Impact of Low-latency TLB Shootdowns

Low-latency TLB Shootdown

A TLB shootdown invalidates remote address mappings.

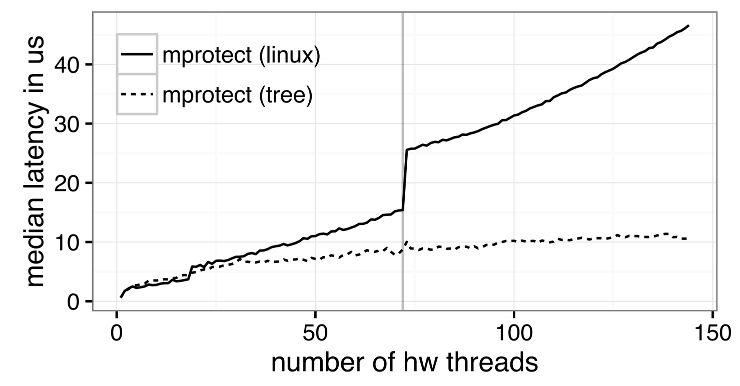
Implementations often propagate invalidations sequencially
→ ***linear scaling with the number of receivers***.

Tree-based propagation can achieve logarithmic scaling
→ **lower latency for large numbers of receivers.**

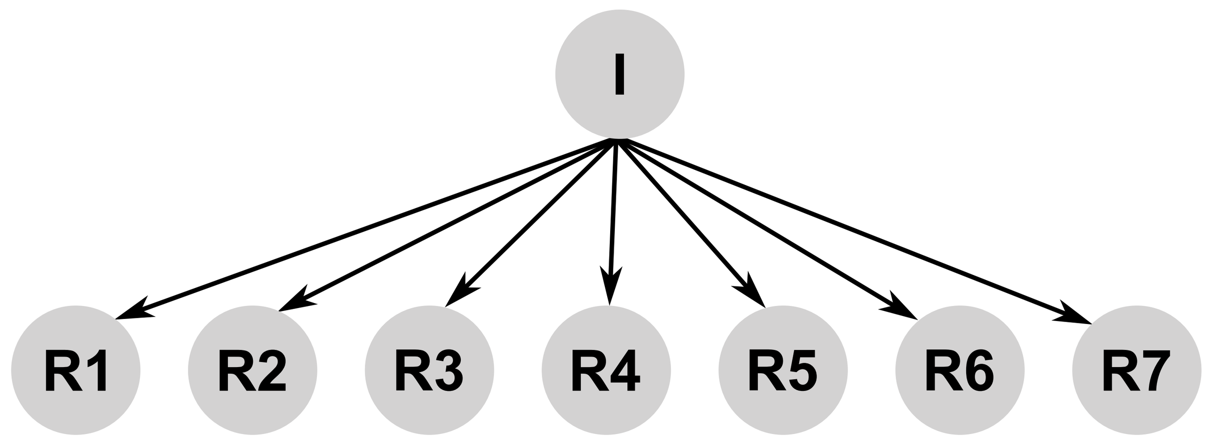
Microbenchmark - mprotect

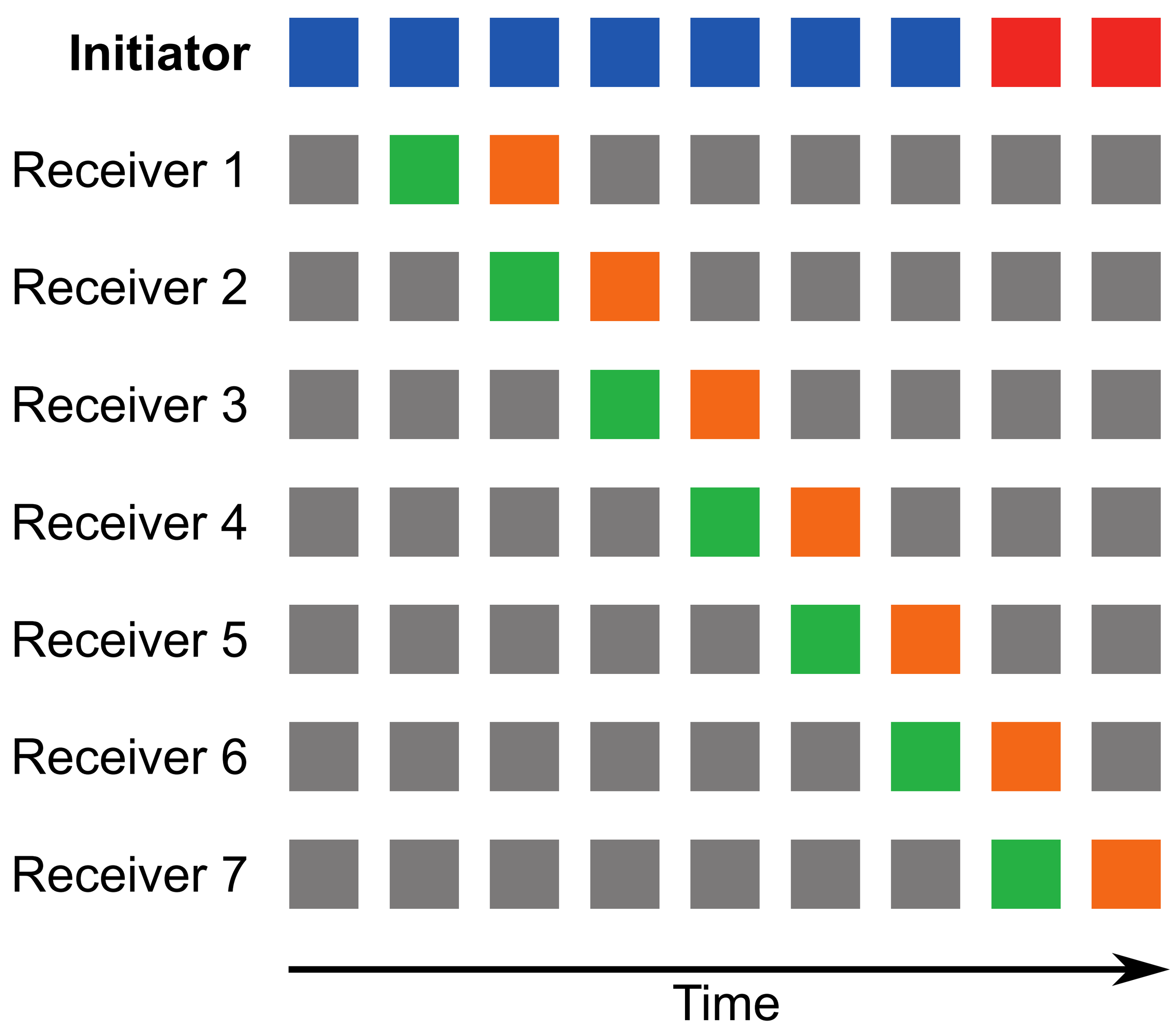
The removal of write protection from a page triggers a TLB shootdown. This plot shows the scaling of mprotect latency with the number of hw threads used.

The tree-based mechanism reduces the latency for large numbers of receivers.

