Subject: Re: [PATCH 8/8] Per-container pages reclamation
Posted by xemul on Tue, 24 Apr 2007 10:29:58 GMT
View Forum Message <> Reply to Message

Balbir Singh wrote:
> Pavel Emelianov wrote:
>> Implement try_to_free_pages_in_container() to free the
>> pages in container that has run out of memory.
>>
>> The scan_control->isolate_pages() function isolates the
>> container pages only.
>>
>
> Pavel,
>
> I've just started playing around with these patches, I preferred
> the approach of v1. Please see below
>
>> +static unsigned long isolate_container_pages(unsigned long nr_to_scan,
>> +		struct list_head *src, struct list_head *dst,
>> +		unsigned long *scanned, struct zone *zone)
>> +{
>> +	unsigned long nr_taken = 0;
>> +	struct page *page;
>> +	struct page_container *pc;
>> +	unsigned long scan;
>> +	LIST_HEAD(pc_list);
>> +
>> +	for (scan = 0; scan < nr_to_scan && !list_empty(src); scan++) {
>> +		pc = list_entry(src->prev, struct page_container, list);
>> +		page = pc->page;
>> +		if (page_zone(page) != zone)
>> +			continue;
>
> shrink_zone() will walk all pages looking for pages belonging to this

No. shrink_zone() will walk container pages looking for pages in the desired zone.
Scann through the full zone is done on global memory shortage.

> container and this slows down the reclaim quite a bit. Although we've
> reused code, we've ended up walking the entire list of the zone to
> find pages belonging to a particular container, this was the same
> problem I had with my RSS controller patches.
>
>> +
>> +		list_move(&pc->list, &pc_list);
>> +
>

>