Memory Bandwidth Regulation for Multiframe Task Sets

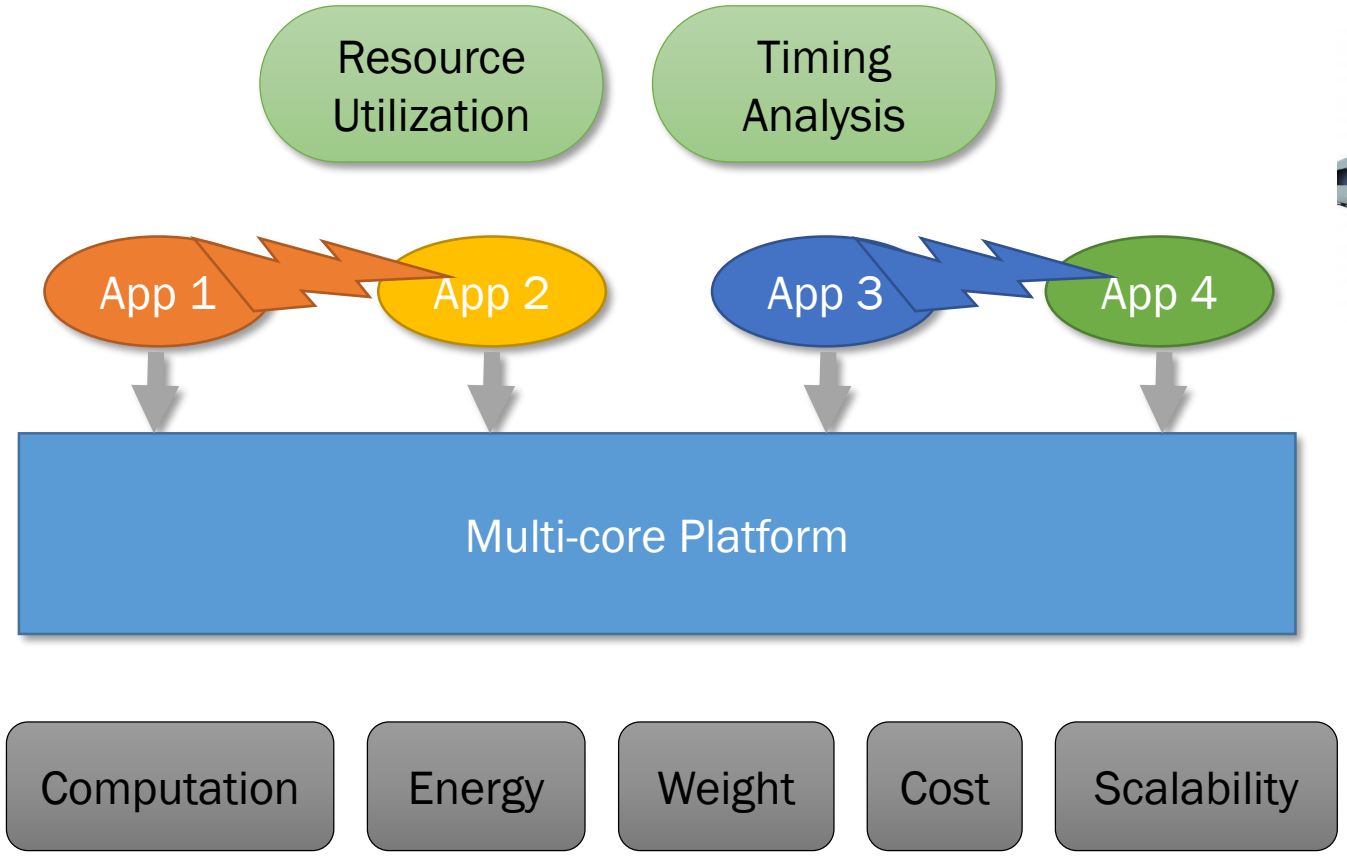
Muhammad Ali Awan, Pedro F. Souto, Konstantinos Bletsas, Benny Akesson, and Eduardo Tovar

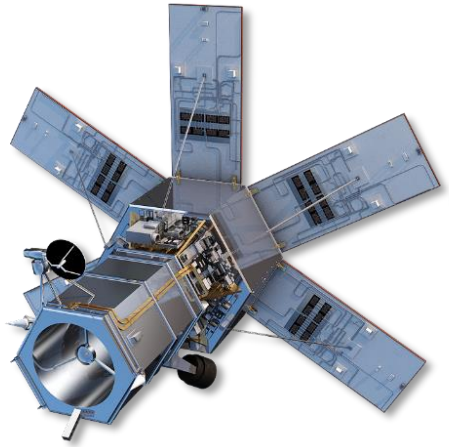
CISTER – Research Centre in Real-Time & Embedded Computing Systems

