Carnegie
Mellon
University

CENTER FOR
ADVANCING
ELECTRONICS
DRESDEN

# Hard Real-Time Multiprocessor Scheduling

## (Part II)

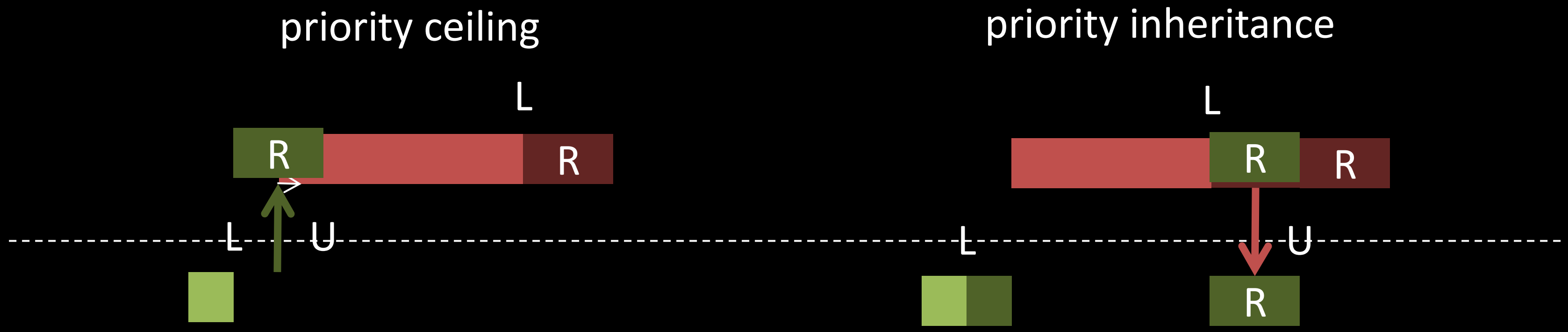
Marcus Völp

TECHNISCHE
UNIVERSITÄT
DRESDEN

# Lessons Learned from UP

two concepts to counteract priority inversion

- priority inheritance
- priority ceiling



# Locking Protocols

[Brandenburg ’10,

Brandenburg ‘13]

- Where to execute the critical section?

  local vs. remote

- How to order conflicting critical sections?

  FIFO vs. priorities

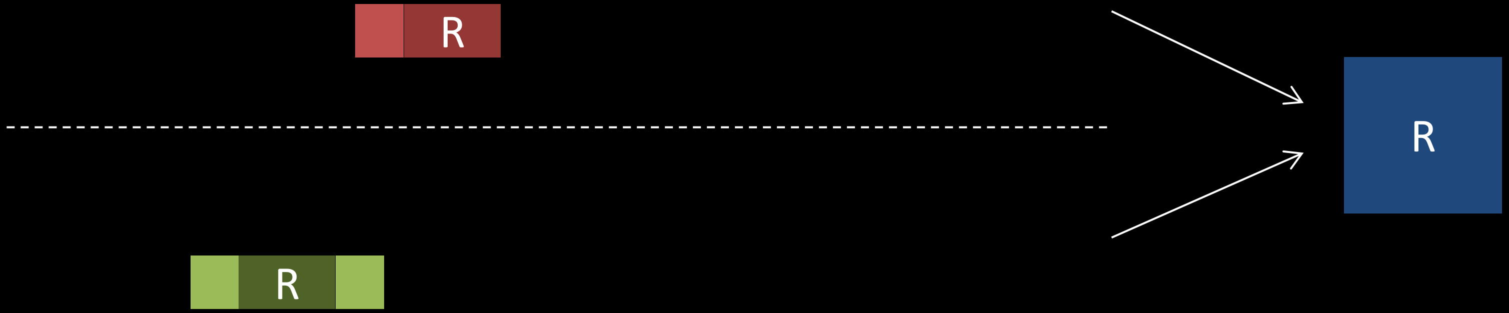
- How unrelated tasks affect the lock holders?
- Whether and how unrelated tasks are affected?

# Where to execute?

[Brandenburg ’10,

Brandenburg ‘13]

local vs. remote

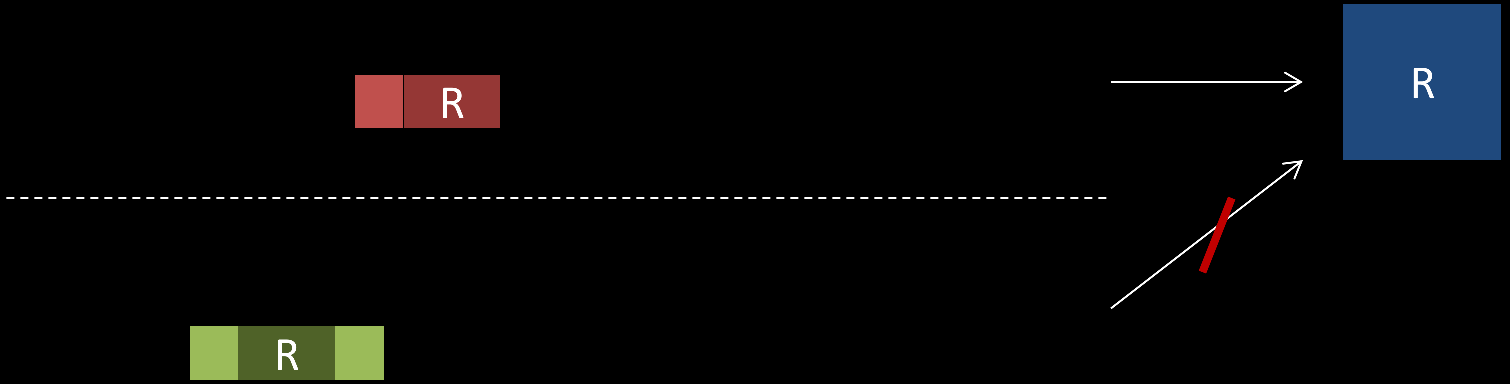


# Where to execute?

[Brandenburg ’10,

Brandenburg ‘13]

local vs. remote

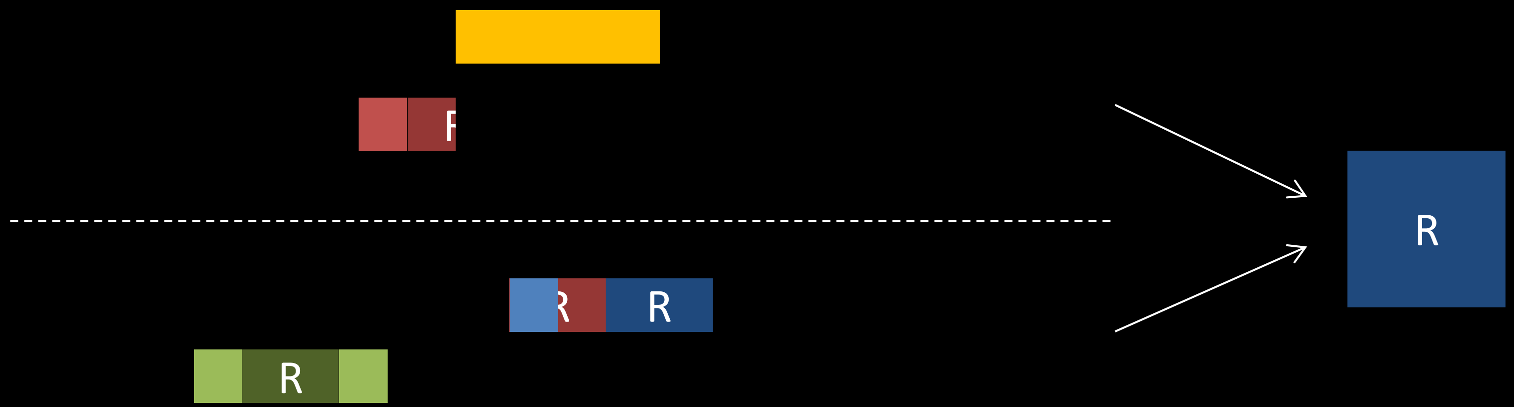


# Where to execute?

[Brandenburg ’10,

Brandenburg ‘13]

