Carnegie
Mellon
University

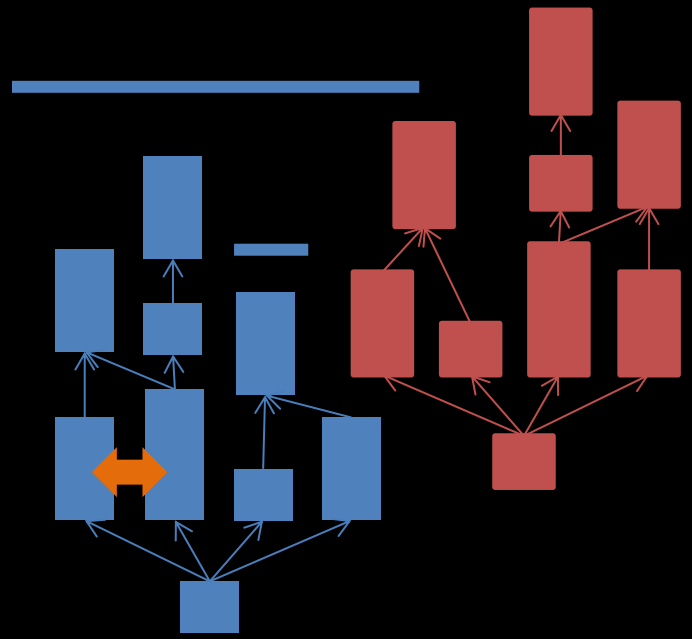
cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

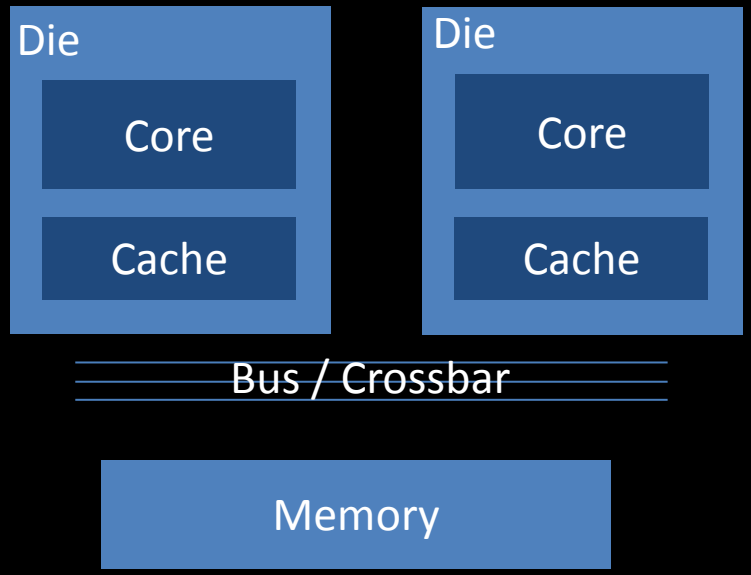
Hard Real-Time Multiprocessor Scheduling

Marcus Völp

TECHNISCHE UNIVERSITÄT DRESDEN

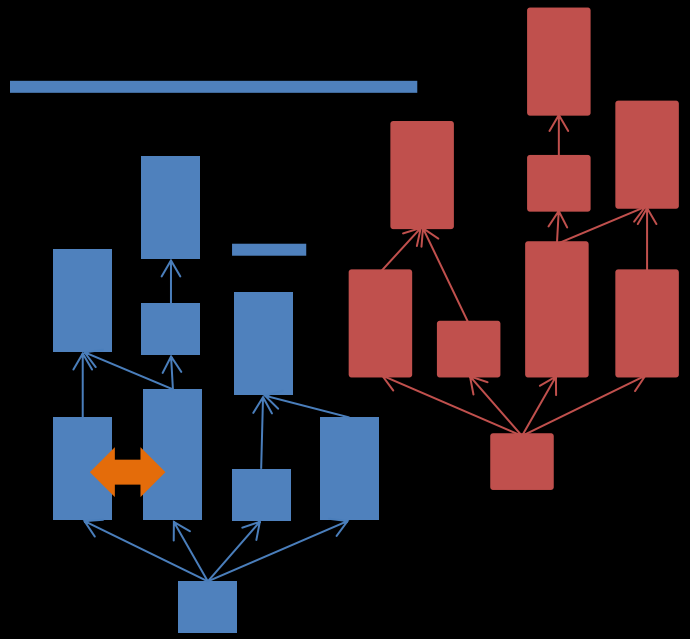
The Challenge

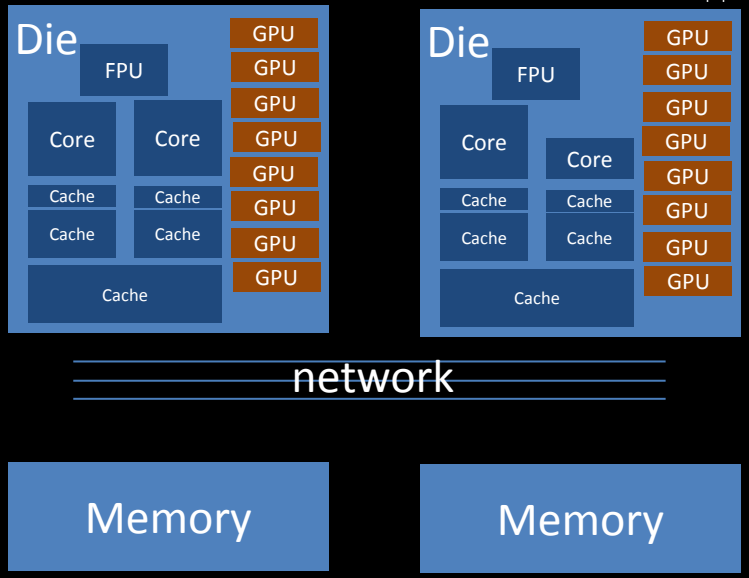




The Challenge

hdwallpaperscool.com





Outline

Today:

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

Next week:

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

Outline

Today:

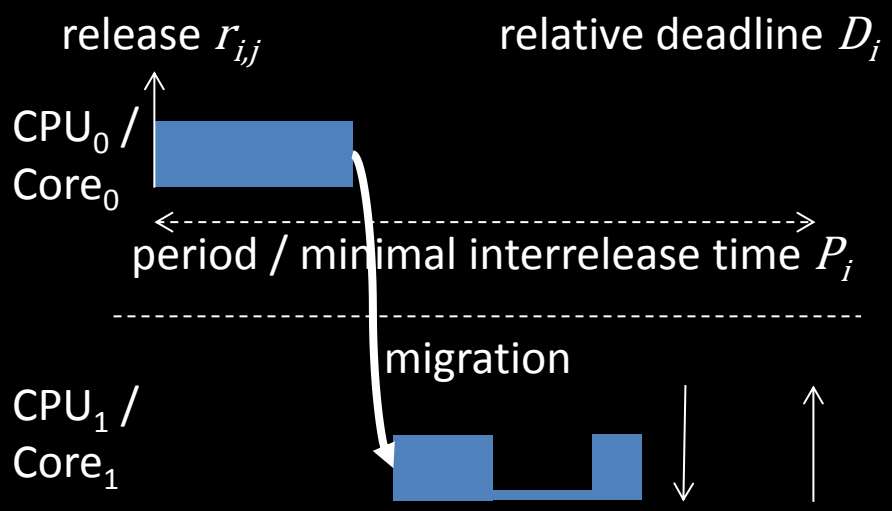
- Terminology and Notations
- Anomalies and
  I
- P
- Global Scheduling
- Optimal MP Scheduling
- Practical Matters

Next week:

- Exercise:
  UP Resource Protocols
  Dynamic Tasks
- Peek and Poke into other
  Bleeding Edge Research

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

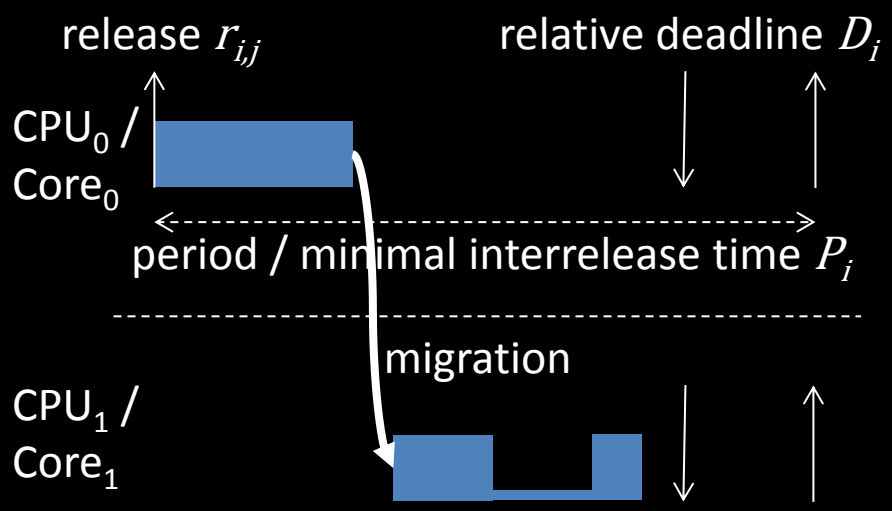
Terminology



- number of priorities
  - FTP: fixed task priority
  - FJP: fixed job priority
  - Dynamic: priorities may change during execution
- migration
  - no migration (partitioned)
  - task-level (at job boundaries)
  - job-level (arbitrary)

global

Terminology



- deadlines
  - implicit: $D_i = P_i$
  - constrained: $D_i \leq P_i$
  - arbitrary
- metrics
  - utility $U_i = \frac{C_i}{P_i}$
  - density $\partial_i = \frac{C_i}{min(D_i, P_i)}$