Outline

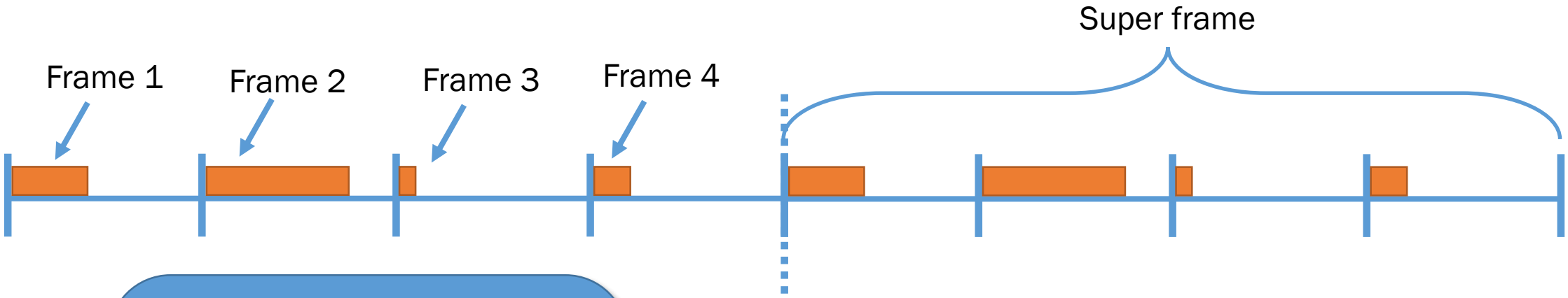
- Motivation and system model
- Background
- Our approach
- Computation time
- Heuristics
- Evaluation
- Conclusions

Motivation


Resource Utilization
Timing Analysis
App 1
App 2
App 3
App 4
Multi-core Platform
Computation
Energy
Weight
Cost
Scalability



Task Model



Multiframe task-model

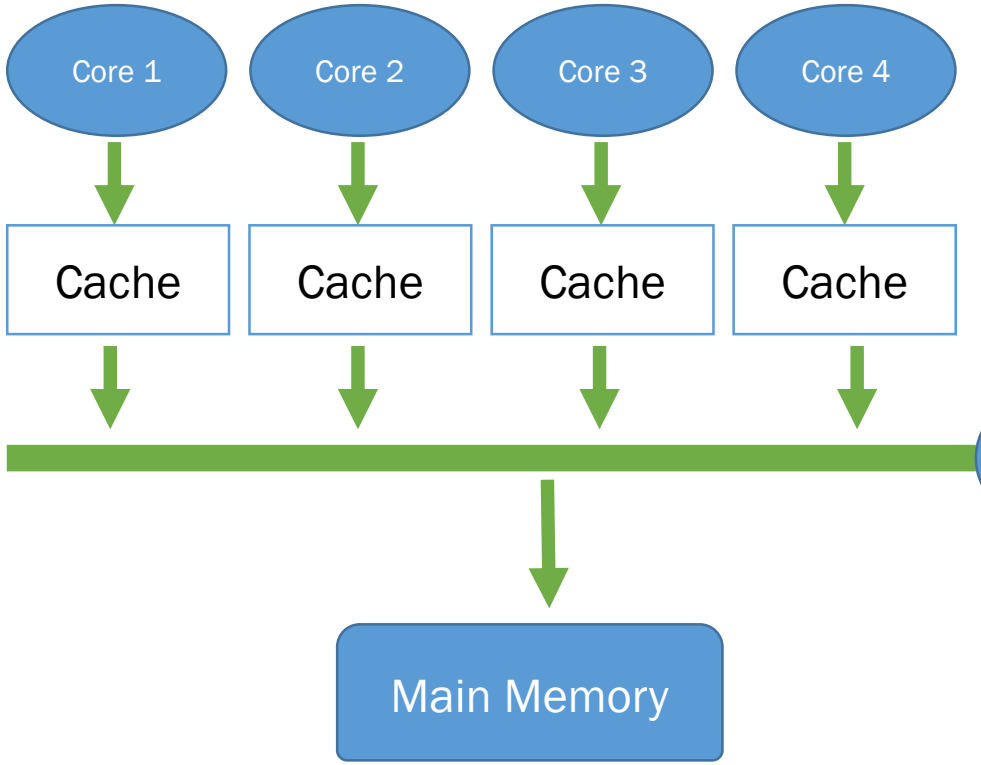
Preemptive fixed priority scheduling

Independent sporadic tasks

No-migration

Deadline monotonic

Platform Model



Identical multicore platform

Partitioned/private last-level cache

Prefetchers and speculative units are disabled

Round-robin bus arbitration
(interconnect + memory controller)

