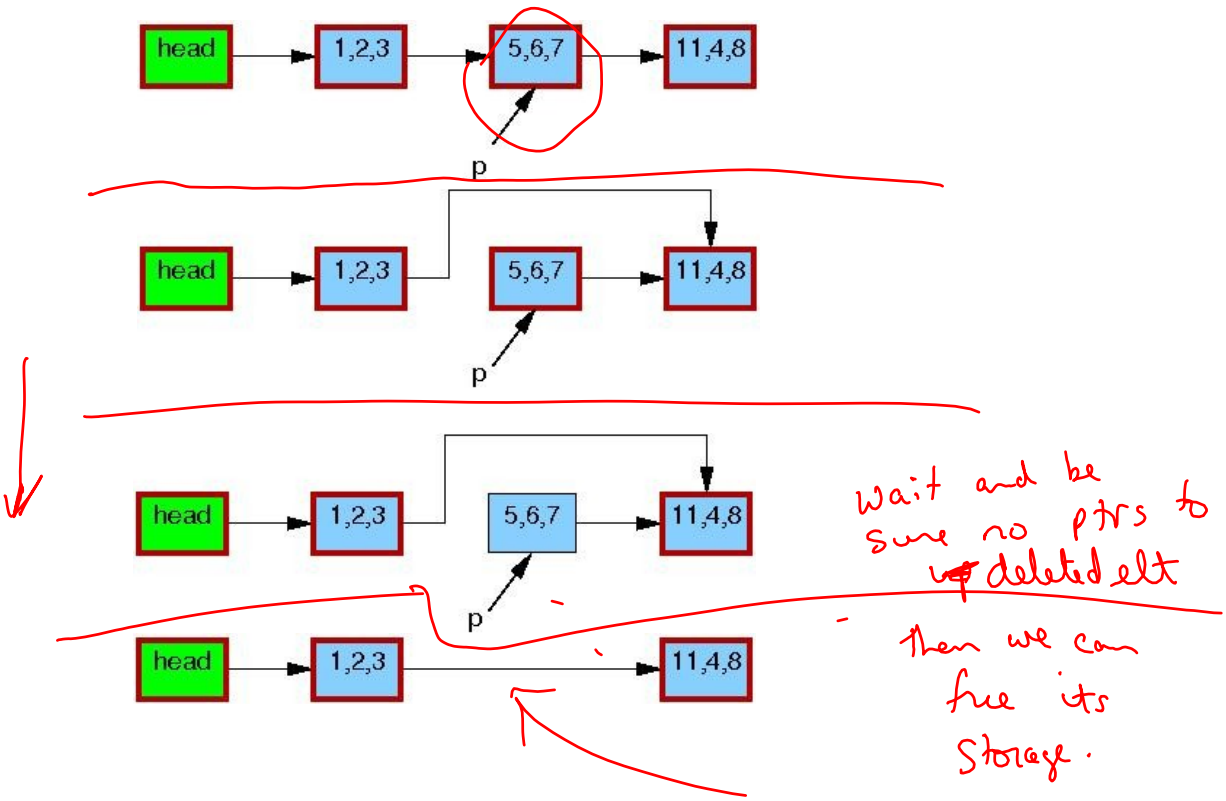
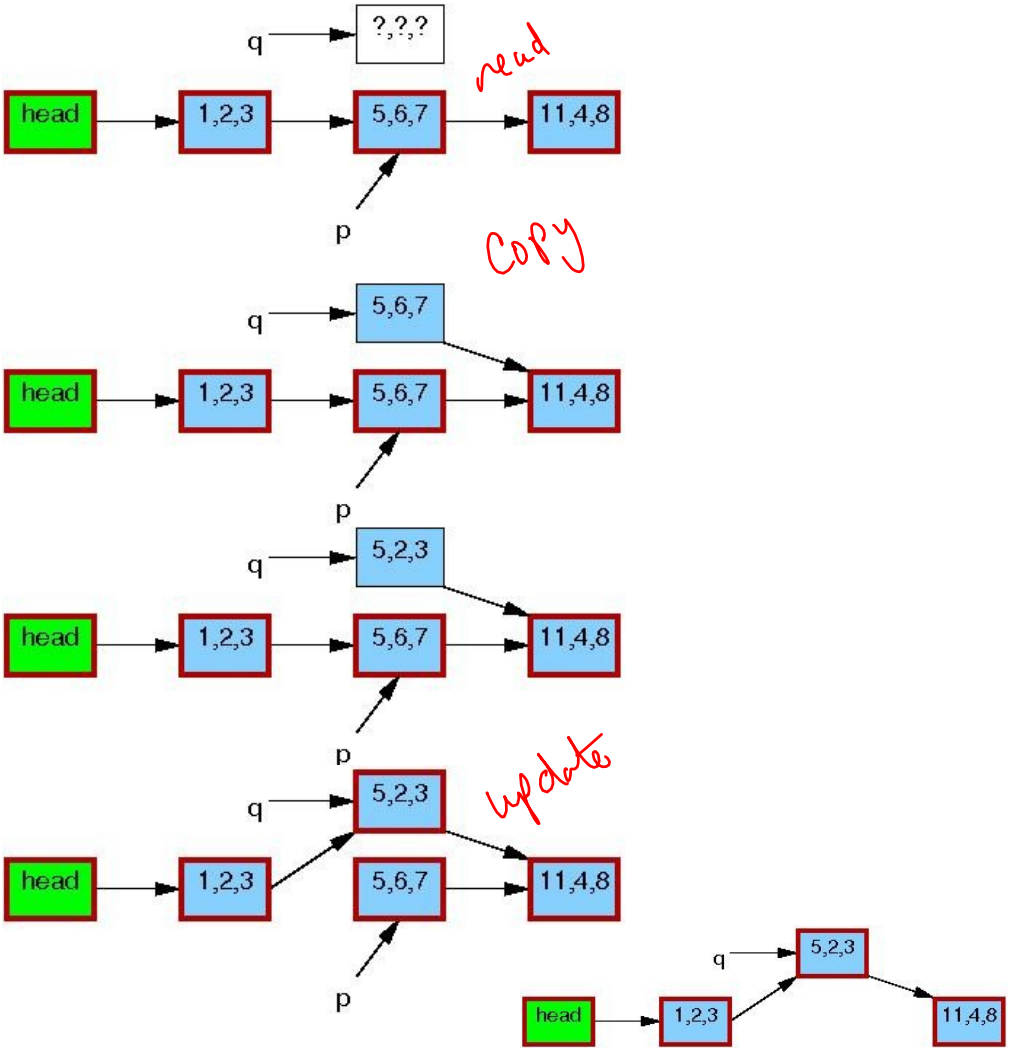
Read-Copy-Update (RCU)