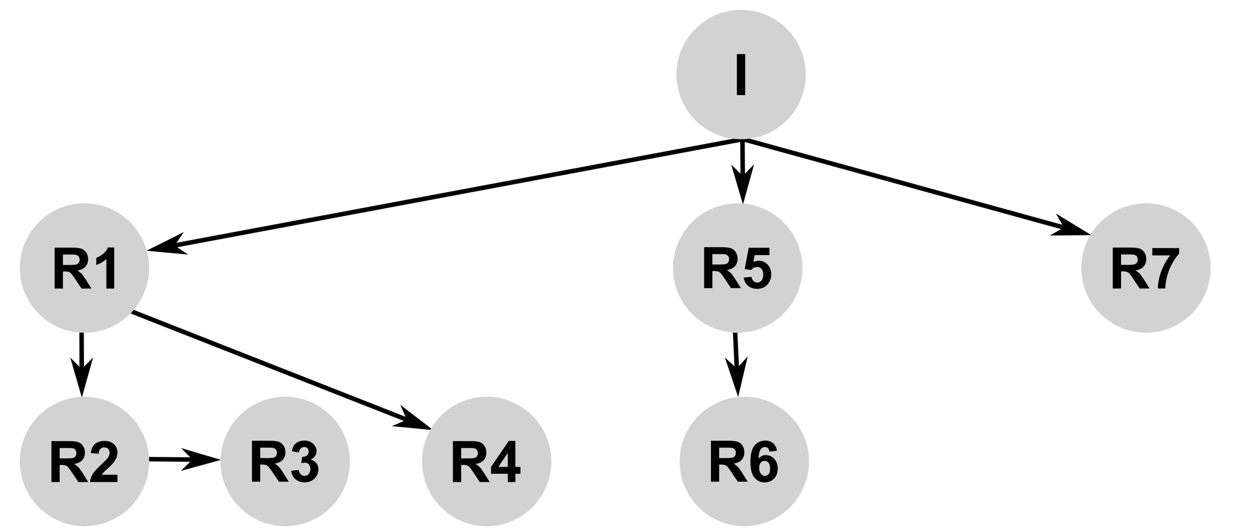


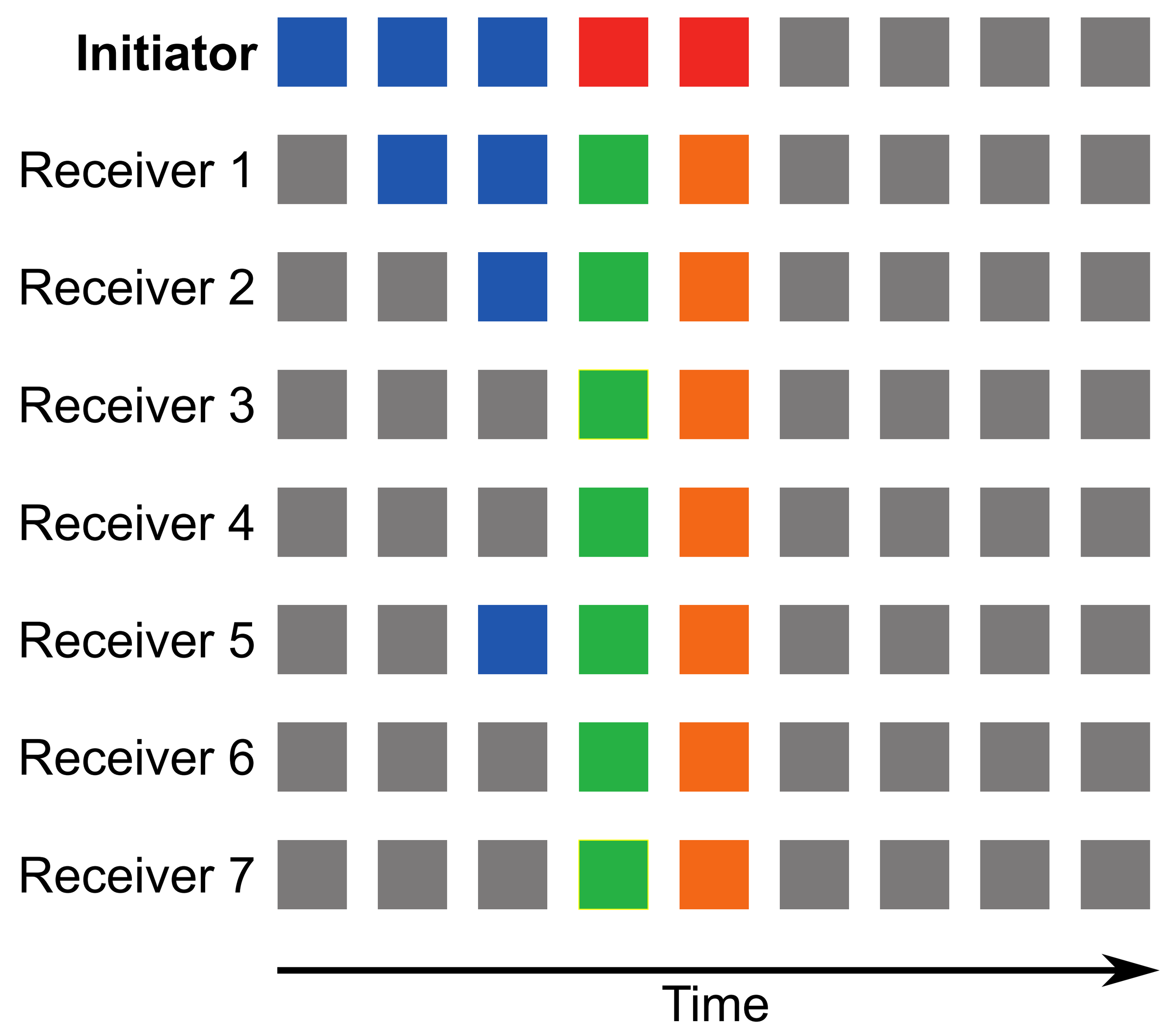
Sequential Propagation in Linux





Binomial Tree





41x application **7x** send **7x** invalidate **7x** acknowledge **2x** wait

- higher latency for TLB shootdown
- all receivers preempted for same time

- lower latency for TLB shootdown
- some receivers (1, 2, 5) preempted for longer

The same amount of work is needed for the propagation of the TLB shootdown.

Memory-intensive Applications

Memory pressure leads to evictions from the page cache, making TLB shootdowns necessary.
Mature cache management algorithms (as used by the page cache) often employ empty page pools and prefetching.
This removes the TLB shootdown latency from the critical path.

Lower shootdown latency alone therefore can not increase throughput.

Multithreaded MapReduce

MapReduce frameworks map large files into memory, making them memory-intensive.
The popular wordcount benchmark of the Phoenix framework uses copy-on-write, triggering a TLB shootdown for every page in the input file. The work balancing in the mapping phase hides progress differences between threads.

Redistribution of the propagation effort therefore does not increase throughput.

Other Parallel Applications

Most applications avoid interaction with the virtual memory subsystem.
Parallel benchmark suites (PARSEC, Splash-2) therefore can only confirm that replacing the propagation mechanism does not degrade the performance.

A significant change in throughput is unlikely when solely the propagation mechanism is changed.

b-tu
Brandenburg University of Technology Cottbus - Senftenberg

Robert Kuban, Randolf Rotta, Jörg Nolte
Distributed Systems & Operating Systems Group
Brandenburg University of Technology Cottbus–Senftenberg,
Cottbus,Germany
Email: {robert.kuban, randolf.rotta, joerg.nolte}@b-tu.de

DFG Deutsche Forschungsgemeinschaft
This work was supported by the German Research Foundation (DFG) under project no. 224615364.