Performance monitoring counters

Constant memory access time

Multiple outstanding memory requests

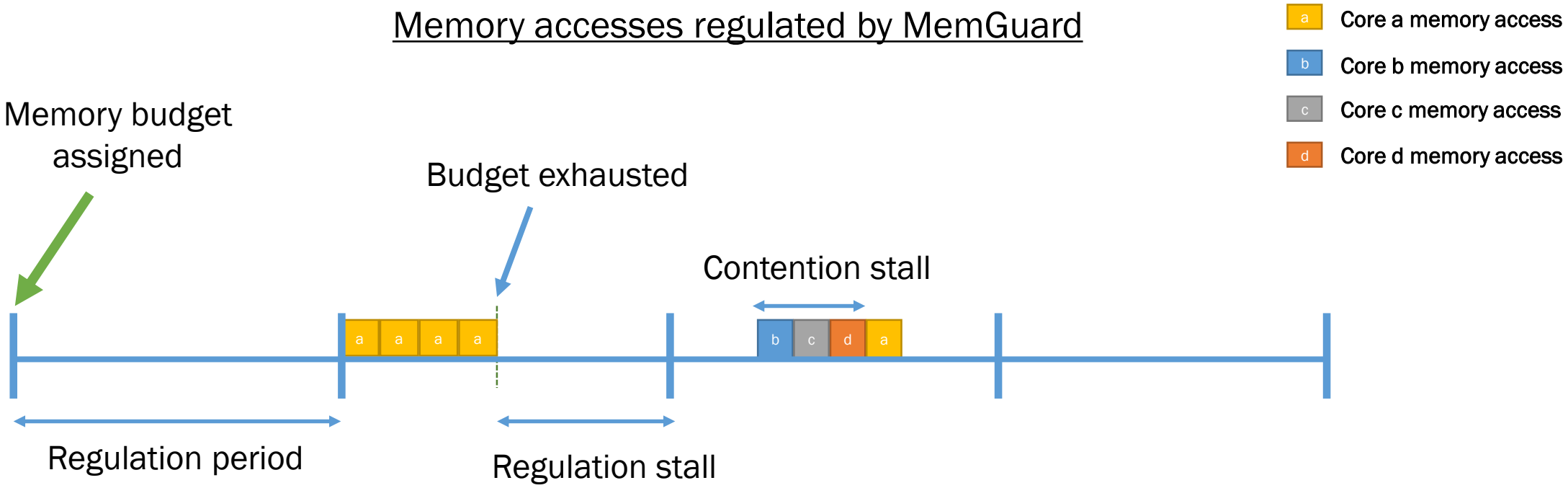
WCET

CPU computation

Memory accesses

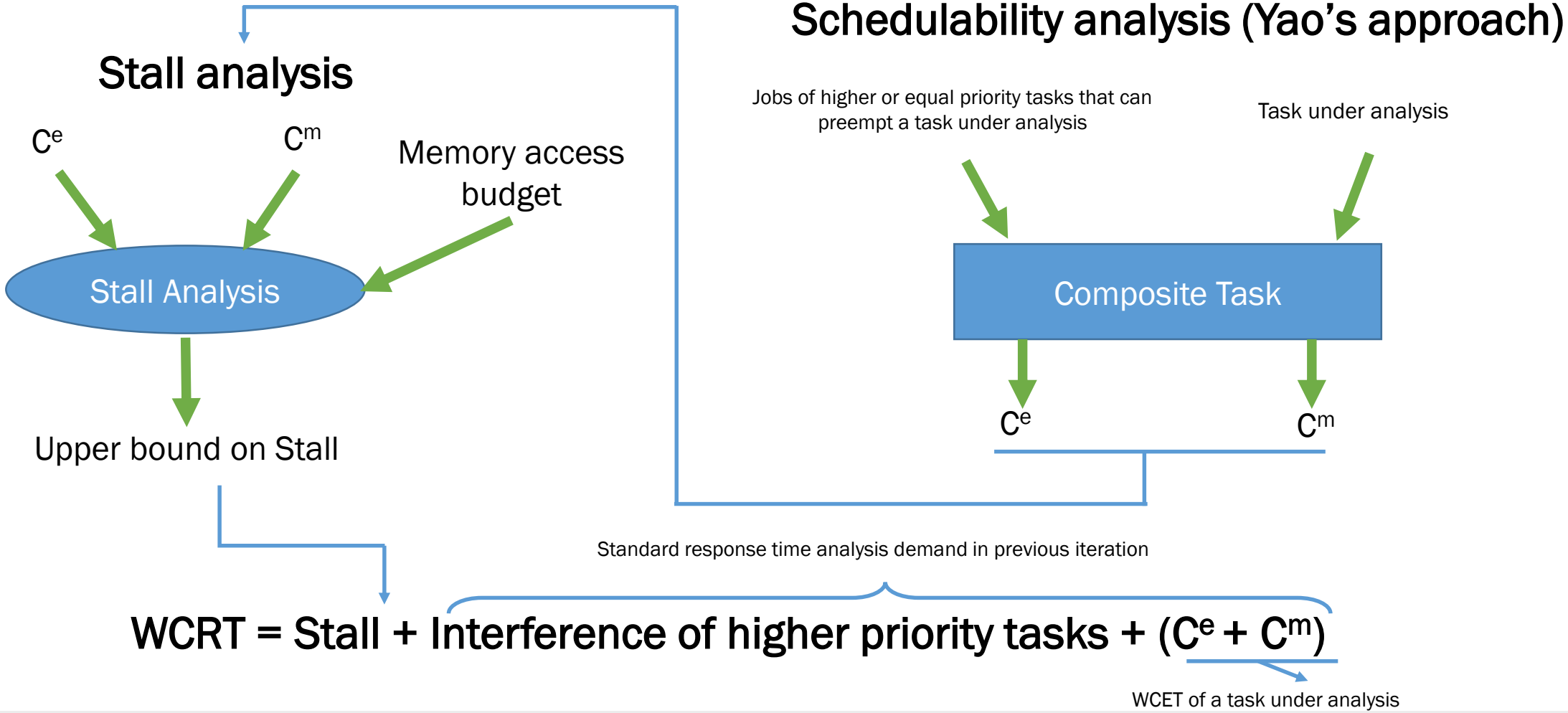
Computation and memory accesses do not overlap in time

