Enhanced Race-To-Halt: A Leakage-Aware Energy Management Approach for Dynamic Priority Systems

Muhammad Ali Awan Stefan M. Petters

CISTER Research Unit, ISEP/IPP, Porto, Portugal

ISEP
INSTITUTO SUPERIOR
DE ENGENHARIA DO PORTO
POLITÉCNICO DO PORTO

CISTER
Research Centre in
Real-Time Computing Systems
FCT Research Unit 608

Outline

- Motivation
- Power Saving Strategies
- State-of-the-Art
- Break-Even-Time and Static Limit
- Slack Management Algorithm
- Enhanced Race-To-Halt Algorithm
- Offline and Online Overhead
- Evaluation
- Future Directions and Conclusion

Motivation

Mobile

Battery Powered

Thermal Issues

Power Saving Strategies

Race-to-halt

1. Go Rab
2. Preemp
3. Reduce
4. High ov
   in and c
5. Effectiv
   interval

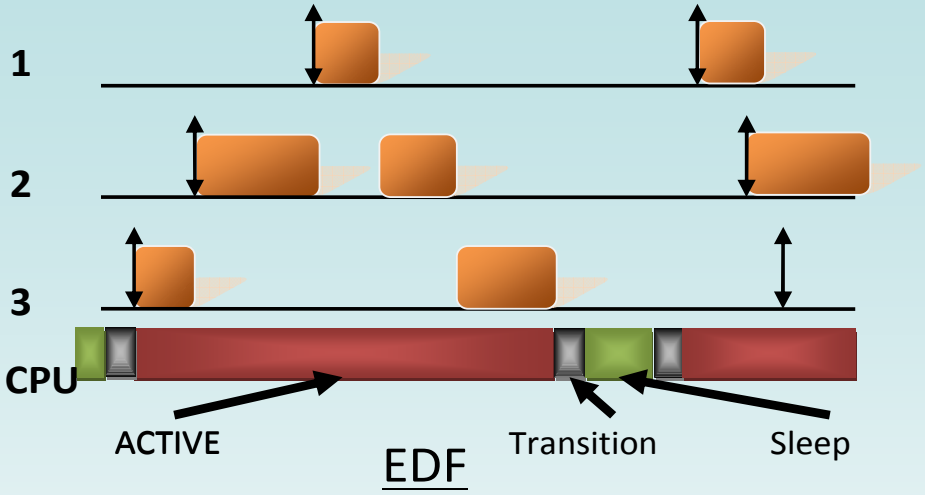
equency
uce dynamic

erhead
ls needed)
riendly

DVFS

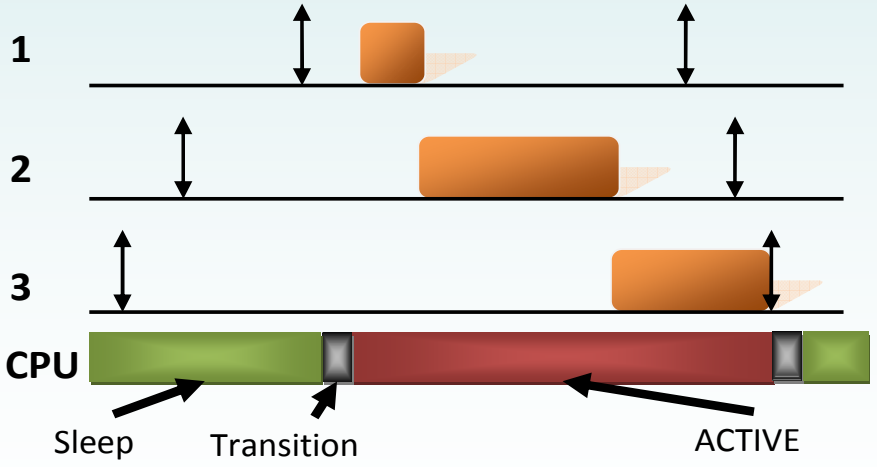
State-of-the-Art

Earliest Deadline First Algorithm



EDF

Procrastination Algorithm



LC-EDF

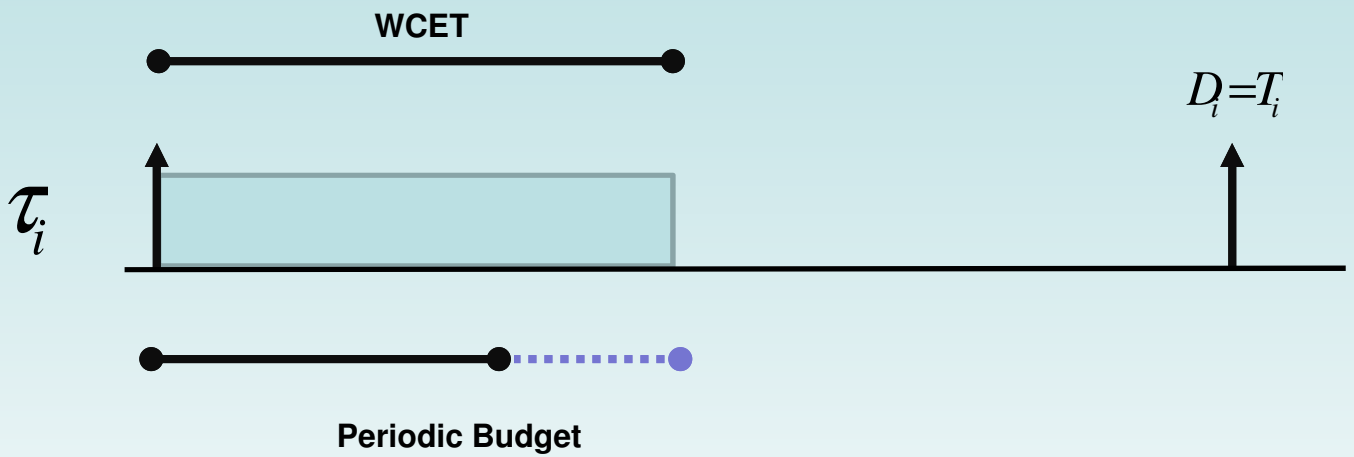
Variant of Procrastination

1. Applied to static priority algorithms (LC-DP)
2. Combines sleep states with DVFS

Assumptions

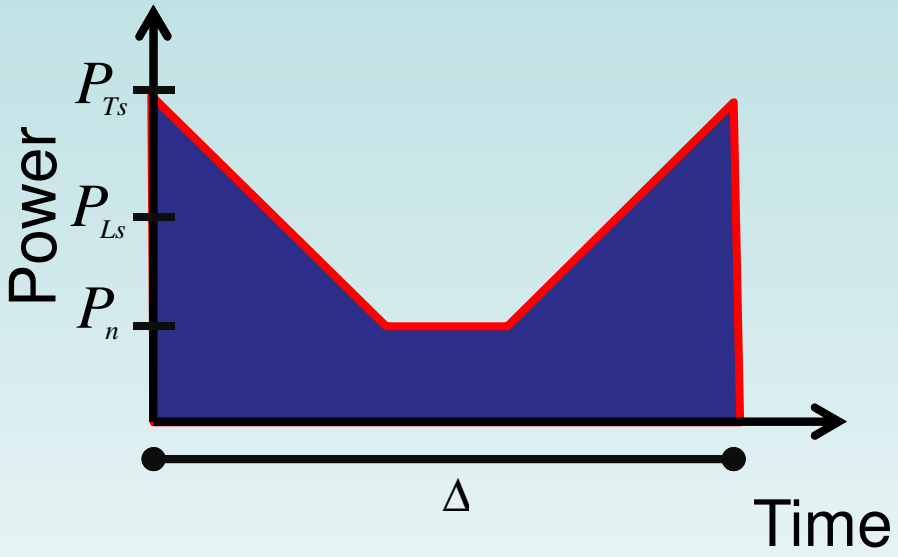
1. External specialized hardware
2. Convex power model
3. Continuous spectrum of available frequencies
4. Negligible time/energy overhead
5. Dynamic power >> static power consumption
6. Future task release information

