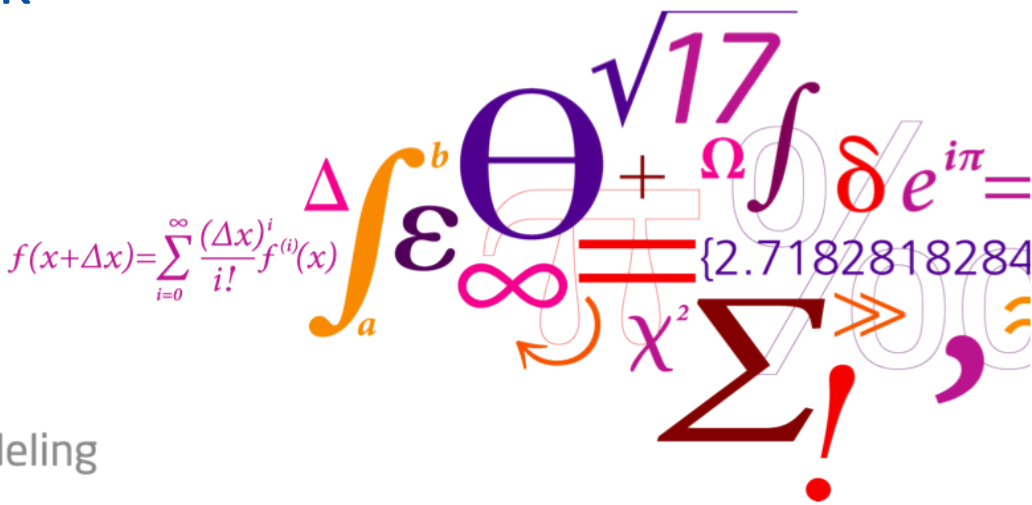
Optimization of Time-Partitions for Mixed-Criticality Real-Time Distributed Embedded Systems

Domițian Tămaș-Selicean and Paul Pop
Technical University of Denmark



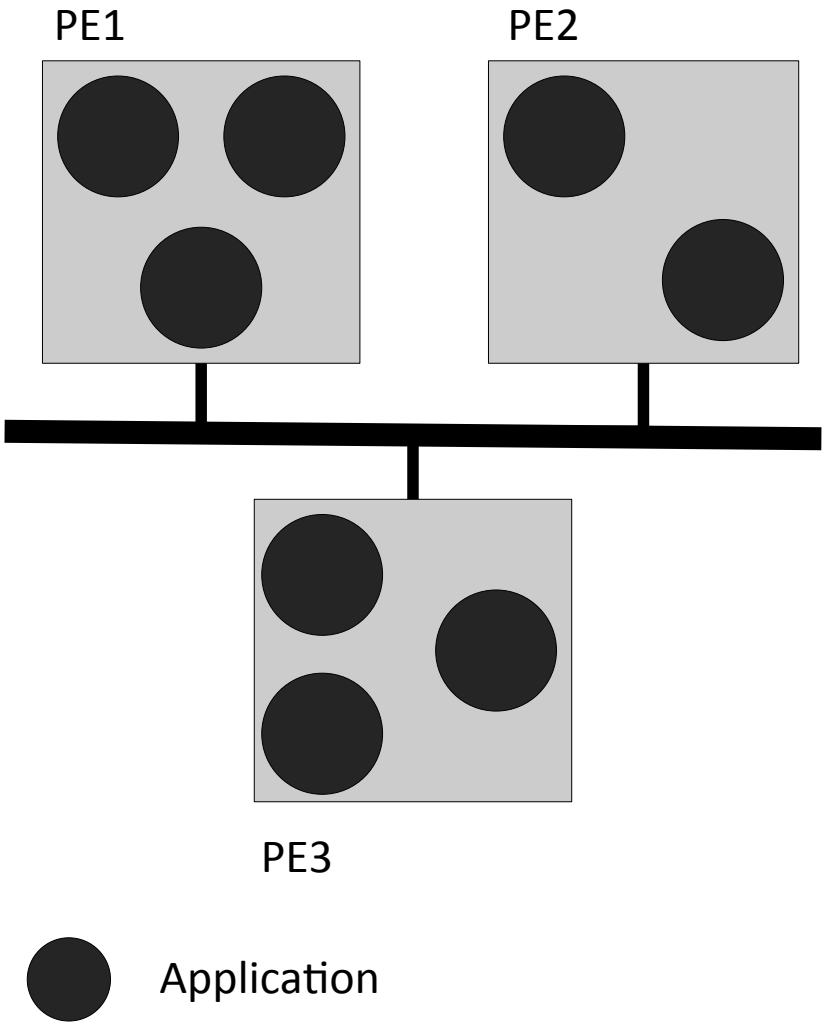
DTU Informatics
Department of Informatics and Mathematical Modeling

Outline

- Motivation
- System and application models
- Problem formulation and example
- Optimization strategy
- Experimental results
- Conclusions and future work