local vs. remote

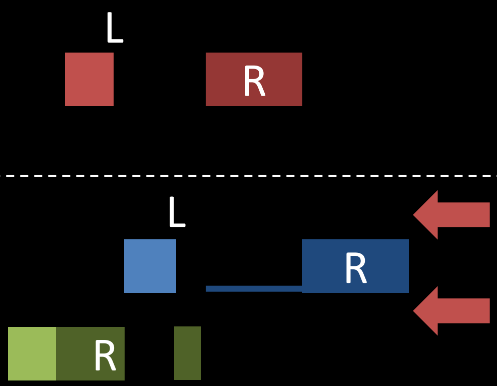


# How to order conflicts?

[Brandenburg ’10,

Brandenburg ‘13]

FIFO vs. priorities



FIFO: O(n)
Priorities: O(higher prioritized on other CPUs)

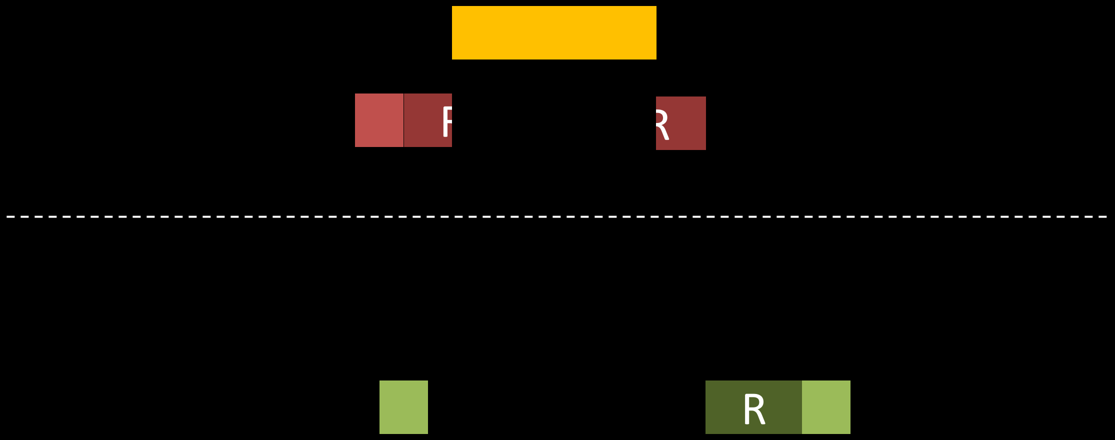
O(1) ?     O(m) ?

# Unrelated tasks affect the lock

[Brandenburg ’10,

Brandenburg ‘13]

Block local vs. remote



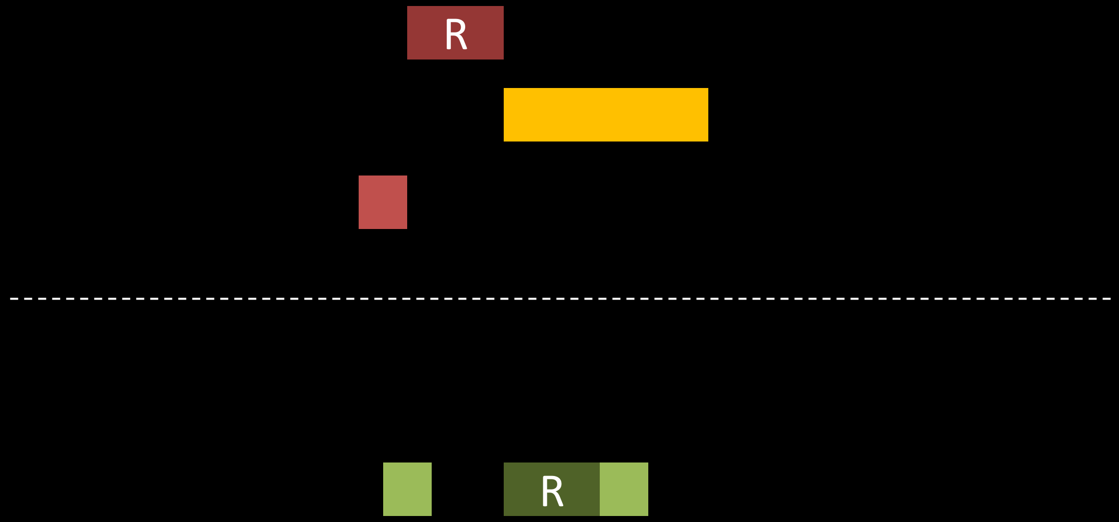
$$\sum_{\tau_j \in H(i)} C_j$$

# Unrelated tasks affect the lock

[Brandenburg ’10,

Brandenburg ‘13]

Block local vs. remote



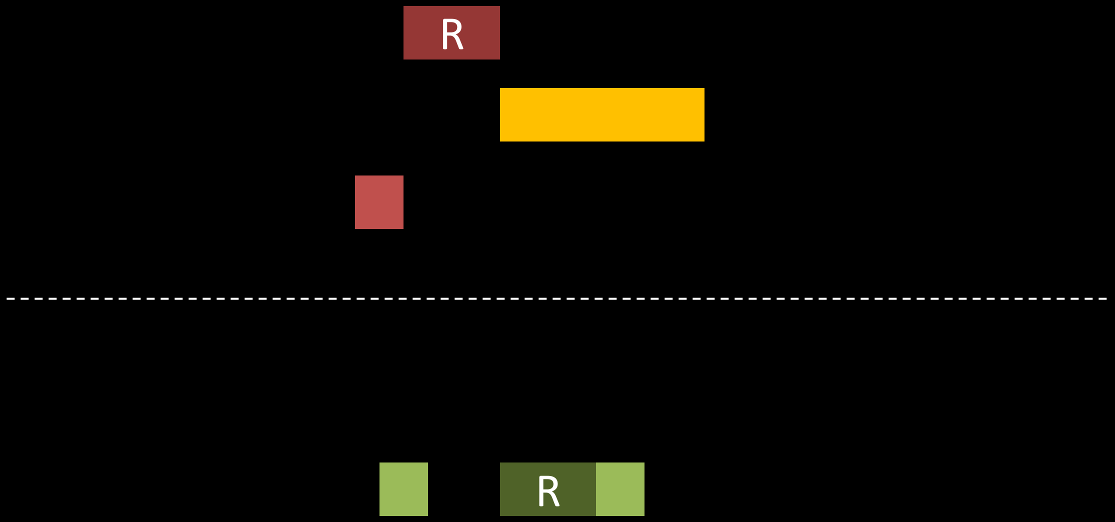
$$\sum_{\tau_j \in H(i)} C_j$$

# How unrelated threads are affected

[Brandenburg ’10,

Brandenburg ‘13]

