

Concurrent Systems

Nebenläufige Systeme

XV. Remote Core Locking

Wolfgang Schröder-Preikschat, Timo Hönig

February 6, 2018



- Final Sprint Winter Term 2017/2018
 - Lecture 12 (January 23, Transactional Memory): self-study
 - Lecture 13 (January 30, Progress Guarantees): self-study
 - Lecture 14 (February 6, Remote Core Locking, Pickings): as scheduled



- Final Sprint Winter Term 2017/2018
 - Lecture 12 (January 23, Transactional Memory): self-study
 - Lecture 13 (January 30, Progress Guarantees): self-study
 - Lecture 14 (February 6, Remote Core Locking, Pickings): as scheduled
- Preparation for Lecture 14
 - read „Remote Core Locking” (Lozi et al., USENIX 2012)
 - read „Remote Core Locking” (Lozi et al., USENIX 2012), at least **twice**
 - RCL: Locking algorithm which improves the performance of critical sections on multicore architectures
 - analyse and study the paper during the next two weeks
 - exchange and discuss your findings and perception in Lecture 14



- Final Sprint Winter Term 2017/2018
 - Lecture 12 (January 23, Transactional Memory): self-study
 - Lecture 13 (January 30, Progress Guarantees): self-study
 - Lecture 14 (February 6, Remote Core Locking, Pickings): as scheduled
- Preparation for Lecture 14
 - read „Remote Core Locking” (Lozi et al., USENIX 2012)
 - read „Remote Core Locking” (Lozi et al., USENIX 2012), at least **twice**
 - RCL: Locking algorithm which improves the performance of critical sections on multicore architectures
 - analyse and study the paper during the next two weeks
 - exchange and discuss your findings and perception in Lecture 14
- Background and Guiding Hints
 - technical analysis of the paper, critical reflection on its approach
 - What are the {contributions,limitations} of RCL?
 - What would you consider as additional, further research?