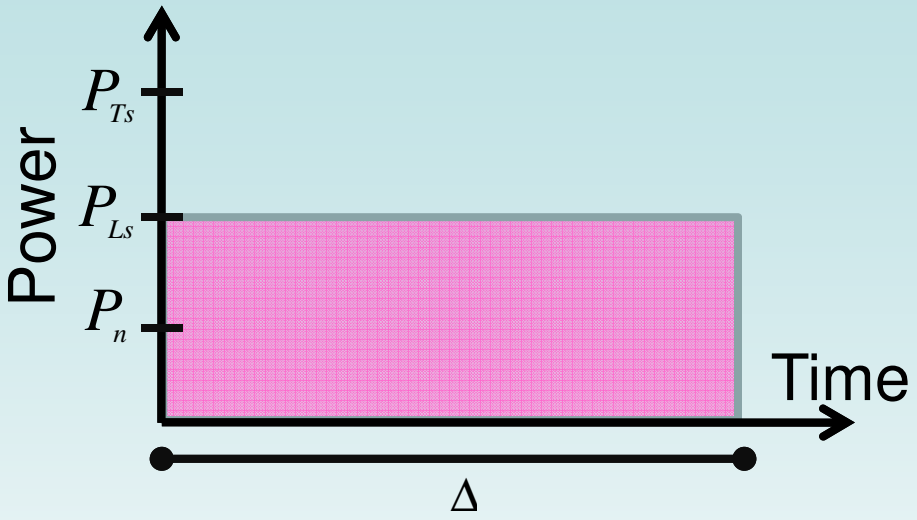
System Model

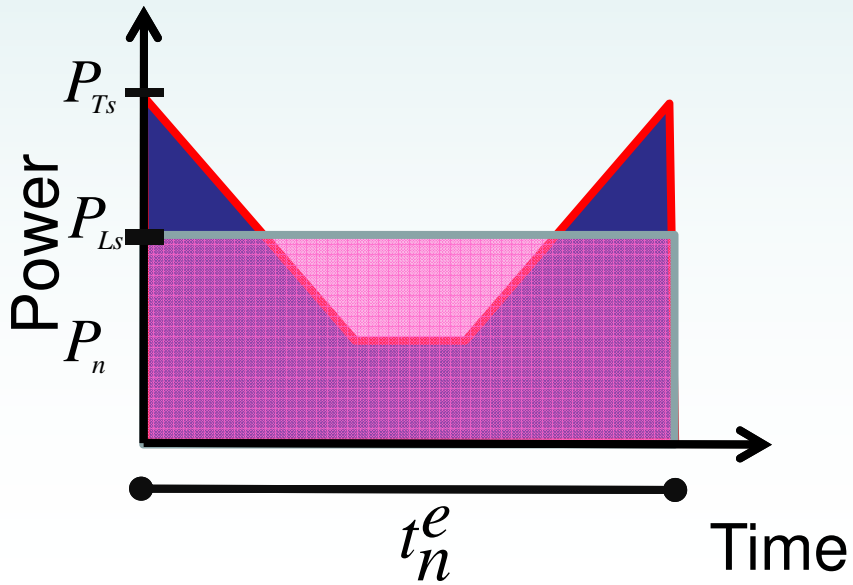


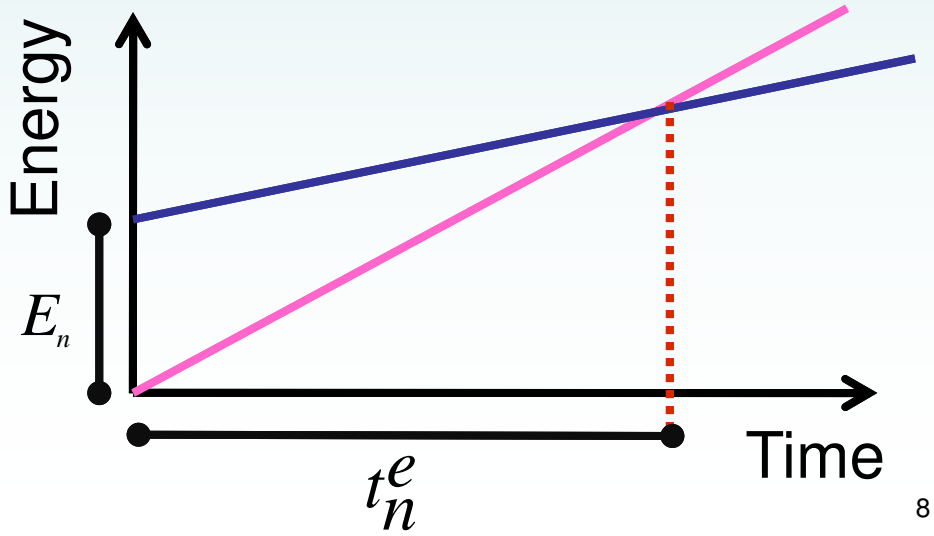
- RBED Frame Work
  - For HRT/SRT ---> **Periodic Budget = WCET**
  - For BE Tasks ---> **Periodic Budget ≤ WCET**