CptS 483/580
April 21, 2010

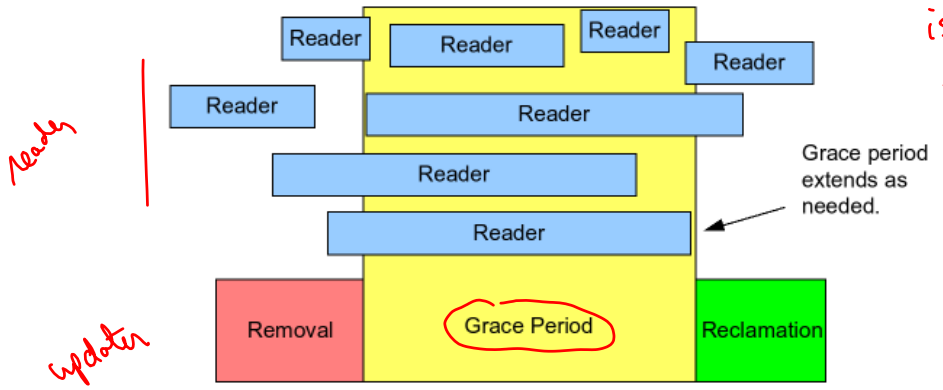


Deleting a List Element

List element update



Read-copy-update: high-performance synchronization for read-mostly workloads.
Paul McKenney
Linux Weekly News (3 articles from late 2007-2008)
See http://www.rdrop.com/users/paulmck/RCU/
Figures here are from those three articles.



any reader that started before the grace period is finished by the end of the grace period.
At end of GP, it is safe to reclaim storage of the removed element

synchronize_rcu
when does GP end?

