

Re: [PATCH 1/5] Swap Migration V5: LRU operations

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-12/msg08577.html>

- *From:* Marcelo Tosatti <marcelo.tosatti@xxxxxxxxxxxxx>
 - *Date:* Sat, 31 Dec 2005 05:32:31 -0200
-

On Tue, Nov 15, 2005 at 10:46:05AM -0800, Andrew Morton wrote:

> Christoph Lameter <clameter@xxxxxxxxxxxxx> wrote:

>>

>> On Tue, 15 Nov 2005, Andrew Morton wrote:

>>

>>> But `lru_add_drain_per_cpu()` will be called from interrupt context: the IPI
>>> handler.

>>

>> Ahh.. thought you meant the `lru_add_drain` run on the local processor.

>>

>>> I'm asking whether it is safe for the IPI handler to reenale interrupts on
>>> all architectures. It might be so, but I don't recall ever having seen it
>>> discussed, nor have I seen code which does it.

>>

>> `smp_call_function` is also used by the slab allocator to drain the
>> pages. All the spinlocks in there and those of the page allocator (called
>> for freeing pages) use `spin_lock_irqsave`. Why is this not used for
>> `lru_add_drain()` and friends?

>

> It's a microoptimisation – `lru_add_drain()` is always called with local irqs
> enabled, so no need for `irqsave`.

>

> I don't think `spin_lock_irqsave()` is notably more expensive than
> `spin_lock_irq()` – the cost is in the irq disabling and in the atomic
> operation.

Pardon me, but `spin_lock_irqsave()` needs to write data to the stack, which is likely to be cache-cold, so you have to fault the cacheline in from slow memory.

And thats much slower than the atomic operation, isnt it?

–

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>

- Prev by Date: **Re: bad pmd filemap.c, oops; 2.4.30 and 2.4.32**
- Next by Date: **Re: [PATCH] strict VM overcommit accounting for 2.4.32/2.4.33-pre1**
- Previous by thread: **PROBLEM: Linux ATAPI CDROM ->FIX: SAMSUNG CD-ROM SC-140**
- Next by thread: **Integer types**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**