Lessons Learned from UP

Fixed Task Priority:

- Rate monotonic / Deadline monotonic
- Liu-Layland Criterion $U_{RMS}(n) = n\sqrt[n]{2} - 1 \leq 0.693$

Fixed Job Priority:

- EDF is optimal

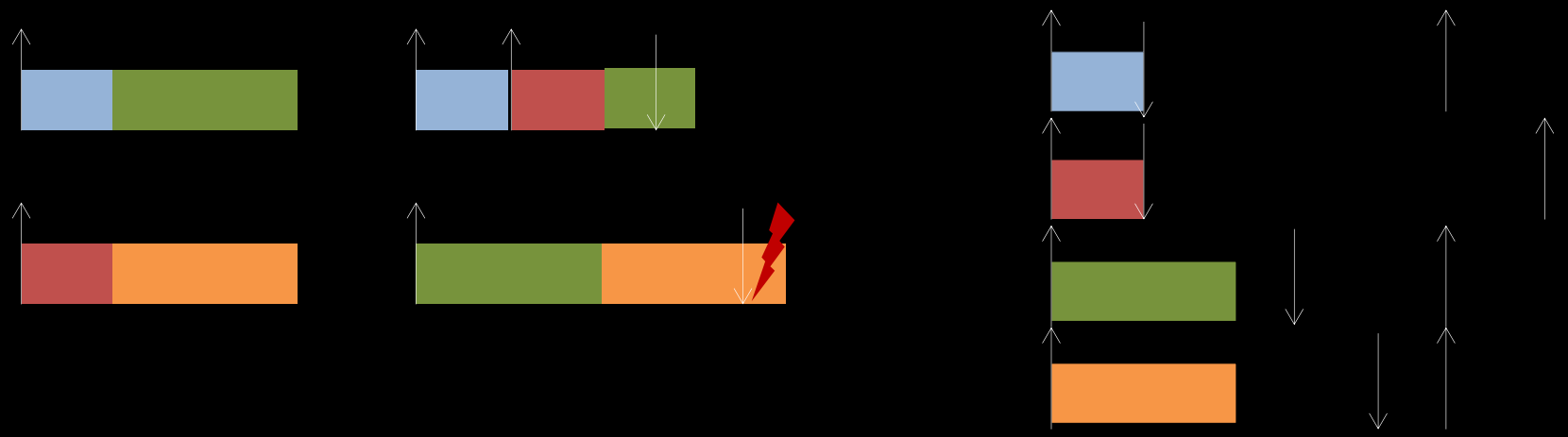
Simultaneous release is critical instance:

- Simultaneous arrival sequence (sporadic)
- Response Time Analysis

Response times depend on <u>set</u> but not on the order of high-priority tasks

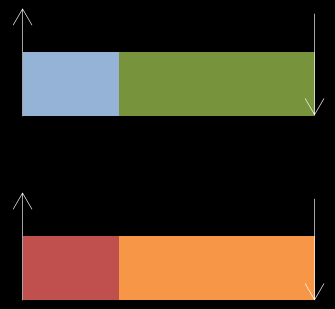
Anomalies

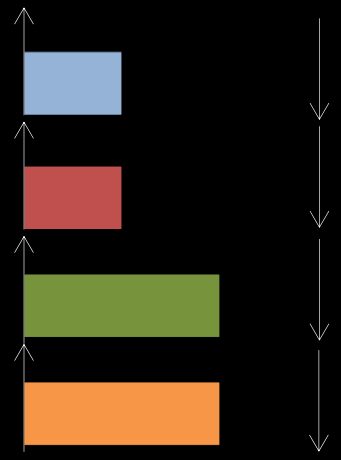
Simultaneous release is not a critical instance [Lauzac ’98]



Anomalies

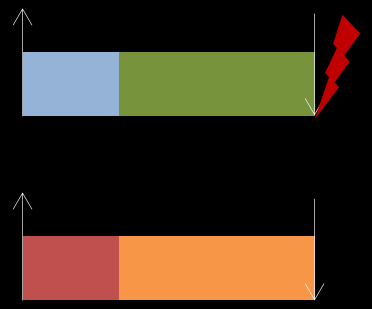
Response time depends on priority ordering of higher priority threads

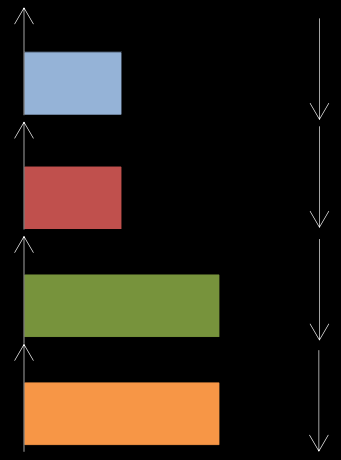




Anomalies

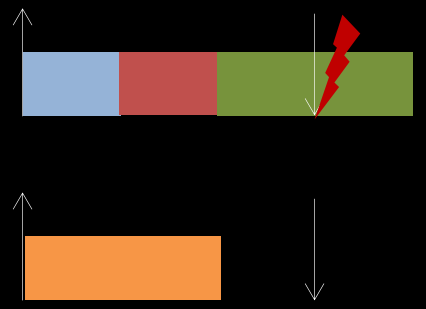
Response time depends on priority ordering of higher priority threads

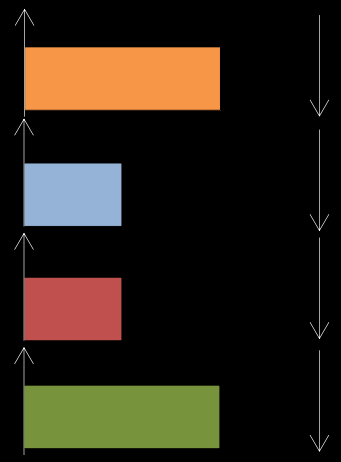




Anomalies

Response time depends on priority ordering of higher prioritized threads





Sustainability

How robust is the schedule against parameter changes?

- decrease execution time of task (predictability)
  - otherwise, WCET won't work for admission
- increase minimal interrelease time
  - otherwise, assuming more frequent releases is no safe approximation
- increase relative deadline
  - otherwise, assuming earlier deadlines is not safe

Global FTP and Global EDF are not sustainable if no. of CPUs > 1

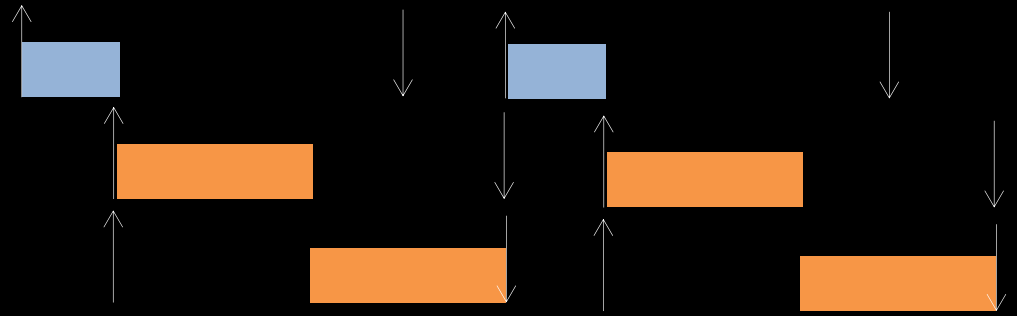
All pre-emptive FJP / FTP algorithms are predictable

Mission Impossible

[Hong '88] No optimal online multiprocessor scheduling algorithm for arbitrary jobs unless all jobs have the same deadlines.

[Dertouzos '89] This holds even if execution times are known precisely.

