Maximizing Level of Confidence for Non-Equidistant Checkpointing

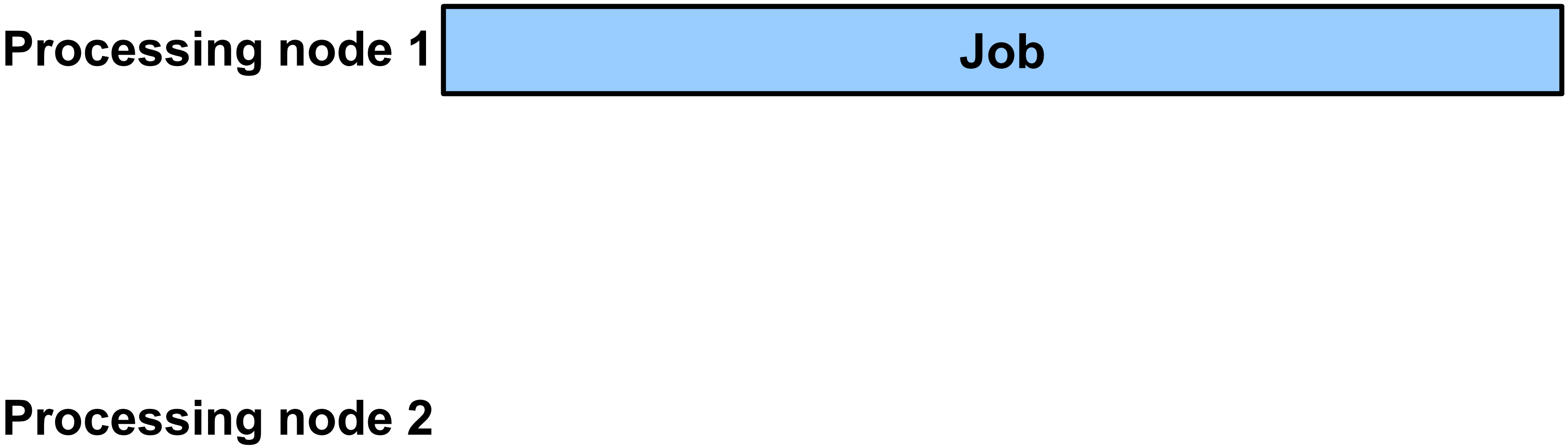
DIMITAR NIKOLOV AND ERIK LARSSON

Roll-back Recovery with Checkpointing

- Efficient technique that copes with soft errors
- Capable of detection and recovery of soft errors
- Checkpoint, an intermediate state of a job
- **Error detection:** compare checkpoints from two processors on which a job is concurrently running
- **Error recovery:** re-execute a portion of the job
- **Introduces a time overhead**

