

# Percpu rw semaphores

Percpu rw semaphores is a new read-write semaphore design that is optimized for locking for reading.

The problem with traditional read-write semaphores is that when multiple cores take the lock for reading, the cache line containing the semaphore is bouncing between L1 caches of the cores, causing performance degradation.

Locking for reading is very fast, it uses RCU and it avoids any atomic instruction in the lock and unlock path. On the other hand, locking for writing is very expensive, it calls `synchronize_rcu()` that can take hundreds of milliseconds.

The lock is declared with "struct `percpu_rw_semaphore`" type. The lock is initialized `percpu_init_rwsem`, it returns 0 on success and -ENOMEM on allocation failure. The lock must be freed with `percpu_free_rwsem` to avoid memory leak.

The lock is locked for read with `percpu_down_read`, `percpu_up_read` and for write with `percpu_down_write`, `percpu_up_write`.

The idea of using RCU for optimized rw-lock was introduced by Eric Dumazet <[eric.dumazet@gmail.com](mailto:eric.dumazet@gmail.com)>. The code was written by Mikulas Patocka <[mpatocka@redhat.com](mailto:mpatocka@redhat.com)>