

Tightening Critical Section Bounds in Mixed-Criticality Systems through Preemptible Hardware Transactional Memory

Tightening Critical Section Bounds in **Mixed-Criticality Systems** through Preemptible Hardware Transactional Memory

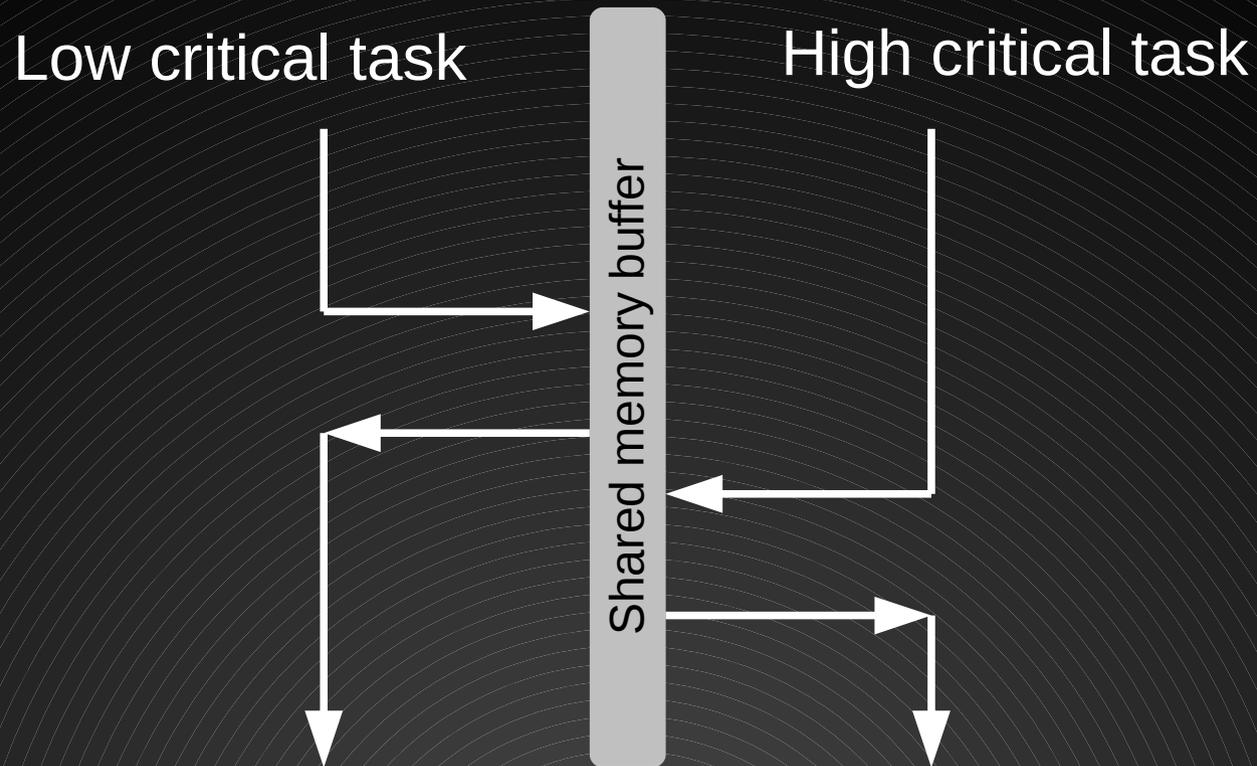
Tightening **Critical Section** Bounds in Mixed-Criticality Systems through Preemptible Hardware Transactional Memory

Tightening Critical Section Bounds in Mixed-Criticality Systems through Preemptible Hardware Transactional Memory

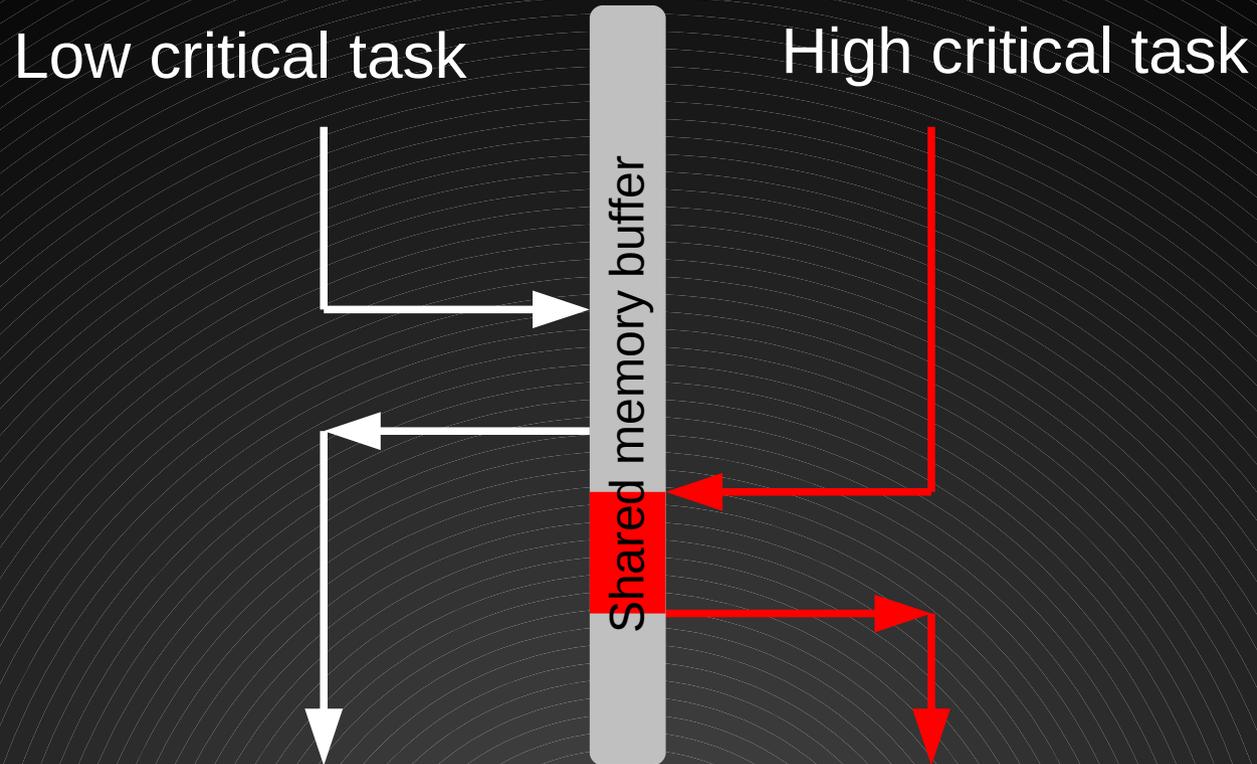
Tightening Critical Section Bounds in Mixed-Criticality Systems through **Preemptible** **Hardware Transactional Memory**