Memory Access Regulation Model



Uneven memory bandwidth across cores

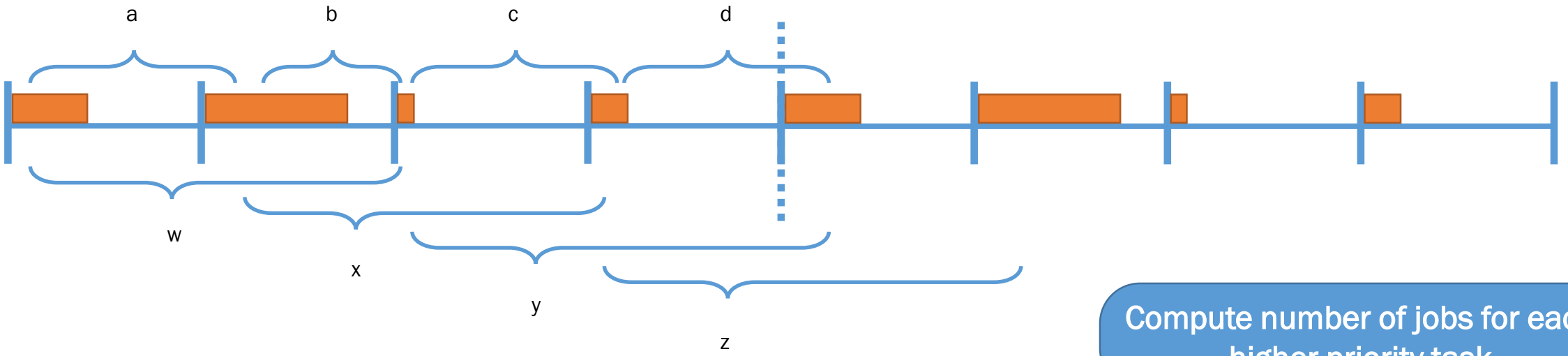
Memory-aware schedulability analysis



Schedulability analysis of Multiframe tasks

Maximum cumulative execution requirement (Baruah's approach)

For two frames = Max (a, b, c, d)

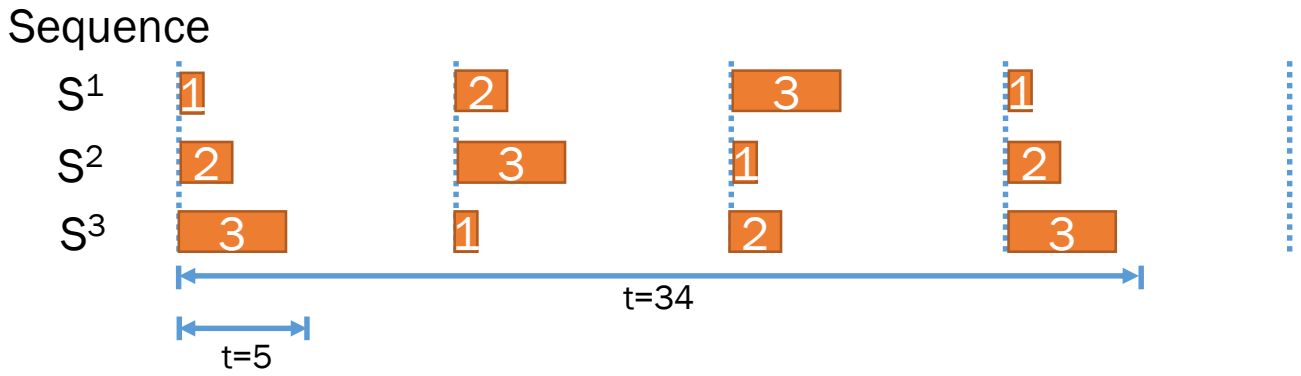


For three frames = Max (w, x, y, z)

Compute number of jobs for each higher priority task

Sum their maximum cumulative execution requirement

Why SOTA solutions do not work



Task parameters

- Period of 10
- Three frames

F1 1 C = 1 (C^m = 0 , C^e = 1)

F2 2 C = 2 (C^m = 2 , C^e = 0)

F3 3 C = 3 (C^m = 1 , C^e = 2)

t = 34

Sequence	Total WCET	Total memory accesses
S^1	7	3
S^2	8	5
S^3	9	4

t = 5

Sequence	Total WCET	Total memory accesses
S^1	1	0
S^2	2	2
S^3	3	1

WCRT = Stall + Interference of higher priority tasks + (C^e + C^m)

Main contributions

Worst-case memory stall for multiframe task-model

Stall-aware schedulability analysis for multiframe task-model

Five memory bandwidth and task-to-core allocation heuristics

Our proposed schedulability analysis

$$WCRT^{F1} = Stall + Interference\ of\ higher\ priority\ tasks + (C^{e,F1} + C^{m,F1})$$

$$WCRT^{F2} = Stall + Interference\ of\ higher\ priority\ tasks + (C^{e,F2} + C^{m,F2})$$