Break-Even-Time

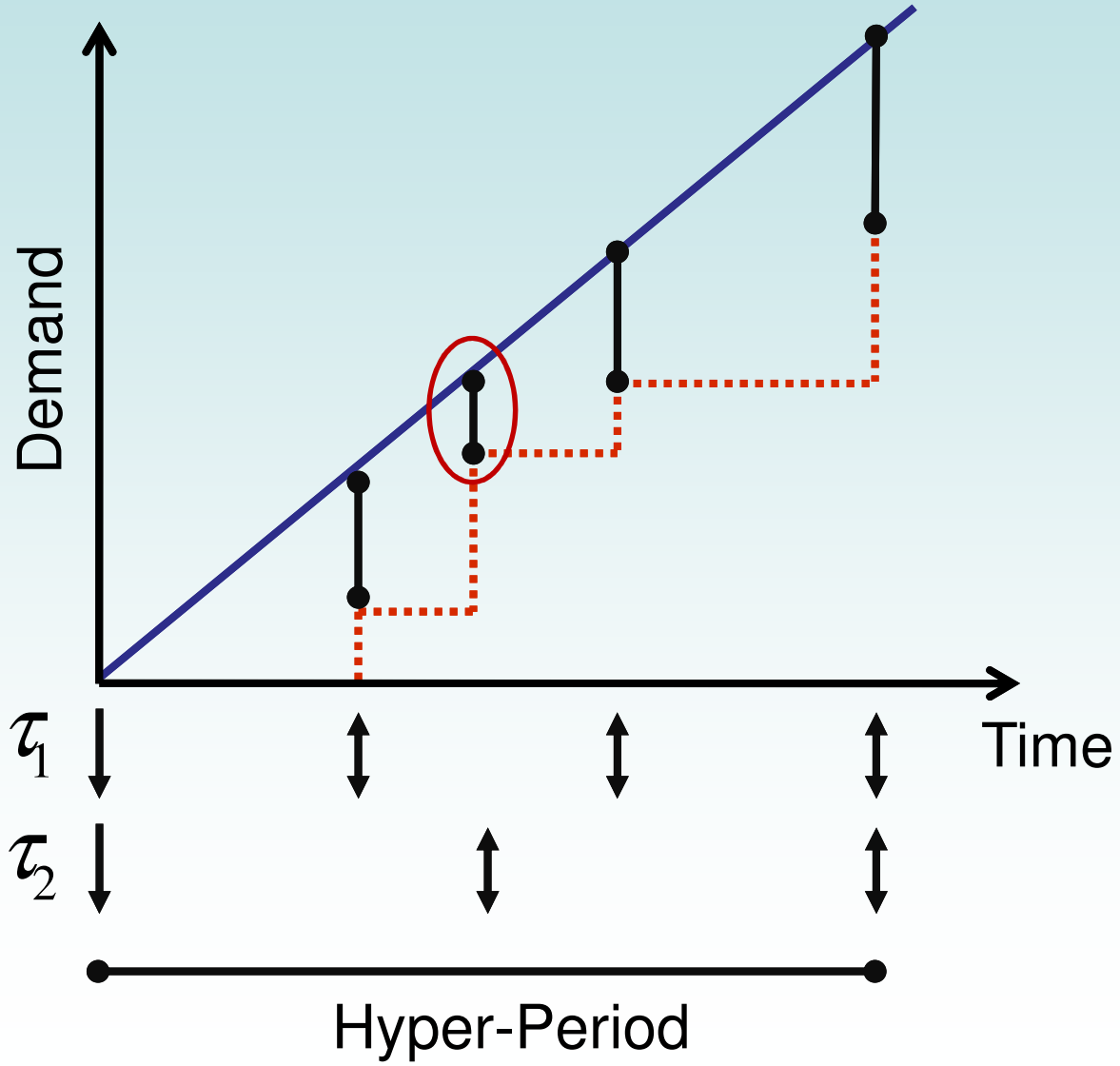








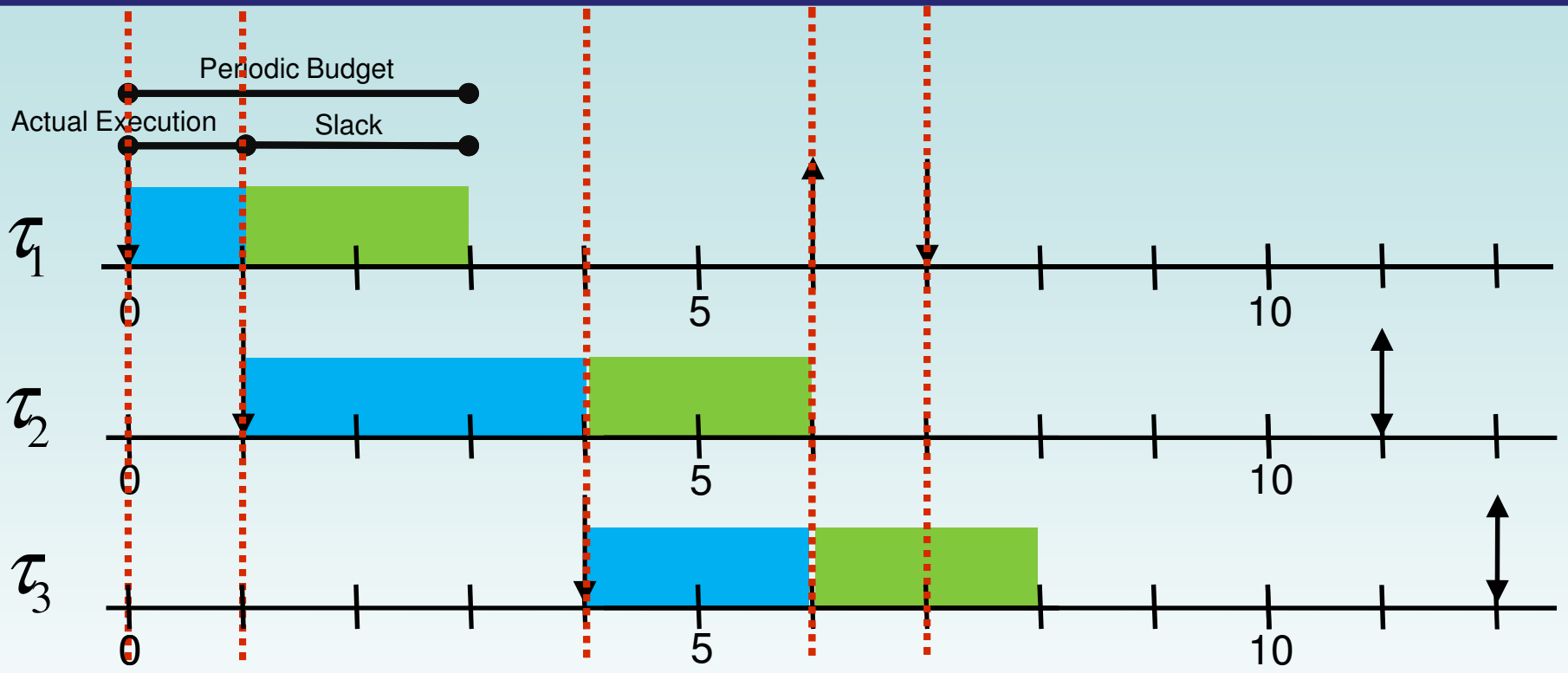
Static Limit for Sleep State



Slack Management

- Static Slack
  - Spare capacity in the schedule
- Dynamic Slack
  - Execution Slack
    - Difference between WCET and actual execution
  - Sporadic Slack
    - Inter-arrival delay (Sporadic Delay)

Slack Management Algorithm



Slack Container

Slack Size	Slack Deadline
	6 12

TaskSet

$\tau_1 (3, 6)$

$\tau_2 (3, 10)$

$\tau_3 (2, 12)$

ERTH Algorithm

$$\textbf{Condition 1=}\ Slack \geq Static\ Limit\ \&\&\ RT\ Task$$

$$\textbf{Condition 2=}\ Slack \geq Static\ Limit\ \&\&\ BE\ Task$$

- Initiate sleep for static limit

SYSTEM IDLE

If(Condition 1)