- Consolidate multiple tasks with *different* certification requirements into a single system
- Tasks can have multiple WCETs, reflecting their certification/validation level
- Two tasks H (1, 2, 2) and L (1, —, 2) with (c_{Low} , c_{High} , period)

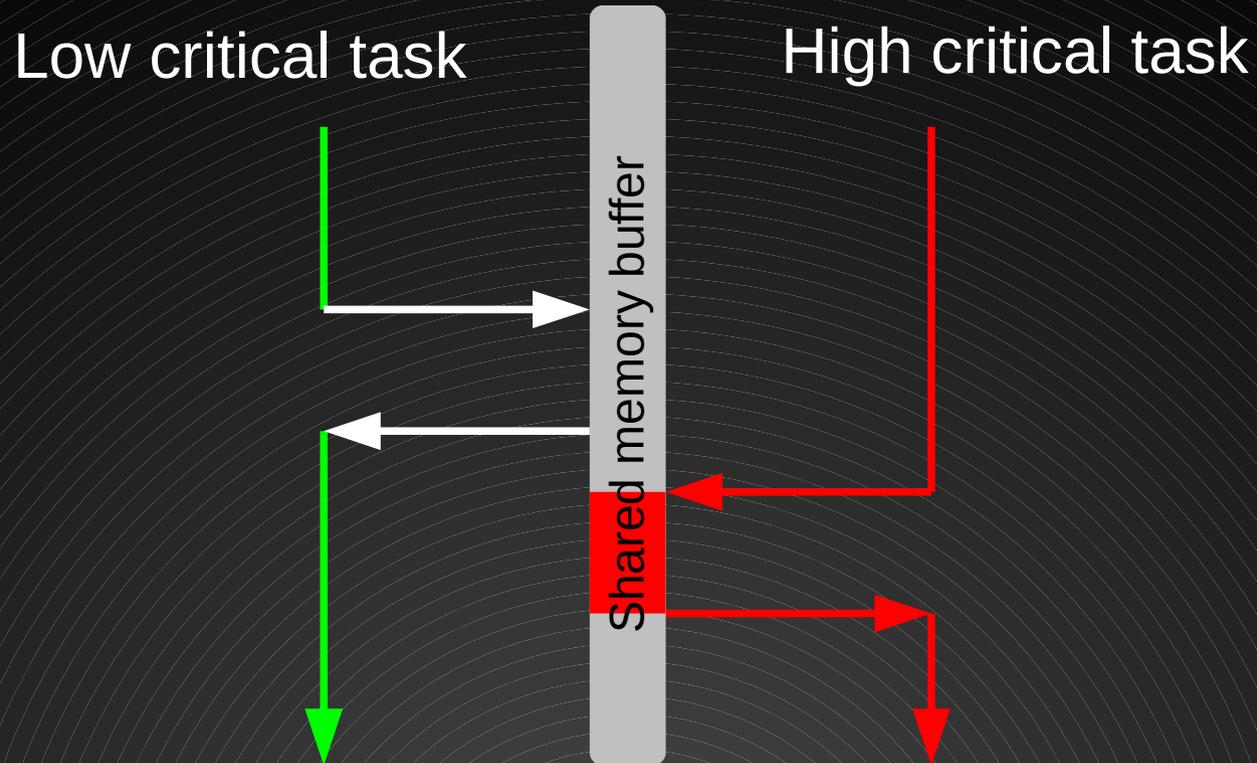
Critical Sections



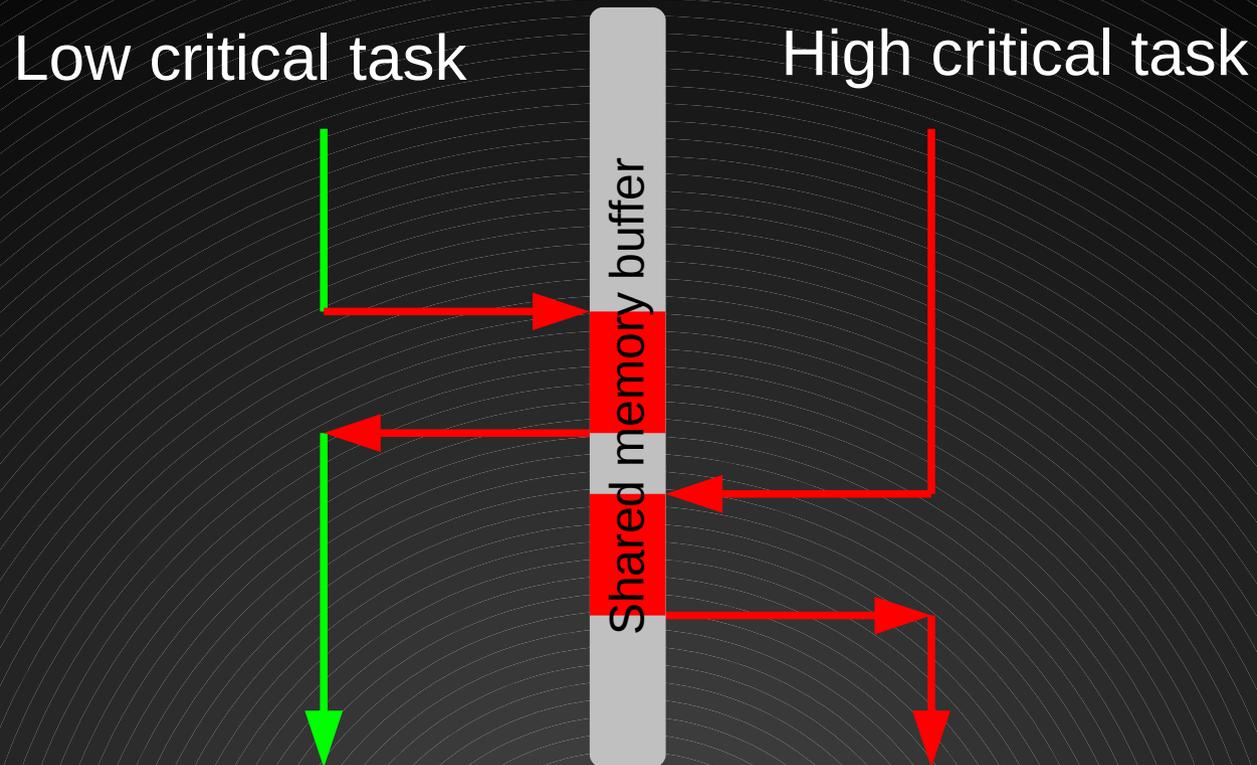
Critical Sections



Critical Sections



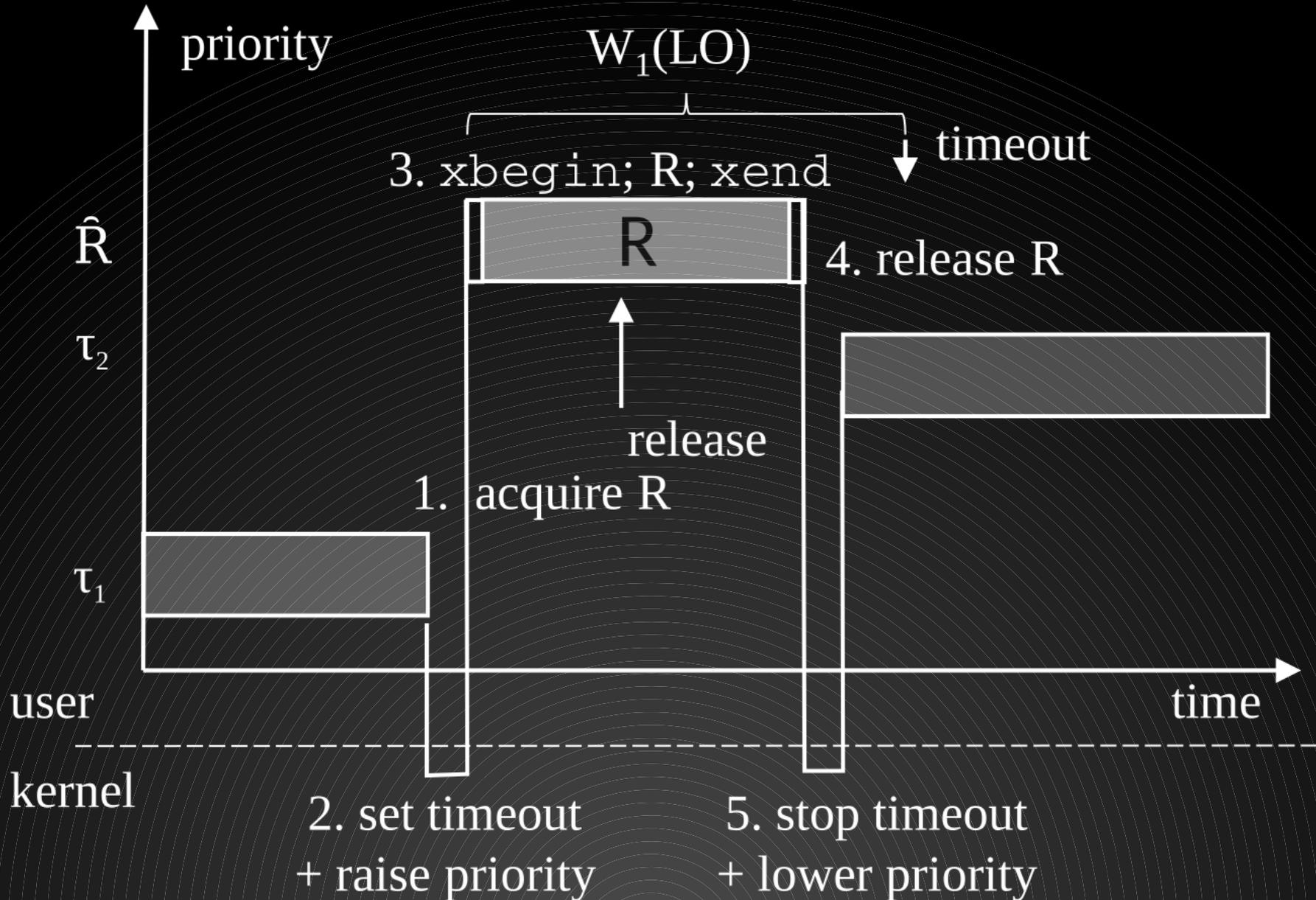
Critical Sections



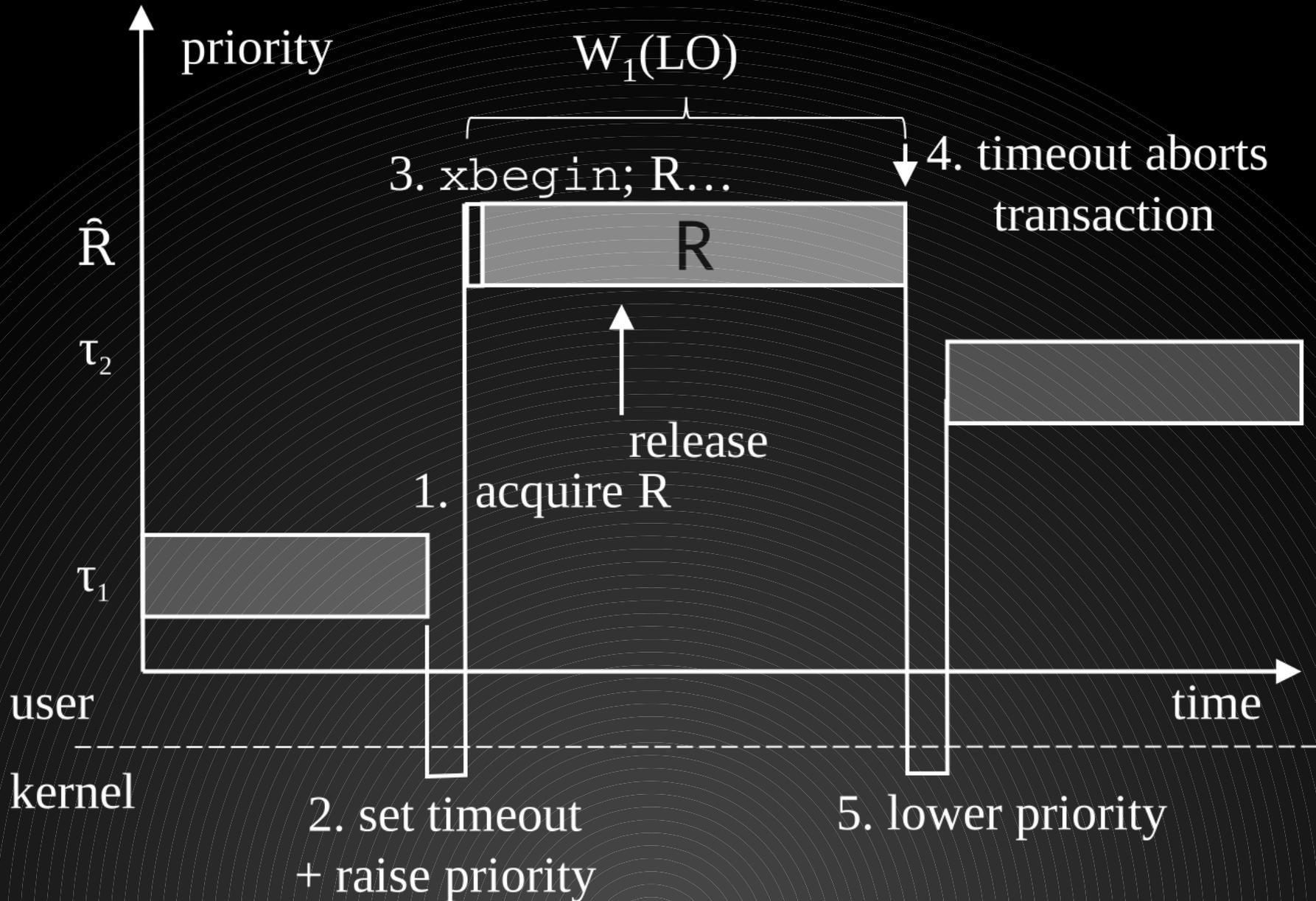
- Low critical tasks are dropped, if the system changes into high criticality mode
- But: shared resource cannot be revoked, thus accessing it has to be evaluated equal to the highest criticality of any task accessing it (compare: ceiling priority)
- Revocation using transactional memory

- Accumulate changes locally, commit or abort atomically, i.e. all-or-nothing
- HTM not used for synchronization (locks are still used), but to quickly abort low-criticality tasks even if they are in a critical section