Note: update visibility is an issue that the Linux RCU interface also addresses with higher-level pointer read/update instructions that incorporate memory barriers

Primitives for RCU

```
rcu_read_lock();  /* read -side */
rcu_read_unlock();
synchronize_rcu(); /* write side */
```

synchronize_rcu() - implements the grace period after which removed elements may be freed.

Another approach is with a **call-back** procedure call_rcu(f,p) which calls f(p) when system determines that the grace period has expired.

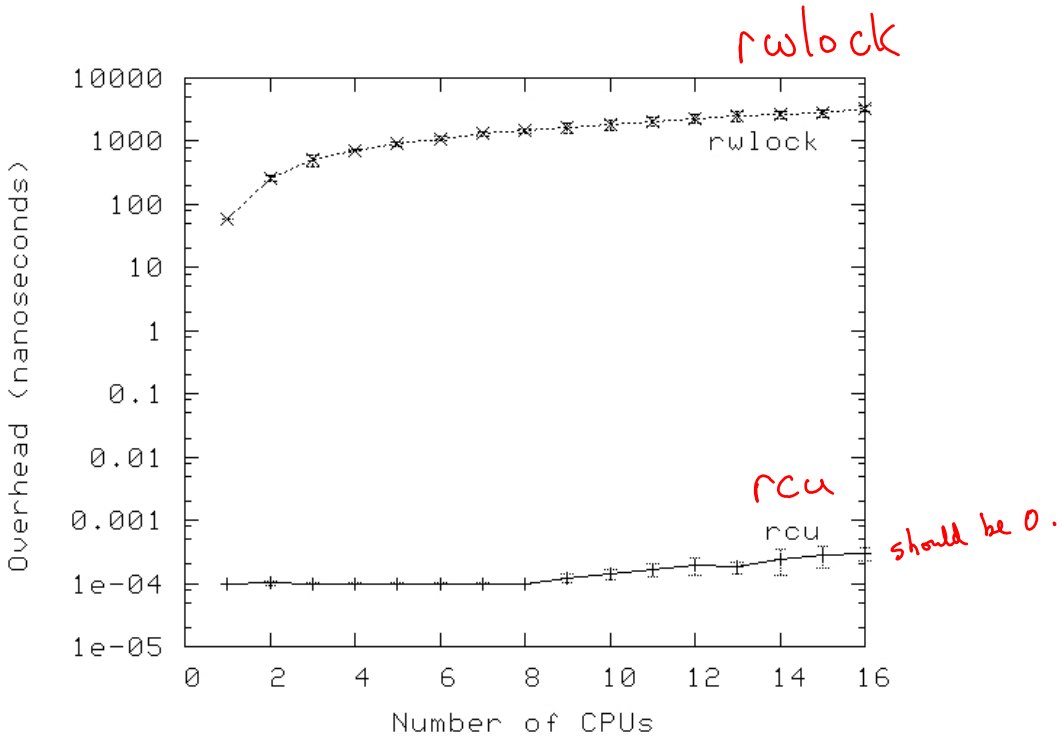
Implementing synchronize_rcu() – various mechanisms: if pre-emption is disallowed in lock...unlock sections sufficient to detect that all contexts have run once.

With garbage-collection: these techniques are (nearly) free – have to worry about update visibility.