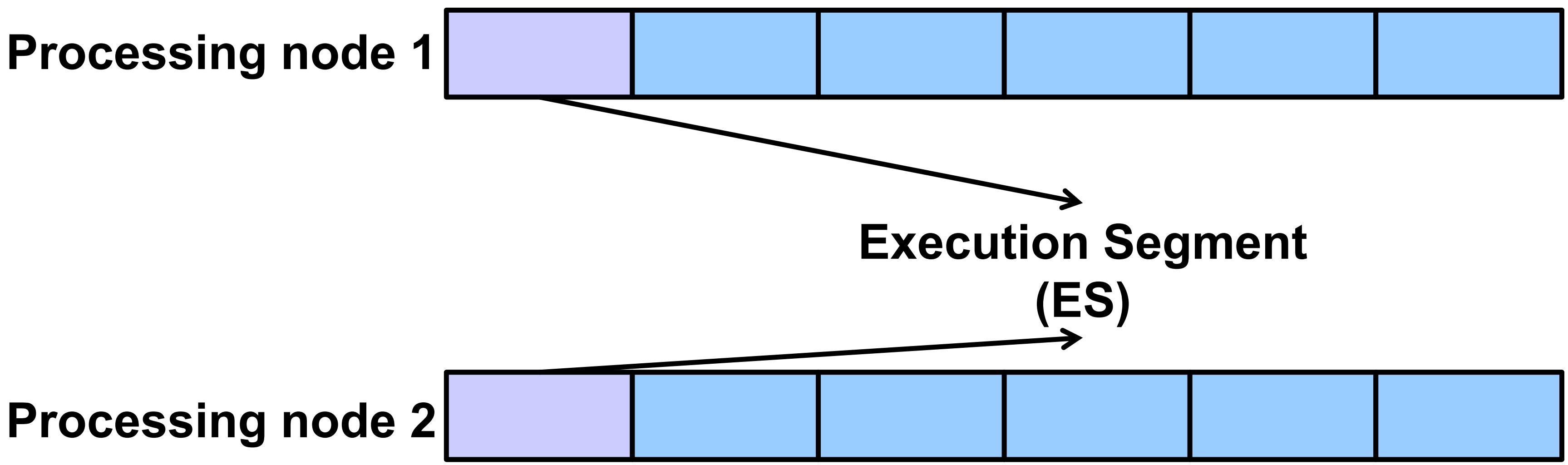
Roll-back Recovery with Checkpointing

- Job is duplicated and executed concurrently on two processing nodes



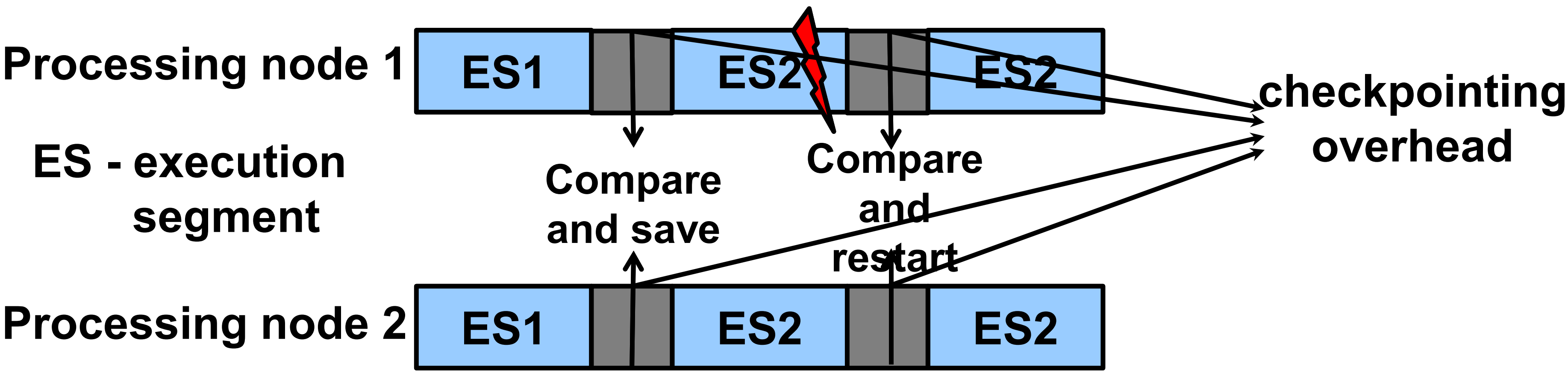
Roll-back Recovery with Checkpointing

- Job is divided into execution segments, based on the number of checkpoints



Roll-back Recovery with Checkpointing

- After each execution segment, the checkpoints from both processing nodes are compared
- If the checkpoints match, they are saved, and the job continues with its execution
- If the checkpoints do not match, the job is restarted from the latest saved checkpoint