Ceiling Priority + Commit

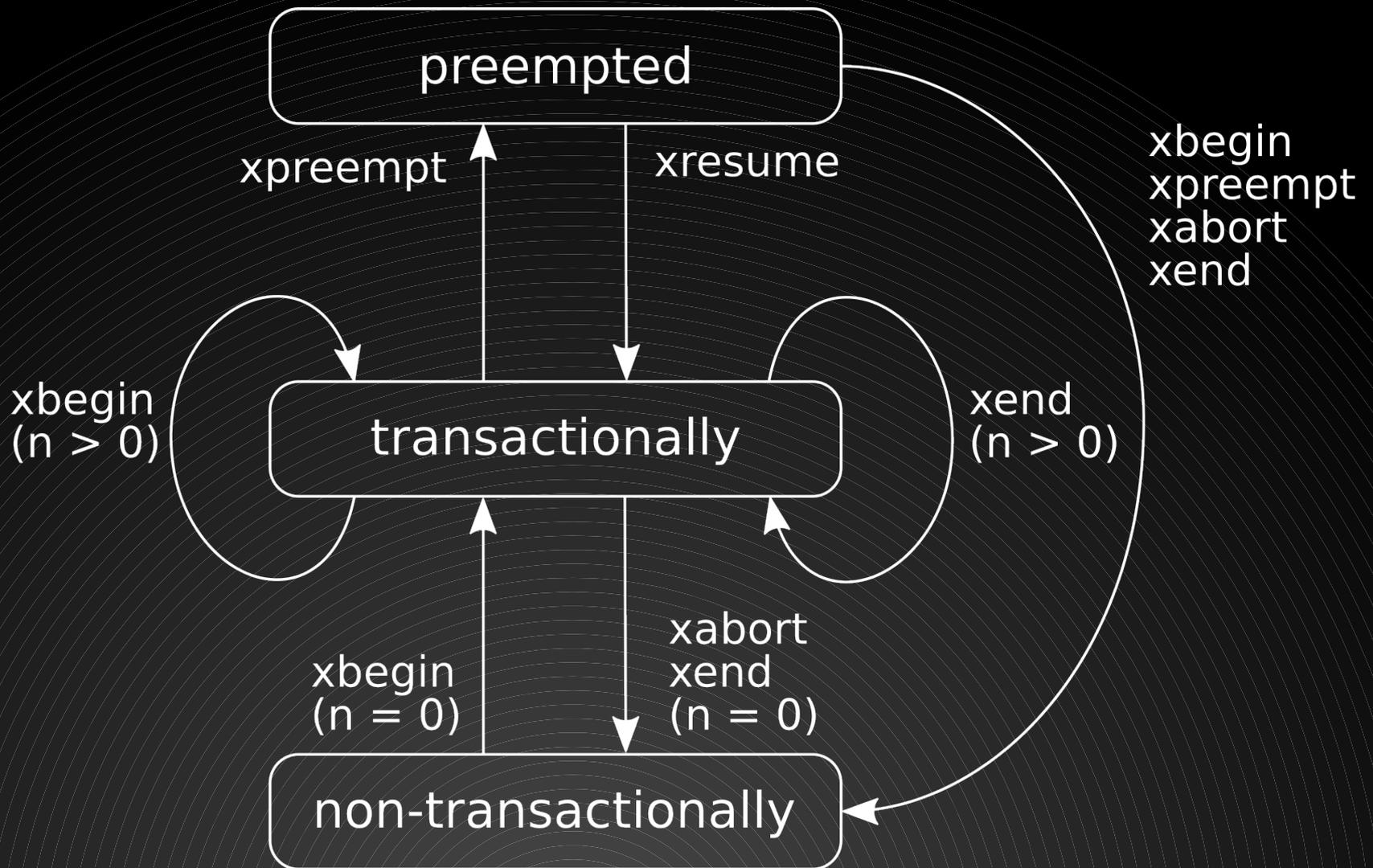


Ceiling Priority + Timeout



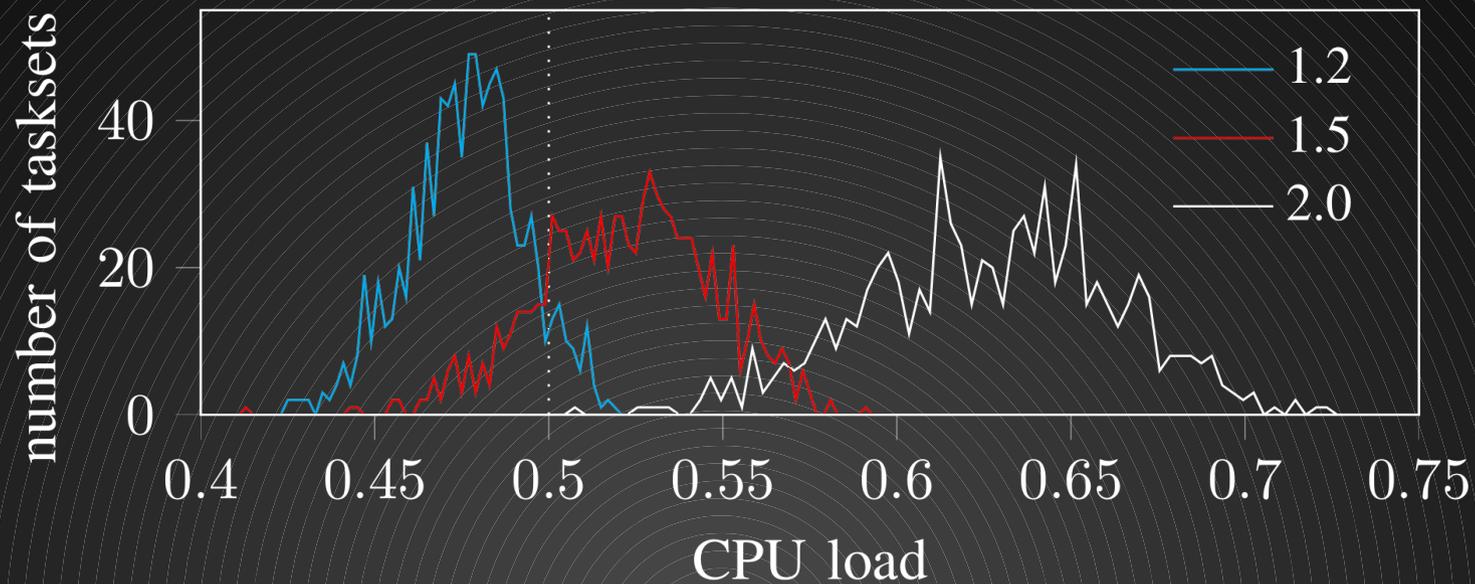
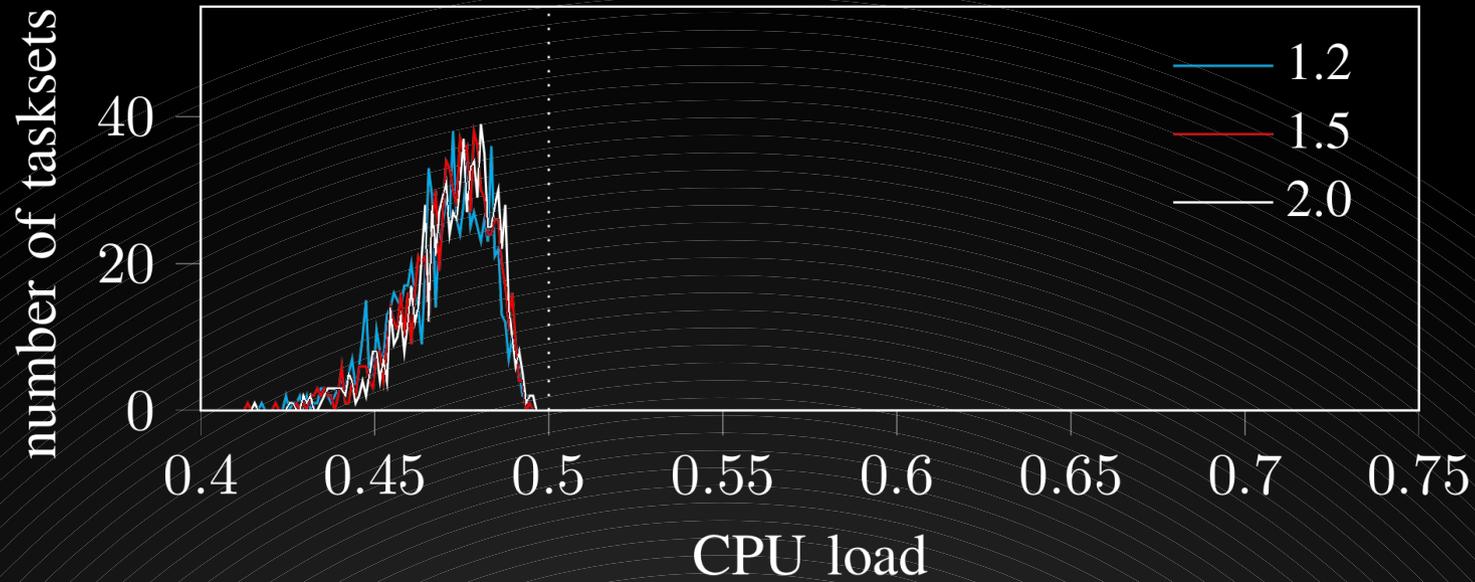
- Cycle-accurate Out-of-Order CPU
- Added Intel-like HTM support (XBegin/XEnd)
- L1 Data cache lines with 'T'-Bit
- Transactional mode in CPU and Cache
- Adapted snoop logic to properly handle remote accesses to transactional cache lines

