
Subject: Re: [-mm PATCH 1/9] Memory controller resource counters (v6)

Posted by [Alexey Dobriyan](#) on Mon, 20 Aug 2007 08:20:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Fri, Aug 17, 2007 at 02:12:38PM +0530, Balbir Singh wrote:

```
> --- /dev/null
> +++ linux-2.6.23-rc2-mm2-balbir/kernel/res_counter.c
> +void res_counter_init(struct res_counter *counter)
> +{
> + spin_lock_init(&counter->lock);
> + counter->limit = (unsigned long)LONG_MAX;
```

why cast?

```
> +int res_counter_charge_locked(struct res_counter *counter, unsigned long val)
> +{
> + if (counter->usage > (counter->limit - val)) {
```

() aren't needed.

```
> + counter->failcnt++;
> + return -ENOMEM;
> + }
> +
> + counter->usage += val;
> + return 0;
> +}
> +
> +int res_counter_charge(struct res_counter *counter, unsigned long val)
> +{
> + int ret;
> + unsigned long flags;
> +
> + spin_lock_irqsave(&counter->lock, flags);
> + ret = res_counter_charge_locked(counter, val);
> + spin_unlock_irqrestore(&counter->lock, flags);
> + return ret;
> +}
> +
> +void res_counter_uncharge_locked(struct res_counter *counter, unsigned long val)
> +{
> + if (WARN_ON(counter->usage < val))
> + val = counter->usage;
```

explicit if and WARN_ON(1) is clearer. I should send a patch banning such type of usage soon.

```
> +
```

```

> + counter->usage -= val;
> +}
> +
> +void res_counter_uncharge(struct res_counter *counter, unsigned long val)
> +{
> + unsigned long flags;
> +
> + spin_lock_irqsave(&counter->lock, flags);
> + res_counter_uncharge_locked(counter, val);
> + spin_unlock_irqrestore(&counter->lock, flags);
> +}

```

```

> +ssize_t res_counter_write(struct res_counter *counter, int member,
> + const char __user *userbuf, size_t nbytes, loff_t *pos)
> +{
> + int ret;
> + char *buf, *end;
> + unsigned long tmp, *val;
> +
> + buf = kmalloc(nbytes + 1, GFP_KERNEL);

```

please, switch to fixed buffer, allocating memory depending on size told by userspace will be later. Ditto for other proc writing functions.

```

> + ret = -ENOMEM;
> + if (buf == NULL)
> + goto out;
> +
> + buf[nbytes] = '\0';
> + ret = -EFAULT;
> + if (copy_from_user(buf, userbuf, nbytes))
> + goto out_free;
> +
> + ret = -EINVAL;
> + tmp = simple_strtoul(buf, &end, 10);
> + if (*end != '\0')
> + goto out_free;
> +
> + val = res_counter_member(counter, member);
> + *val = tmp;
> + ret = nbytes;
> +out_free:
> + kfree(buf);
> +out:
> + return ret;
> +}

```

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