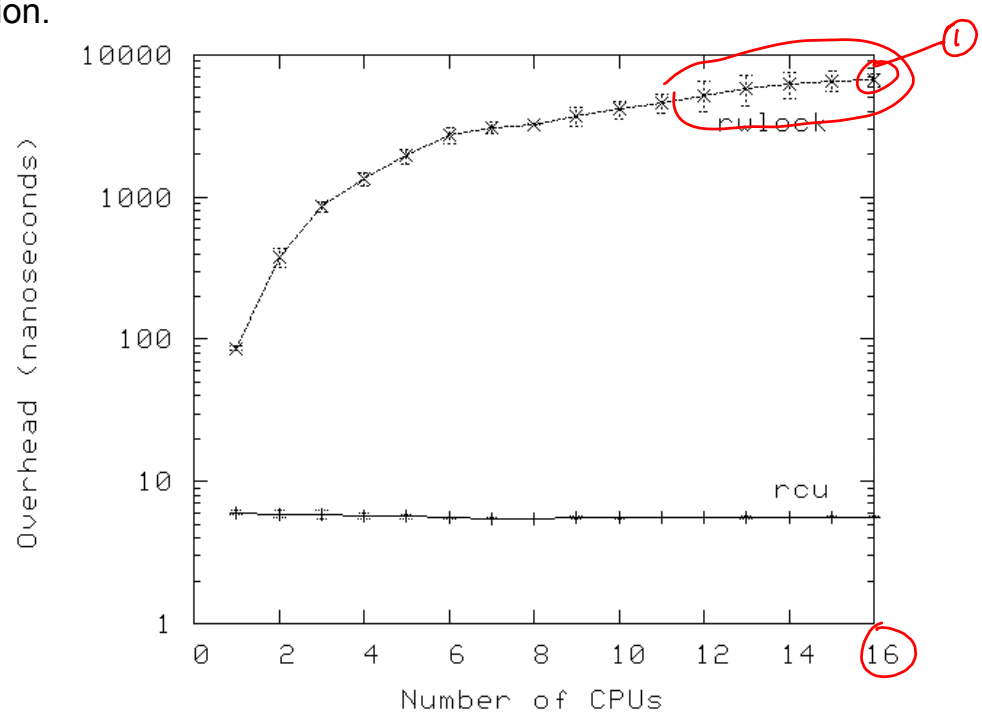
Note: locks (or TM, or …) are still needed to protect against concurrent updates!

Micro-benchmarks

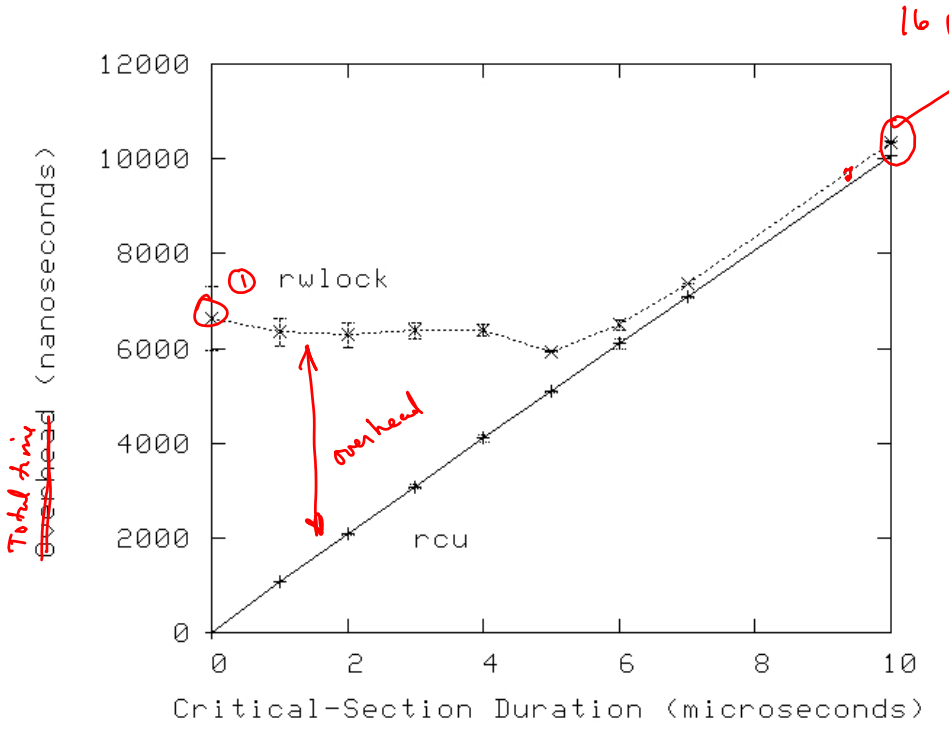
Overhead – non-preemptive kernel: 0



Overhead – preemptive kernel: non-zero cost because read_lock_rcu() has to turn off pre-emption.