- But: Intel-like transactions are fragile, abort on various reasons, e.g. exceptions and interrupts
- This must not happen, so we added XPreempt/XResume to pause/unpause transactions

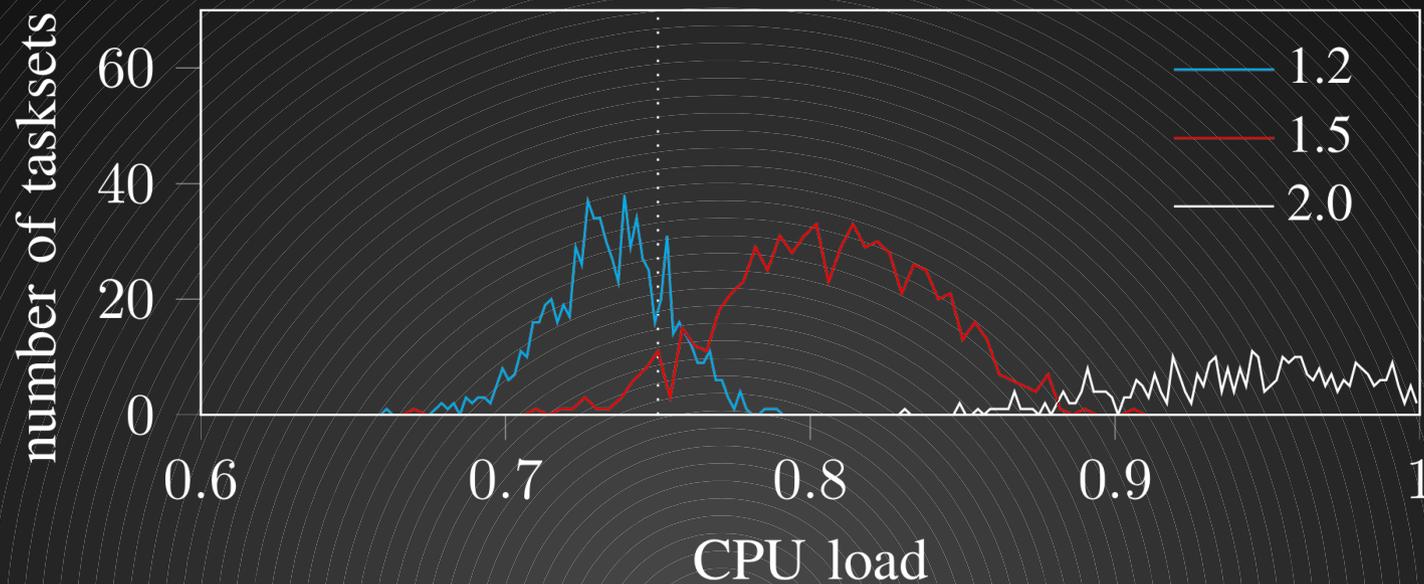
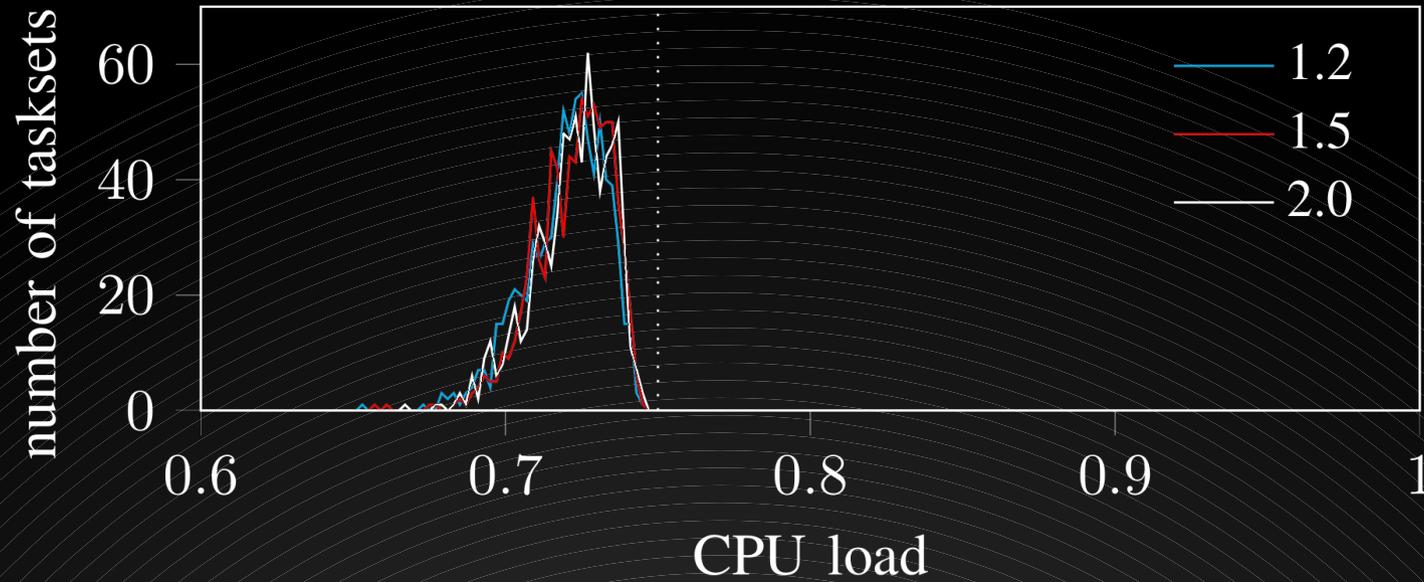