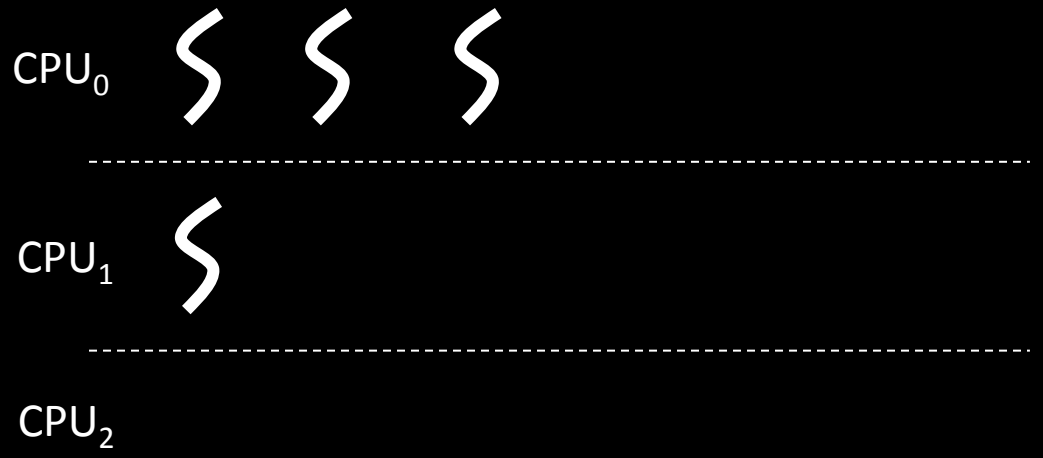
[Fisher '07] No optimal online scheduling algorithm for sporadic tasksets with constrained or arbitrary deadlines.



Partitioned vs. Global

Partitioned Scheduling

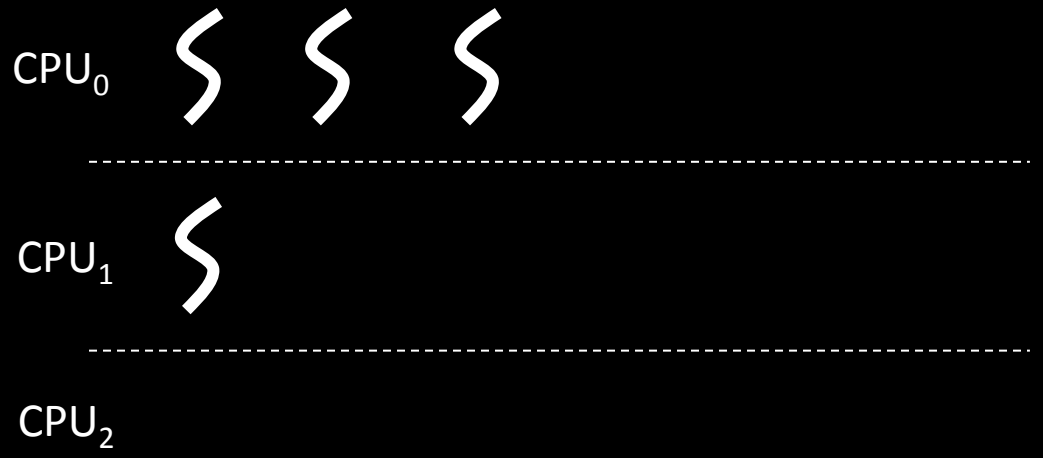
- split workload into chunks of total utilization $U \leq n\sqrt[n]{2} - 1 \leq 0.693$
- reuse uniprocessor scheduler $U \leq 1$



Partitioned vs. Global

Partitioned Scheduling

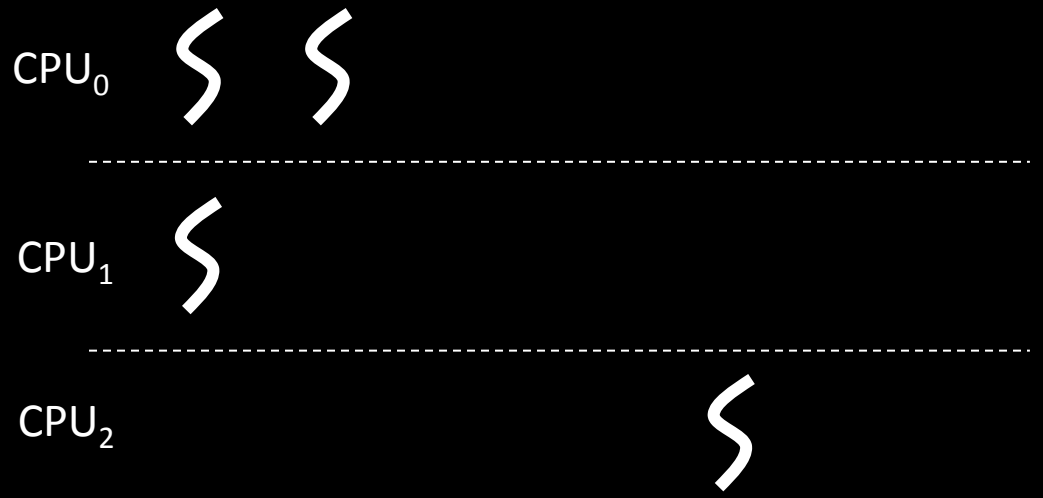
- split workload into chunks of total utilization $U \leq n\sqrt[n]{2} - 1 \leq 0.693$
- reuse uniprocessor scheduler $U \leq 1$



Partitioned vs. Global

Partitioned Scheduling

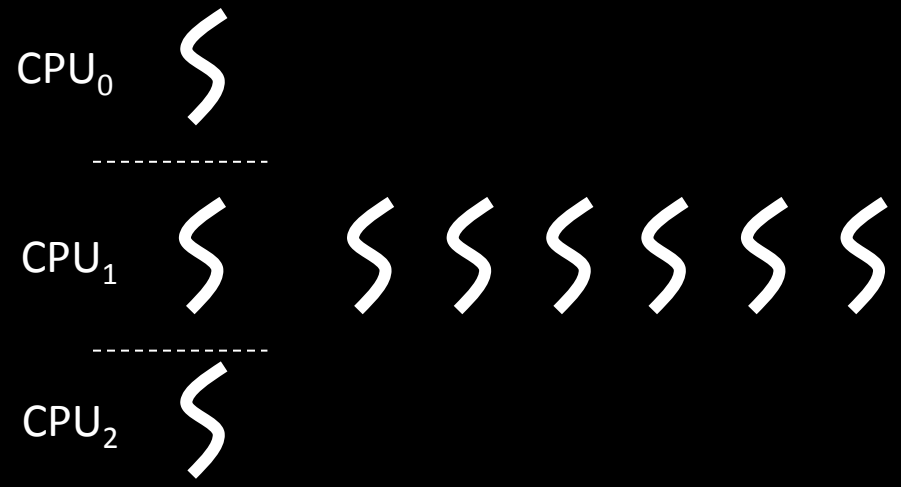
- split workload into chunks of total utilization $U \leq n\sqrt[n]{2} - 1 \leq 0.693$
- reuse uniprocessor scheduler $U \leq 1$



Partitioned vs. Global

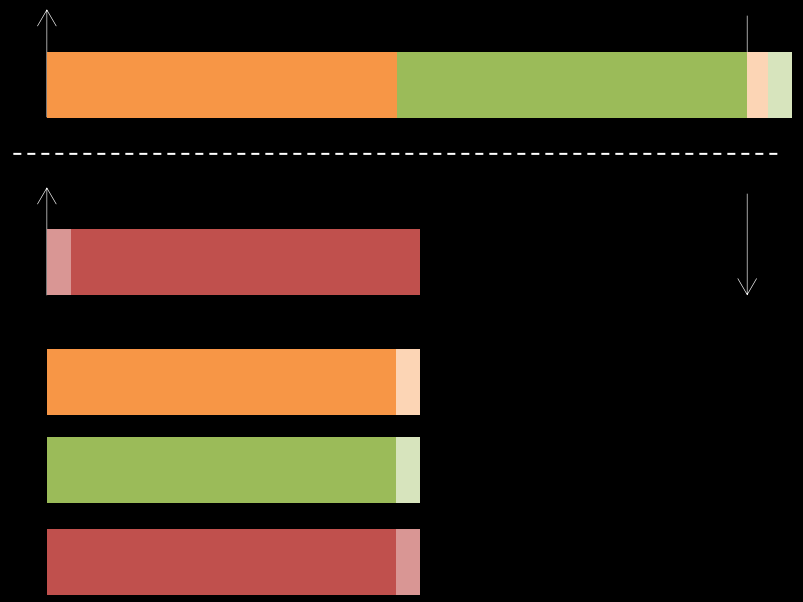
Global Scheduling

- just one ready queue



Partitioned Scheduling

[Anderson '01]