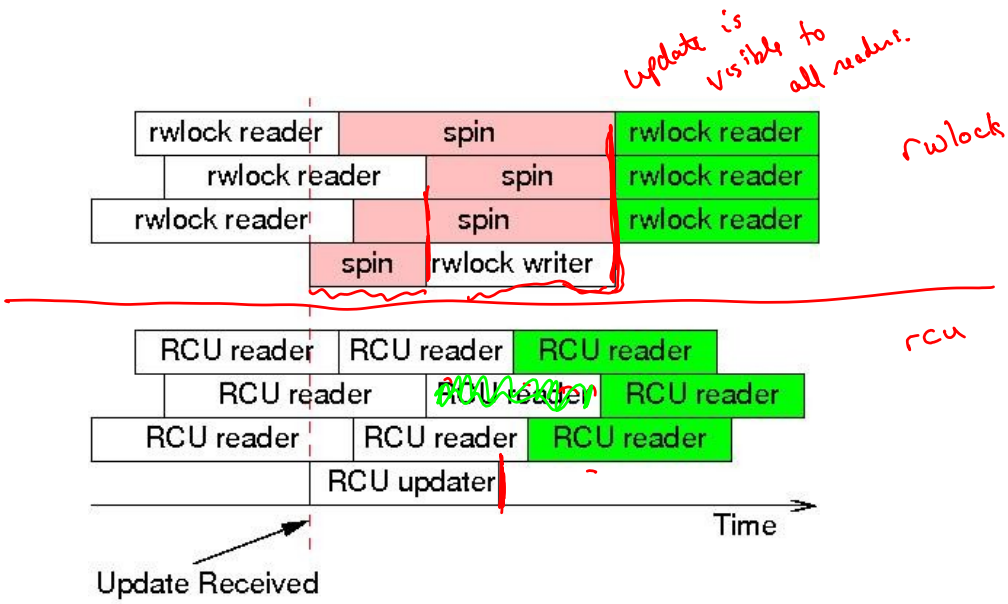
Overhead: Pre-emptive kernel – 16 CPUs, increasing CS size