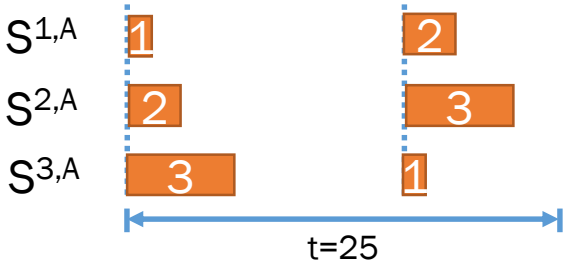
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$WCRT^{Fn} = Stall + Interference\ of\ higher\ priority\ tasks + (C^{e,Fn} + C^{m,Fn})$$

$$\mathbf{WCRT = Max(WCRT^{F1}, WCRT^{F2}, \ldots, WCRT^{Fn})}$$

Example

Task A
(3 frames, Period =20)



$S^{1,A} = \{F^{1,A}, F^{2,A}\}$

$S^{2,A} = \{F^{2,A}, F^{3,A}\}$

$S^{3,A} = \{F^{3,A}, F^{1,A}\}$

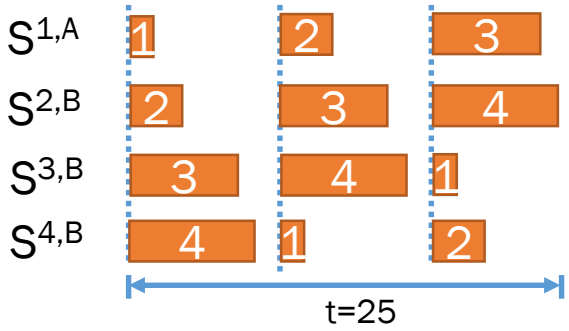
Cartesian product of sequences =

$\{ (S^{1,A}, S^{1,B}), (S^{1,A}, S^{2,B}), (S^{1,A}, S^{3,B}), (S^{1,A}, S^{4,B}),$
$(S^{2,A}, S^{1,B}), (S^{2,A}, S^{2,B}), (S^{2,A}, S^{3,B}), (S^{2,A}, S^{4,B}),$
$(S^{3,A}, S^{1,B}), (S^{3,A}, S^{2,B}), (S^{3,A}, S^{3,B}), (S^{3,A}, S^{4,B}) \}$

Or

$\{\Theta^1, \Theta^2, \Theta^3, \Theta^4,$
$\Theta^5, \Theta^6, \Theta^7, \Theta^8,$
$\Theta^9, \Theta^{10}, \Theta^{11}, \Theta^{12}\}$

Task B
(4 frames, Period = 10)



$S^{1,B} = \{F^{1,B}, F^{2,B}, F^{3,B}\}$

$S^{2,B} = \{F^{2,B}, F^{3,B}, F^{4,B}\}$

$S^{3,B} = \{F^{3,B}, F^{4,B}, F^{1,B}\}$

$S^{4,B} = \{F^{4,B}, F^{1,B}, F^{2,B}\}$

In $(k+1)^{th}$ iteration, for each tuple Θ^x compute stall and interference

$WCRT^{Fx \mid (k+1)}$ = Stall (Θ^x) + Interference of higher priority tasks (Θ^x) + ($C^{e,Fx} + C^{m,Fx}$), where x = {1, 2, …, 12}

Select tuple that gives the maximum memory regulation stall and interference.

Reducing computational cost

No. of Frames

For any two frames

$F^i \geq F^j$, iff

$$C^{e,i} \geq C^{e,j} \quad \text{and} \quad C^{m,i} \geq C^{m,j}$$

Or

$$(C^{e,i}, C^{m,i}) \geq (C^{e,j} \geq C^{m,j})$$

It is sufficient to check only WCRT of F^i

No. of Sequences

For any two sequences

$S^x \geq S^y$, iff

$$C^e_{S^x} \geq C^e_{S^y} \quad \text{and} \quad C^m_{S^x} \geq C^m_{S^y}$$

Or

$$(C^e_{S^x}, C^m_{S^x}) \geq (C^e_{S^y} C^m_{S^y})$$

It is sufficient to check only sequence S^x

No. of tuples

For any two tuple

$\Theta^x \geq \Theta^y$, iff

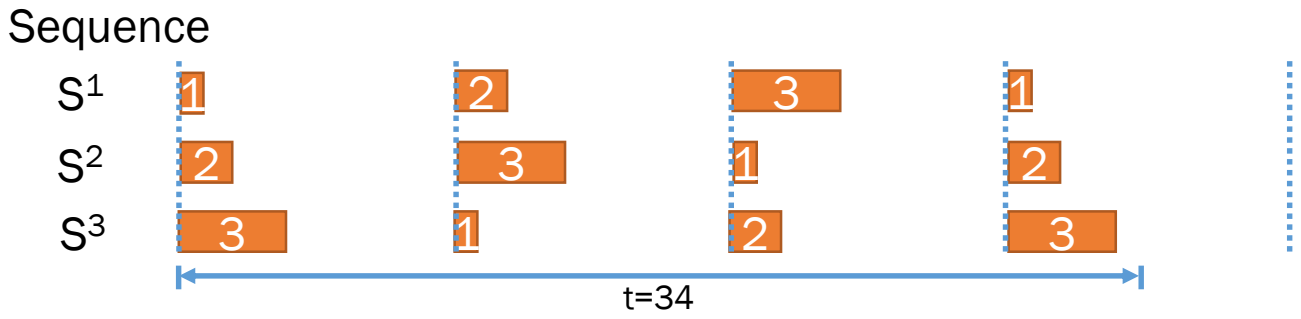
$$C^e_{\Theta^x} \geq C^e_{\Theta^y} \quad \text{and} \quad C^m_{\Theta^x} \geq C^m_{\Theta^y}$$