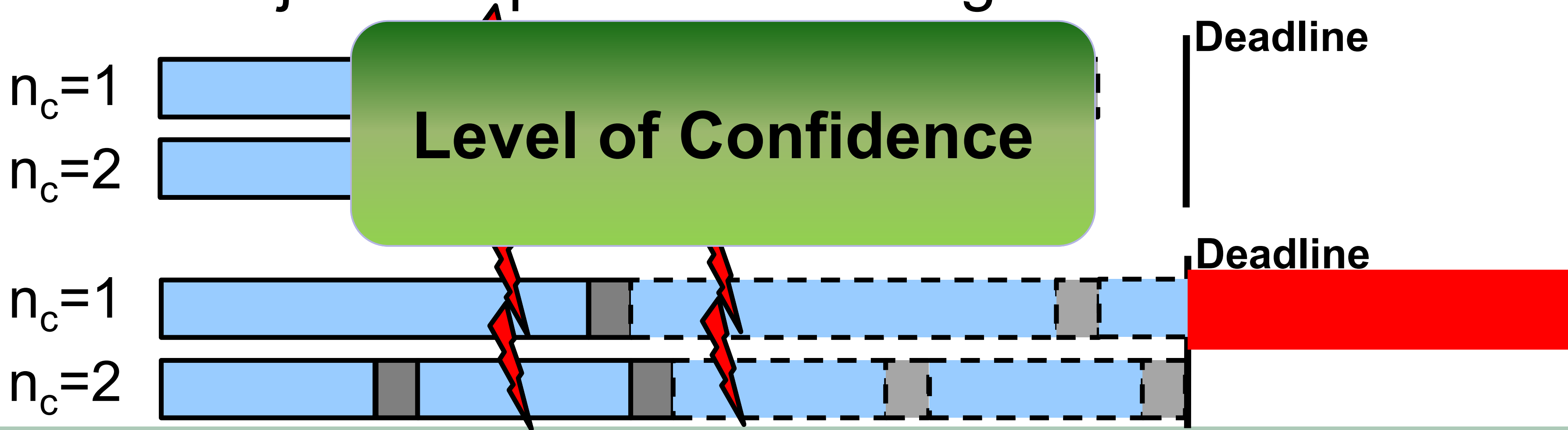
Roll-back Recovery with Checkpointing in Real-Time Systems

- The number of checkpoints, $\mathbf{n_c}$, may be the reason to cause deadline violation

To what extent a deadline is met?

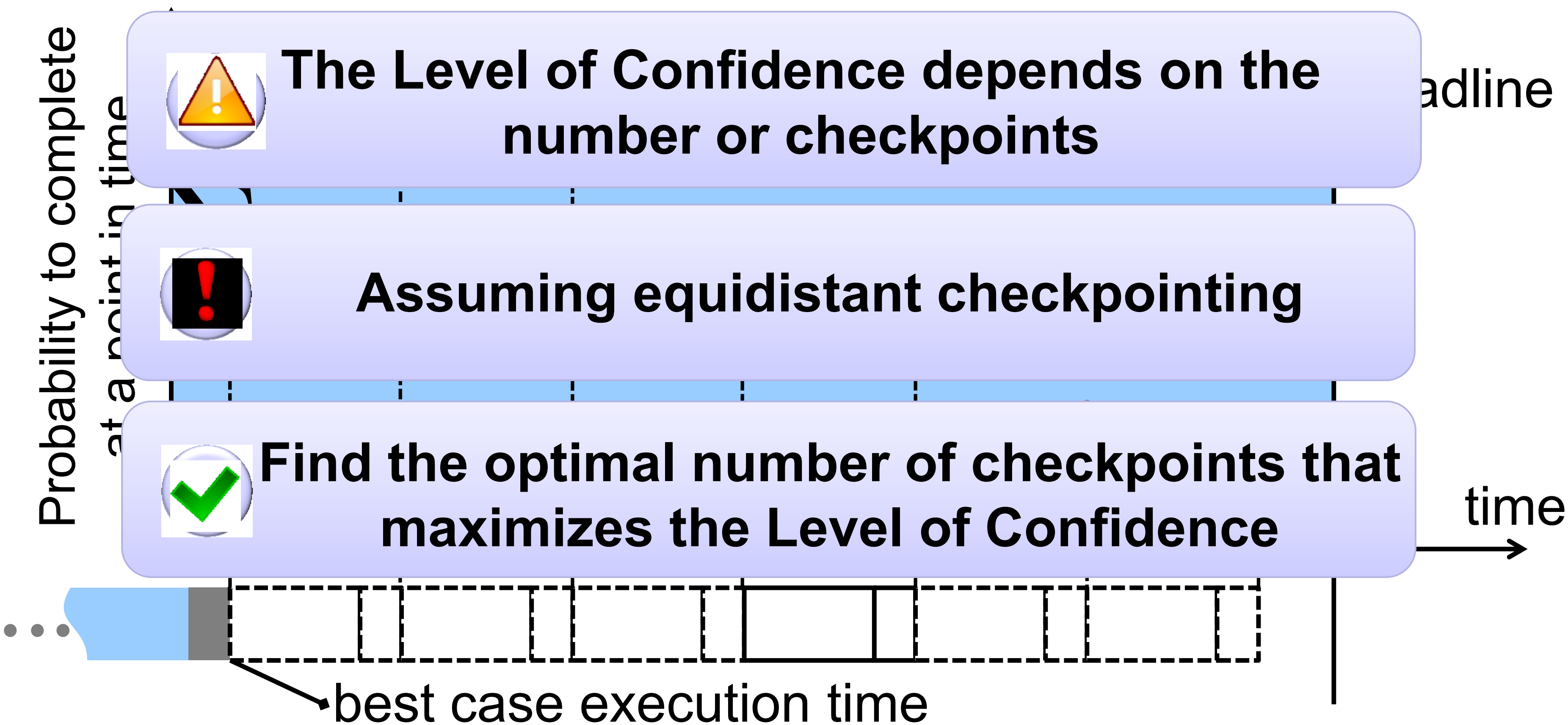
- The number of checkpoints impacts the probability that the job completes before a given deadline

