

On 08/23, Kirill Korotaev wrote:

```
>
> +struct beancounter *beancounter_findcreate(uid_t uid, int mask)
> +{
> + struct beancounter *new_bc, *bc;
> + unsigned long flags;
> + struct hlist_head *slot;
> + struct hlist_node *pos;
> +
> + slot = &bc_hash[bc_hash_fun(uid)];
> + new_bc = NULL;
> +
> +retry:
> + spin_lock_irqsave(&bc_hash_lock, flags);
> + hlist_for_each_entry (bc, pos, slot, hash)
> + if (bc->bc_id == uid)
> + break;
> +
> + if (pos != NULL) {
> + get_beancounter(bc);
> + spin_unlock_irqrestore(&bc_hash_lock, flags);
> +
> + if (new_bc != NULL)
> + kmem_cache_free(bc_cachep, new_bc);
> + return bc;
> + }
> +
> + if (!(mask & BC_ALLOC))
> + goto out_unlock;
```

Very minor nit: it is not clear why we are doing this check under bc_hash_lock. I'd suggest to do

```
if (!(mask & BC_ALLOC))
    goto out;
```

after unlock(bc_hash_lock) and kill out_unlock label.

```
> + if (new_bc != NULL)
> + goto out_install;
> +
> + spin_unlock_irqrestore(&bc_hash_lock, flags);
> +
> + new_bc = kmem_cache_alloc(bc_cachep,
```

```
> + mask & BC_ALLOC_ATOMIC ? GFP_ATOMIC : GFP_KERNEL);  
> + if (new_bc == NULL)  
> + goto out;  
> +  
> + memcpy(new_bc, &default_beancounter, sizeof(*new_bc));
```

May be it is just me, but I need a couple of seconds to parse this 'memcpy'.
How about

```
*new_bc = default_beancounter;
```

?

Oleg.