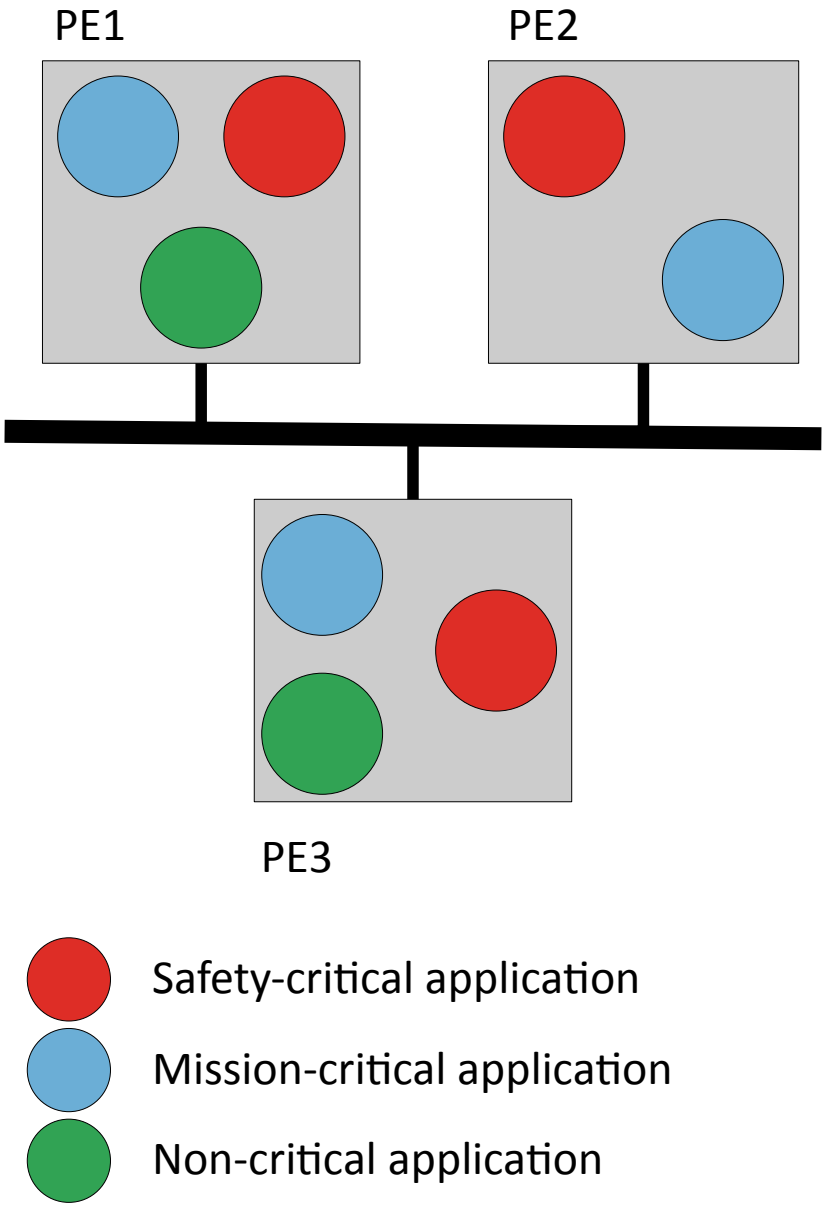
Motivation

- Real time applications implemented using distributed systems



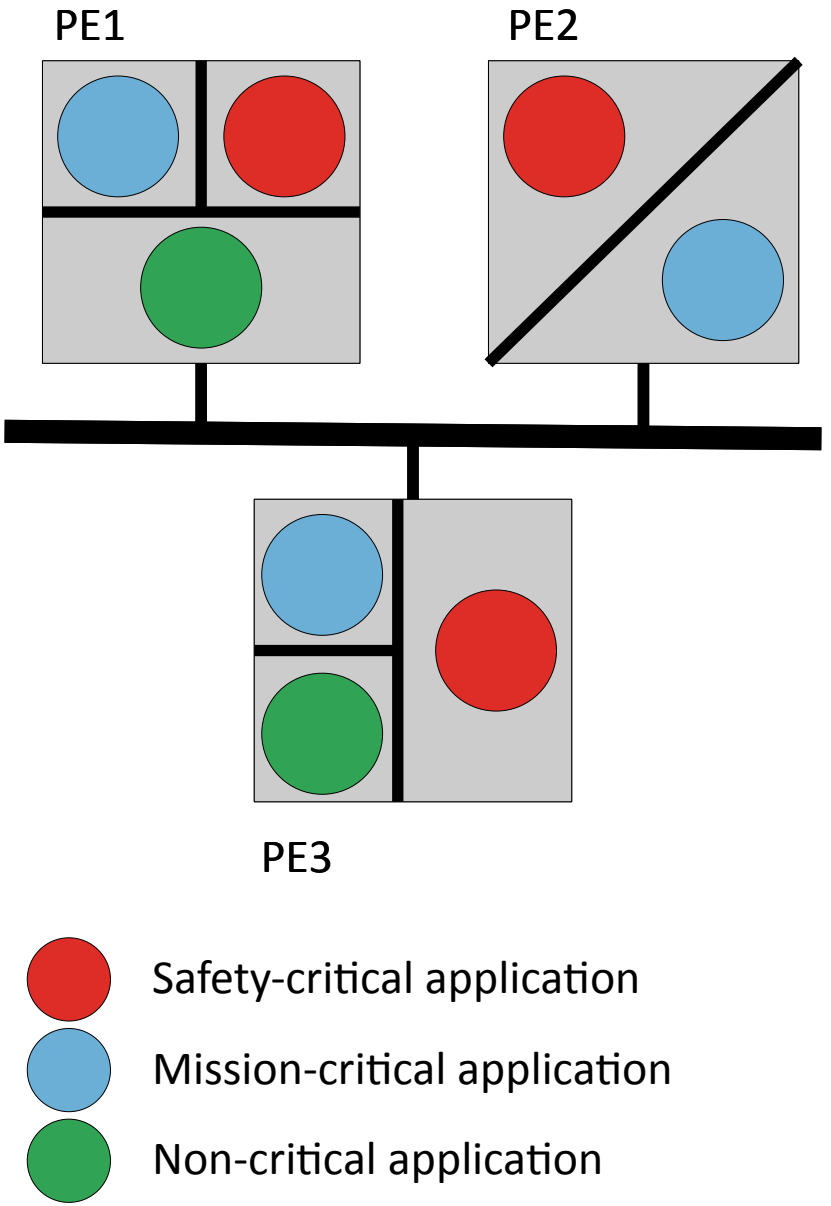
Motivation

- Real time applications implemented using distributed systems
- Mixed criticality applications share the same architecture

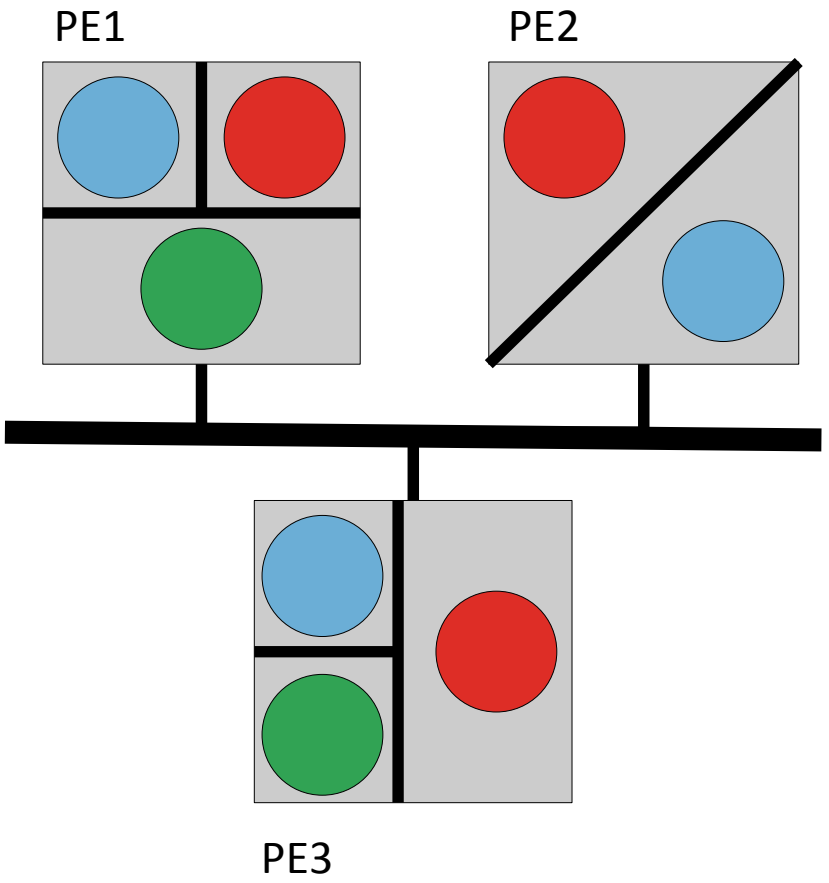


Motivation

- Real time applications implemented using distributed systems
- Mixed criticality applications share the same architecture
- Solution: partitioned architectures such as IMA

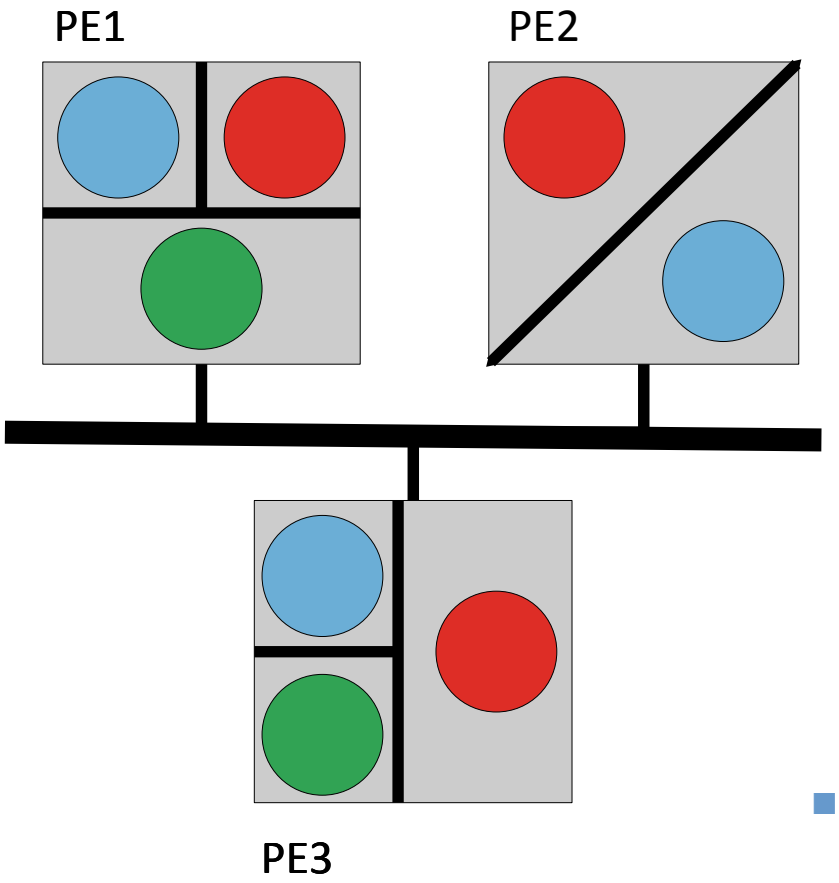


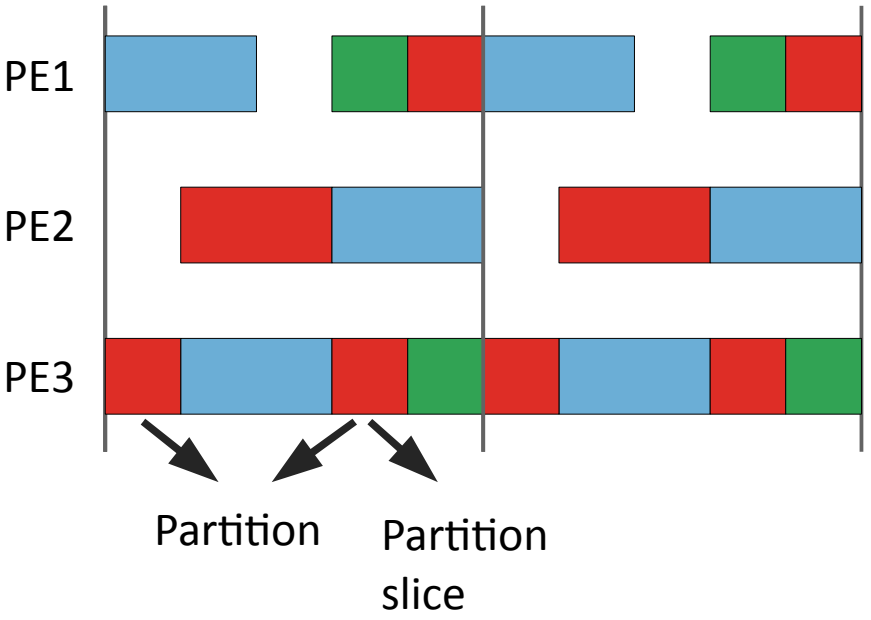
System Model



- Partition = virtual dedicated machine
- Partitioned architecture
  - Spatial partitioning
    - Protects one application from another
  - Time partitioning
    - Partitions the CPU time among partitions

System Model : Time Partitions





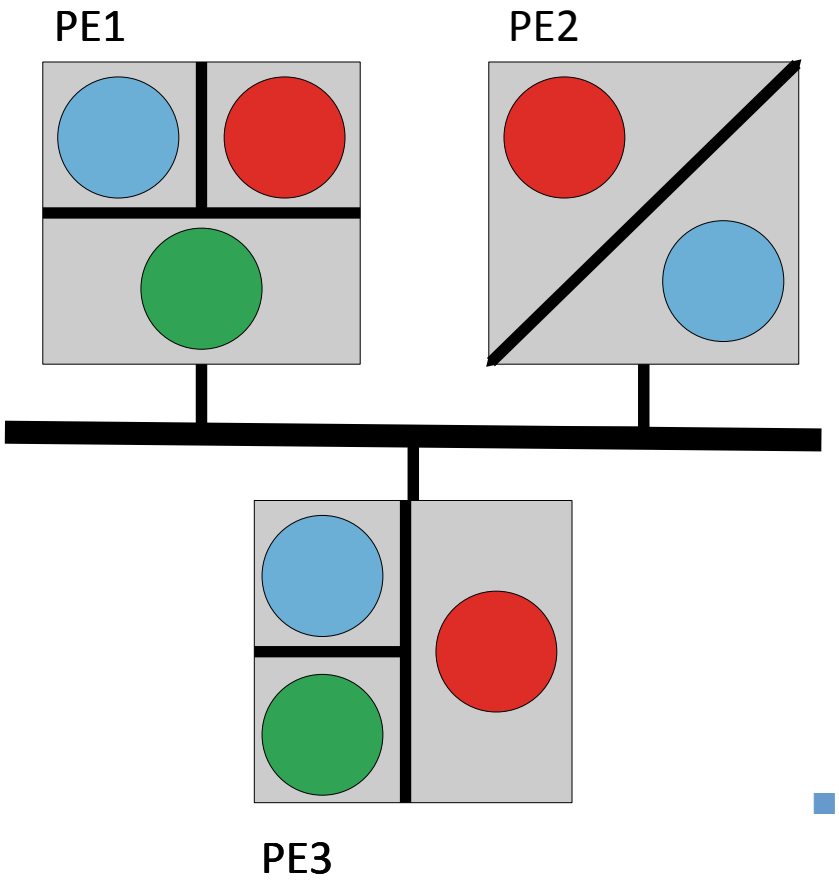
- Temporal partitioning
  - Static partition table