Level of Confidence



Level of Confidence

- Probability that a job completes before a deadline



The Level of Confidence depends on the number or checkpoints

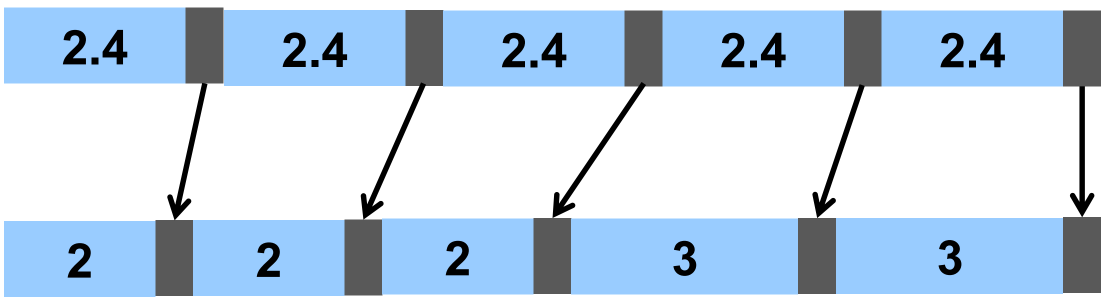
Assuming equidistant checkpointing

Find the optimal number of checkpoints that maximizes the Level of Confidence

Motivation

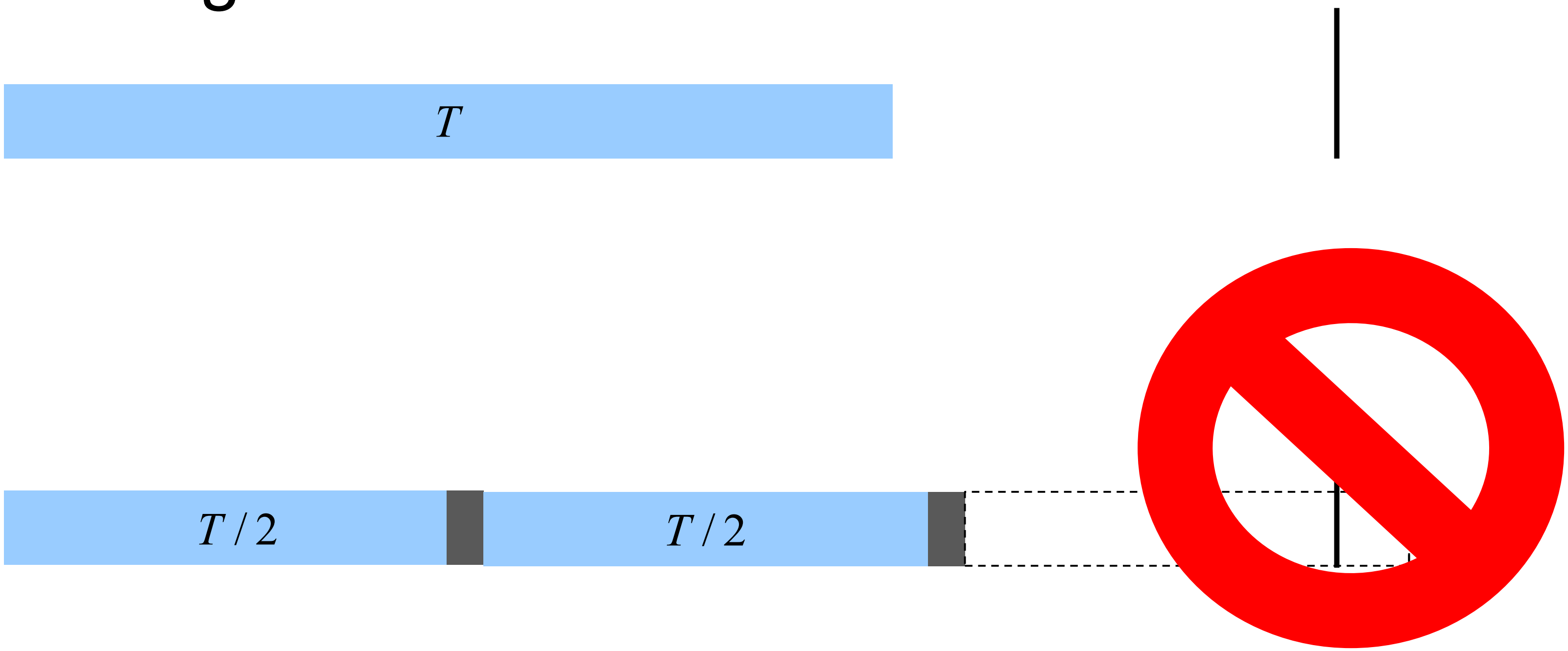
- A checkpoint cannot be taken before an on-going instruction completes
- Assumption: each instruction takes one time unit to complete
- I… y?
- Consider that $n_c=5$ and the processing time $T=12$ time unites

