Subject: Re: [PATCH 06/11] sysfs: Implement sysfs tagged directory support.
Posted by Tejun Heo on Mon, 23 Jun 2008 02:05:21 GMT
View Forum Message <> Reply to Message

Hello,

> Index: linux-mm/fs/sysfs/file.c
> ===
> --- linux-mm.orig/fs/sysfs/file.c
> +++ linux-mm/fs/sysfs/file.c
> @@ -460,9 +460,9 @@ void sysfs_notify(struct kobject *k, cha
> mutex_lock(&sysfs_mutex);
>
> if (sd && dir)
> - sd = sysfs_find_dirent(sd, dir);
> + sd = sysfs_find_dirent(sd, NULL, dir);
> if (sd && attr)
> - sd = sysfs_find_dirent(sd, attr);
> + sd = sysfs_find_dirent(sd, NULL, attr);
> if (sd) {
> struct sysfs_open_dirent *od;
>

As only directories can be tagged, I suppose handling tags explicitly
isn't necessary here, right? Can we please add a comment explaning
that?

> Index: linux-mm/fs/sysfs/inode.c
> ===
> --- linux-mm.orig/fs/sysfs/inode.c
> +++ linux-mm/fs/sysfs/inode.c
> @@ -217,17 +217,20 @@ struct inode * sysfs_get_inode(struct sy
> return inode;
> }
>
> -int sysfs_hash_and_remove(struct sysfs_dirent *dir_sd, const char *name)
> +int sysfs_hash_and_remove(struct kobject *kobj, struct sysfs_dirent *dir_sd,
> + const char *name)
> {
> struct sysfs_addrm_cxt acxt;
> struct sysfs_dirent *sd;
> + const void *tag;
>
> if (!dir_sd)
> return -ENOENT;
>
> sysfs_addrm_start(&acxt, dir_sd);
> + tag = sysfs_removal_tag(kobj, dir_sd);

>
> - sd = sysfs_find_dirent(dir_sd, name);
> + sd = sysfs_find_dirent(dir_sd, tag, name);
>   if (sd)
>   sysfs_remove_one(&acxt, sd);

Taking both @kobj and @dir_sd is ugly but it isn't your fault. I'll
clean things up later.

> Index: linux-mm/include/linux/sysfs.h
> ===
> --- linux-mm.orig/include/linux/sysfs.h
> +++ linux-mm/include/linux/sysfs.h
> @@ -80,6 +80,14 @@ struct sysfs_ops {
>   ssize_t (*store)(struct kobject *,struct attribute *,const char *, size_t);
> };
>
> +struct sysfs_tag_info {
> +};
> +
> +struct sysfs_tagged_dir_operations {
> + const void *(*sb_tag)(struct sysfs_tag_info *info);
> + const void *(*kobject_tag)(struct kobject *kobj);
> +};

As before, I can't bring myself to like this interface. Is computing
tags dynamically really necessary? Can't we do the followings?

tag = sysfs_allocate_tag(s);
sysfs_enable_tag(kobj (or sd), tag);
sysfs_sb_show_tag(sb, tag);

Where tags are allocated using ida and each sb has bitmap of enabled
tags so that sysfs ops can simply use something like the following to
test whether it's enabled.

```
bool sysfs_tag_enabled(sb, tag)
{
 return sysfs_info(sb)->tag_map & (1 << tag);
}
```

Tags which can change dynamically seems too confusing to me and it
makes things difficult to verify as it's unclear how those tags are
gonna to change.

Thanks.

--

tejun

_____

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