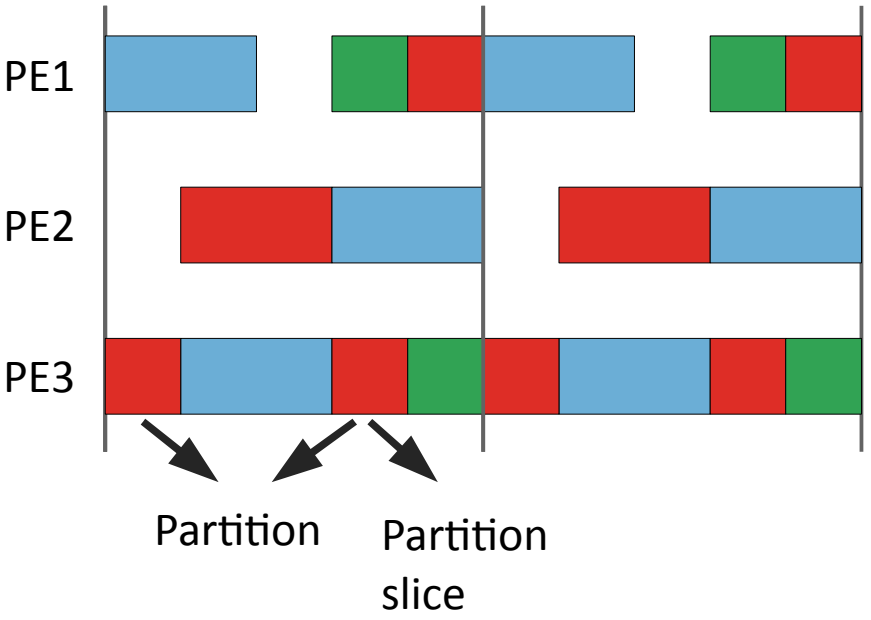
Safety-critical application

Mission-critical application

Non-critical application

System Model : Time Partitions





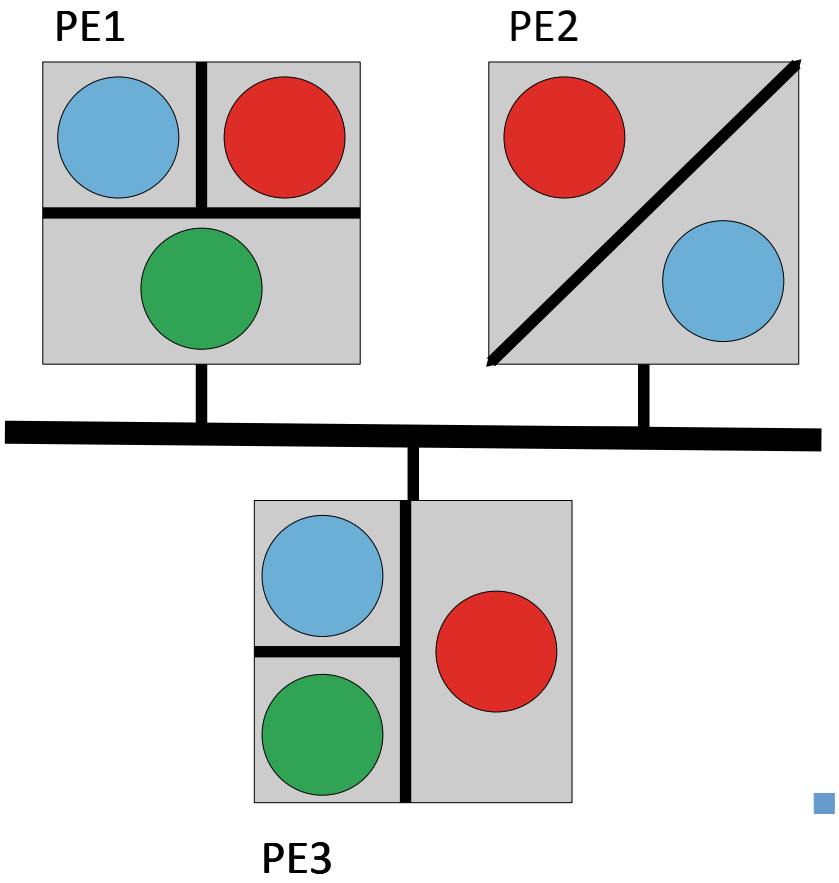
Safety-critical application

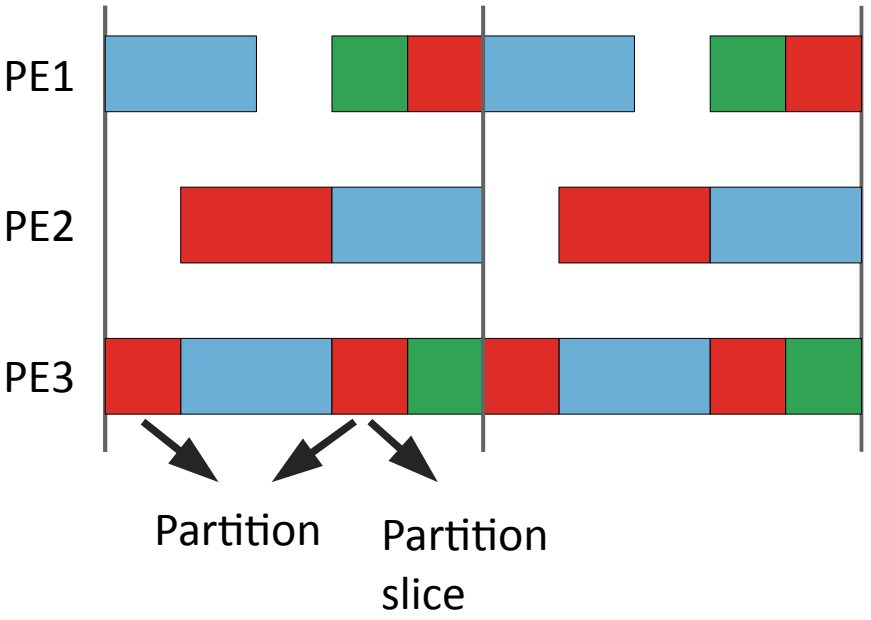
Mission-critical application

Non-critical application

- Temporal partitioning
  - Static partition table
    - For each PE

System Model : Time Partitions





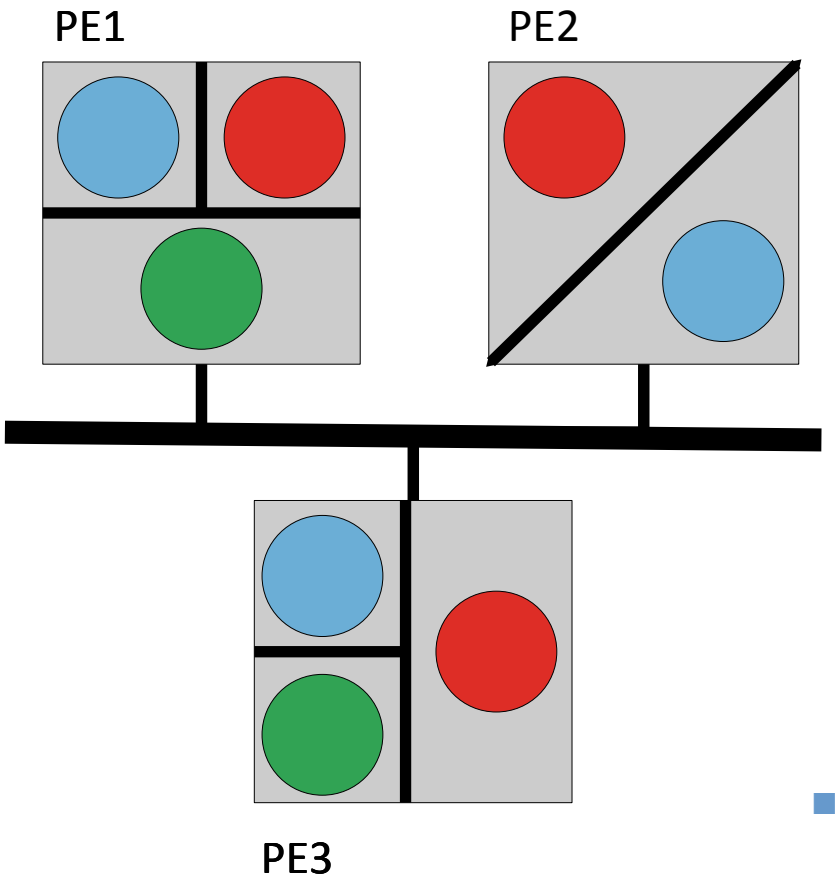
Safety-critical application

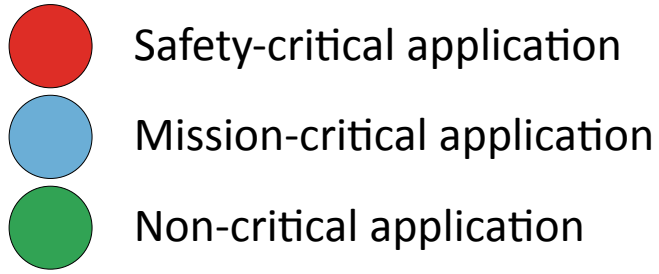
Mission-critical application

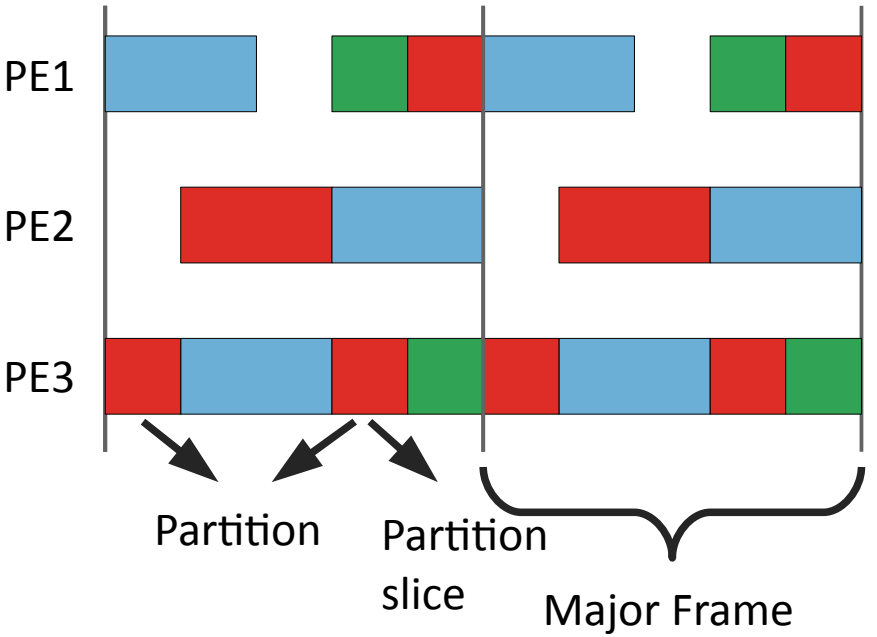
Non-critical application

- Temporal partitioning
  - Static partition table
    - For each PE
    - Each partition can have its own scheduling policy

System Model : Time Partitions

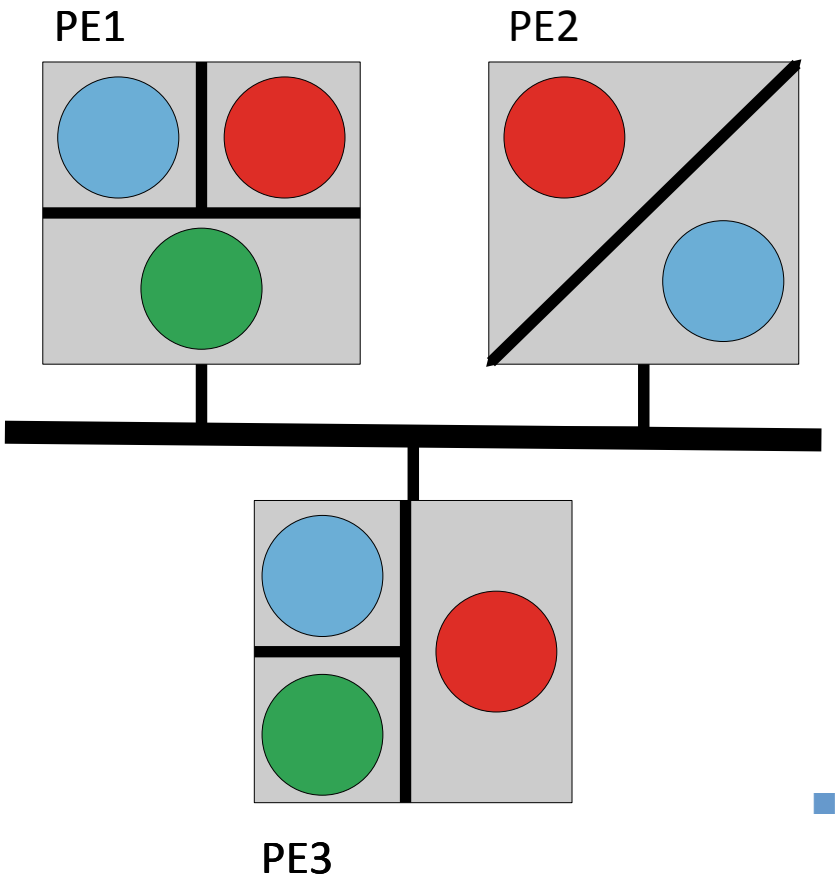