- Sleep for Static Limit

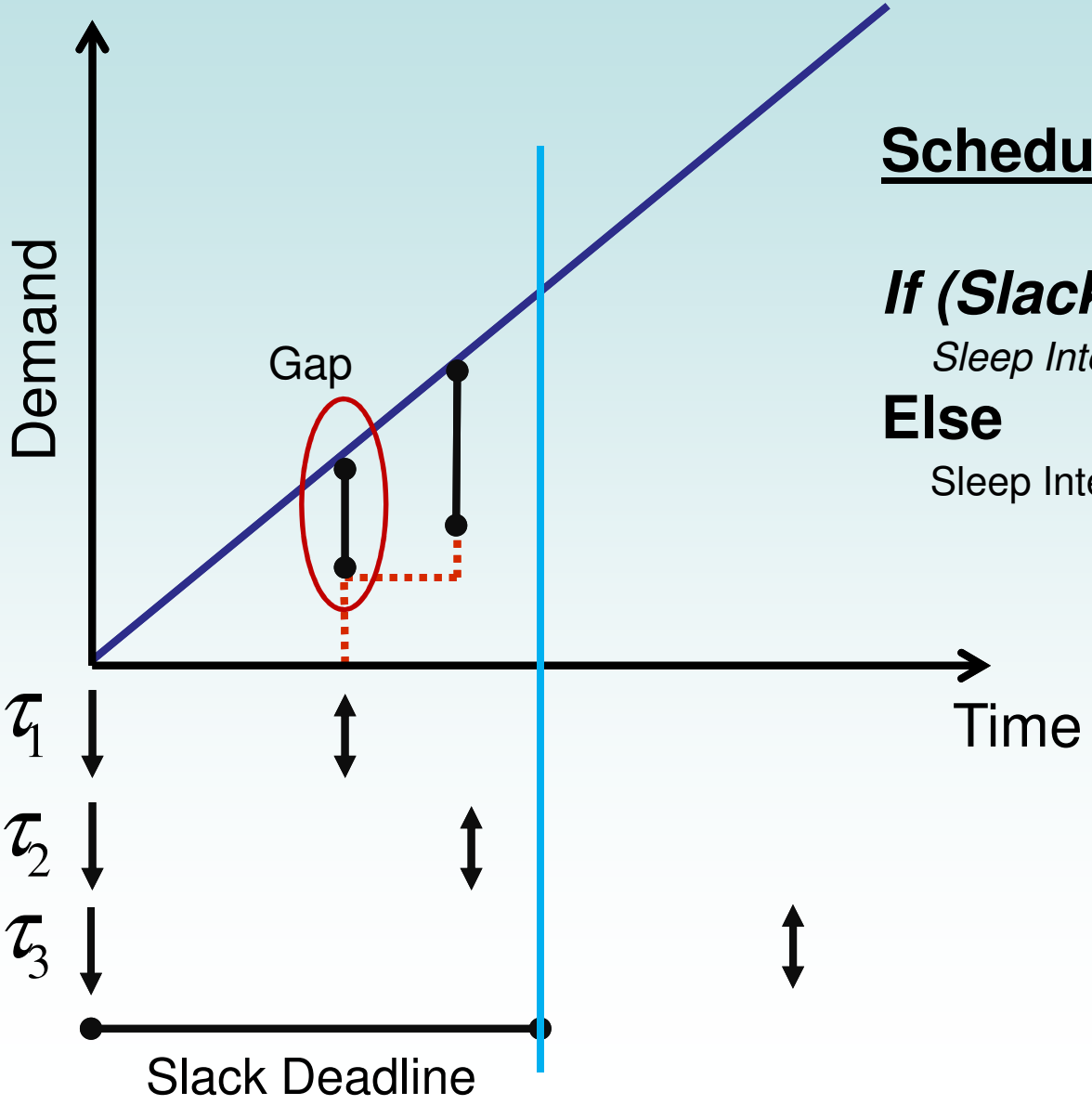
RT TASK

If(Condition 2)

- Evaluate sleep interval

BE TASK

Sleep Interval for BE Task



Schedulability:

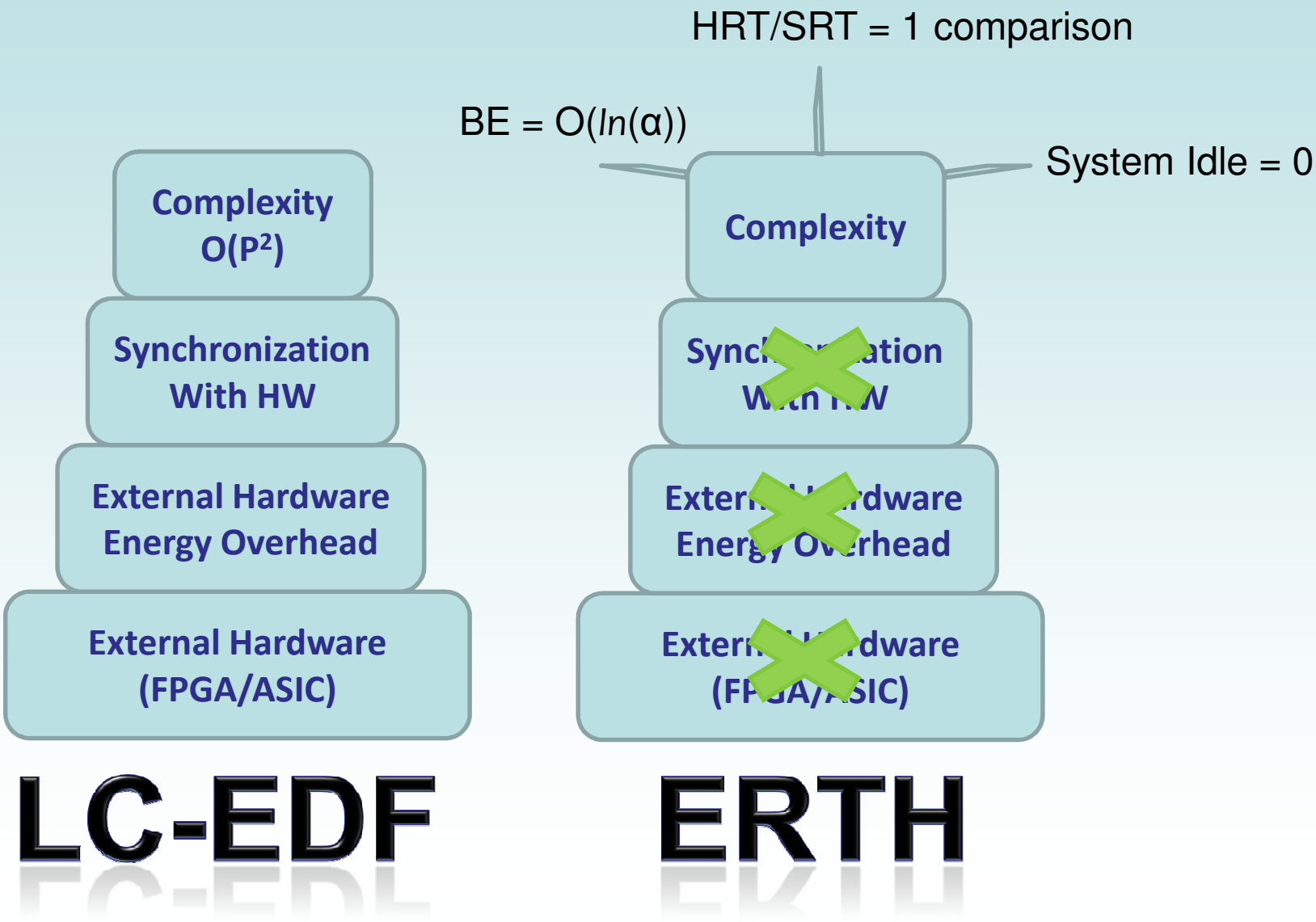
If (Slack < Gap)