Or

$$(C^e_{\Theta^x}, C^m_{\Theta^x}) \geq (C^e_{\Theta^y} C^m_{\Theta^y})$$

It is sufficient to check only tuple Θ^x

Tightness vs computational cost



Task parameters

- Period of 10
- Three frames

F^1	1	$C = 1$ ($C^m = 0$, $C^e = 1$)
F^2	2	$C = 2$ ($C^m = 2$, $C^e = 0$)
F^3	3	$C = 3$ ($C^m = 1$, $C^e = 2$)

t = 34

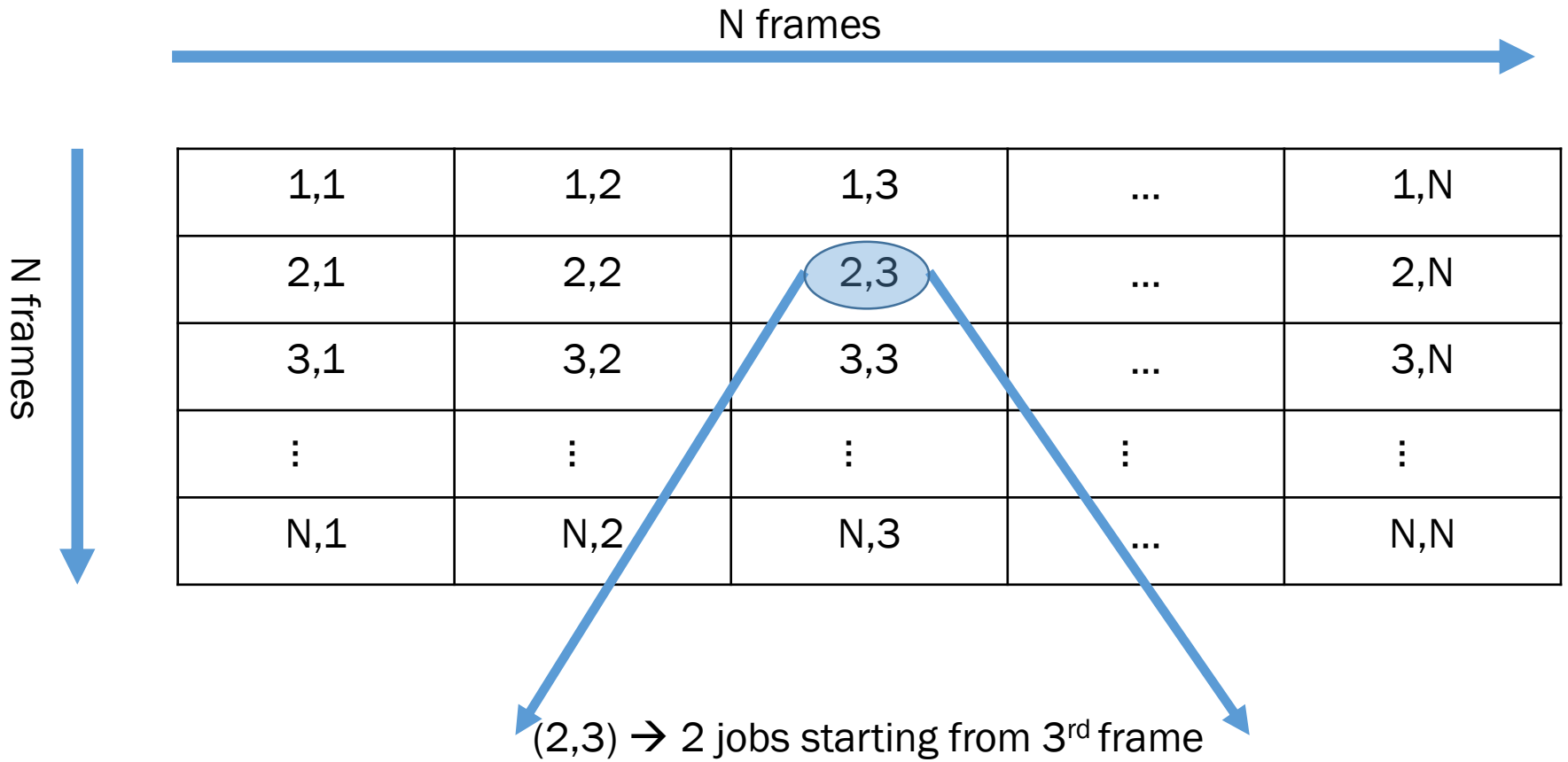
Sequence	Total WCET	Total memory accesses	Total CPU Computation
S^1	7	3	4
S^2	8	5	3
S^3	9	4	5

Sequence	Total WCET	Total memory accesses	Total CPU Computation
S^{12}	8	5	4
S^3	9	4	5