Non-equidistant checkpointing has to be considered



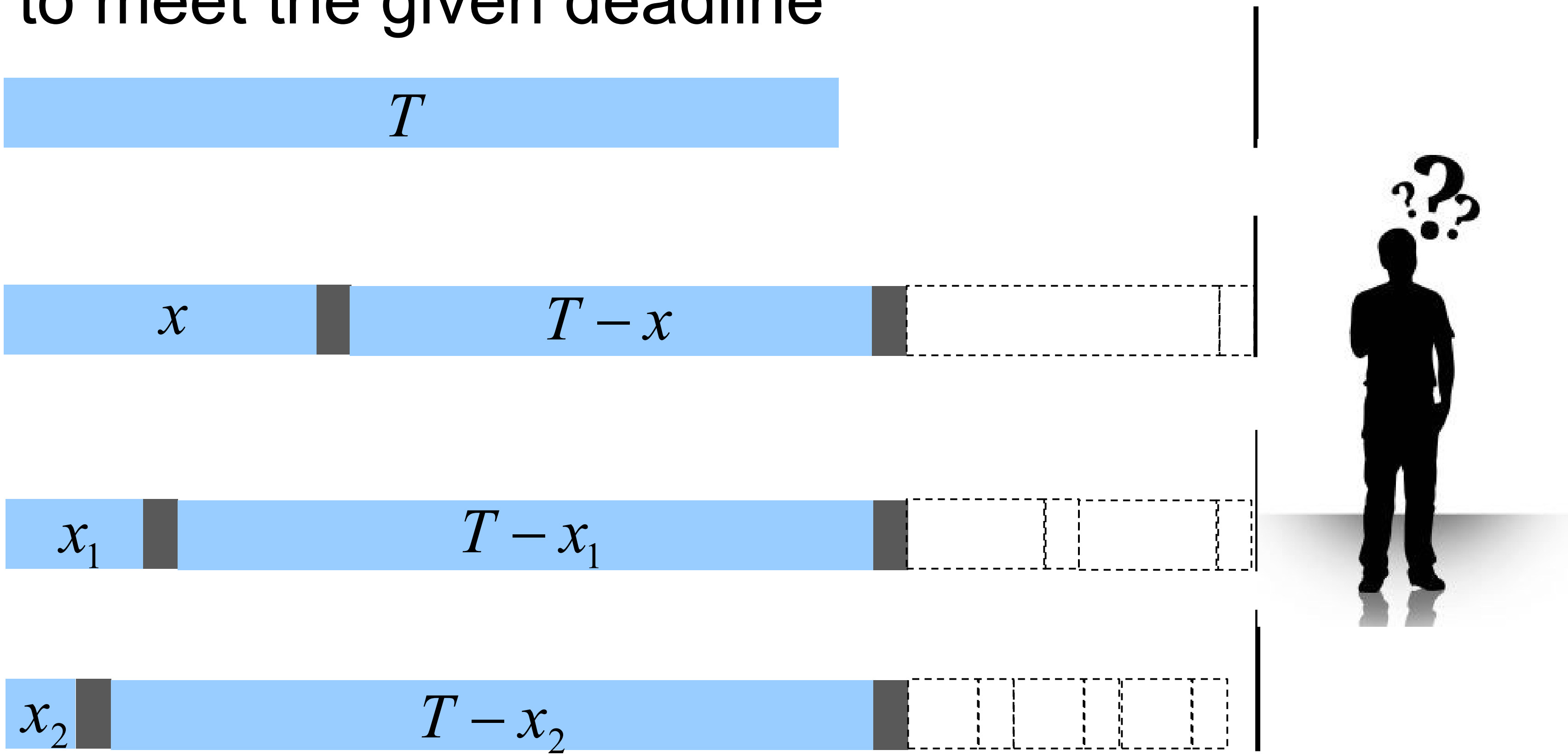
Motivation

- Given that n_c=2, what is the Level of Confidence to meet a given deadline



Motivation

- Given that n_c=2, what is the Level of Confidence to meet the given deadline



Problem Formulation

- Given the following inputs:
  - T, processing time
  - D, deadline
  - τ , checkpointing overhead
  - P_T, probability that no soft errors occur in interval T
  - n_c, number of checkpoints

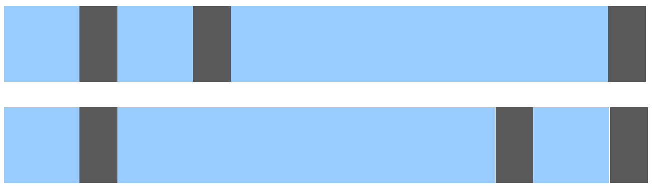
find the optimal distribution of the given number of checkpoints that maximizes the Level of Confidence

Overview

- Exhaustive Search
- Clustered Checkpointing
- Results
- Conclusion

Exhaustive Search