Block local vs. remote



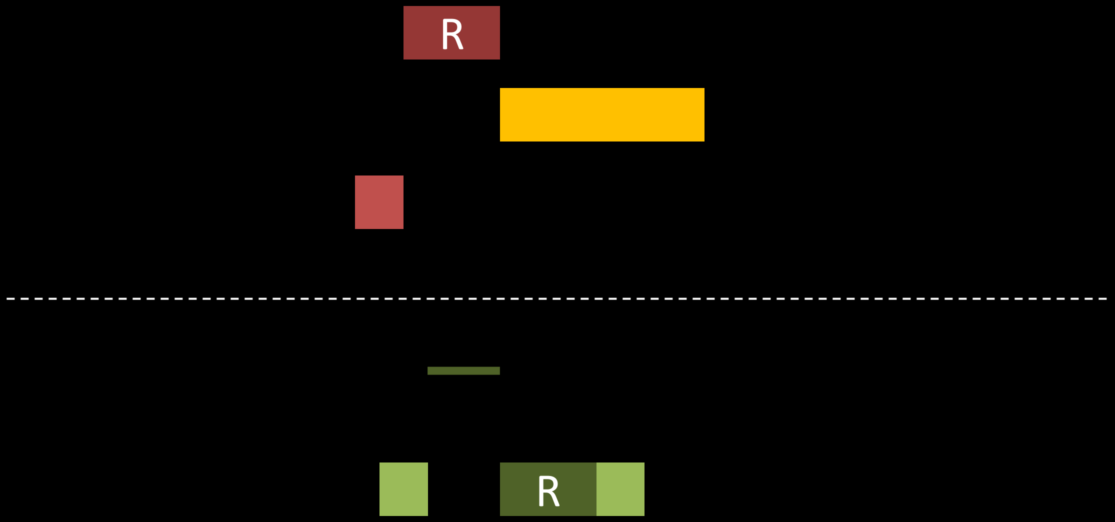
$$\sum_{\tau_j \in H(i)} C_j$$

# How unrelated threads are affected

[Brandenburg ’10,

Brandenburg ‘13]