Sequence	Total WCET	Total memory accesses	Total CPU Computation
S^{123}	9	5	5

Implementation details

For each task



Task-to-core and Memory Bandwidth allocation

Even First-Fit
- Each Core has equal memory bandwidth share
- First-fit bin packing for task-to-core analysis

Five Heuristics

Uneven First-Fit
- Initially each Core has equal memory bandwidth share
- Trim-off memory bandwidth, if tasks are not schedulable with equal memory bandwidth
- Use this trimmed bandwidth to schedule remaining tasks

Priority assignment:
Deadline monotonic

Memory-fit
- Assign a task to a core that requires minimum memory bandwidth

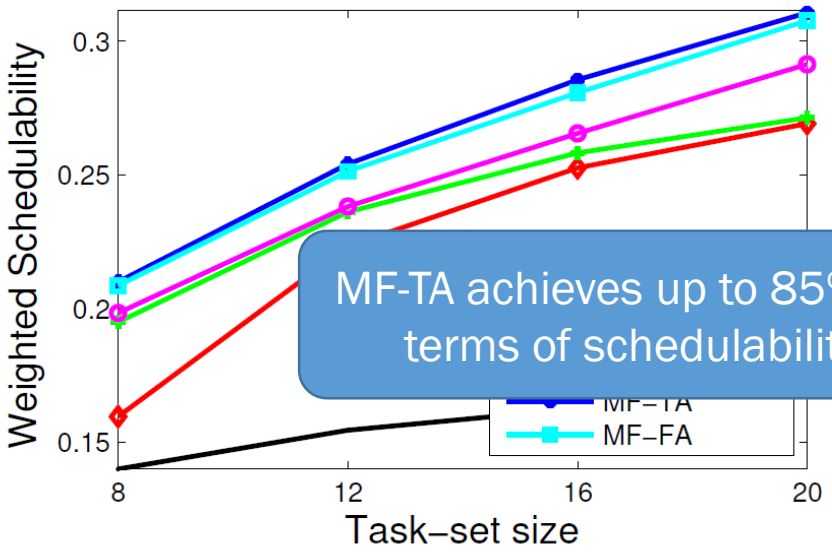
Memory density worst-fit
- Sort cores in non-increasing order of energy density
- Assign task to a core with that gives minimum increase in memory density

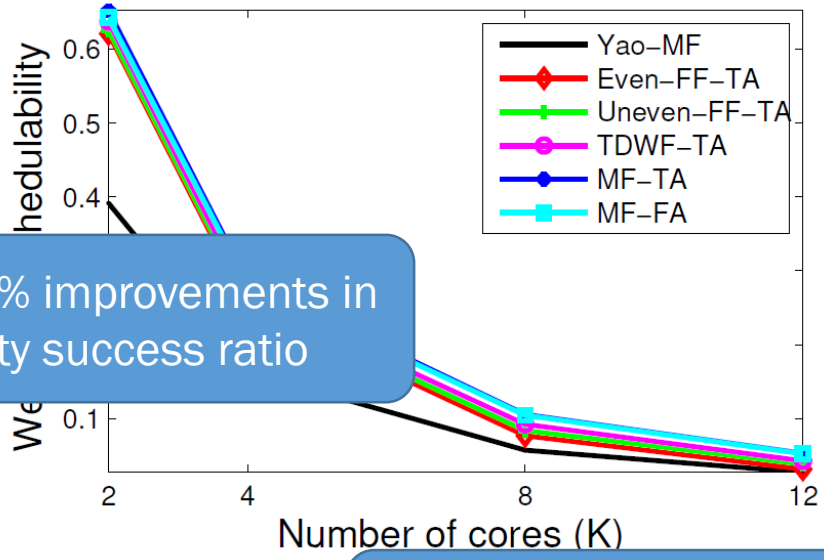
Total density worst-fit
- Similar to memory density worst-fit, except it uses total density instead of memory density