Sleep Interval = Slack

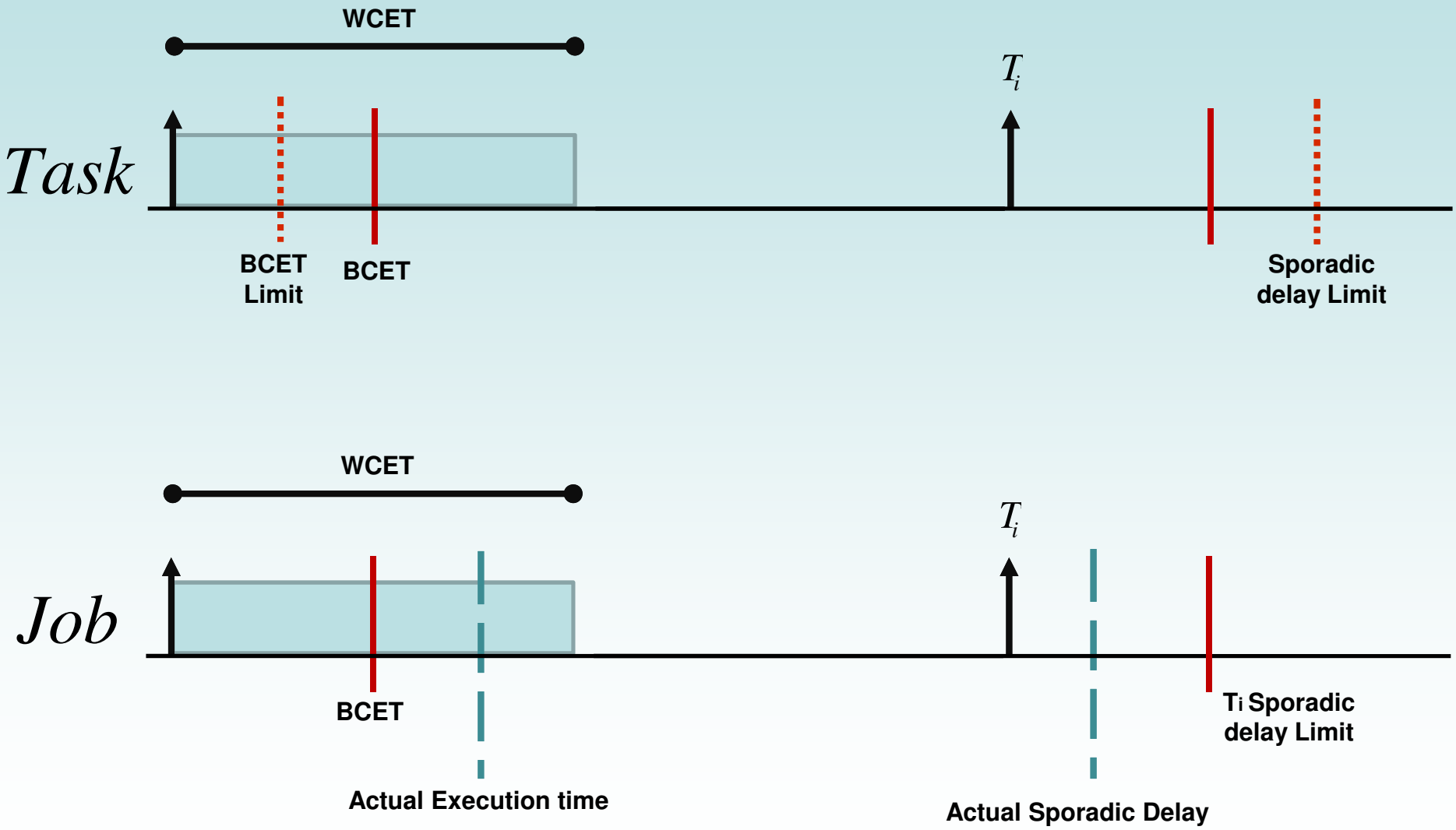
Else

Sleep Interval = Gap

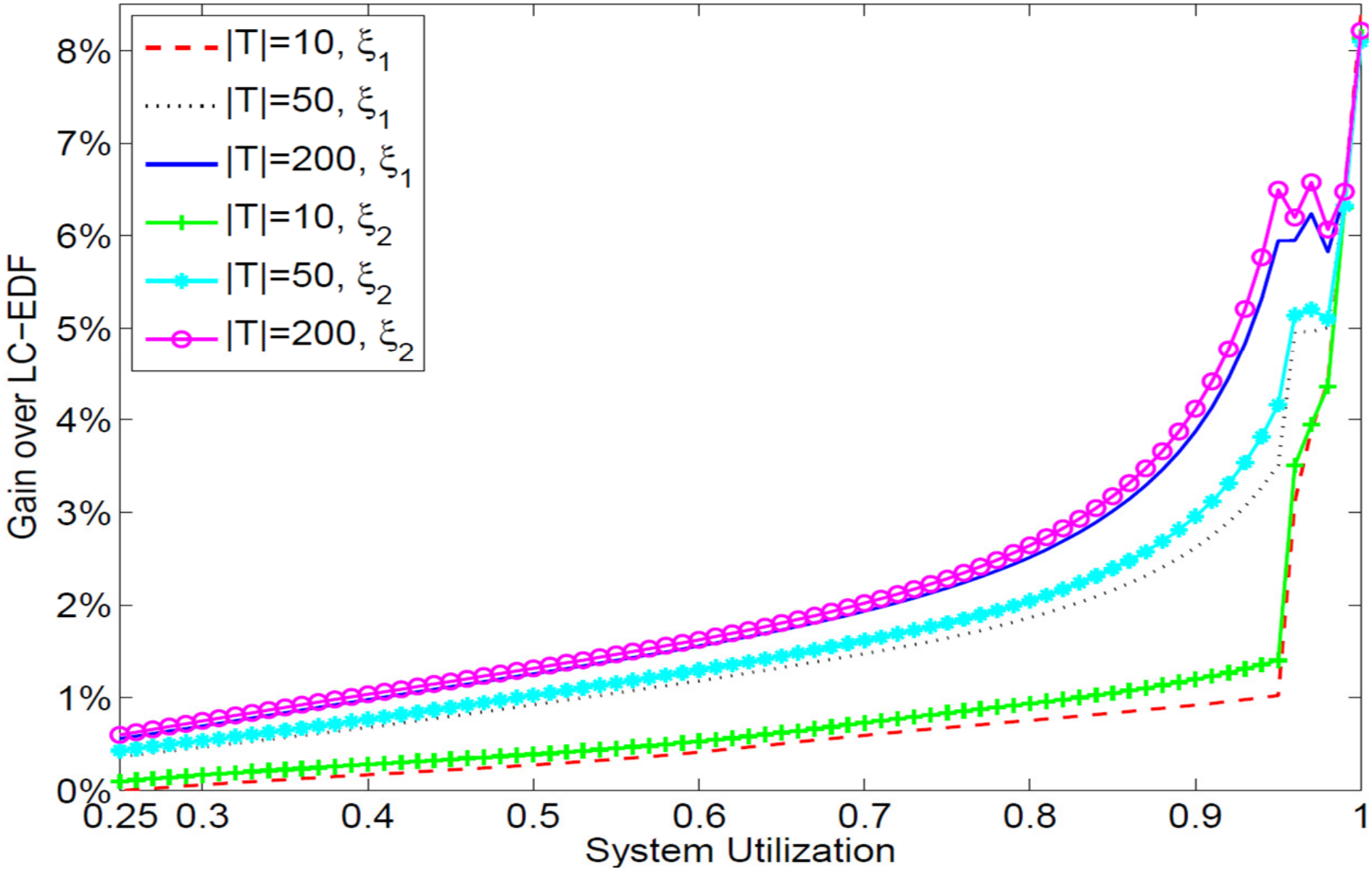
Offline and Online Overhead



Experimental Setup -1



Results-1


|T|=10, ξ1
|T|=50, ξ1
|T|=200, ξ1
|T|=10, ξ2
|T|=50, ξ2
|T|=200, ξ2
Gain over LC−EDF
0%
1%
2%
3%
4%
5%
6%
7%
8%
0.25
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
System Utilization

Progress Till Now

- Work in progress paper to WTR
  - 24/05/2010
- Work in progress paper to RTAS
  - 15/03/2011
- Conference paper to ECRTS
  - 06/06/2011
- Conference paper to RTCSA 2011
  - (Together With Borislav submitted on 22/04/2011)

Future Directions

- Submit the Journal version of this paper
  - Extension of the same algorithm with prior release information
  - Journal of Real-Time Systems 01/09/2011
- Extend this technique for Device power management (DPM)
  - RTAS 10/10/2012

Future Directions

- Offline analysis to select most efficient sleep state based on available workload
  - Almost Complete
- Bounding Number of preemptions with sleep states
  - ECRTS 01/2012
- Power management for partitioned Multicores
  - RTSS 05/2012

Conclusion

- DVFS is diminishing
- Sleep states are emerging
  - Reduced overhead (Energy/Time)
- ERTH
  - Reduces online overhead
  - Energy gains
  - A practical approach

Questions and Comments