Block local vs. remote



$$\sum_{\tau_j \in H(i)} C_j$$

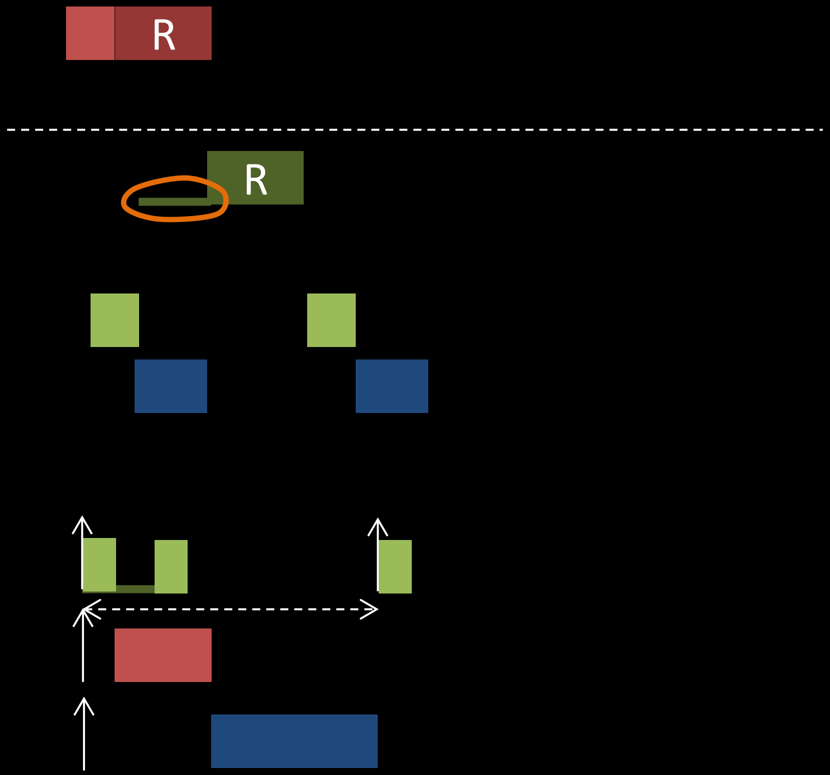
# RT-Locking Terminology

schedulability analysis

- suspension-aware
- suspension-oblivious

blocking

- priority-inversion blocking
- deferral blocking



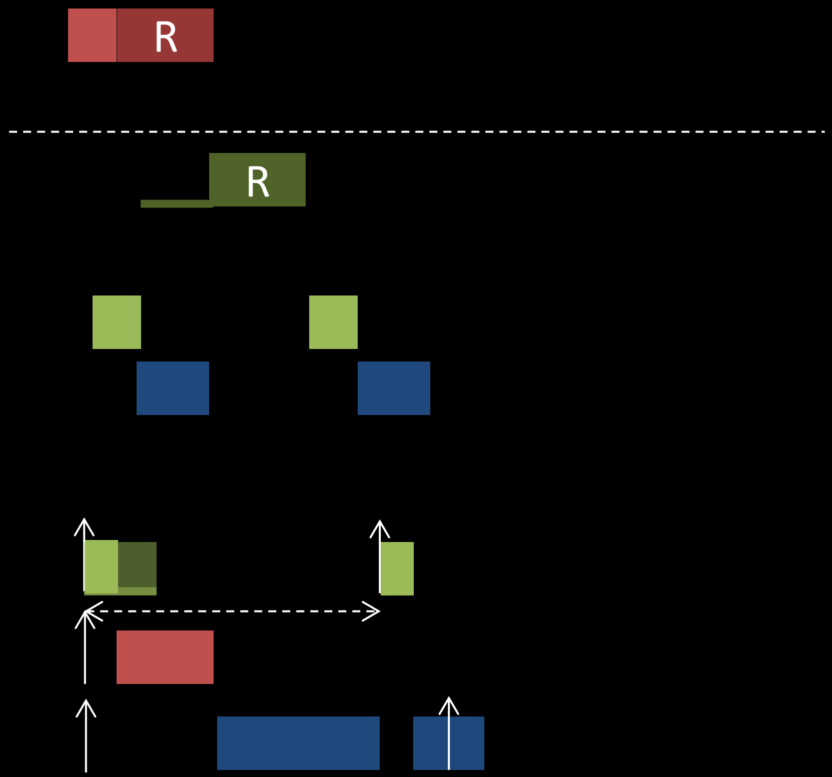
# RT-Locking Terminology

schedulability analysis

- suspension-aware
- suspension-oblivious

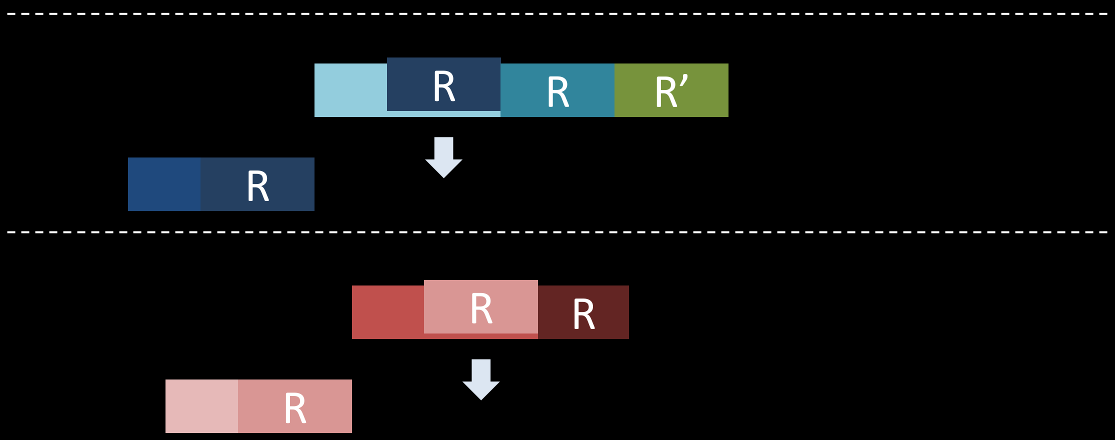
blocking

- priority-inversion blocking
- deferral blocking



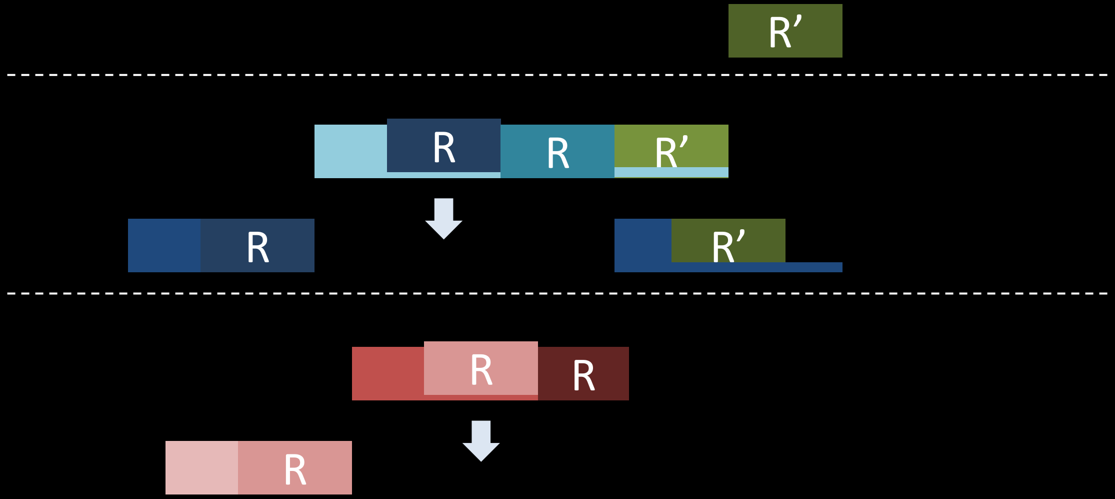
# Reuse UP Protocols?

Distributed Priority-Ceiling Protocol (DPCP) [Rajkuma, Sha, Lehocky ‘88]



# Reuse UP Protocols?

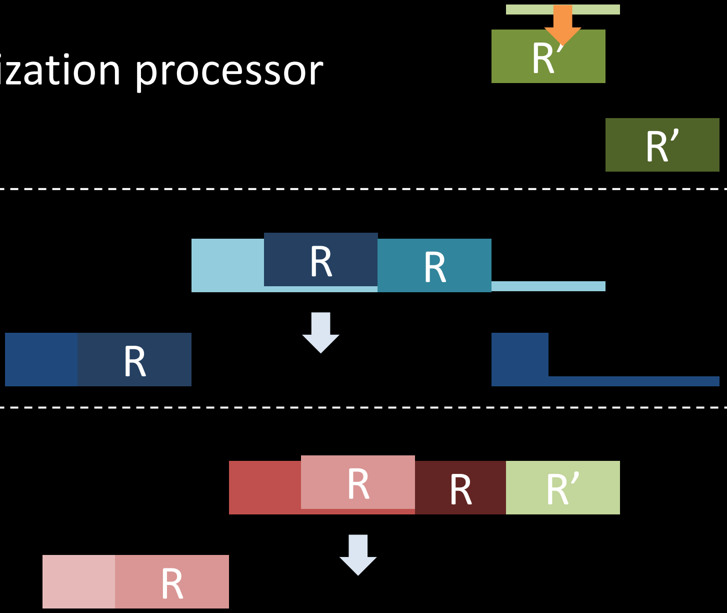
Distributed Priority-Ceiling Protocol (DPCP) [Rajkuma, Sha, Lehocky ‘88]



# Reuse UP Protocols?

Distributed Priority-Ceiling Protocol (DPCP) [Rajkuma, Sha, Lehocky '88]

synchronization processor

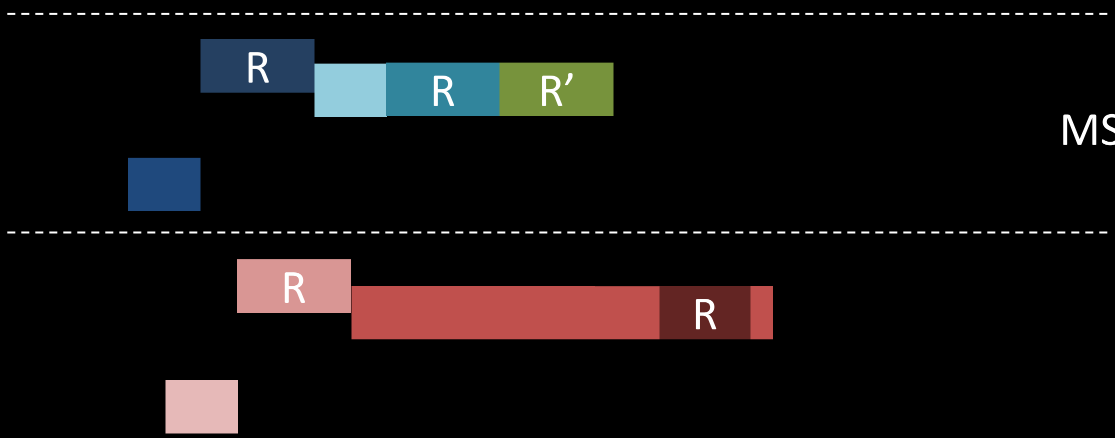


DPCP = $\text{BPCP}^{m+1}$

# Reuse UP Protocols?

Multiprocessor Stack Resource Protocol (MSRP) [Gai, Lipari, Natale ‘01]

synchronization processor



MSRP = $\text{MSRP}^{m+1}$

# Björn Brandenburg

Improved Analysis and Evaluation of Real-Time Semaphore Protocols for Partitioned Fixed Priority Scheduling ‘13