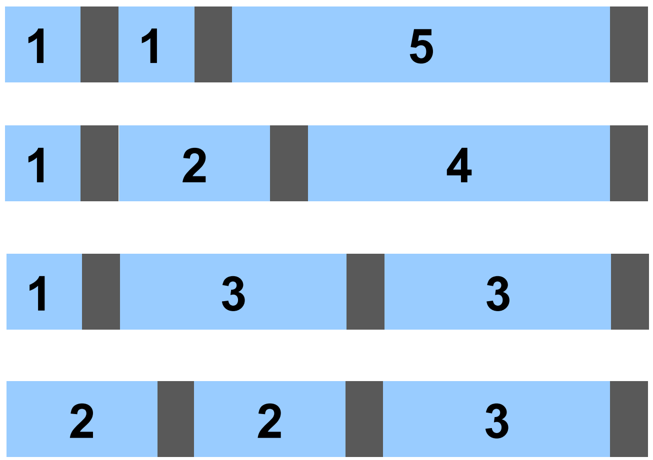
- Explore all possible distributions of a number of checkpoints and for each distribution calculate the Level of Confidence with respect to the deadline
- Guarantees to find the optimal solution
- Some distributions are equivalent



- Sufficient to explore distributions where the execution segments are ordered in ascending order according to their size

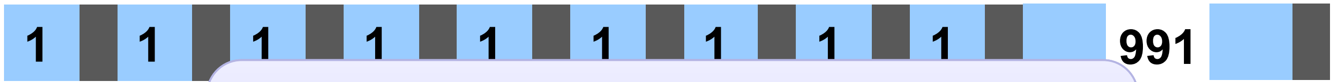
Exhaustive Search

- How many ways to distribute n_c=3 checkpoints for a job with a processing time is T=7 time units, assuming that a checkpoint can be taken only at integer time units?



Exhaustive Search

- How many ways to distribute n_c=10 checkpoints when the processing time is T=1000?



Not efficient!!!