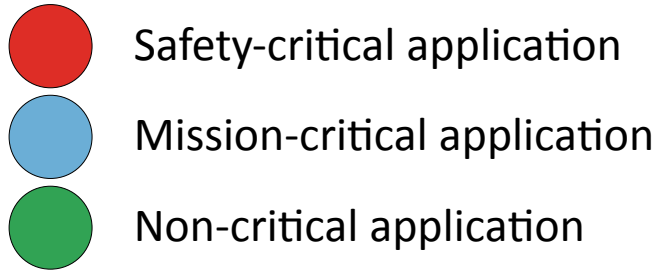


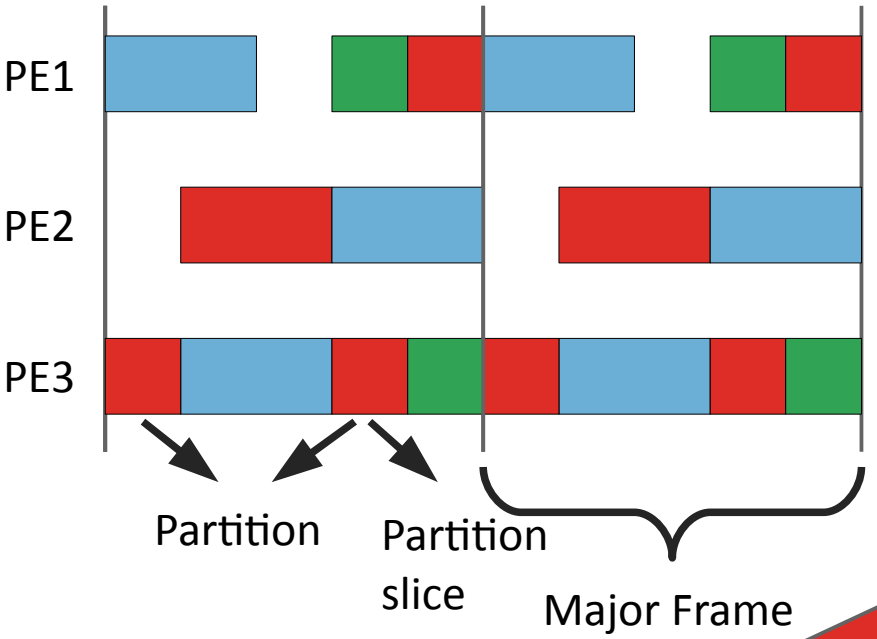


- Temporal partitioning
  - Static partition table
    - For each PE
    - Each partition can have its own scheduling policy
    - Repeated with a period MF

System Model : Time Partitions





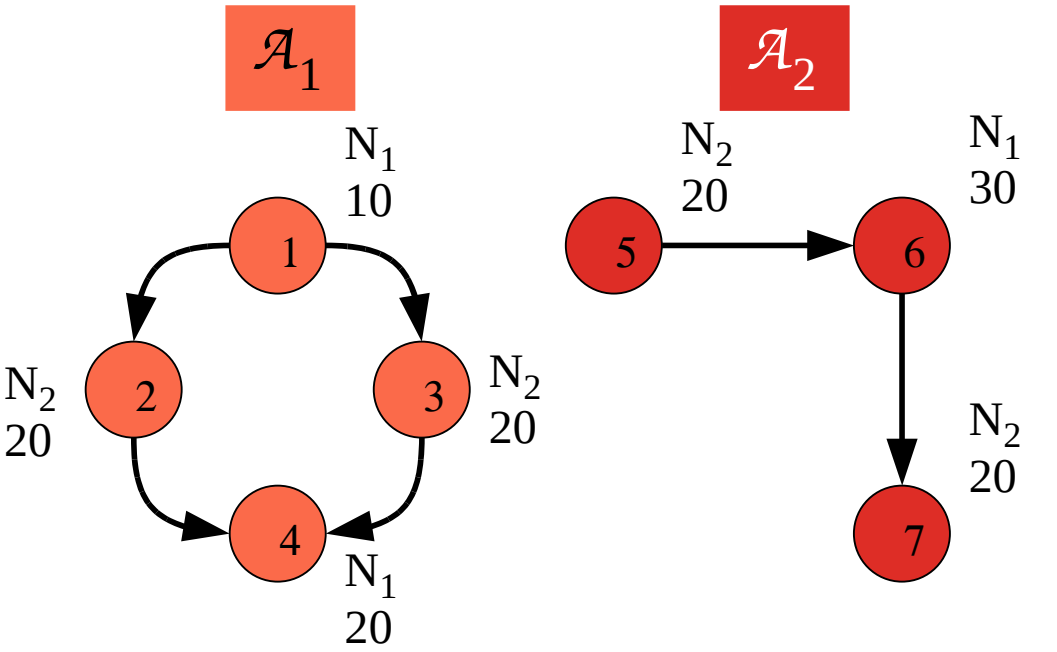


- Temporal partitioning
  - Static partition
    - For ea
    - can have its own
      ling policy
      Repeated with a period MF

Problem: optimize time partitions

Application Model

Safety-Critical Applications
Static Cyclic Scheduling



$T_{\mathcal{A}_1} = D_{\mathcal{A}_1} = 100$

$T_{\mathcal{A}_2} = D_{\mathcal{A}_2} = 120$

Non-Critical Applications
Fixed Priority Scheduling

$\mathcal{A}_3$

	C_i	D_i	T_i	Priority	PE
8	10	30	60	1	N_1
9	20	60	120	2	N_2
10	10	100	120	3	N_1
11	10	120	240	4	N_2

