
Subject: Re: [RFC][only for review] memory controller bacground reclaim [3/5]
high/low watermark support in
Posted by [KAMEZAWA Hiroyuki](#) on Thu, 29 Nov 2007 01:18:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 28 Nov 2007 14:12:53 +0300

Pavel Emelyanov <xemul@openvz.org> wrote:

```
> > /*
> > @@ -73,6 +88,8 @@
> > RES_USAGE,
> > RES_LIMIT,
> > RES_FAILCNT,
> > + RES_HIGH_WATERMARK,
> > + RES_LOW_WATERMARK,
> >
> > I'd prefer some shorter names. Like RES_HWMARK and RES_LWMARK.
> >
> > Hmm, ok.
```

```
> > counter->usage += val;
> > +
> > + if (counter->usage > counter->high_watermark) {
> > + counter->watermark_state = RES_WATERMARK_ABOVE_HIGH;
> > + return 0;
> > + }
> >
> > "else" would look much better here :)
> > I agree and will fix. thanks.
```

```
> >
> > + if (counter->usage > counter->low_watermark)
> > + counter->watermark_state = RES_WATERMARK_ABOVE_LOW;
> > +
> > return 0;
> > }
> >
> > @@ -47,6 +59,13 @@
> > val = counter->usage;
> >
> > counter->usage -= val;
> > +
> > + if (counter->usage < counter->low_watermark) {
> > + counter->watermark_state = RES_WATERMARK_BELOW_LOW;
> > + return;
> > + }
> >
> > and here
```

```

>
here, too.

>> + if (counter->usage < counter->high_watermark)
>> + counter->watermark_state = RES_WATERMARK_ABOVE_LOW;
>> }
>>
>> void res_counter_uncharge(struct res_counter *counter, unsigned long val)
>> @@ -69,6 +88,10 @@
>>     return &counter->limit;
>>     case RES_FAILCNT:
>>         return &counter->failcnt;
>> + case RES_HIGH_WATERMARK:
>> +     return &counter->high_watermark;
>> + case RES_LOW_WATERMARK:
>> +     return &counter->low_watermark;
>> };
>>
>> BUG();
>> @@ -117,7 +140,7 @@
>> {
>>     int ret;
>>     char *buf, *end;
>> - unsigned long long flags, tmp, *val;
>> + unsigned long long flags, tmp, *val, limit, low, high;
>>
>>     buf = kmalloc(nbytes + 1, GFP_KERNEL);
>>     ret = -ENOMEM;
>> @@ -141,6 +164,26 @@
>>     goto out_free;
>> }
>> spin_lock_irqsave(&counter->lock, flags);
>> + /*
>> +  * High/Low watermark should be changed automatically AMAP.
>> +  */
>> + switch (member) {
>> + case RES_HIGH_WATERMARK:
>> +     limit = res_counter_get(counter, RES_LIMIT);
>> +     if (tmp > limit)
>> +         goto out_free;
>> +     low = res_counter_get(counter, RES_LOW_WATERMARK);
>> +     if (tmp <= low)
>> +         goto out_free;
>> +     break;
>> + case RES_LOW_WATERMARK:
>> +     high = res_counter_get(counter, RES_HIGH_WATERMARK);
>> +     if (tmp >= high)
>> +         goto out_free;

```

```
> > + break;
>
> Why there's no checks for limit? Smth like
>
> case RES_LIMIT:
>   high = res_counter_get(counter, RES_HIGH_WATERMARK);
>   if (tmp < high)
>     goto out_free;
>
Ok, I'll drop automatic adjustment patch and add check for LIMIT.
```

Thanks,
-Kame

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