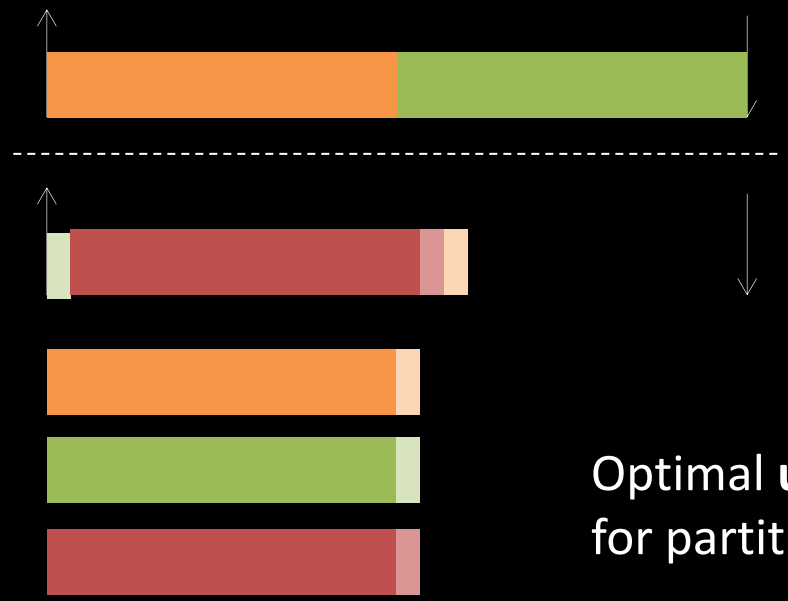
m CPUs, m+1 Tasks

Period $P_i = 2$

WCET $C_i = 1 + \varepsilon$

Partitioned Scheduling

[Anderson '01]



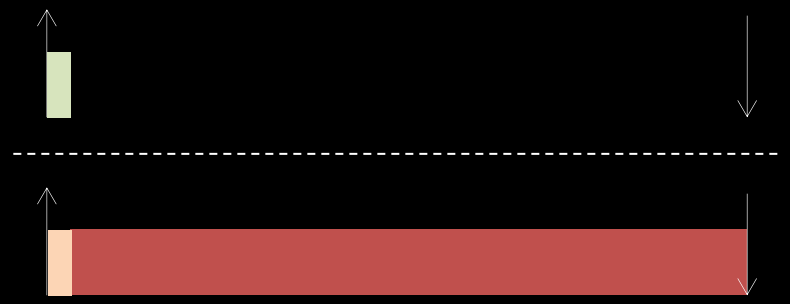
m CPUs, m+1 Tasks

Period $P_i = 2$
WCET $C_i = 1 + \varepsilon$

Optimal **utilization bound** for partitioned schedulers:

$$U_{opt} = \frac{m+1}{2}$$

Global EDF



m+1 CPUs

m+1 Tasks with

Period $P_i = 1$, $U_i = \varepsilon$

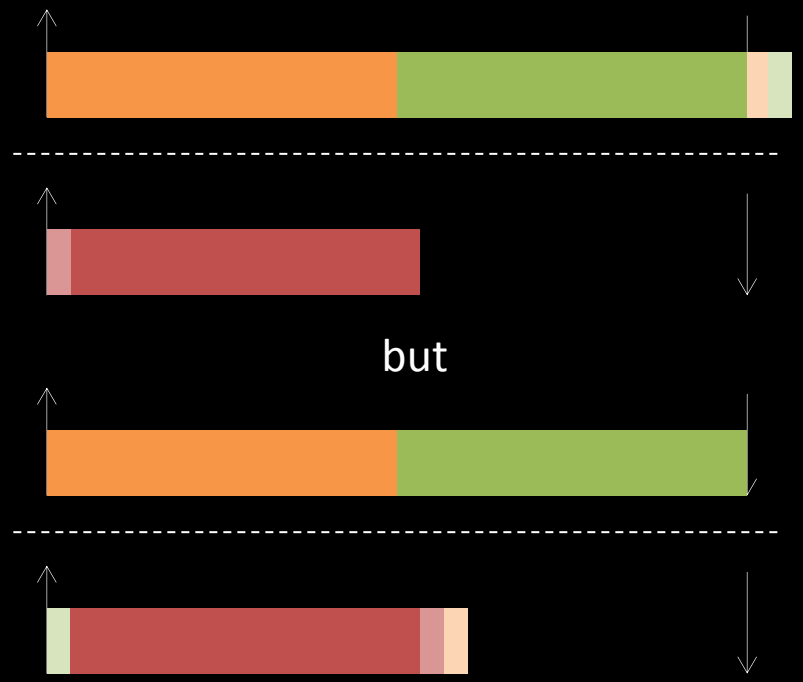
1 Task with

Period $P_0 = 2$, $U_0 > (2 - \varepsilon) / 2$

Utilization bound: $U_{GEDF} = 1 + m\varepsilon$

Dhall Effect does not occur if Ui < 41%

Global Scheduling



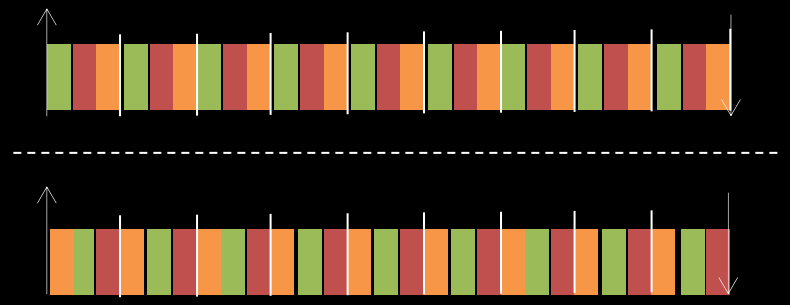
m CPUs, m+1 Tasks

Period $P_i = 2$
WCET $C_i = 1 + \varepsilon$

Optimal **utilization bound** for global schedulers: $U_{opt} = \frac{m+1}{2}$

$U_{opt} = m$?

PFAIR [Baruah '96]



Divide timeline into quanta q of length t. At each quanta, allocate tasks such that the accumulated processor time is either $\lfloor tu_i \rfloor$ or $\lceil tu_i \rceil$.

Optimal **utilization bound** for PFAIR schedulers and periodic implicit deadline constraint tasksets: $U_{opt} = \mathrm{m}$

Very high preemption and migration costs.

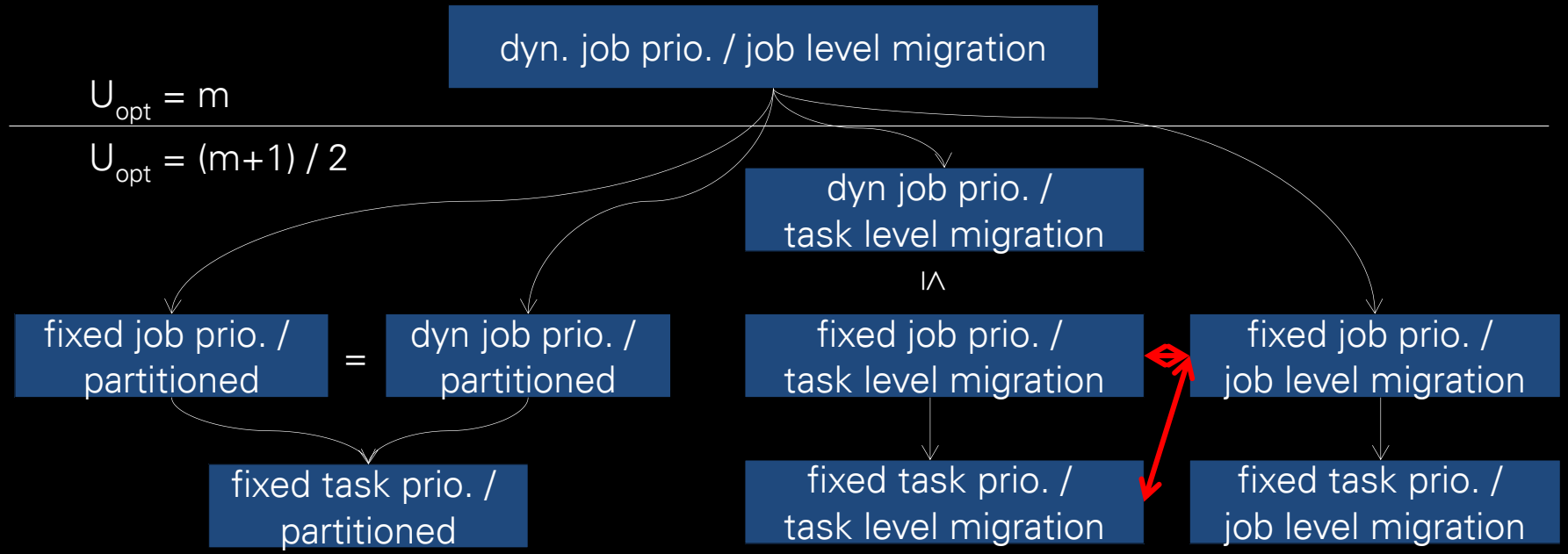
Design Space

dyn job prio. / partitioned	dyn job prio. / task level migration	dyn. job prio. / job level migration
fixed job prio. / partitioned	fixed job prio. / task level migration	fixed job prio. / job level migration
fixed task prio. / partitioned	fixed task prio. / task level migration	fixed task prio. / job level migration

more dynamic priorities

more dynamic migration

Design Space



A → B => A can schedule any taskset that B can schedule and more

A ↔ B => dominance is not yet known

PDMS-HPTS-DS

[Lakshmanan '09]

Can we improve on Anderson's utilization bound by migration some (few) jobs?