Problem Formulation

- Given
  - A set of applications of mixed criticality levels
  - A set of N processing elements (PEs)
  - Mapping of tasks to PEs
  - The size of the Major Frame and of the Application Cycle

Problem Formulation

- Given
  - A set of applications of mixed criticality levels
  - A set of N processing elements (PEs)
  - Mapping of tasks to PEs
  - The size of the Major Frame and of the Application Cycle

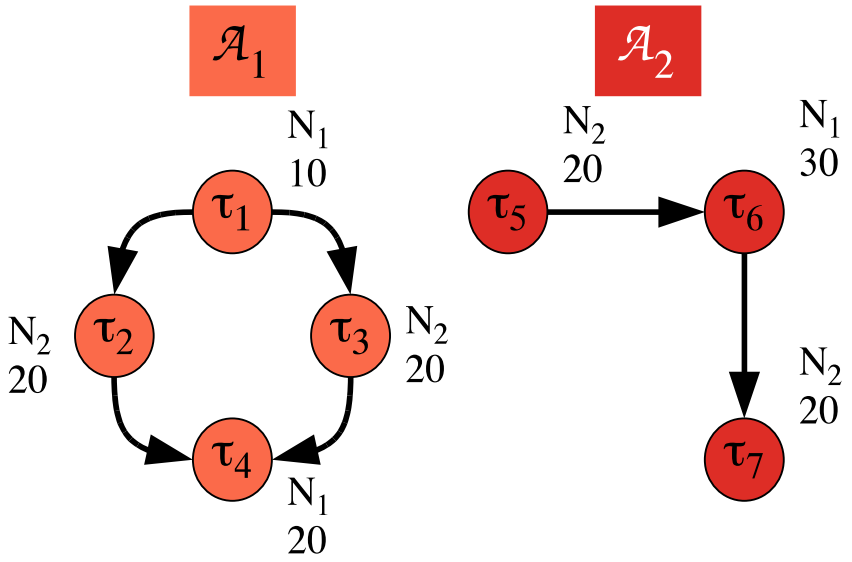
- Determine
  - The allocation of applications to partitions
  - Sequence and length of partition slices for each PE
  - Schedule for the SC applications

Problem Formulation

- Given
  - A set of applications of mixed criticality levels
  - A set of N processing elements (PEs)
  - Mapping of tasks to PEs
  - The size of the Major Frame and of the Application Cycle

- Determine
  - The allocation of applications to partitions
  - Sequence and length of partition slices for each PE
  - Schedule for the SC applications
- Such that
  - SC and NC applications are schedulable
  - The available slack is maximized (for upgrades)

Motivational Example



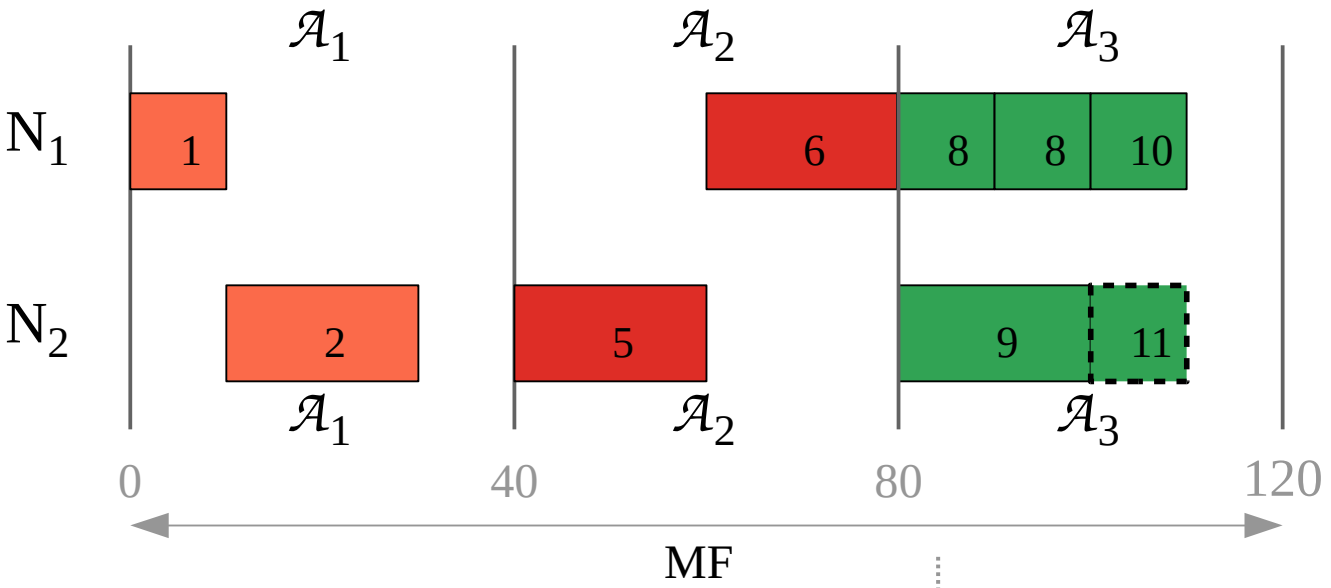
$T_{\mathcal{A}_1} = D_{\mathcal{A}_1} = 100$

$T_{\mathcal{A}_2} = D_{\mathcal{A}_2} = 120$

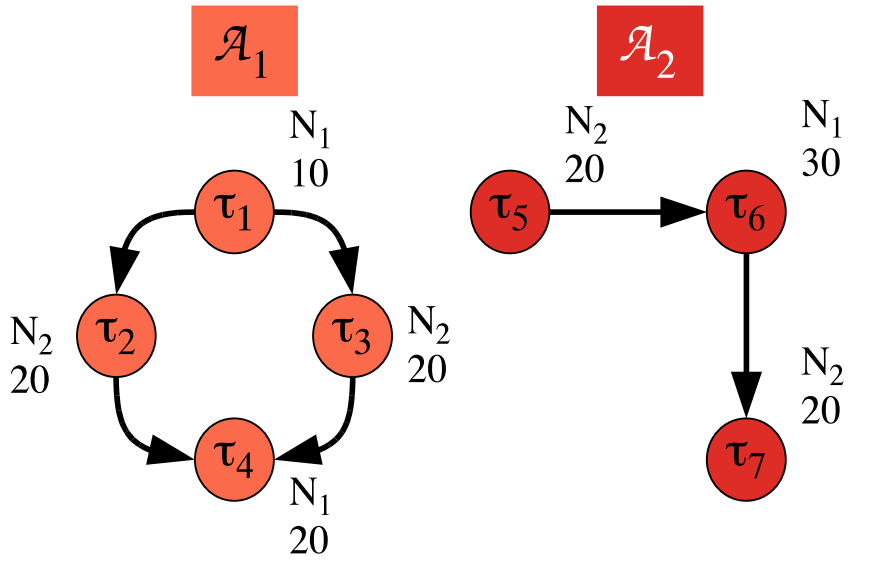
$\mathcal{A}_3$

	C_i	D_i	T_i	Priority	PE
τ_8	10	30	60	1	N_1
τ_9	20	60	120	2	N_2
τ_{10}	10	100	120	3	N_1
τ_{11}	10	120	240	4	N_2

Motivational Example



All applications miss their deadlines. Deadlines are missed by 3, 4, 6, 7, 8 (both jobs), 9, 10



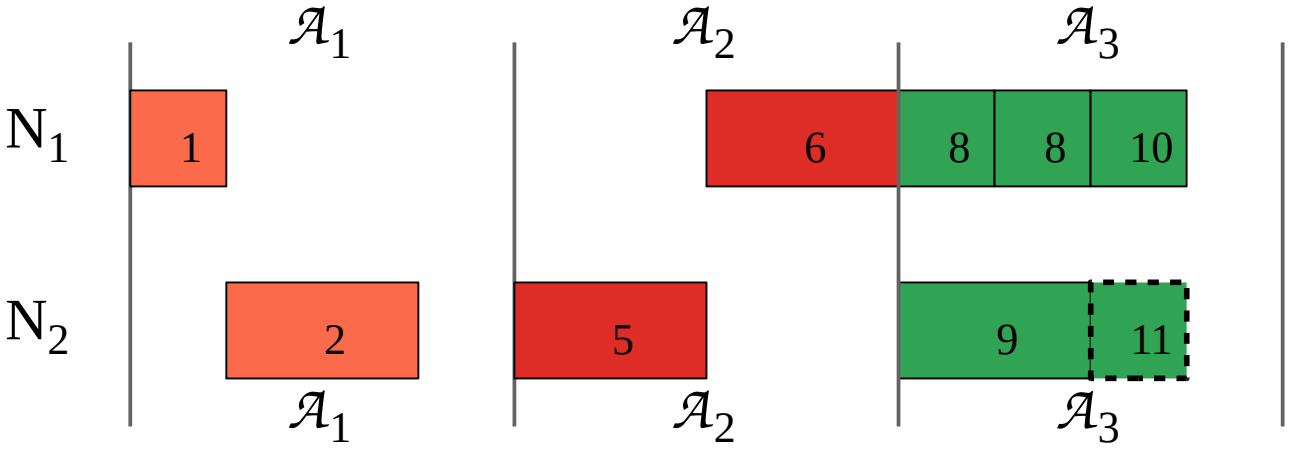
$T_{\mathcal{A}_1} = D_{\mathcal{A}_1} = 100$

