Subject: Re: [PATCH] Send quota messages via netlink
Posted by Jan Kara on Wed, 29 Aug 2007 12:26:47 GMT
View Forum Message <> Reply to Message

On Tue 28-08-07 21:13:35, Andrew Morton wrote:
> On Tue, 28 Aug 2007 16:13:18 +0200 Jan Kara <jack@suse.cz> wrote:
>
> > Hello,
> >
> > I'm sending rediffed patch implementing sending of quota messages via netlink
> > interface (some rationale in patch description). I've already posted it to
> > LKML some time ago and there were no objections, so I guess it's fine to put
> > it to -mm. Andrew, would you be so kind? Thanks.
> > Userspace deamon reading the messages from the kernel and sending them to
> > dbus and/or user console is also written (it's part of quota-tools). The
> > only remaining problem is there are a few changes needed to libnl needed for
> > the userspace daemon. They were basically acked by the maintainer but it
> > seems he has not merged the patches yet. So this will take a bit more time.
> >
>
> So it's a new kernel->userspace interface.
>
> But we have no description of the interface :(
Oops, forgotten about it. I'll write one. Do we have some standard place
where to document such interfaces? I could create some file in
Documentation/filesystems/ but that seems a bit superfluous...

> > +/* Send warning to userspace about user which exceeded quota */
> > +static void send_warning(const struct dquot *dquot, const char warntype)
> > +{
> > + static unsigned long seq;
> > + struct sk_buff *skb;
> > + void *msg_head;
> > + int ret;
> > +
> > + skb = genlmsg_new(QUOTA_NL_MSG_SIZE, GFP_NOFS);
> > + if (!skb) {
> > + printk(KERN_ERR
> > + "VFS: Not enough memory to send quota warning.\n");
> > + return;
> > + }
> > + msg_head = genlmsg_put(skb, 0, seq++, &quota_genl_family, 0,
QUOTA_NL_C_WARNING);
> > + if (!msg_head) {
> > + printk(KERN_ERR
> > + "VFS: Cannot store netlink header in quota warning.\n");
> > + goto err_out;
> > + }

> > + ret = nla_put_u32(skb, QUOTA_NL_A_QTYPE, dquot->dq_type);
> > + if (ret)
> > + goto attr_err_out;
> > + ret = nla_put_u64(skb, QUOTA_NL_A_EXCESS_ID, dquot->dq_id);
> > + if (ret)
> > + goto attr_err_out;
> > + ret = nla_put_u32(skb, QUOTA_NL_A_WARNING, warntype);
> > + if (ret)
> > + goto attr_err_out;
> > + ret = nla_put_u32(skb, QUOTA_NL_A_DEV_MAJOR,
> > + MAJOR(dquot->dq_sb->s_dev));
> > + if (ret)
> > + goto attr_err_out;
> > + ret = nla_put_u32(skb, QUOTA_NL_A_DEV_MINOR,
> > + MINOR(dquot->dq_sb->s_dev));
> > + if (ret)
> > + goto attr_err_out;
> > + ret = nla_put_u64(skb, QUOTA_NL_A_CAUSED_ID, current->user->uid);
> > + if (ret)
> > + goto attr_err_out;
> > + genlmsg_end(skb, msg_head);
> > +
> > + ret = genlmsg_multicast(skb, 0, quota_genl_family.id, GFP_NOFS);
> > + if (ret < 0 && ret != -ESRCH)
> > + printk(KERN_ERR
> > +   "VFS: Failed to send notification message: %d\n", ret);
> > + return;
> > +attr_err_out:
> > + printk(KERN_ERR "VFS: Failed to compose quota message: %d\n", ret);
> > +err_out:
> > + kfree_skb(skb);
> > +}
> > +#endif
>
> This is it. Normally netlink payloads are represented as a struct. How
> come this one is built-by-hand?
  I use "generic netlink", which is in fact a layer built on top of
netlink. As far as I've read it's documentation, creating a message
argument by argument is the preferred way. As David writes, this way
we can add new arguments without worries about backward compatibility,
alignment issues or such things.

> It doesn't appear to be versioned. Should it be?
  We don't need a version for future additions. Also each attribute sent
has its identifier (e.g. QUOTA_NL_A_CAUSED_ID) and userspace checks these
identifiers and unknown attributes are ignored. But in case we would like
to remove some attribute, versioning would be probably useful so that
userspace won't break silently... So I'll add it.

> Does it have (or need) reserved-set-to-zero space for expansion? Again,
> hard to tell..
  No, we don't need it as I wrote above.

> I guess it's OK to send a major and minor out of the kernel like this.
> What's it for? To represent a filesytem? I wonder if there's a more
> modern and useful way of describing the fs. Path to mountpoint or
> something?
  I also find major/minor pair a bit old-fashioned. But the identifying it
by a mountpoint is problematic - quota does not care about namespaces and
such and so it works with superblocks. It's not trivial to get a mountpoint
from a superblock (and generally it's frown upon, isn't it?). Also if a
filesystem is mounted on several places, we have to pick one (OK, userspace
has to do this choice anyway when displaying the message but still...).

> I suspect the namespace virtualisation guys would be interested in a new
> interface which is sending current->user->uid up to userspace. uids are
> per-namespace now. What are the implications? (cc's added)
  I know there's something going on in this area but I don't know any
details. If somebody has some advice what should be passed into userspace
so that user/group can be idenitified, it is welcome.

> Is it worth adding a comment explaining why GFP_NOFS is used here?
  Probably yes. Added.

  Thanks for all your comments.

        Honza
--
Jan Kara <jack@suse.cz>
SuSE CR Labs