

Page Queue Lock Contention Page Replacement Algorithm

Max Laier

<max.laier@isilon.com>

<milaier@freebsd.org>

Problems

(1) The VM page queue lock is a major source of contention

(2) The simple LRU page replacement algorithm for the INACTIVE queue falls down easily (scan pushout, ...)

WIP!!! Research Project!!!

- Very much in flux
- Currently akin to CLOCK-Pro (NetBSD, Linux), but adapted to our vm locking model and with more bells and whistles (which may be special to our environment)
- Promising first results for medium memory sizes (4-12G)
- No benefit/performance hit for large memory sizes (>40G), due to contention

Breaking up the lock

- One queue per CPU + 1
- Still a single page daemon
- Scan queues round-robin
- Use marker pages in order to keep LRU order (within $\frac{1}{2}$ the average scan count)
- Lightweight, simple, and cuts contention more than 10 fold***

- Contention on the vm page queue lock, that is
- Contention moves to the free lock, and the vm object lock instead (and some others which are private to us, but similar locks exist in UFS and other backing stores)
- Additional work to reduce these underway ...

Status

- Work in progress ... really ... profiling benchmarks are running as we speak!
- Plans to share patches with the community as soon as we have them ... (working for us)
- Some of this will be rather specialized, but we will probably make most of it available for discussion

Questions?

Max Laier

<max.laier@isilon.com>

<mlaier@freebsd.org>