$T_{\mathcal{A}_2} = D_{\mathcal{A}_2} = 120$

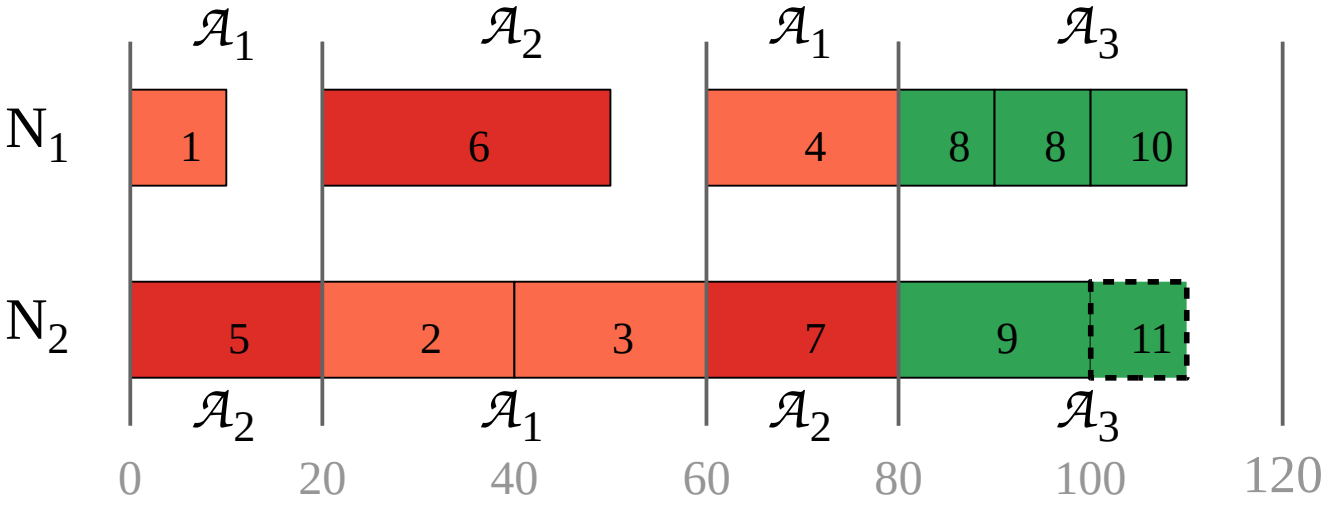
$\mathcal{A}_3$

	C_i	D_i	T_i	Priority	PE
τ_8	10	30	60	1	N_1
τ_9	20	60	120	2	N_2
τ_{10}	10	100	120	3	N_1
τ_{11}	10	120	240	4	N_2

Motivational Example

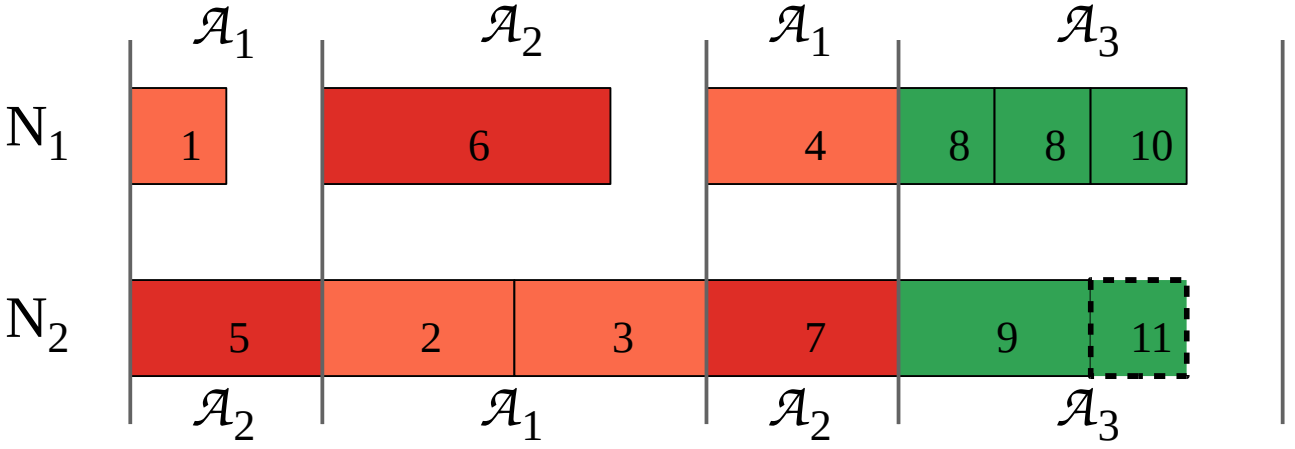


All applications miss their deadlines. Deadlines are missed by 3, 4, 6, 7, 8 (both jobs), 9, 10

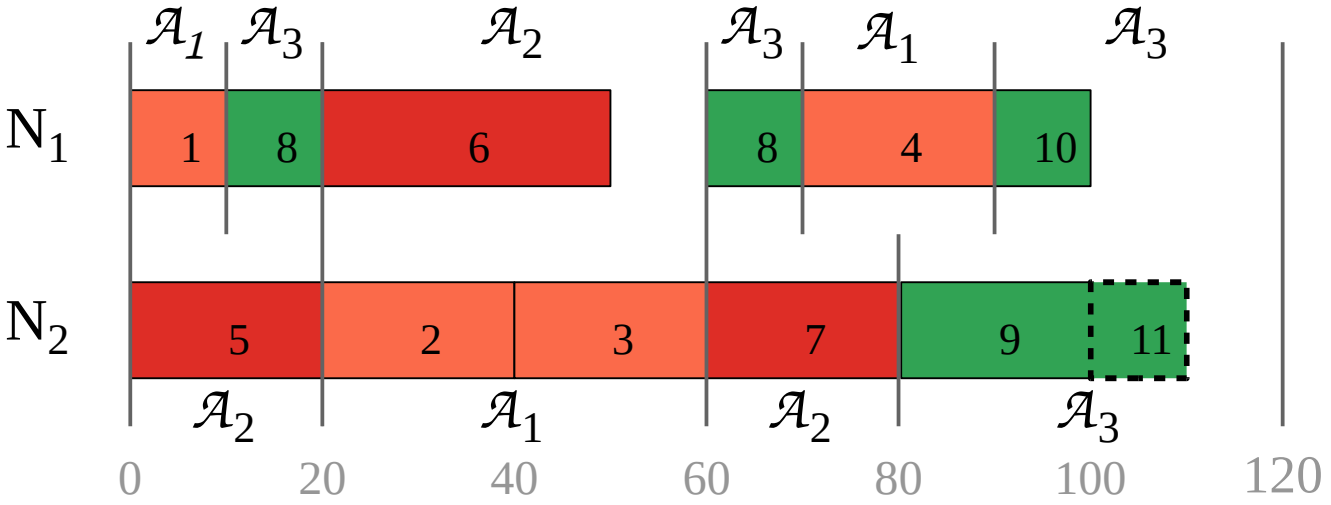


SC applications meet their deadlines. NC application miss theirs. Deadlines are missed by: 8, 9 and 10

Motivational Example

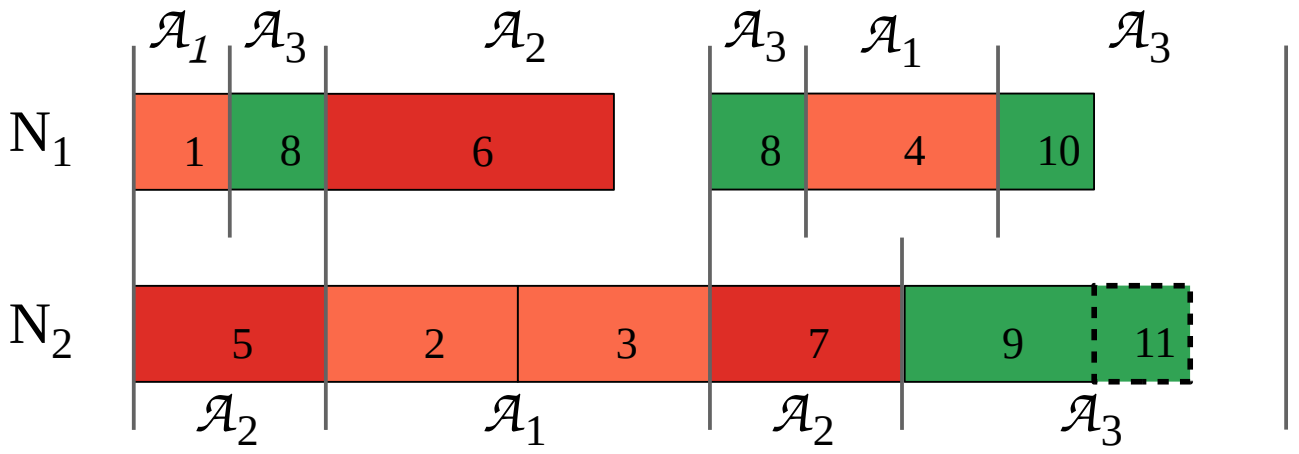


SC applications meet their deadlines. NC application miss theirs. Deadlines are missed by: $_8$, $_9$ and $_{10}$