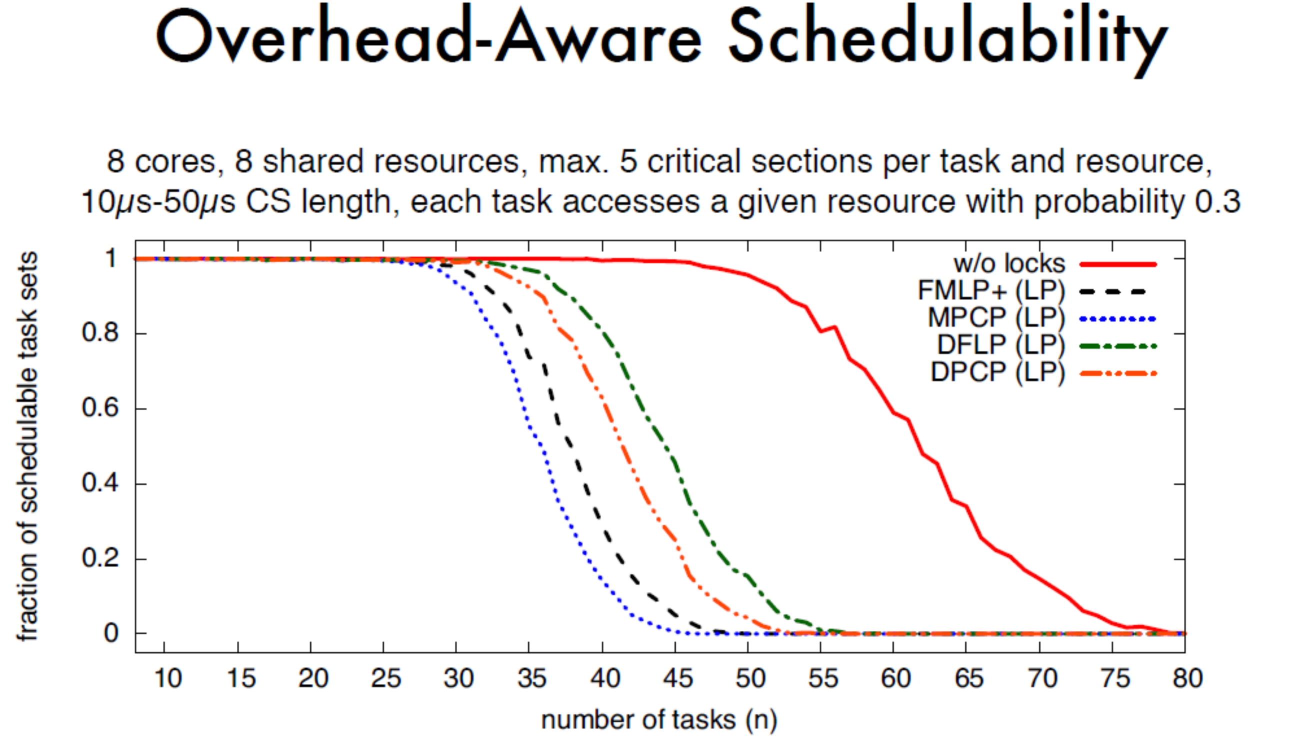
hdwallpaperscool.com

| | MPCP<br>[Rajkumar ‘90,<br>Lakshmanan ’09] | FMLP<br>[Brandenburg ’11] | DPCP<br>[Rajkumar ’88] | DFLP<br>[Brandenburg ’11] |
|---|---|---|---|---|
| wait queue | priority | FIFO | priority | FIFO |
| protocol type | shared mem.<br>(local) | shared mem.<br>(local) | distributed<br>(sync. CPU) | distributed<br>(sync. CPU) |
| asymptotically optimal (wrt. maximal blocking) | No | Yes | No | Yes |

# Björn Brandenburg

Improved Analysis and Evaluation of Real-Time Semaphore Protocols for Partitioned Fixed Priority Scheduling ‘13

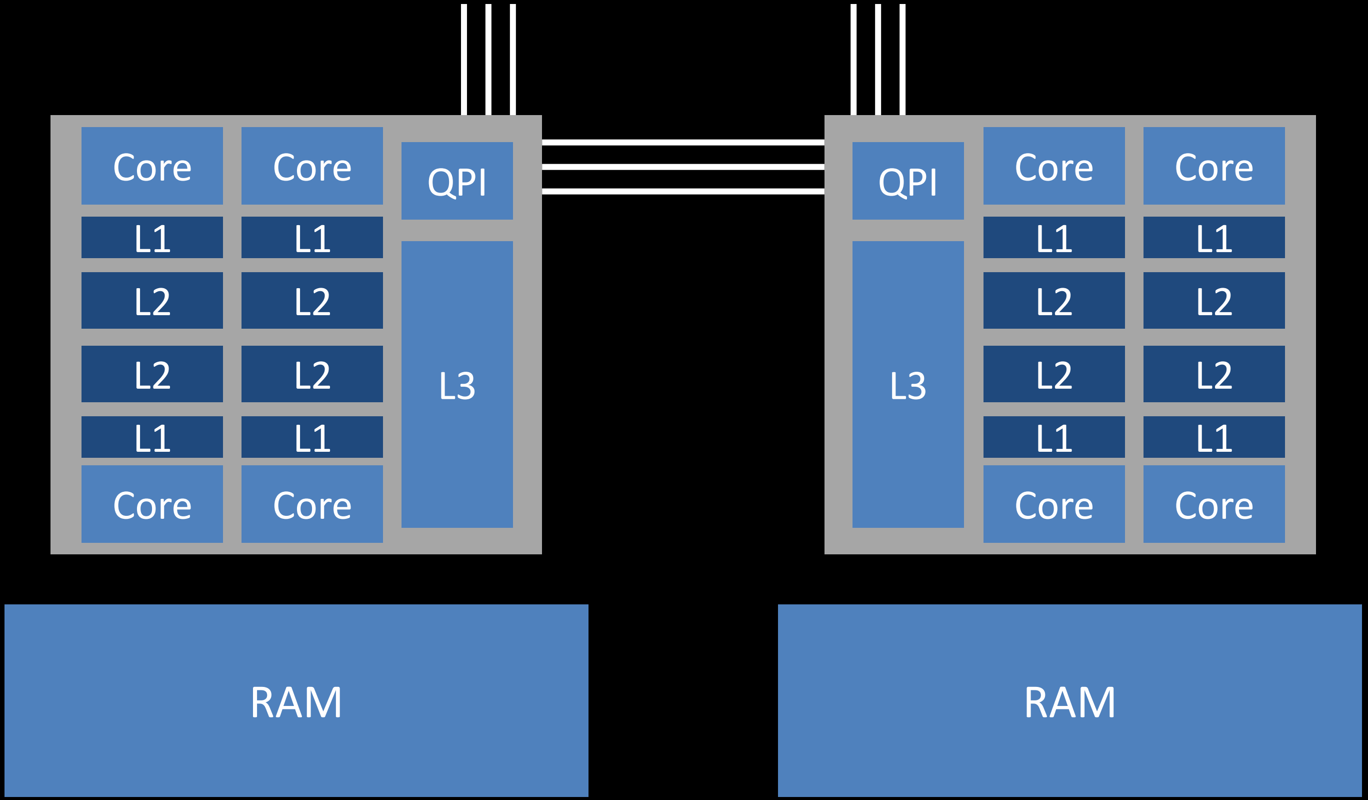
hdwallpaperscool.com



## Overhead-Aware Schedulability

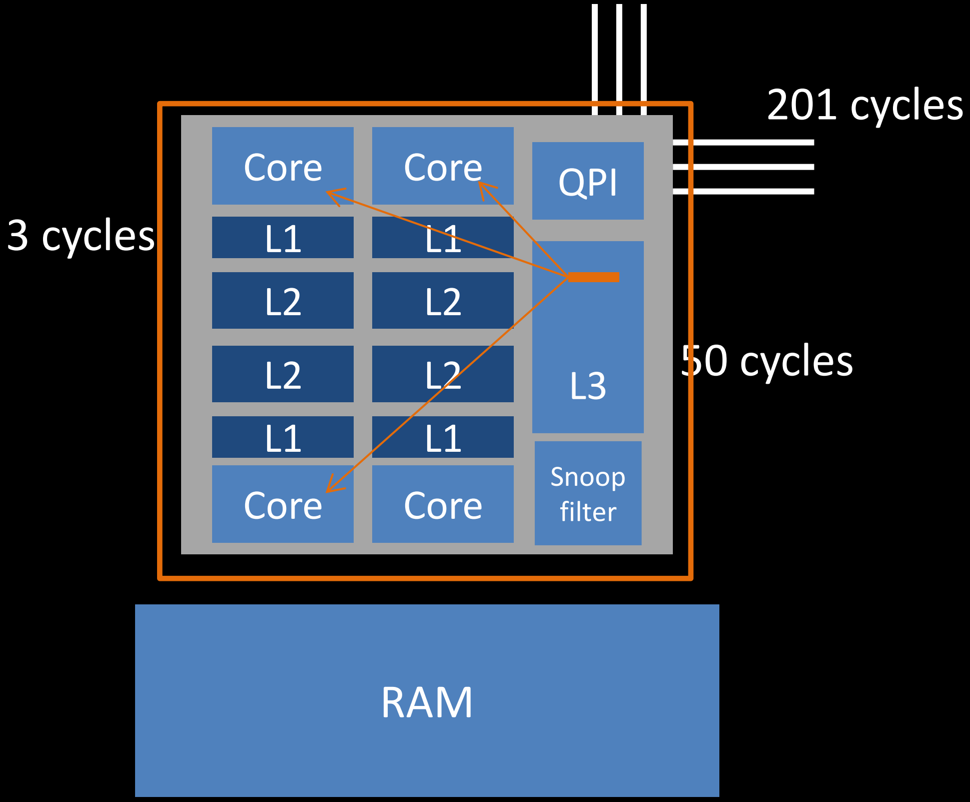
8 cores, 8 shared resources, max. 5 critical sections per task and resource, 10$\mu$s-50$\mu$s CS length, each task accesses a given resource with probability 0.3

