Subject: Re: [RFC][PATCH 5/7] UBC: kernel memory accounting (core)
Posted by dev on Thu, 17 Aug 2006 13:29:43 GMT
View Forum Message <> Reply to Message

Dave Hansen wrote:
> On Wed, 2006-08-16 at 19:40 +0400, Kirill Korotaev wrote:
>

```
>>--- ./include/linux/mm.h.kmemcore	2006-08-16 19:10:38.000000000
>>+0400
>>+++ ./include/linux/mm.h	2006-08-16 19:10:51.000000000 +0400
>>@@ -274,8 +274,14 @@ struct page {
>> 	unsigned int gfp_mask;
>> 	unsigned long trace[8];
>> #endif
>>+#ifdef CONFIG_USER_RESOURCE
>>+	union {
>>+		struct user_beancounter *page_ub;
>>+	} bc;
>>+#endif
>> };
```

>
>
> Is everybody OK with adding this accounting to the 'struct page'? Is
> there any kind of noticeable performance penalty for this? I thought
> that we had this aligned pretty well on cacheline boundaries.
When I discussed this with Hugh Dickins on summit we agreed
that +4 bytes on page struct for kernel using accounting
are ok and almost unavoidable.

it can be stored not on the struct page, but in this
case you need to introduce some kind of hash to lookup ub
quickly from page, which is slower for accounting-enabled kernels.

> How many things actually use this? Can we have the slab ubcs without
> the struct page pointer?
slab doesn't use this pointer on the page.
It is used for pages allocated by buddy
alocator implicitly (e.g. LDT pages, page tables, ...).

Kirill