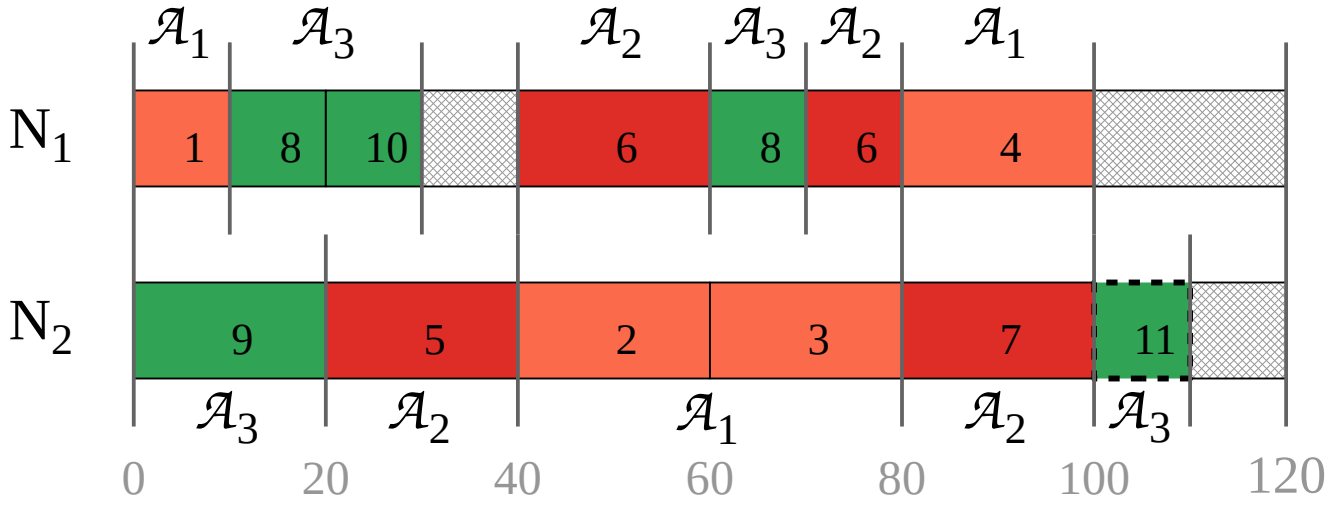


Deadlines are missed only by the NC task $_9$ on N_2. All other tasks meet their deadlines

Motivational Example



Deadlines are missed only by the NC task 9 on N_2. All other tasks meet their deadlines



Deadlines are met for the all the applications

Optimization Strategy

- Time Partition Optimization (TPO) strategy:
  - Simulated Annealing meta-heuristic
    - The allocation of applications to partitions
    - Sequence and length of partition slices for each PE
  - List scheduling
    - Schedule for the SC applications

Optimization Strategy

- Time Partition Optimization (TPO) strategy:
  - Simulated Annealing meta-heuristic
    - The allocation of applications to partitions
    - Sequence and length of partition slices for each PE
  - List scheduling
    - Schedule for the SC applications

- Simulated Annealing
  - Minimizes the cost function
  - Explores the solution space using design transformations

Optimization Strategy: Cost Function

- Degree of schedulability
  - Captures the difference between the worst case response time and the deadline

The response time for SC is obtained through List Scheduling.

The response time for NC is obtained using a Response Time Analysis.

Optimization Strategy: Cost Function

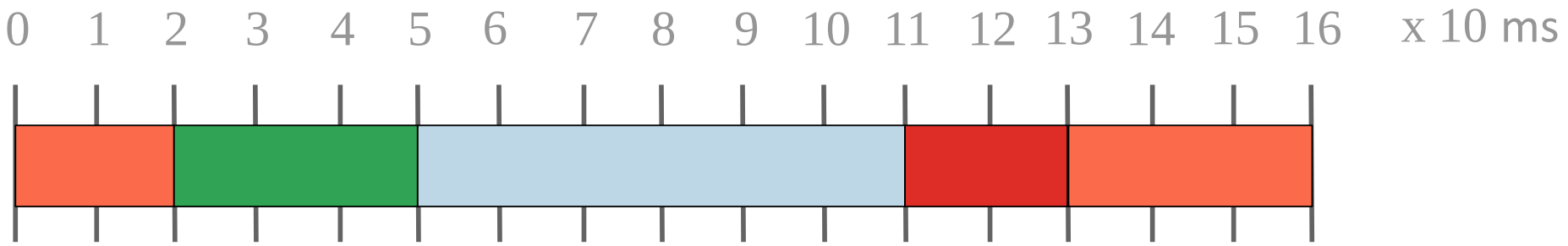
- Degree of schedulability
  - Captures the difference between the worst case response time and the deadline

The response time for SC is obtained through List Scheduling.

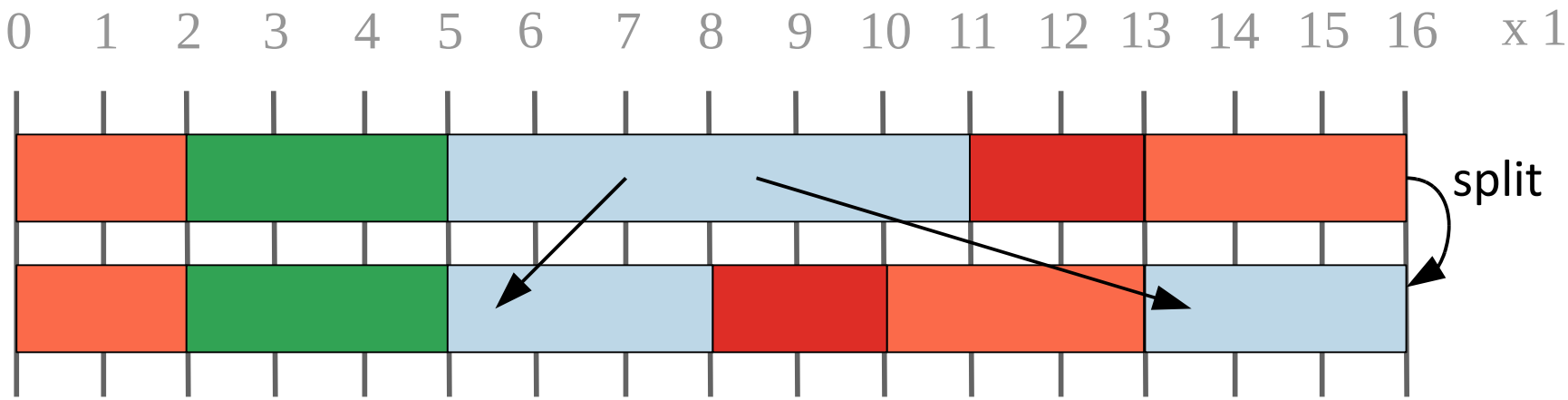
The response time for NC is obtained using a Response Time Analysis.

- Cost Function
  - Weighted combination of the degree of schedulability for SC and NC applications

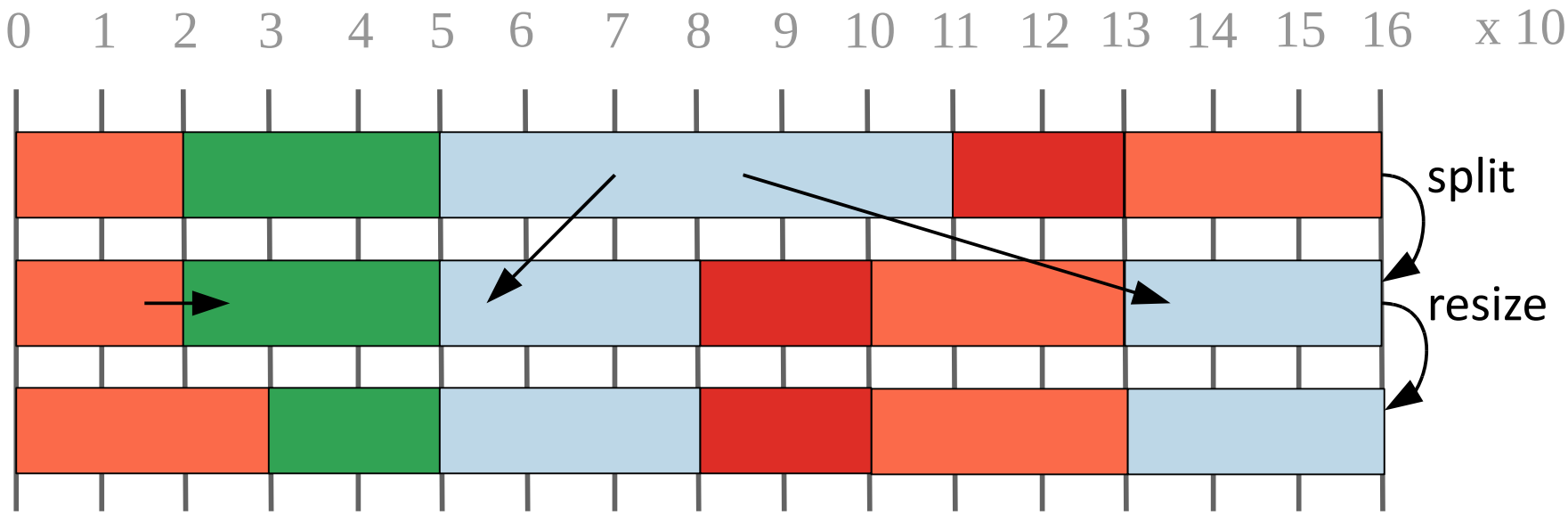
Optimization Strategy: Design Transformations