- PDMS partitioned deadline monotonic scheduling
- HPTS highest priority task split
- DS allocate tasks according to highest density first

PDMS-HPTS-DS

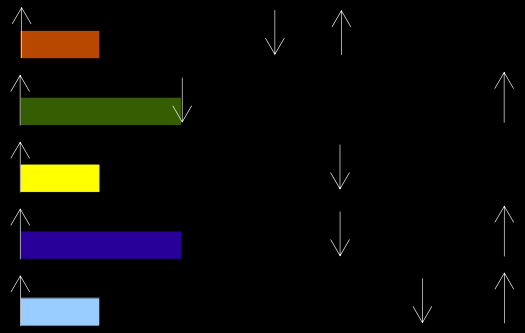
τ_1 = (4,3,1)

τ_2 = (6,2,2)

τ_3 = (4,4,1)

τ_4 = (6,4,2)

τ_5 = (6,5,1)



$u_1 = 0.25$ $\delta_1 = 1/3 = 0.33$

$u_2 = 0.33$ $\delta_2 = 1$

$u_3 = 0.25$ $\delta_3 = 1/4 = 0.25$

$u_4 = 0.33$ $\delta_4 = 1/2 = 0.5$

$u_5 = 0.16$ $\delta_5 = 1/5 = 0.2$

$u_{sum} = 1.33$ => $u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$

CPU_0

CPU_1

PDMS-HPTS-DS

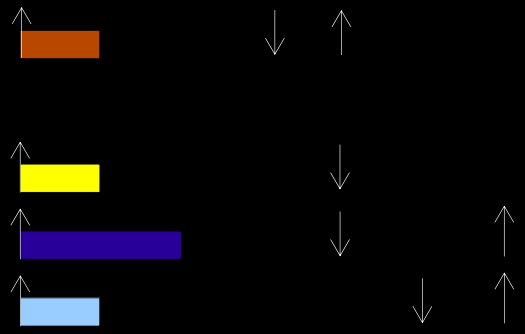
τ_1 = (4,3,1)

τ_2 = (6,2,2)

τ_3 = (4,4,1)

τ_4 = (6,4,2)

τ_5 = (6,5,1)



$u_1 = 0.25$ $\delta_1 = 1/3 = 0.33$

$u_2 = 0.33$ $\delta_2 = 1$

$u_3 = 0.25$ $\delta_3 = 1/4 = 0.25$

$u_4 = 0.33$ $\delta_4 = 1/2 = 0.5$

$u_5 = 0.16$ $\delta_5 = 1/5 = 0.2$

$u_{sum} = 1.33$ => $u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$

CPU_0

CPU_1

PDMS-HPTS-DS

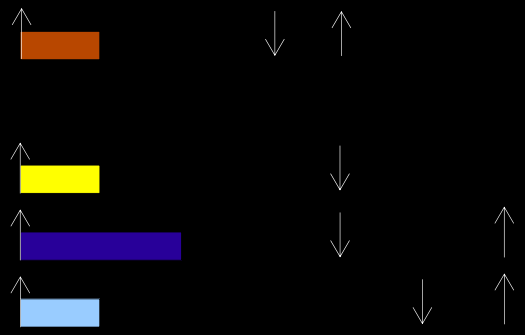
τ_1 = (4,3,1)

τ_2 = (6,2,2)

τ_3 = (4,4,1)

τ_4 = (6,4,2)

τ_5 = (6,5,1)



$u_1 = 0.25$	$\delta_1 = 1/3 = 0.33$
$u_2 = 0.33$	$\delta_2 = 1$
$u_3 = 0.25$	$\delta_3 = 1/4 = 0.25$
$u_4 = 0.33$	$\delta_4 = 1/2 = 0.5$
$u_5 = 0.16$	$\delta_5 = 1/5 = 0.2$

$u_{sum} = 1.33 \Rightarrow u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$

CPU_0

CPU_1

PDMS-HPTS-DS

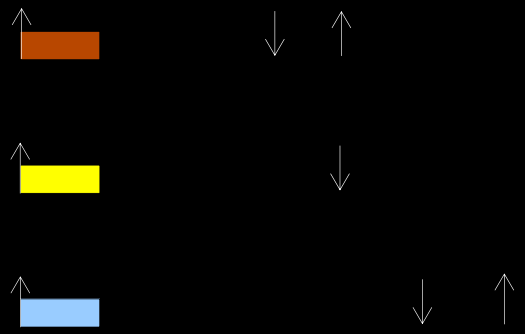
τ_1 = (4,3,1)

τ_2 = (6,2,2)

τ_3 = (4,4,1)

τ_4 = (6,4,2)

τ_5 = (6,5,1)

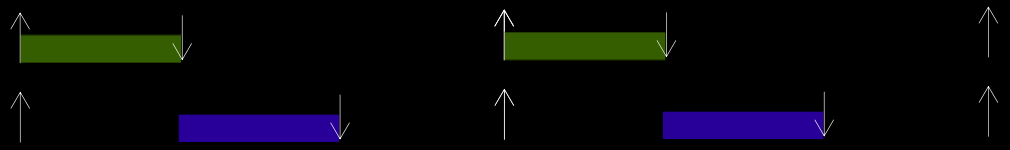


$u_1 = 0.25$	$\delta_1 = 1/3 = 0.33$
$u_2 = 0.33$	$\delta_2 = 1$
$u_3 = 0.25$	$\delta_3 = 1/4 = 0.25$
$u_4 = 0.33$	$\delta_4 = 1/2 = 0.5$
$u_5 = 0.16$	$\delta_5 = 1/5 = 0.2$

$u_{sum} = 1.33 \Rightarrow u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$

CPU_0



CPU_1

PDMS-HPTS-DS

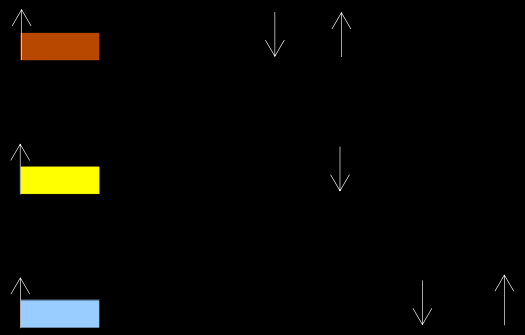
τ_1 = (4,3,1)

τ_2 = (6,2,2)

τ_3 = (4,4,1)

τ_4 = (6,4,2)

τ_5 = (6,5,1)

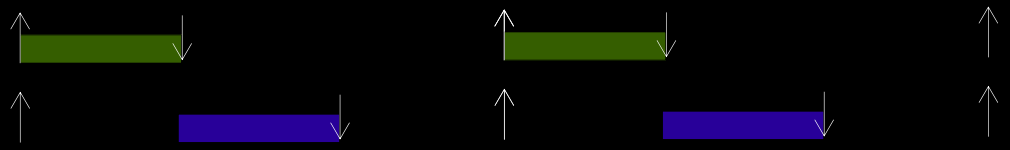


$u_1 = 0.25$	$\delta_1 = 1/3 = 0.33$
$u_2 = 0.33$	$\delta_2 = 1$
$u_3 = 0.25$	$\delta_3 = 1/4 = 0.25$
$u_4 = 0.33$	$\delta_4 = 1/2 = 0.5$
$u_5 = 0.16$	$\delta_5 = 1/5 = 0.2$

$u_{sum} = 1.33 \Rightarrow u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$

CPU_0



CPU_1

PDMS-HPTS-DS

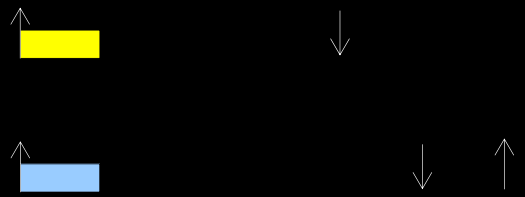
τ_1 = (4,3,1)

τ_2 = (6,2,2)

τ_3 = (4,4,1)

τ_4 = (6,4,2)

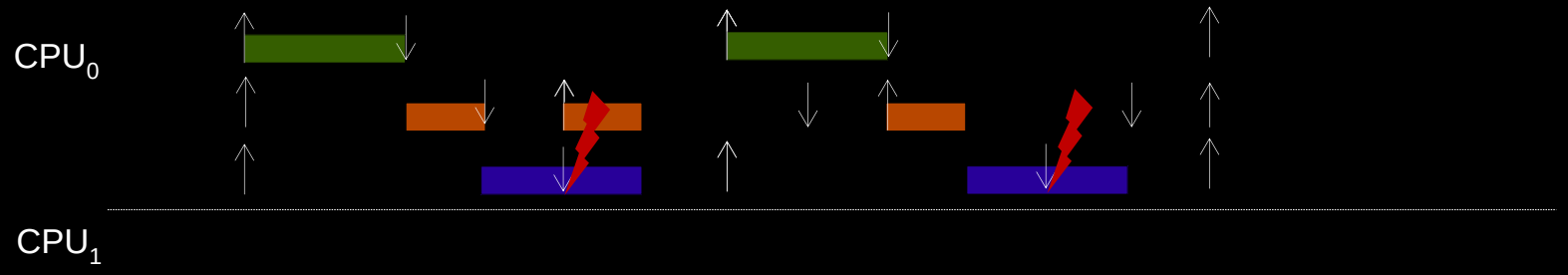
τ_5 = (6,5,1)