88674569666532253

Overview

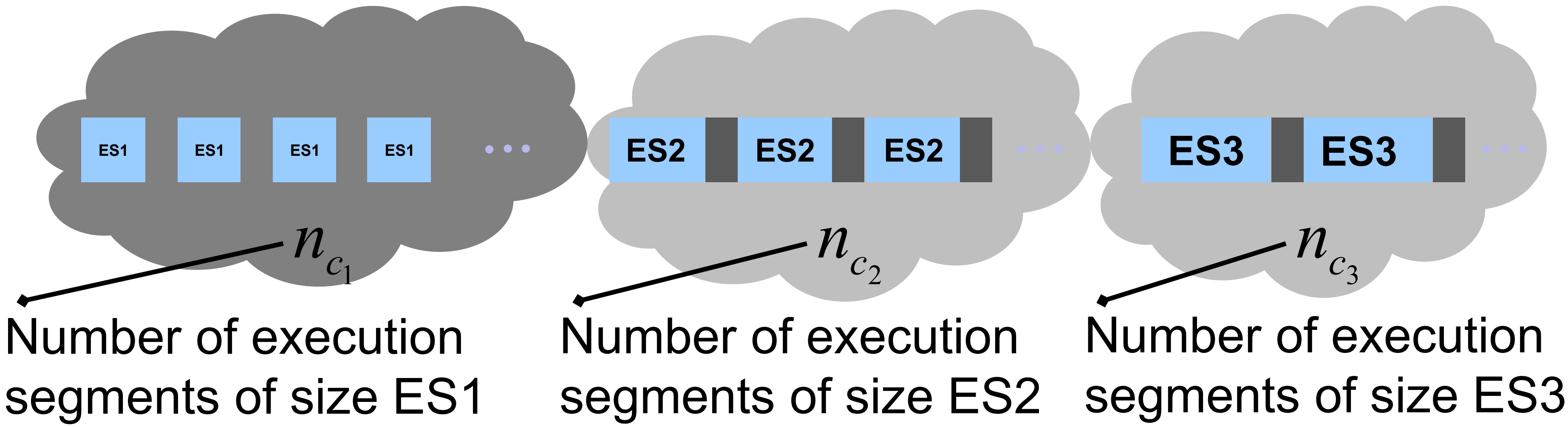
- Exhaustive Search
- Clustered Checkpointing
- Results
- Conclusion

Clustered Checkpointing

- It is a heuristic, thus does not guarantee optimality
- Explores only a limited set of distributions of a given number of checkpoints
- Number of distributions is of the same order as the processing time of the job which is given as input
- Arrives at a solution at much shorter computation time when compared against exhaustive search
- In most cases, provides the optimal solution

Clustered Checkpointing

- Explores only distributions made out of clusters, such that all the execution segments that belong to the same cluster have the same size
- Distributions made out of at most 3 clusters



Overview

- Exhaustive Search
- Clustered Checkpointing
- Results
- Conclusion

Results

- Comparing the achieved Level of Confidence (LoC) at various number of checkpoints for:
  - EQC (equidistant checkpointing)
  - EXS (Exhaustive search, *optimal solution)
  - CC (Clustered Checkpointing, heuristic)
- Given inputs:
  - T =100 t.u. (time units)
  - D =150 t.u.
  - τ =2 t.u.
  - P_T = 0.99999

Equidistant vs Non-Equidistant Checkpointing

n_c	EQC	Non-Equidistant Checkpointing	
		EXS	CC
2	1.99999e-5 $\tilde{n}_c = [50^2]$		
3	2.66678e-10 $\tilde{n}_c = [33^2, 34^1]$		
9	[illegible]		
10	1.75999e-15 $\tilde{n}_c = [10^{10}]$		

n_c	EQC	Non-Equidistant Checkpointing	
		EXS	CC
11	1.71943e-15 $\tilde{n}_c = [9^{10}, 10^1]$		
12	1.68642e-15 $\tilde{n}_c = [8^8, 9^4]$		
[illegible]	[illegible]		
19	2.10599e-10 $\tilde{n}_c = [5^{14}, 6^5]$		

Non-Equidistant Checkpointing provides higher or at least equal LoC, but never worse, when compared against Equidistant Checkpointing

Numbers represent the probability to miss the deadline, thus lower values indicate higher LoC

Exhaustive Search vs. Clustered Checkpointing

n_c		Non-Equidistant Checkpointing	
		EXS	CC
2			
9			
10			

n_c		Non-Equidistant Checkpointing	
		EXS	CC
11			1.58908e-15 $\tilde{n}_c = [5^4, 11^4, 12^3]$
18			
19			

Clustered Checkpointing finds, in most cases, the optimal distribution of checkpoints in much shorter computation time than Exhaustive Search