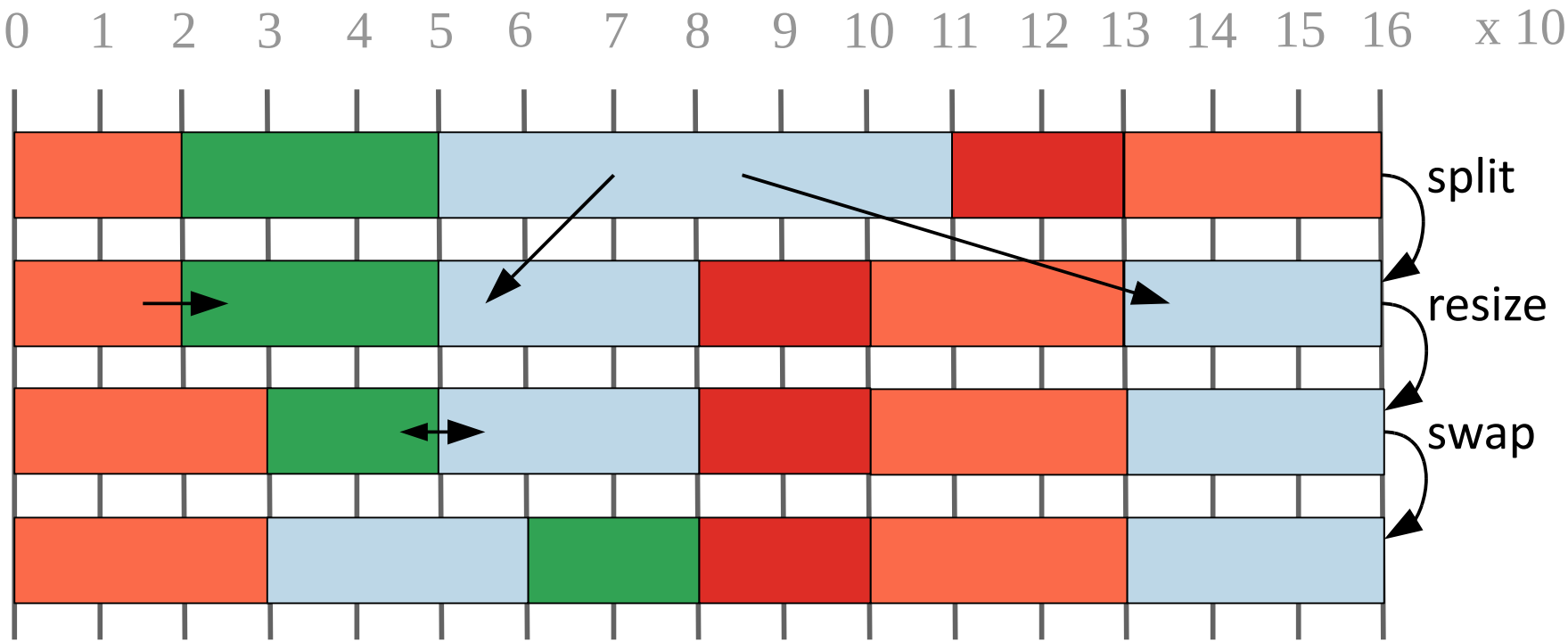
Optimization Strategy: Design Transformations



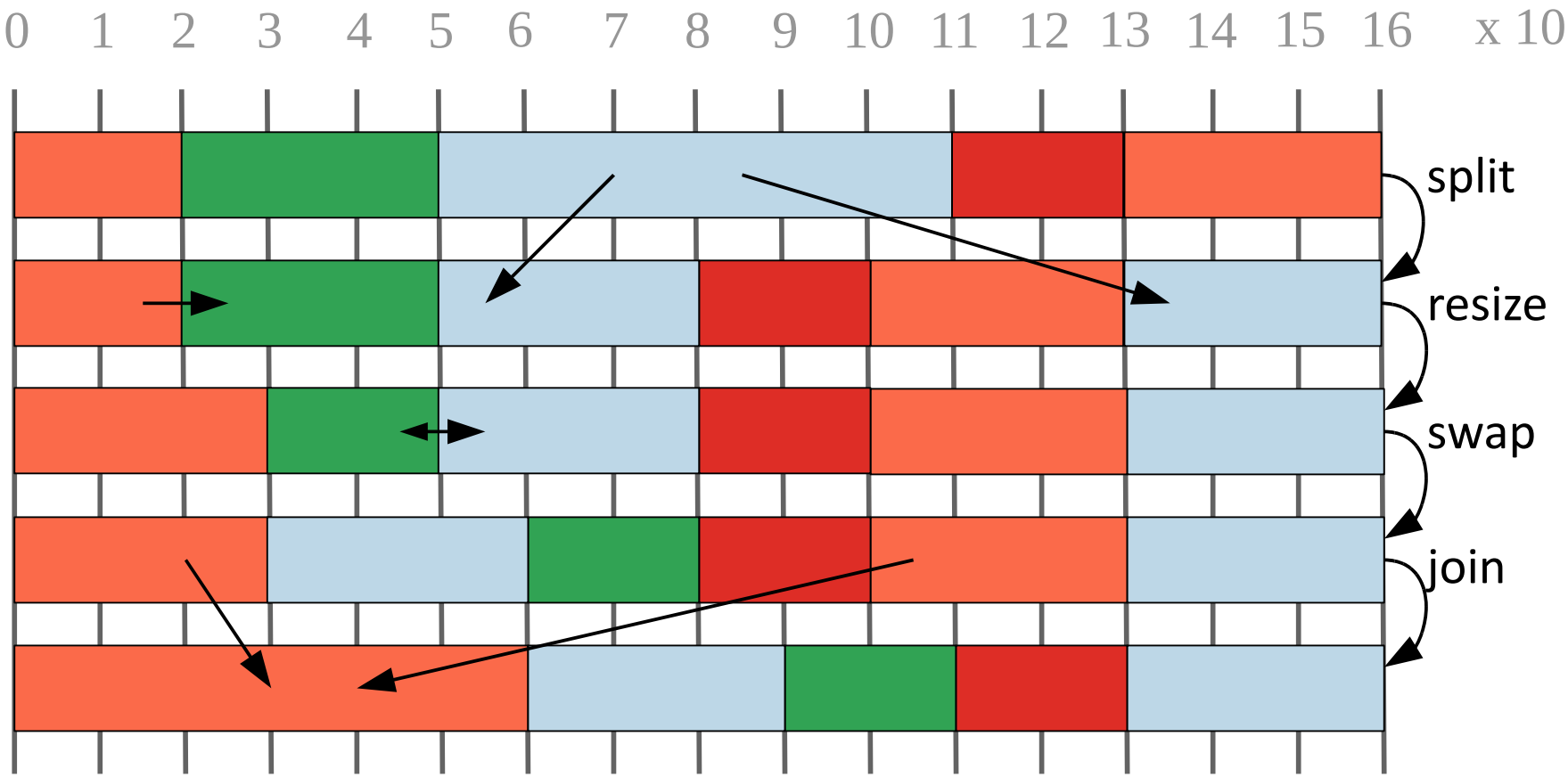
Optimization Strategy: Design Transformations



Optimization Strategy: Design Transformations



Optimization Strategy: Design Transformations



Experimental Results

- Benchmarks
  - 10 synthetic
  - 2 real life test cases from E3S
- Evaluated using
  - Straightforward Solution (SS)
    - Allocates to each application a partition size proportional to the utilization of the tasks mapped on that PE
  - Time Partition Optimization (TPO)
    - For 120 minutes
    - $w_{SC} = 400$
    - $w_{NC} = 100$

Experimental Results

Set	SC		NC	PE	SS		TPO (120 min.)	
	Apps	Tasks	Tasks		Sched. SC Apps	Sched. NC tasks	Sched. SC Apps	Sched. NC Tasks
1	3	15	5	2	1 of 3	All	All	All
	3	20	6	3	1 of 3	All	All	All
	4	34	6	4	None	All	All	All
	4	40	10	5	None	All	All	All
	5	53	9	6	3 of 5	All	All	All

Experimental Results

Set	SC		NC	PE	SS		TPO (120 min.)	
	Apps	Tasks	Tasks		Sched. SC Apps	Sched. NC tasks	Sched. SC Apps	Sched. NC Tasks
1	3	15	5	2	1 of 3	All	All	All
	3	20	6	3	1 of 3	All	All	All
	4	34	6	4	None	All	All	All
	4	40	10	5	None	All	All	All
	5	53	9	6	3 of 5	All	All	All
2	1	6	6	4	All	All	All	All
	2	12	6	4	All	All	All	All
	3	20	6	4	None	5 of 6	All	All
	4	30	6	4	1 of 4	All	All	All
	5	34	6	4	2 of 5	5 of 6	All	All

Experimental Results

Set	SC		NC	PE	SS		TPO (120 min.)	
	Apps	Tasks	Tasks		Sched. SC Apps	Sched. NC tasks	Sched. SC Apps	Sched. NC Tasks
1	3	15	5	2	1 of 3	All	All	All
	3	20	6	3	1 of 3	All	All	All
	4	34	6	4	None	All	All	All
	4	40	10	5	None	All	All	All
	5	53	9	6	3 of 5	All	All	All
2	1	6	6	4	All	All	All	All
	2	12	6	4	All	All	All	All
	3	20	6	4	None	5 of 6	All	All
	4	30	6	4	1 of 4	All	All	All
	5	34	6	4	2 of 5	5 of 6	All	All
3	3	19	5	3	None	All	All	All
	4	19	6	3	All	All	All	All

Experimental Results

Set	SC		NC	PE	SS		TPO (120 min.)				
	Apps	Tasks	Tasks		Sched. SC Apps	Sched. NC tasks	Sched. SC Apps	Sched. NC Tasks	Δ_{SC}	Δ_{NC}	Avg. % increase in degree of sched.
1	3	15	5	2	1 of 3	All	All	All	1709.76	-44.00	832.88
	3	20	6	3	1 of 3	All	All	All	107.94	-53.23	27.36
	4	34	6	4	None	All	All	All	169.68	7.14	88.41
	4	40	10	5	None	All	All	All	147.54	-0.40	73.57
	5	53	9	6	3 of 5	All	All	All	542.78	14.66	278.72
2	1	6	6	4	All	All	All	All	78.38	0.00	39.19
	2	12	6	4	All	All	All	All	59.20	-2.87	28.17
	3	20	6	4	None	5 of 6	All	All	518.06	1453.85	985.96
	4	30	6	4	1 of 4	All	All	All	211.66	0.00	105.83
	5	34	6	4	2 of 5	5 of 6	All	All	466.36	673.33	569.85
3	3	19	5	3	None	All	All	All	227.33	0.57	113.95
	4	19	6	3	All	All	All	All	135.29	-11.56	61.87

Conclusions

- Mixed criticality systems, with safety-critical and non-critical applications running on the same processor, are implemented using a partitioned architecture.
  - The safety-critical applications are scheduled using Static Cyclic Scheduling.
  - The non-critical applications are scheduled using Fixed Priority Scheduling.
- We proposed a SA based optimization of time partitions.

Optimization of time partitions is needed for mixed criticality applications implemented on partitioned systems.

Future work

- Map tasks to PEs
- Allow partition slice sharing among applications
- Consider separation requirements
- Consider certification costs

Thank you!

Domițian Tămaș-Selicean
dota@imm.dtu.dk

Paul Pop
paul.pop@imm.dtu.dk