$u_1 = 0.25$	$\delta_1 = 1/3 = 0.33$
$u_2 = 0.33$	$\delta_2 = 1$
$u_3 = 0.25$	$\delta_3 = 1/4 = 0.25$
$u_4 = 0.33$	$\delta_4 = 1/2 = 0.5$
$u_5 = 0.16$	$\delta_5 = 1/5 = 0.2$

$u_{sum} = 1.33$ => $u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$



PDMS-HPTS-DS

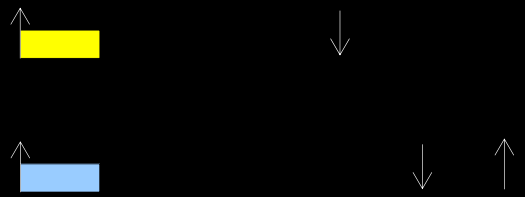
τ_1 = (4,3,1)

τ_2 = (6,2,2)

τ_3 = (4,4,1)

τ_4 = (6,4,2)

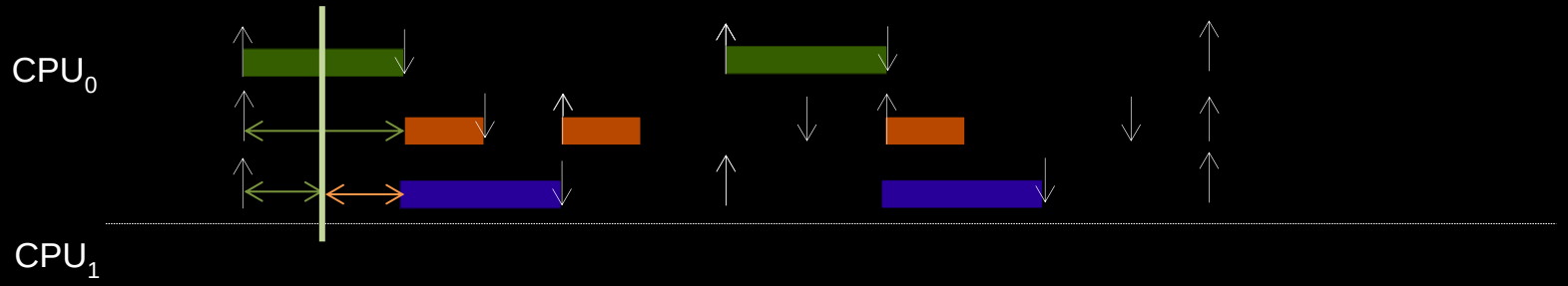
τ_5 = (6,5,1)



$u_1 = 0.25$	$\delta_1 = 1/3 = 0.33$
$u_2 = 0.33$	$\delta_2 = 1$
$u_3 = 0.25$	$\delta_3 = 1/4 = 0.25$
$u_4 = 0.33$	$\delta_4 = 1/2 = 0.5$
$u_5 = 0.16$	$\delta_5 = 1/5 = 0.2$

$u_{sum} = 1.33$ => $u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$



PDMS-HPTS-DS

$\tau_1 = (4,3,1)$

$\tau_2 = (6,2,2)$

$\tau_3 = (4,4,1)$

$\tau_4 = (6,4,2)$

$\tau_5 = (6,5,1)$

$u_1 = 0.25 \quad \delta_1 = 1/3 = 0.33$

$u_2 = 0.33 \quad \delta_2 = 1$

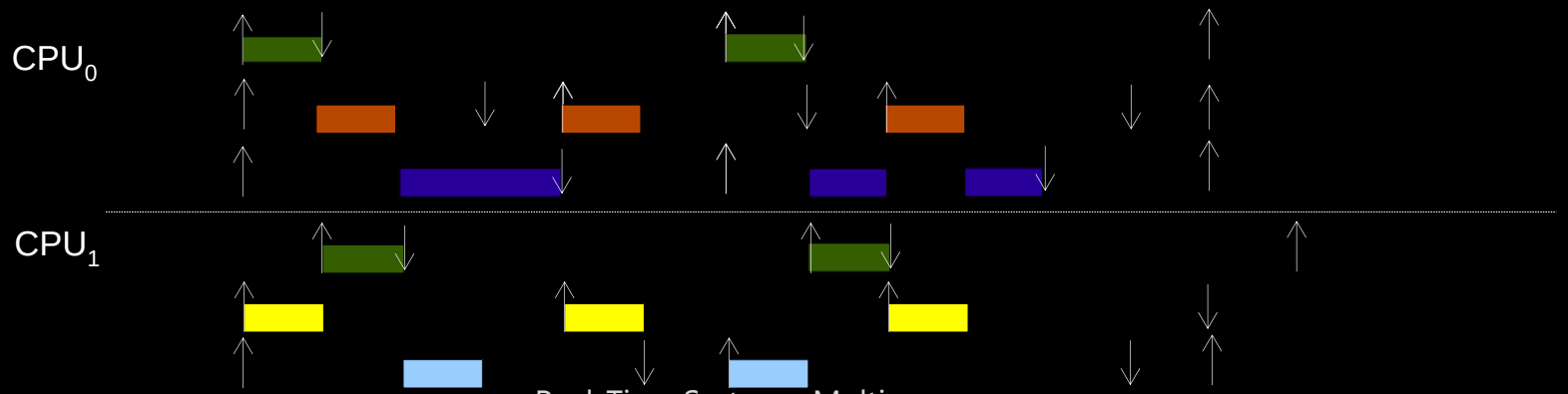
$u_3 = 0.25 \quad \delta_3 = 1/4 = 0.25$

$u_4 = 0.33 \quad \delta_4 = 1/2 = 0.5$

$u_5 = 0.16 \quad \delta_5 = 1/5 = 0.2$

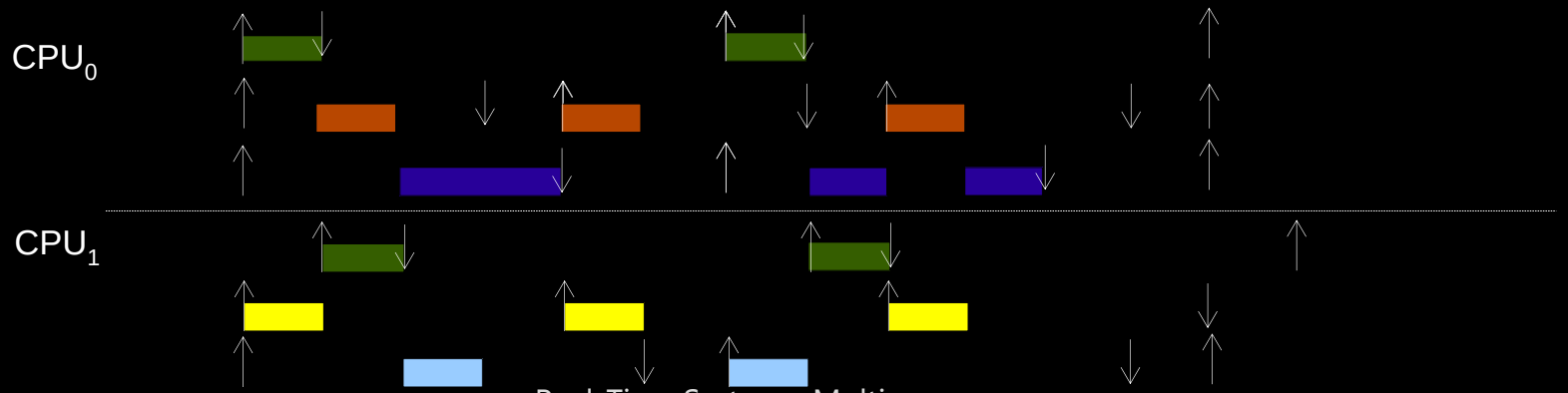
$u_{sum} = 1.33 \Rightarrow u_{sum} / 2 = 0.66\%$

$$\partial_i = \frac{C_i}{min(D_i, P_i)}$$



PDMS-HPTS-DS

$U_{PDMS\text{-}HPTS\text{-}DS}$ = m 69.3 % if all tasks have a utilization U_i < 41%

