

# Re: missing madvise functionality

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-04/msg02234.html>

---

- *From:* Nick Piggin <[nickpiggin@xxxxxxxxxxxxx](mailto:nickpiggin@xxxxxxxxxxxxx)>
  - *Date:* Fri, 06 Apr 2007 12:59:42 +1000
- 

Ulrich Drepper wrote:

Nick Piggin wrote:

Cool. According to my thinking, `madvise(MADV_DONTNEED)` even in today's kernels using `down_write(mmap_sem)` for `MADV_DONTNEED` is better than `mmap/mprotect`, which have more fundamental locking requirements, more overhead and no benefits (except debugging, I suppose).

It's a tiny bit faster, see

<http://people.redhat.com/drepper/dontneed.png>

I just ran it once so the graph is not smooth. This is on a UP dual core machine. Maybe tomorrow I'll turn on the big 4p machine.

Hmm, I saw an improvement, but that was just on a raw syscall test with a single page chunk. Real-world use I guess will get progressively less dramatic as other overheads start being introduced.

Multi-thread performance probably won't get a whole lot better (it does eliminate 1 `down_write(mmap_sem)`, but one remains) until you use my `madvise` patch.

I would have to see dramatically different results on the big machine to make me change the `libc` code. The reason is that there is a big drawback.

So far, when we allocate a new arena, we allocate address space with `PROT_NONE` and only when we need memory the protection is changed to `PROT_READ|PROT_WRITE`. This is the advantage of catching wild pointer accesses.

Re: missing madvise functionality

Sure, yes. And I guess you'd always want to keep that options around as a debugging aid.

--

SUSE Labs, Novell Inc.

-

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

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

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