- Random-generated task sets (up to 10 tasks) using UUnifast algorithm
- Periods between 4 and 144, resulting in a hyperperiod of 5184
- Dual-criticality systems, where High-WCET is 1.2 / 1.5 / 2.0 times the respective Low-WCET
- System load up to 0.5 / 0.75 / 1.0 and 1.5 for multi processor setup

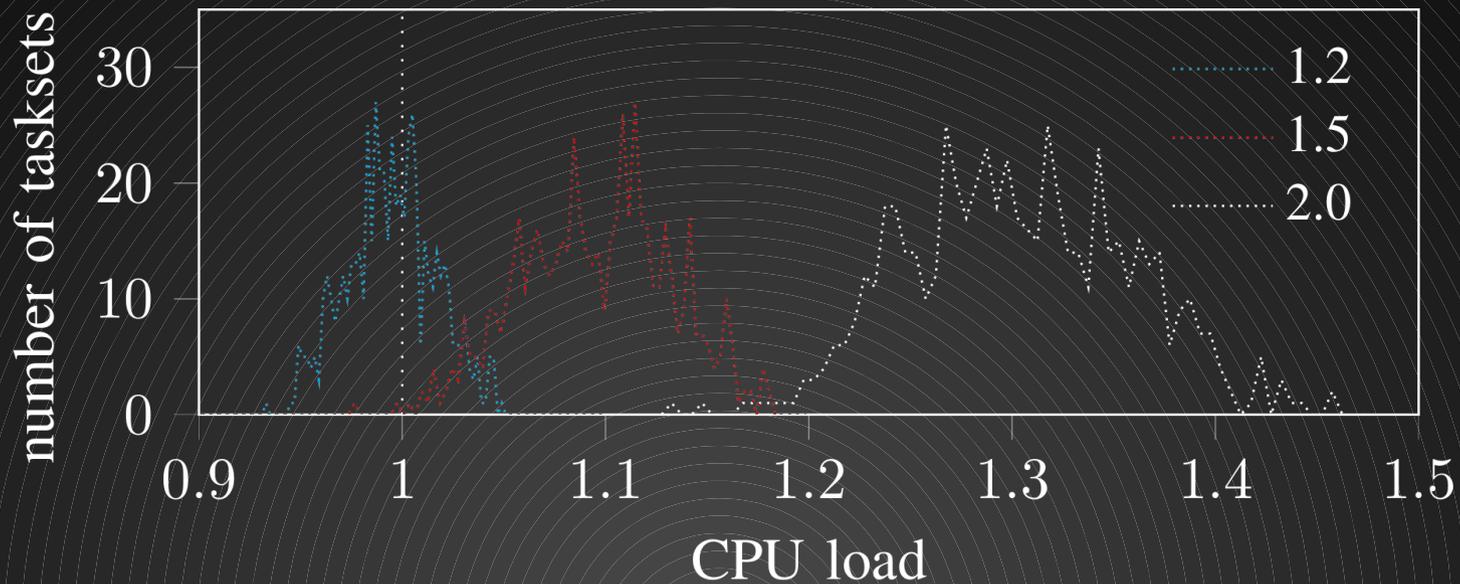
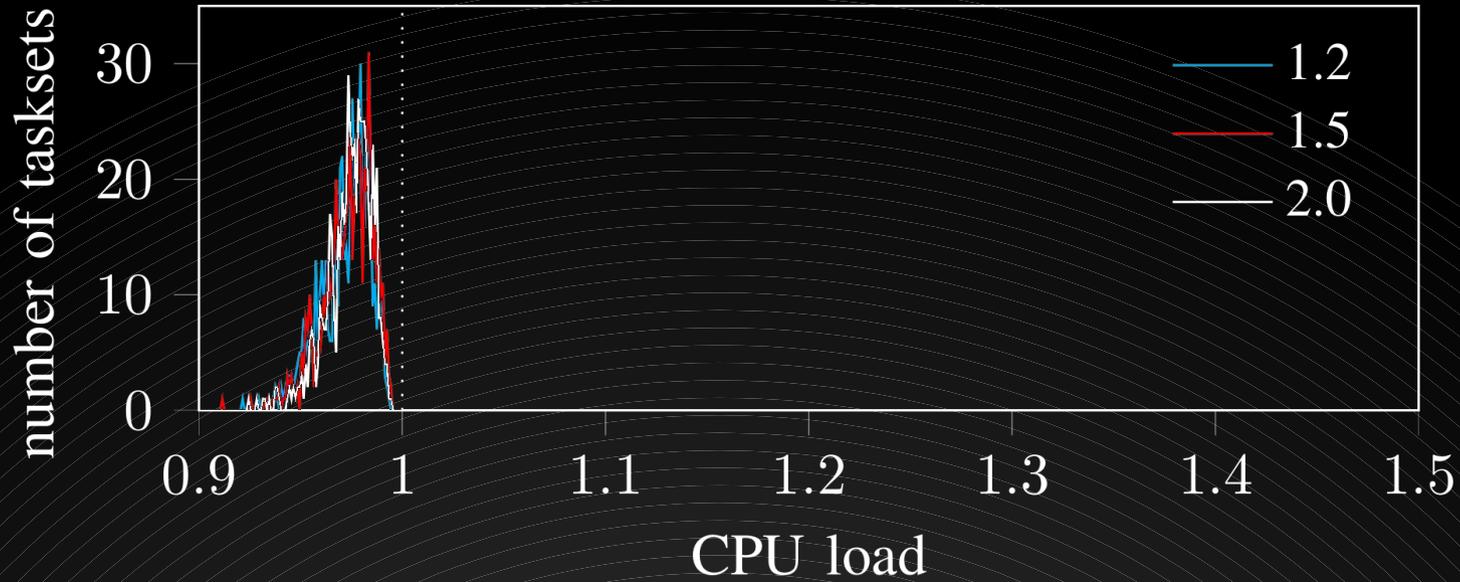
50% System Load