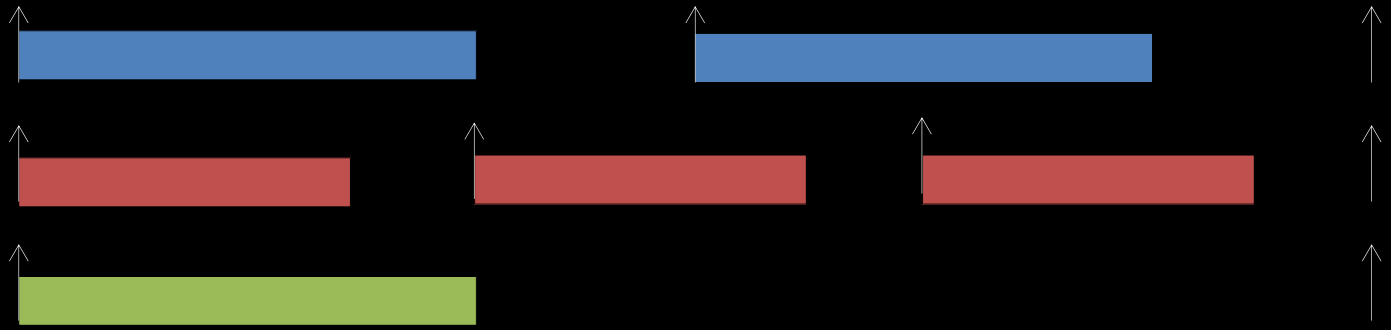


DP-FAIR [Levin '10]

impossibility result [Hong '88]:
no optimal MP scheduling algorithm if ~~not all tasks have the same deadline~~.


hdwallpaperscool.com

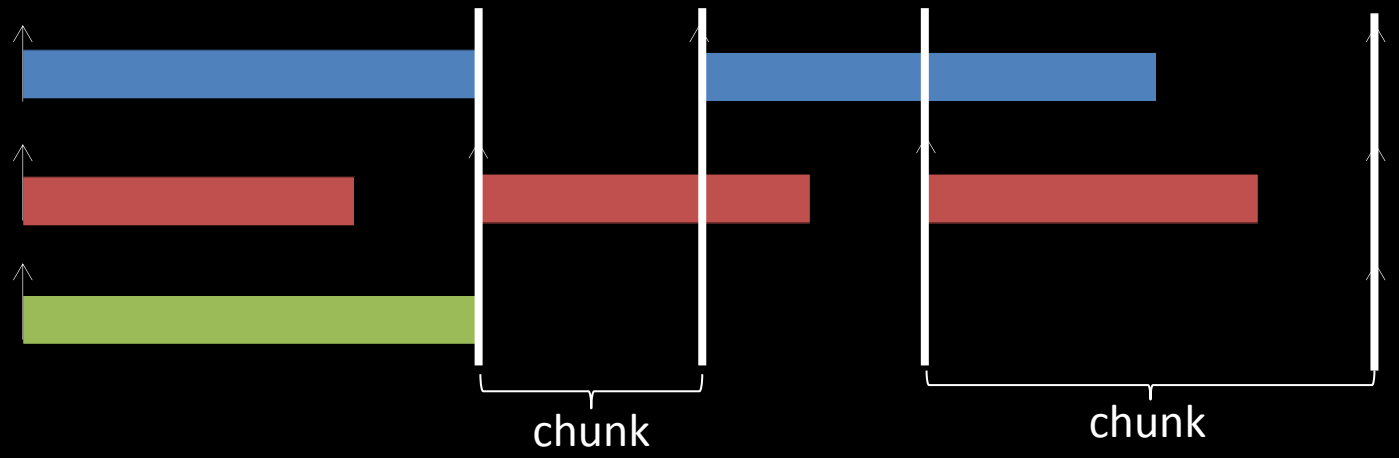
- deadline partitioning



DP-FAIR

impossibility result [Hong '88]:
no optimal MP scheduling algorithm if ~~not all tasks have the same deadline~~.

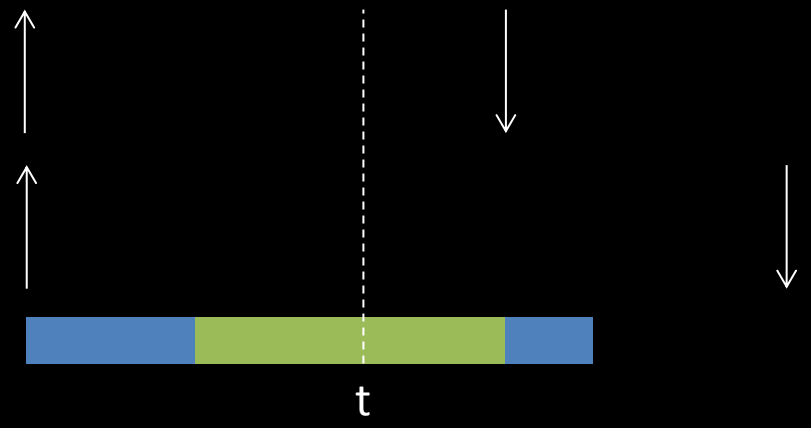
- deadline partitioning



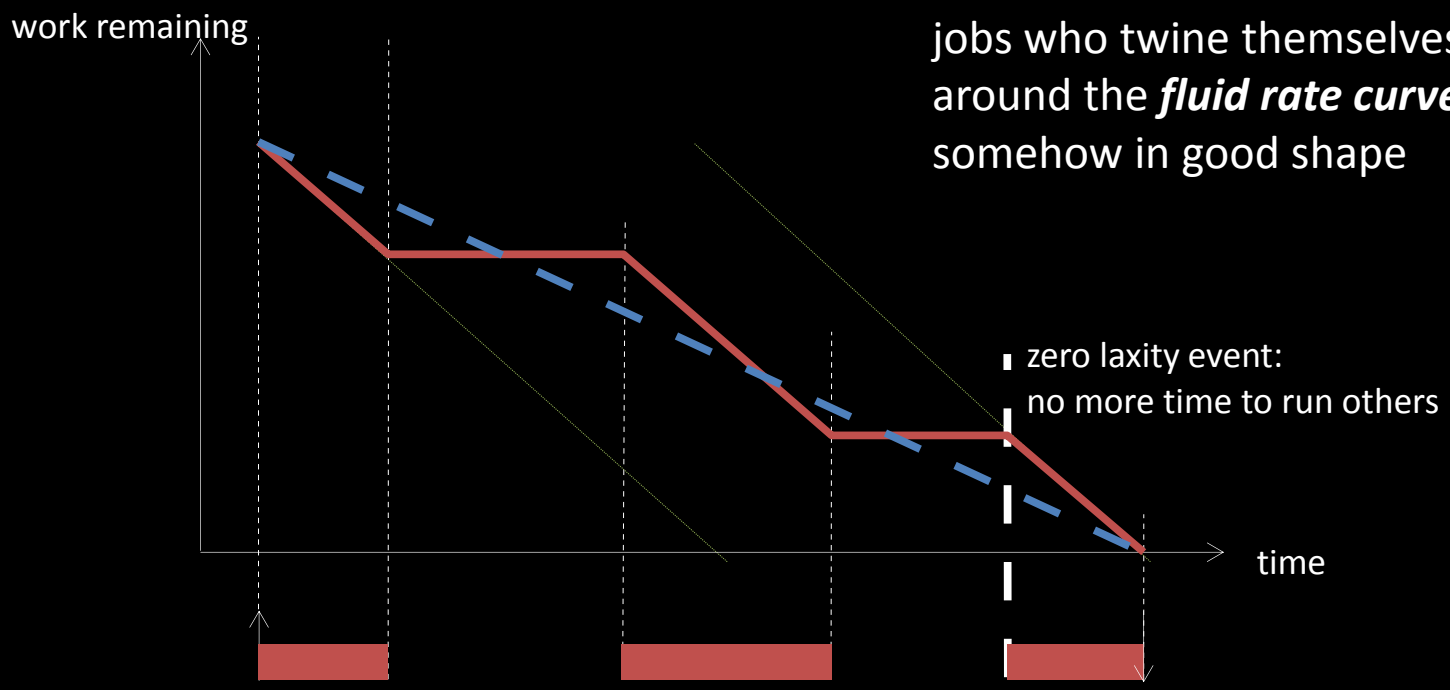
DP-FAIR

impossibility result [Hong '88]:
no optimal MP scheduling algorithm if ~~not all tasks have the same deadline~~.