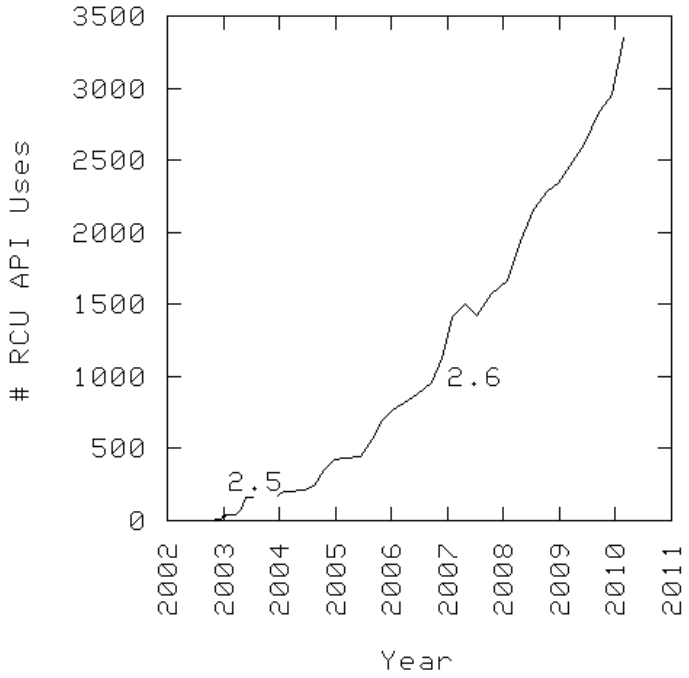


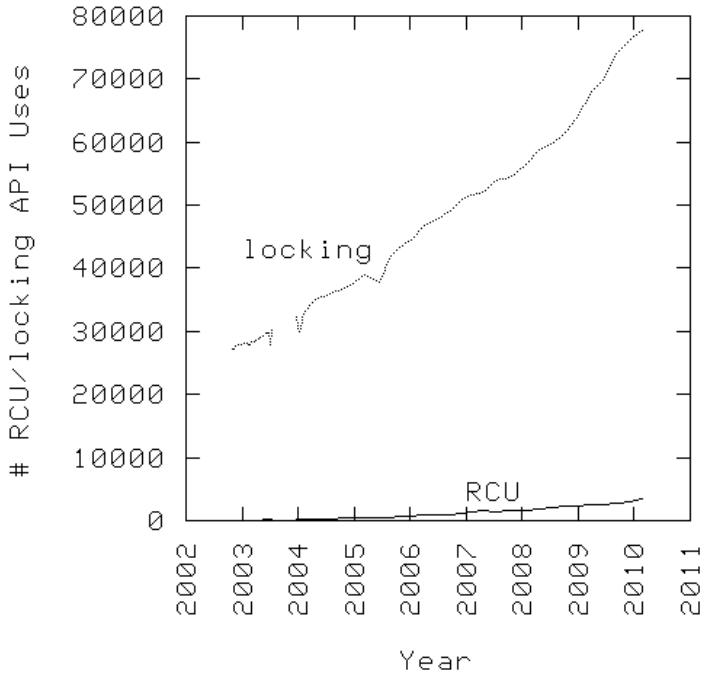
Why does the overhead and variability of rwlock decrease with increasing CS duration?

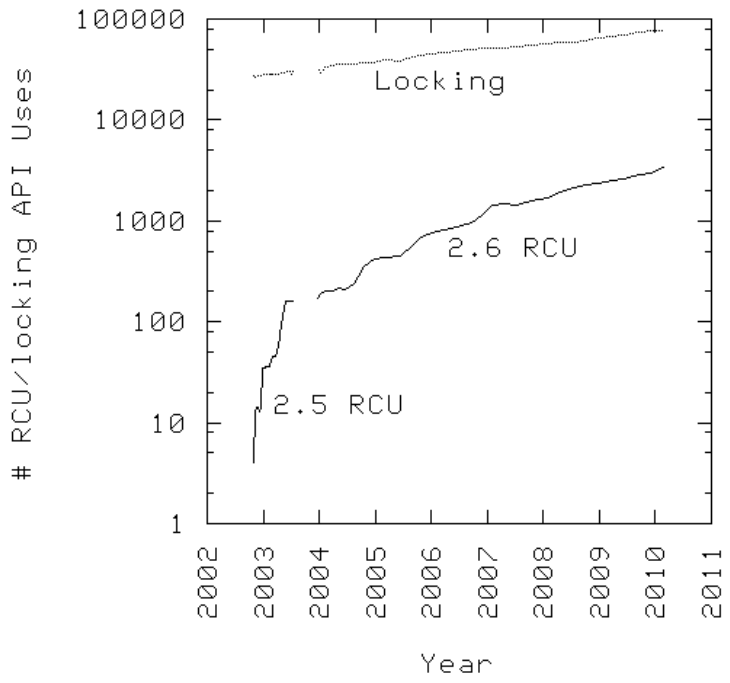
Macro-effect

RW-lock vs RCU




3500
3000
2500
2000
1500
1000
500
0
RCU API Uses
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Year
2.5
2.6


80000
70000
60000
50000
40000
30000
20000
10000
0
RCU/locking API Uses
locking
RCU
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Year


100000
10000
1000
100
10
1
RCU/locking API Uses
Locking
2.6 RCU
2.5 RCU
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Year