Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)
Posted by Dave Hansen on Fri, 18 Aug 2006 19:32:03 GMT
View Forum Message <> Reply to Message

On Fri, 2006-08-18 at 15:59 +0100, Alan Cox wrote:
> Ar Gwe, 2006-08-18 am 12:52 +0400, ysgrifennodd Kirill Korotaev:
> > > hmm, not sure why it is simpler.
> > because introducing additonal lookups/hashes etc. is harder and
> > adds another source for possible mistakes.
> > we can always optimize it out if people insist (by cost of slower accounting).
>
> It ought to be cheap. Given each set of page structs is an array its a
> simple subtract and divide (or with care and people try to pack them
> nicely for cache lines - shift) to get to the parallel accounting array.

I wish page structs were just a simple array. ;)

It will just be a bit more code, but we'll need this for the two other memory models: sparsemem and discontigmem. For discontig, we'll just need pointers in the pg_data_ts and, for sparsemem, we'll likely need another pointer in the 'struct mem_section'.

This will effectively double the memory we need for sparsemem (because we only use one pointer per SECTION_SIZE bytes of memory) but, that should be just fine.

Is there ever any need to go from the accounting structure *back* to the page? I guess that might be the hard part with keeping parallel arrays, if we even need it.

The reverse lookups might introduce a bit more pain with sparsemem and discontig because, right now, we use bits in page->flags to help us go find the containing node or the correct mem_section for the page.

-- Dave