Subject: Re: controlling mmap()'d vs read/write() pages
Posted by Dave Hansen on Fri, 23 Mar 2007 16:41:12 GMT
View Forum Message <> Reply to Message

On Fri, 2007-03-23 at 04:12 -0600, Eric W. Biederman wrote:
> Would any of them work on a system on which every filesystem was on
> ramfs, and there was no swap? If not then they are not memory attacks
> but I/O attacks.

I truly understand your point here. But, I don't think this thought exercise is really helpful here. In a pure sense, nothing is keeping an unmapped page cache file in memory, other than the user's prayers. But, please don't discount their prayers, it's what they want!

I seem to remember a quote attributed to Alan Cox around OLS time last year, something about any memory controller being able to be fair, fast, and accurate. Please pick any two, but only two. Alan, did I get close?

To me, one of the keys of Linux's "global optimizations" is being able to use any memory globally for its most effective purpose, globally (please ignore highmem :). Let's say I have a 1GB container on a machine that is at least 100% committed. I mmap() a 1GB file and touch the entire thing (I never touch it again). I then go open another 1GB file and r/w to it until the end of time. I'm at or below my RSS limit, but that 1GB of RAM could surely be better used for the second file. How do we do this if we only account for a user's RSS? Does this fit into Alan's unfair bucket? ;)

Also, in a practical sense, it is also a *LOT* easier to describe to a customer that they're getting 1GB of RAM than >=20GB/hr of bandwidth from the disk.

-- Dave

P.S. Do we have an quotas on ramfs? If we have an ramfs filesystems, what keeps the containerized users from just filling up RAM?