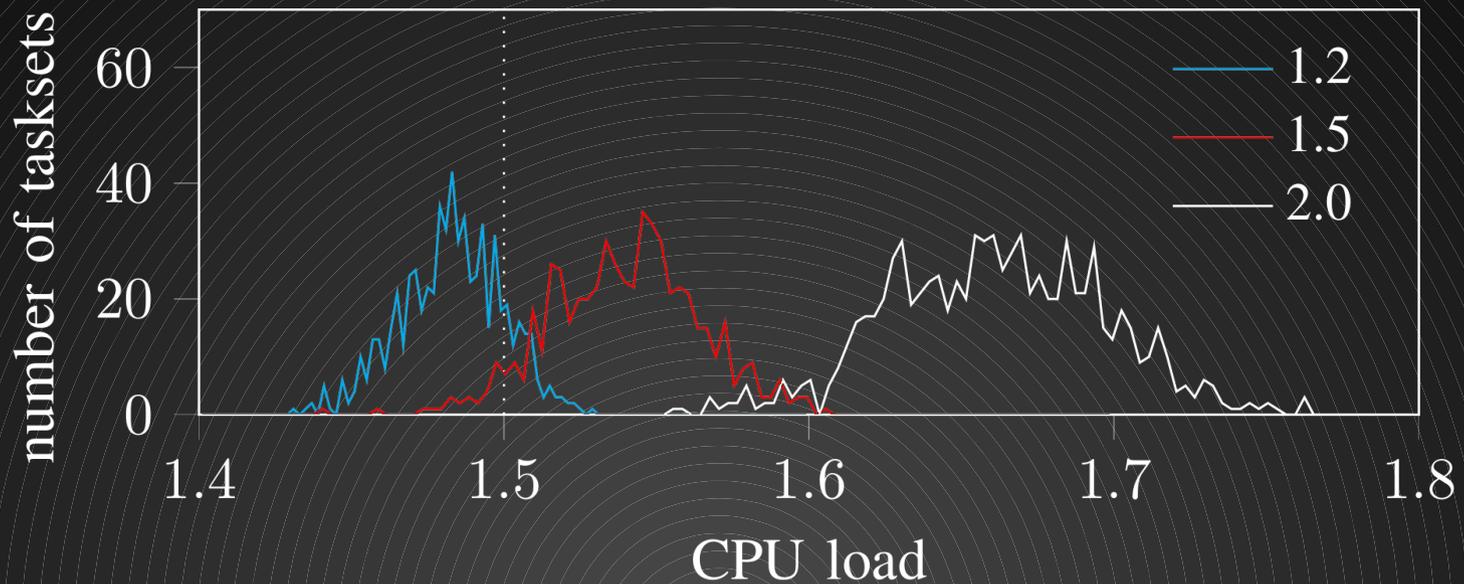
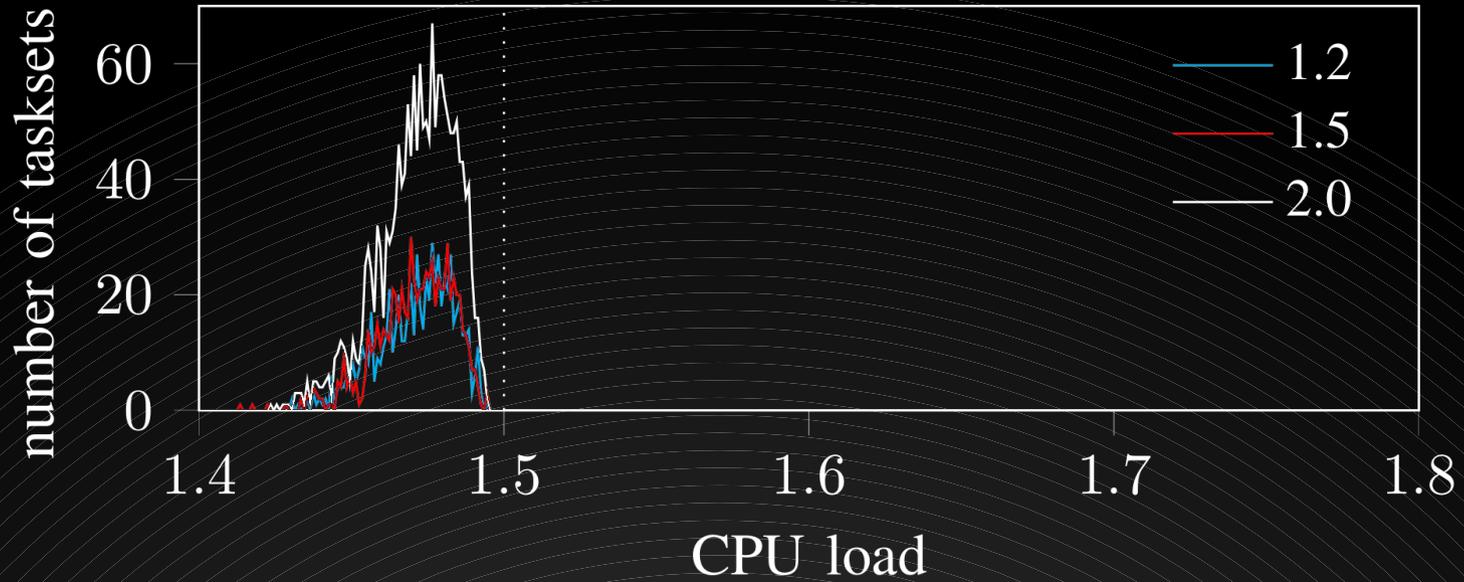
75 % System Load



100% System Load



150 % System Load on SMP



Low-crit to High-crit ratio

		1 : 1.2	1 : 1.5	1 : 2.0
System load	50%	1%	6%	21%
	75%	4%	14%	54%
	100%	45%	99.6%	---
	150%	5%	26%	50%

- Using transactional memory to enforce tighter critical section bounds
- Implementation of preemptible HTM in gem5
- Improved schedulability and reduced system load using HTM by allowing low criticality resource access times being trustworthy