# Caches



hdwallpaperscool.com

# Caches



[Corey: OSDI ‘08]

reduce coherent traffic by

keeping track of cached lines

- inclusive L3:

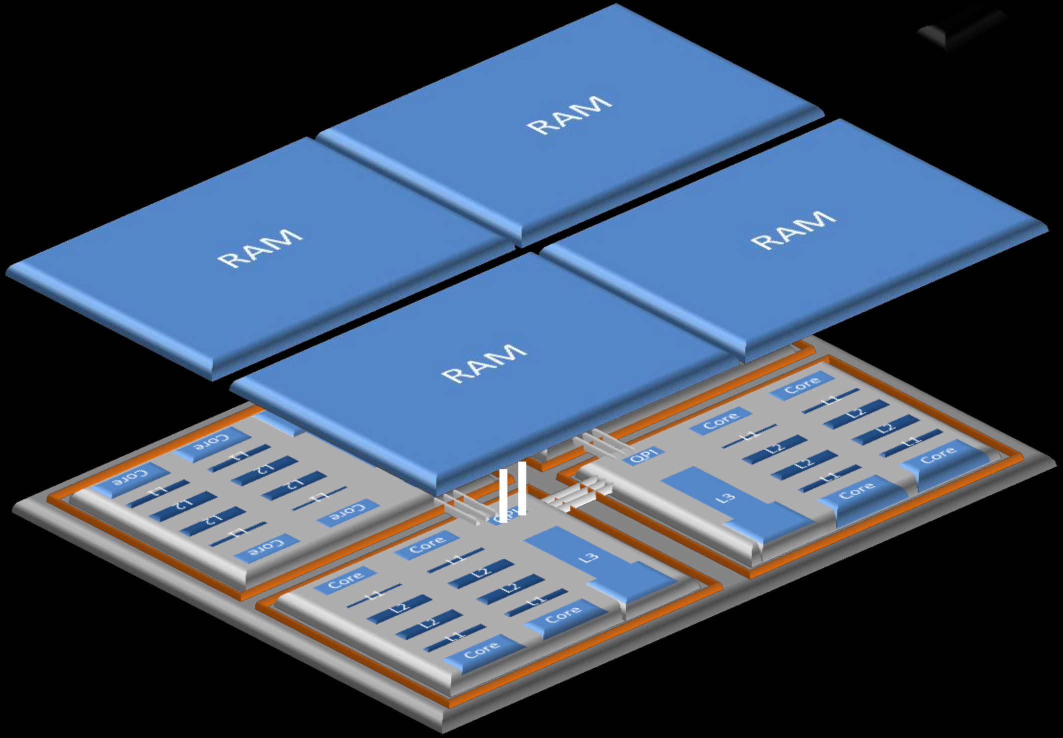
  line ∈ L1, L2 => line ∈ L3

- snoop filter:

  keep address not data

# Caches

[Corey: OSDI '08]



reduce coherent traffic by

keeping track of cached lines

- inclusive L3:

  line ∈ L1, L2 => line ∈ L3

- snoop filter:

  keep address not data

# RT - Locks for Clusters

Real-Time Resource-Sharing under Clustered Scheduling:
Mutex, Reader-Writer, and k-Exclusion Locks
Björn Brandenburg, Jim Anderson '11

hdwallpaperscool.com

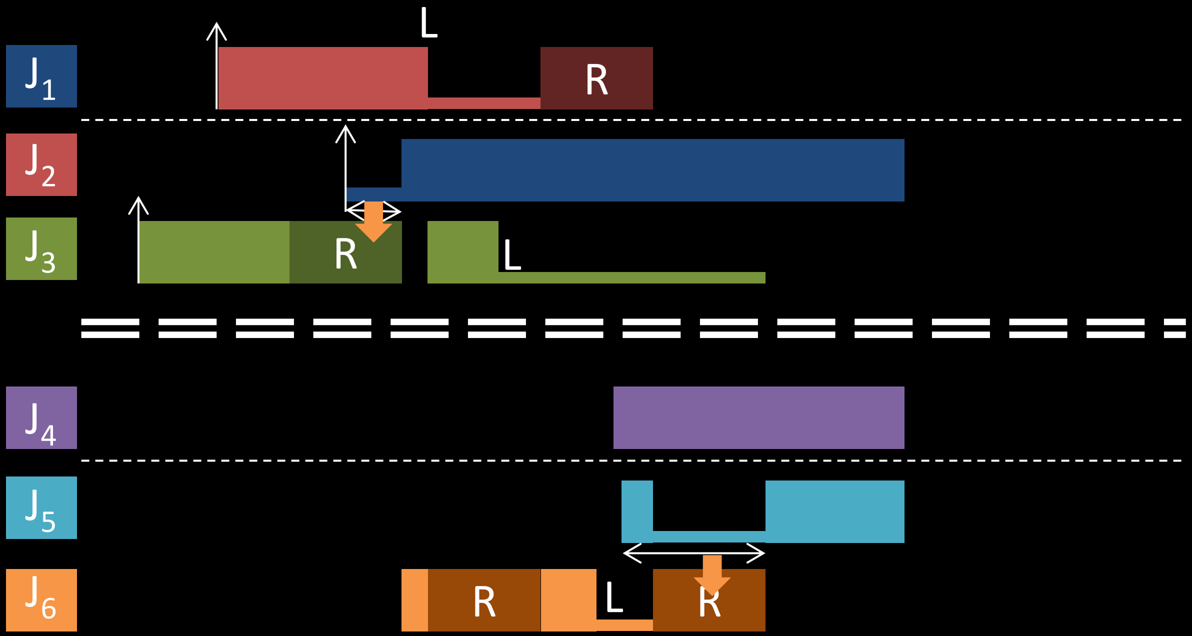
What if

- every task helps out a little bit when it is released?
- only the n-highest prioritized tasks may acquire locks?

