

Summary

The paper presents a specialized locking mechanism called Read-Copy Update which provides concurrent reads even when there are updates with simplified locking protocols, improved performance and increased scalability.

Main Points

The paper claims that traditional locking mechanism used by operating system kernels were complex in design and did not take the full advantages of event driven nature of these systems, which caused loss in performance and poor concurrency.

The Authors presents Read-Copy Update mechanism which reduces the locking overhead of reader's to zero since they are not protecting their data access instead writer's are ensuring that their updates will allow the reader's to safely access their data any time. Removing locking mechanism from reader's significantly improves the reading performance.

The RCU mechanism takes advantage of event-driven nature by allowing concurrent reads in the presence of updates by dividing update in different phases. This is called as "Two-Phase Update" which are as following:

- Generate new states by carrying out updates from new operations but prevent new states to be accessed by existing operations and allow access to the old states. This duration is called grace period where ongoing operations are allowed to continue on old states but new operations are only allowed new states.
- Complete all the updates as soon as each preexisting operations finishes ie grace period expires. That includes unmapping the relevant modules and freeing up associated data.

The grace period detection is one of the challenging part in the RCU mechanisms. Grace period is detected indirectly by using a *quiescent state*. A "quiescent state" provides a barrier like mechanism by forcing to wait until all preexisting operations on all data structures guaranteed to have been completed. Authors discussed multiple ways to implement a *quiescent state* by using context switches, system call entry, trap from user mode and many more methods.

RCU mechanism provides many improvement on traditional locking mechanism such as fewer explicit mutex operations required to protect access to shared data structures. It also not susceptible to deadlock as explicit lock operations.

Limitations/Critique

The main drawbacks of RCU mechanism is that it works well where update operations are less frequent than read operations. Also because of grace period during read operation we can see stale data and RCU is not much of use in systems which are not event-driven.

Proposed Extension

RCU mechanism can be extended to non-event driven systems. Also RCU can be modified to perform well in update intensive systems. Simplified grace period detection can be implemented or improved to increase performance.