when all you get is Xen performance and resource usage
then please elaborate on what you mean by

perfect and non-perfect accounting and limits?
I would be happy to sent a patch with a "non-perfect"
accounting if it really works correct and good and suits all the people needs.

BTW, Xen overhead comes mostly from different things (not resource management) -
inability to share data effectively, emulation overhead etc.

>>And, please, believe me, I would not suggest so much
>>complicated patches If everything was so easy and I
>>had no reasons simply to accept vserver code.
>
>
> no, you are suggesting those patches, because that
> is what your company came up with after being confronted
> with the task (of creating OS-Level virtualization) and
> the arising problems ... so it definitely _is_ a
> solution to those problems, but not necessarily the
> best and definitely not the only one :)
You judge so because you want to.
Have you had some time to compare UBC patches from OVZ
and those sent to LKML (container + RSS)?
You would notice too litle in common.
Patches in LKML has non-OVZ interfaces, no shared pages accounting,
RSS accounting which is not used in OVZ at all.
So do you see any similarities except for stupid and simple
controls like numtask/numfile?

Thanks,
Kirill