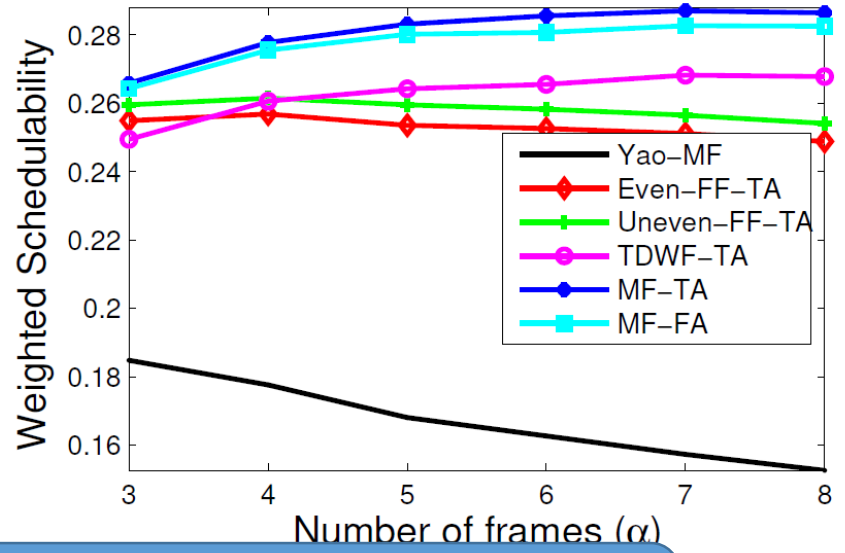
Experimental Setup

- Utilization: UUnifast-discard algorithm
- Inter-arrival time: Log-uniform distribution (10 ms- 1 s)
- WCET of first frame = inter-arrival time × Utilization
- Implicit deadlines (though algorithm works for constrained deadlines)
- Number of frames: Selected randomly
- WCET of other frames
  - Randomly selected with log-uniform distribution
  - Between user define value and WCET of first frame
- Memory accesses of each frame
  - Selected randomly
  - Memory access time is 40 nsec
  - Regulation period length is 100 msec
- 1000 random task-sets per set point

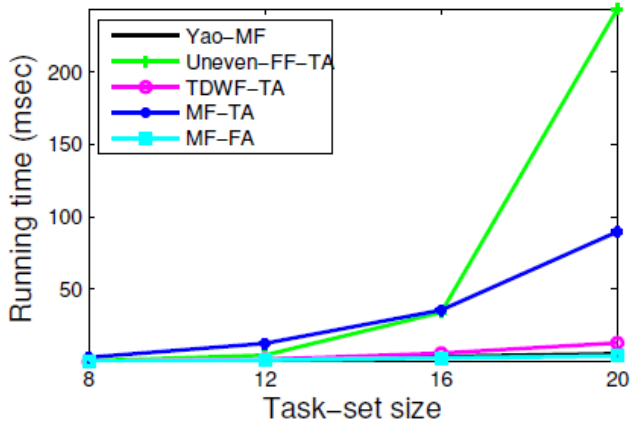
Results



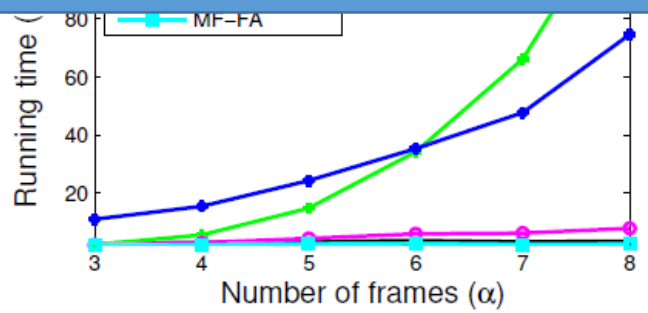




MF-TA achieves up to 85% improvements in terms of schedulability success ratio



MF-FA is 11x times faster than MF-TA with 3.7% decreases in terms of schedulability success ratio



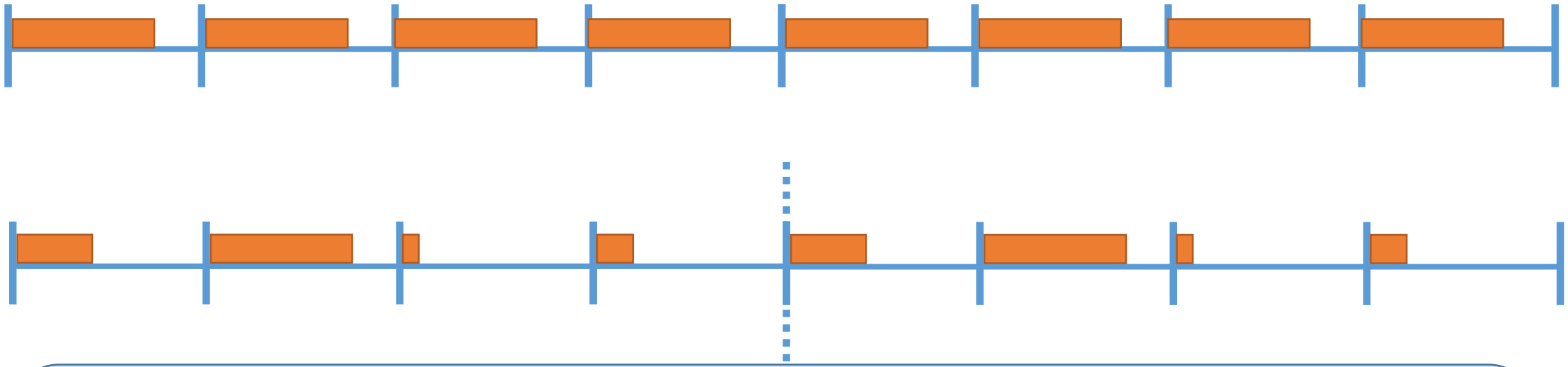
Conclusions

- Stall-aware schedulability analysis for multiframe task sets on multicore platforms
- Provided techniques to reduce the computation time
- Has possibility to trade-off tightness vs computation time
- Improved schedulability success ratio up to 85% when compared to frame-agonistic stall-aware analysis
- Achieved 11-fold speed up with 3.7% loss in schedulability
- Proposed five memory bandwidth and task-to-core allocation heuristics


isep
CISTER - Research Center in
Real-Time & Embedded Computing Systems
New Frontiers
in Computing

Questions ?

Main idea



Multiframe Task-model

Multicore platforms

Memory regulation

Our work