- deadline partitioning
- zero laxity
- fluid rate curve

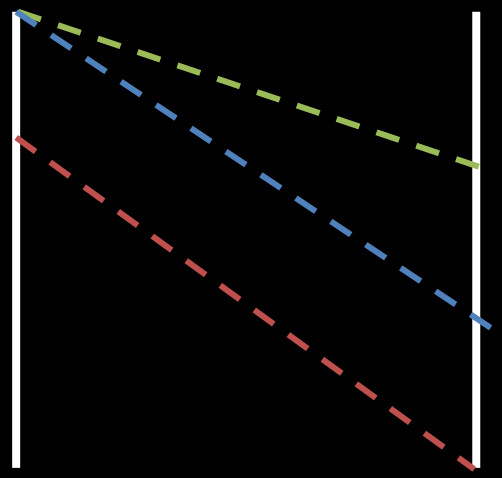


Fluid Rate Curve

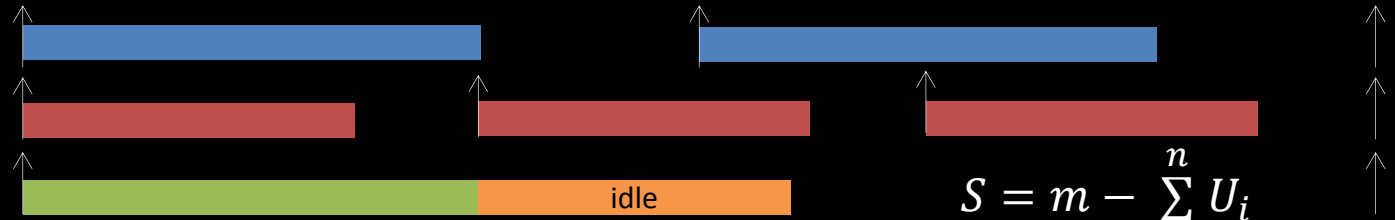


jobs who twine themselves around the ***fluid rate curve*** are somehow in good shape

DP-Wrap

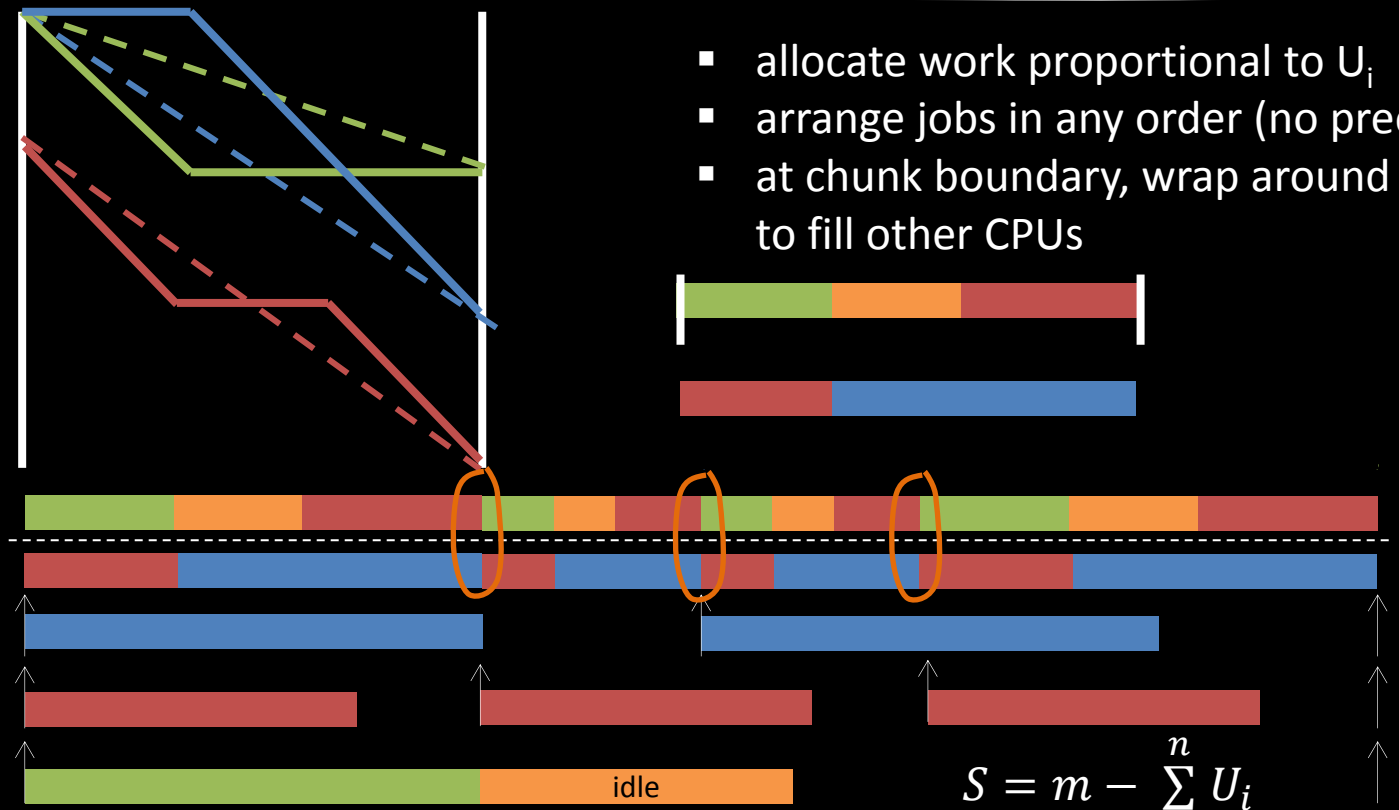


- allocate work proportional to U_i
- arrange jobs in any order (no preemptions)



$$S = m - \sum_{i=0}^{n} U_i$$

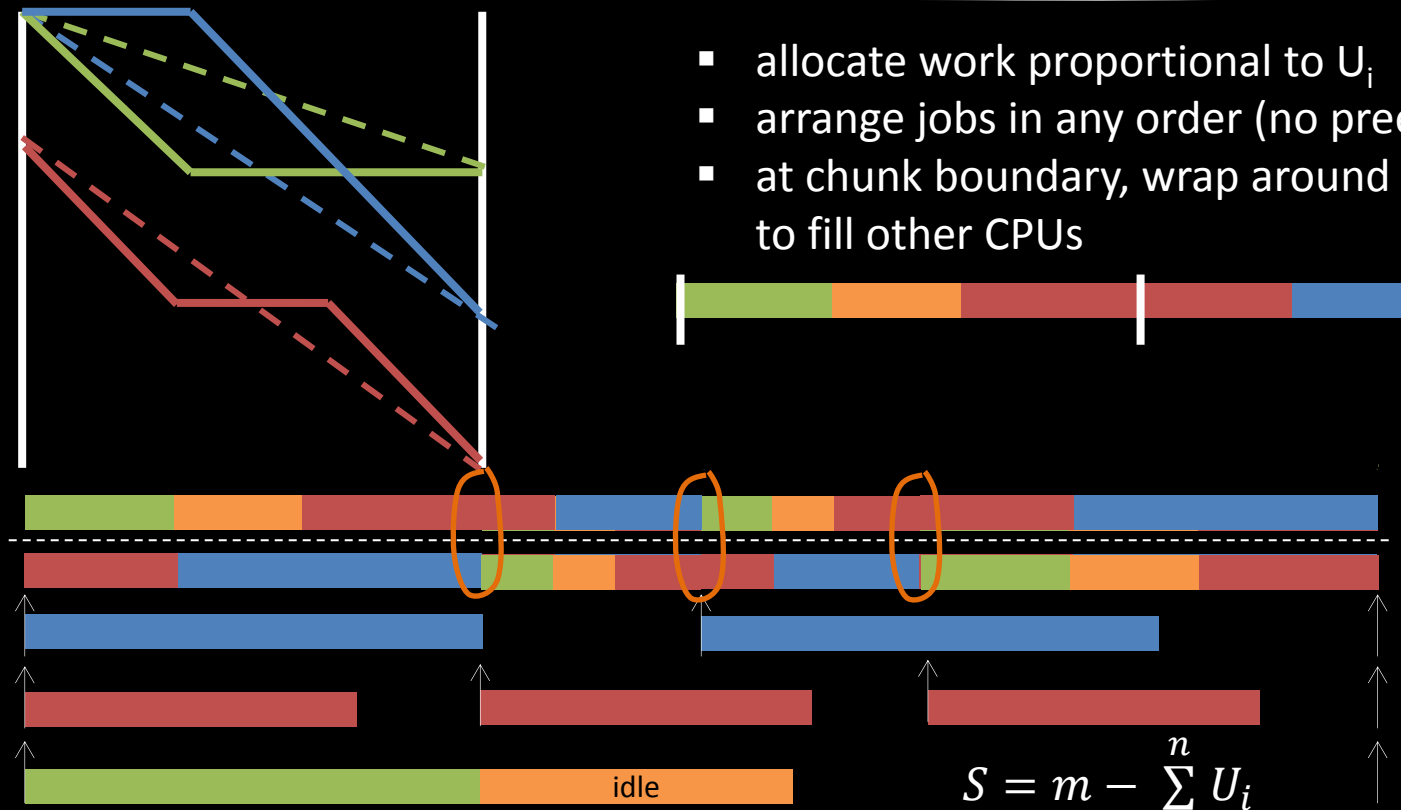
DP-Wrap



- allocate work proportional to U_i
- arrange jobs in any order (no preemptions)
- at chunk boundary, wrap around to fill other CPUs

$$S = m - \sum_{i=0}^{n} U_i$$

DP-Wrap



- allocate work proportional to U_i
- arrange jobs in any order (no preemptions)
- at chunk boundary, wrap around to fill other CPUs

$$S = m - \sum_{i=0}^{n} U_i$$

Outline

Today:

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

Next week:

- Exercise: UP Resource Protocols
- Caches / Memories
- Resources
- 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