
Subject: Re: [PATCH 2/6] BC: beancounters core (API)
Posted by [Oleg Nesterov](#) on Thu, 24 Aug 2006 11:09:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 08/23, Andrew Morton wrote:

```
>
> On Wed, 23 Aug 2006 15:03:07 +0400
> Kirill Korotaev <dev@sw.ru> wrote:
>
> > +void __put_beancounter(struct beancounter *bc)
> > +{
> > + unsigned long flags;
> > +
> > + /* equivalent to atomic_dec_and_lock_irqsave() */
> > + local_irq_save(flags);
> > + if (likely(!atomic_dec_and_lock(&bc->bc_refcount, &bc_hash_lock))) {
> > + local_irq_restore(flags);
> > + if (unlikely(atomic_read(&bc->bc_refcount) < 0))
> > + printk(KERN_ERR "BC: Bad refcount: bc=%p, "
> > + "luid=%d, ref=%d\n",
> > + bc, bc->bc_id,
> > + atomic_read(&bc->bc_refcount));
> > + return;
> > + }
> > +
> > + BUG_ON(bc == &init_bc);
> > + verify_held(bc);
> > + hlist_del(&bc->hash);
> > + spin_unlock_irqrestore(&bc_hash_lock, flags);
> > + kmem_cache_free(bc_cachep, bc);
> > +}
>
> I wonder if it's safe and worthwhile to optimise away the local_irq_save():
```

Suppose ->bc_refcount == 1

```
> if (atomic_dec_and_test(&bc->bc_refcount)) {
```

Yes, preempted or blocks on spin_lock() below.

another cpu locks bc_hash_lock, does get_beancounter() (beancounter_findcreate), then does put_beancounter(), and frees it.

```
> spin_lock_irqsave(&bc_hash_lock, flags);
> if (atomic_read(&bc->bc_refcount) == 0) {
```

Yes,

> free it
>

Already freed.

Oleg.