# RT - Locks for Clusters

Real-Time Resource-Sharing under Clustered Scheduling:
Mutex, Reader-Writer, and k-Exclusion Locks
Björn Brandenburg, Jim Anderson ‘11

hdwallpaperscool.com



# RT - Locks for Clusters

Real-Time Resource-Sharing under Clustered Scheduling:
Mutex, Reader-Writer, and k-Exclusion Locks
Björn Brandenburg, Jim Anderson '11

hdwallpaperscool.com

D1. $J_i$ may issue a request only if it among the n = |CPU| highest prioritized jobs

D2. When released at time ta, Jd becomes Ji's priority donor if a) $J_i$ was the c = |Cluster| highest-priority pending job released prior to $J_d$, b) $J_d$ has one of the c highest base priorities, and c) $J_i$ has issued a request that is incomplete

D3. $J_i$ inherits $J_d$'s priority

D4. If $J_d$ ceases to be among the c highest-priority jobs in its cluster, the newly released job takes over the donor role

D5. / D6. schedule either $J_i$ or its donor $J_d$

D7. a priority donor may not issue requests ($J_d$ suspends in this case while being a priority donor)

D8. If $J_d$ finishes execution while being a donor, $J_d$ remains the donor but is suspended

D9. $J_d$ stops being a donor if a) $J_i$ completes its request, b) $J_i$ becomes one of the c highest prioritized jobs c) another job takes over being the donor of $J_i$

# Outline

Last Week:

- Terminology and Notations
- Anomalies and Impossibility Results
- Partitioned Scheduling
- Global Scheduling
- Optimal MP Scheduling
- Practical Matters

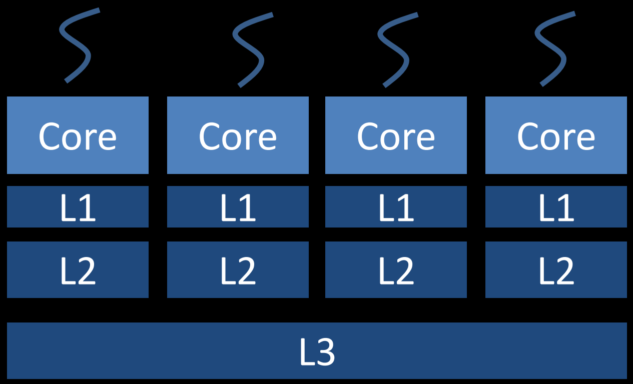
Today:

- Exercise: UP Resource Protocols
- Resources
  - Caches / Memories
- One More Resource
- Dynamic Tasks
- Peek and Poke into other Bleeding Edge Research

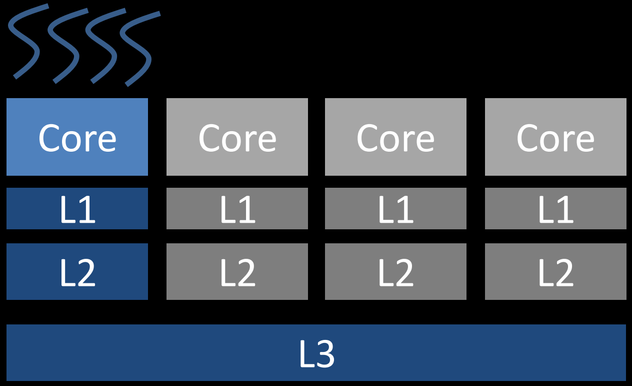
# Caches

A Coordinated Approach for Practical OS-Level Cache Management in Multi-Core Real-Time Systems
Hyoseung Kim, Arvind Kandhalu, Ragunathan (Raj) Rajkumar (ECRTS ’13)

hdwallpaperscool.com



40% slowdown



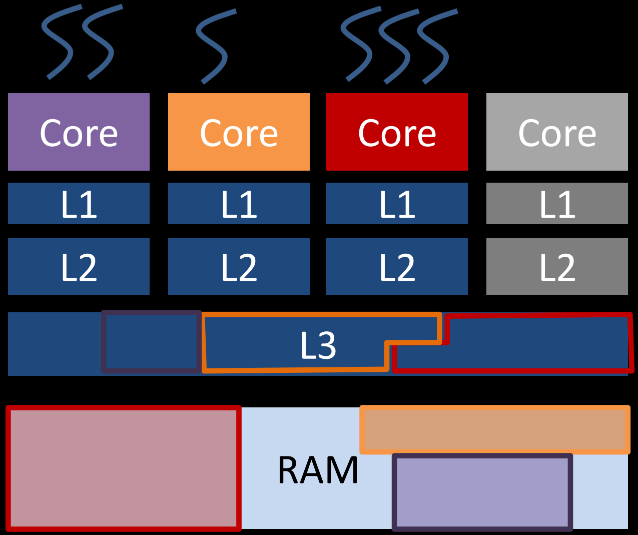
27% slowdown

PARSEC on i7

# Caches

A Coordinated Approach for Practical OS-Level Cache Management in Multi-Core Real-Time Systems
Hyoseung Kim, Arvind Kandhalu, Ragunathan (Raj) Rajkumar (ECRTS ’13)

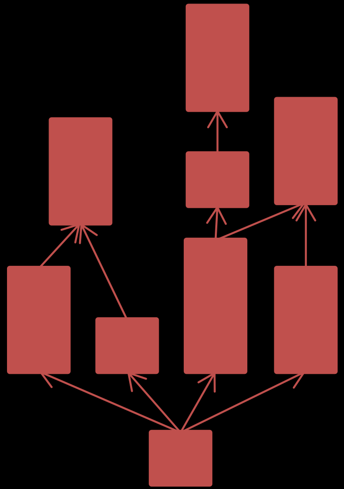
hdwallpaperscool.com

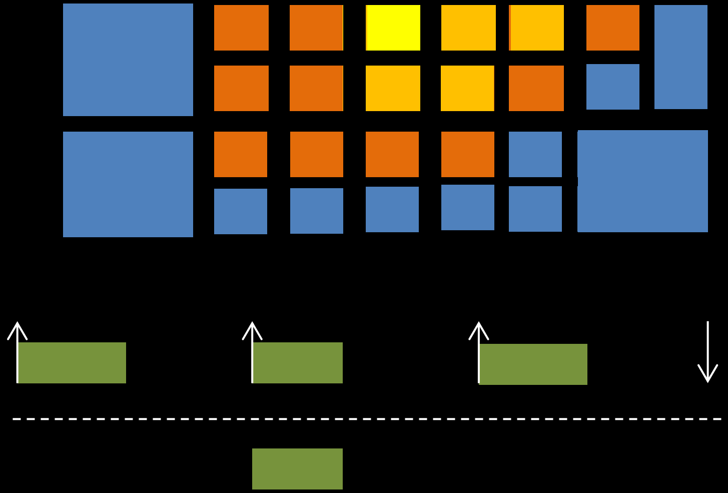


# Dynamic Tasks

Algorithm and complexity for the global scheduling of sporadic tasks on multiprocessors with work-limited parallelism
S. Collette, L. Cucu, J. Goossens [Real-Time and Network Systems, ’07]

hdwallpaperscool.com



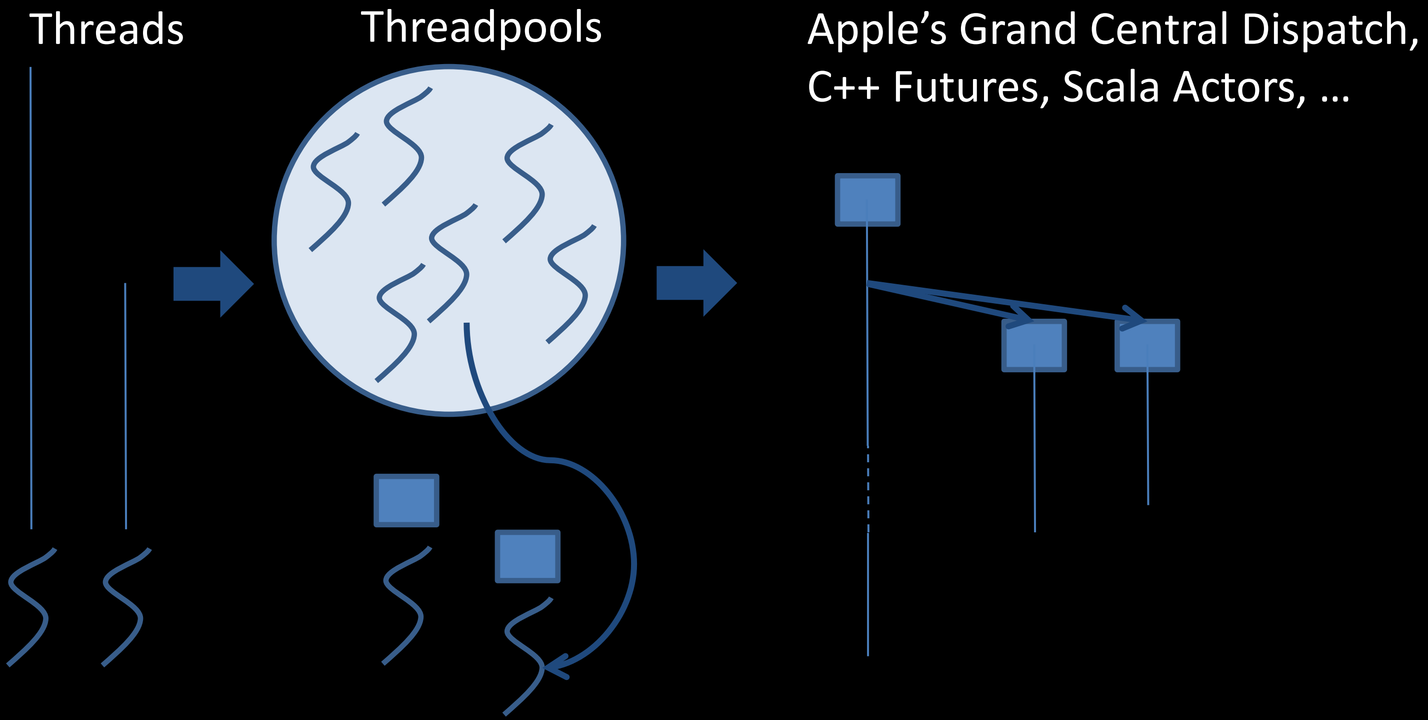


$J_i \overset{\text{def}}{=} (C_i, P_i, \Gamma_i)$

$\Gamma_i \overset{\text{def}}{=} (\gamma_{i,1}, \gamma_{i,2}, \dots, \gamma_{i,m})$

Job $J_i$ with j CPUs completes in $\gamma_{i,j}$ $C_i$

# Dynamic Tasks / Runtime



hdwallpaperscool.com

# Outline

Last Week:

- Terminology and Notations
- Anomalies and Impossibility Results
- Partitioned Scheduling
- Global Scheduling
- Optimal MP Scheduling
- Practical Matters

Today:

- Exercise: UP Resource Protocols
- Resources
  - Caches / Memories
- One More Resource
- Dynamic Tasks
- Peek and Poke into other Bleeding Edge Research

# References

- **[Burns '09]**
  **R. Davis, A. Burns, "A Survey of Hard Real-Time Scheduling Algorithms and Schedulability Analysis Techniques for Multiprocessor Systems", 2009**

- [Colette]
  S. Collette, L. Cucu, J. Goossens, "Algorithm and complexity for the global scheduling of sporadic tasks on multiprocessors with work-limited parallelism", Int. Conf. On Real-Time and Network Systems, 2007

- [Edmonds]
  J. Edmonds, K. Pruhs, "Scalably scheduling processes with arbitrary speedup curves", Symp. On Discrete Algorithms, 2009

- [Levin '10]
  G. Levin, S. Funk, C.Sadowski, I. Pye, S. Brandt, "DP-Fair: A Simple Model for Understanding Optimal Multiprocessor Scheduling", 2010

- [Hong '88]
  K. Hong, J. Leung, "On-line scheduling of real-time tasks", RTSS, 1988

- [Anderson '01]
  B. Anderson, S. Baruah, J. Jonsson, "Static-priority scheduling on multiprocessors", RTSS, 2001

- [Lakshmanan '09]
  K. Lakshmanan, R. Rajkumar, J. Lehoczky, "Partitioned Fixed-Priority Preemptive Scheduling for Multi-Core Processors", ECRTS, 2009

# References

- [Fischer '07]

  N. Fischer, “The multiprocessor real-time scheduling of general task systems”, PhD. Thesis, University of North Carolina at Chapel Hill, 2007

- [Dertouzos '89]
  M. Dertouzos, A. Mok, “Multiprocessor scheduling in a hard real-time environment”, IEEE Trans. on Software Engineering, 1989

- [Dhall]
  S. Dhall, C, Liu, “On a Real-Time Scheduling Problem”, Operations Research, 1978

- [Baruah '06]
  S. Baruah, A. Burns, “Sustainable Scheduling Analysis”, RTSS, 2006

- [Corey '08]
  S. Boyd-Wickizer, H. Chen, R. Chen, Y. Mao, F. Kaashoek, R. Morris, A. Pesterev, L. Stein, M. Wu, Y. Dai, Y. Zhang, Z. Zhang, “Corey: An Operating System for Many Cores”, OSDI 2008

- [Carpenter '04]

  J. Carpenter, S. Funk, P. Holman, A. Srinivasan, J. Anderson, S. Baruah, “A categorization of real-time multiprocessor scheduling problems and algorithms”, Handbook of Scheduling: Algorithms, Models, and Performance Analysis, 2004

- [Whitehead]

  J. Leung, J. Whitehead, “On the Complexity of Fixed-Priority Scheduling of Periodic, Real-Time Tasks”,1982

- [Brandenburg '08]

  B. Brandenburg, J. Calandrino, A. Block, H. Leontyev, J. Anderson, “Real-Time Synchronization on Multiprocessors: To Block or Not to Block, to Suspend or Spin”, RTAS, 2008

# References

- [Gai '03]

  P. Gai, M. Natale, G. Lipari, A. Ferrari, C. Gabellini, P. Marceca, “A comparison of MPCP and MSRP when sharing resources in the Janus multiple-processor on a chip platform”, RTAS, 2003

- [Block]

  A. Block, H. Leontyev, B. Brandenburg, J. Anderson, “A Flexible Real-Time Locking Protocol for Multiprocessors”, RTCSA, 2007

- [MSRP]

  P. Gai, G. Lipari, M. Natale, “Minimizing memory utilization of real-time task sets in single and multi-processor systems-on-a-chip”, RTSS, 2001

- [MPCP]

  R. Rajkumar, L. Sha, J. Lehoczky, “Real-time synchronization protocols for multiprocessors”, RTSS, 1988

- [Lauzac '98]

  S. Lauzac, R. Melhem, D. Mosse, “Comparison of global and partitioning schemes for scheduling rate monotonic tasks on a multiprocessor”, EuroMicro, 1998

- [Baruah '96]

  S. Baruah, N. Cohen, G. Plaxton, D. Varvel, “Proportionate progress: A notion of fairness in resource allocation”, Algorithmica 15, 1996

- [Brandenburg'11]

  B.Brandenburg, J. Anderson, “Real-Time Resource-Sharing under Clustered Scheduling: Mutex, Reader-Writer, and k-Exclusion Locks”, ESORICS 2011

# References

- [Brandenburg '13]

  B. Brandenburg, “Improved Analysis and Evaluation of Real-Time Semaphore Protocols for Partitioned Fixed Priority Scheduling”, RTAS, 2013

- [Brandenburg ‘10]

  B. Brandenburg, J. Anderson, “Optimality Results for Multiprocessor Real-Time Locking”, RTSS, 2010

- [Lauzac '98]

  S. Lauzac, R. Melhem, D. Mosse, “Comparison of global and