Maximal LoC

n_c	EQC	n_c	EQC
2	1.99999e-5 $\tilde{n}_c = [50^2]$	11	1.71943e-15 $\tilde{n}_c = [9^{10}, 10^1]$
3	2.66678e-10 $\tilde{n}_c = [33^2, 34^1]$	12	1.68642e-15 $\tilde{n}_c = [8^8, 9^4]$
4	2.49998e-10 $\tilde{n}_c = [25^4]$	13	1.65807e-15 $\tilde{n}_c = [7^4, 8^9]$
5	2.39998e-10 $\tilde{n}_c = [20^5]$	14	1.63343e-15 $\tilde{n}_c = [7^{12}, 8^2]$
6	2.07477e-15 $\tilde{n}_c = [16^2, 17^4]$	15	1.61335e-15 $\tilde{n}_c = [6^5, 7^{10}]$
7	1.95991e-15 $\tilde{n}_c = [14^5, 15^2]$	16	1.59510e-15 $\tilde{n}_c = [6^{12}, 7^4]$
8	1.87599e-15 $\tilde{n}_c = [12^4, 13^4]$		
9	1.81113e-15 $\tilde{n}_c = [11^8, 12^1]$	18	1.75199e-10 $\tilde{n}_c = [5^8, 6^{10}]$
10	1.75999e-15 $\tilde{n}_c = [10^{10}]$	19	2.10599e-10 $\tilde{n}_c = [5^{14}, 6^5]$

Maximal LoC for EQC achieved for n_c=17

Maximal LoC

n_c	EQC	Non-Equidistant Checkpointing	
		EXS	CC
2	1.99999e-5 $\tilde{n}_c = [50^2]$	**1.12000e-5** $\tilde{n}_c = [44^1, 56^1]$	**1.12000e-5** $\tilde{n}_c = [44^1, 56^1]$
3	2.66678e-10 $\tilde{n}_c = [33^2, 34^1]$	**2.55998e-10** $\tilde{n}_c = [20^1, 40^2]$	**2.55998e-10** $\tilde{n}_c = [20^1, 40^2]$
4	2.49998e-10 $\tilde{n}_c = [25^4]$	**2.09559e-10** $\tilde{n}_c = [19^2, 31^2]$	**2.09559e-10** $\tilde{n}_c = [19^2, 31^2]$
5	2.39998e-10 $\tilde{n}_c = [20^5]$	**1.12000e-10** $\tilde{n}_c = [18^4, 28^1]$	**1.12000e-10** $\tilde{n}_c = [18^4, 28^1]$
6	2.07477e-15 $\tilde{n}_c = [16^2, 17^4]$	**2.07477e-15** $\tilde{n}_c = [16^2, 17^4]$	**2.07477e-15** $\tilde{n}_c = [16^2, 17^4]$
7	1.95991e-15 $\tilde{n}_c = [14^5, 15^2]$	**1.95161e-15** $\tilde{n}_c = [7^1, 15^3, 16^3]$	**1.95161e-15** $\tilde{n}_c = [7^1, 15^3, 16^3]$
8	1.87599e-15 $\tilde{n}_c = [12^4, 13^4]$	**1.83839e-15** $\tilde{n}_c = [7^2, 14^4, 15^2]$	**1.83839e-15** $\tilde{n}_c = [7^2, 14^4, 15^2]$
9	1.81113e-15 $\tilde{n}_c = [11^8, 12^1]$	**1.70906e-15** $\tilde{n}_c = [6^3, 13^2, 14^4]$	**1.70906e-15** $\tilde{n}_c = [6^3, 13^2, 14^4]$
	1.75999e-15 $\tilde{n}_c = [10^{10}]$		

n_c	EQC	Non-Equidistant Checkpointing	
		EXS	CC
11	1.71943e-15 $\tilde{n}_c = [9^{10}, 10^1]$	**1.58463e-15** $\tilde{n}_c = [5^4, 8^1, 12^6]$	**1.58908e-15** $\tilde{n}_c = [5^4, 11^4, 12^3]$
	1.68642e-15 $\tilde{n}_c = [8^8, 9^4]$		
	1.65807e-15 $\tilde{n}_c = [7^4, 8^9]$		
	1.63343e-15 $\tilde{n}_c = [7^{12}, 8^2]$		
	1.61335e-15 $\tilde{n}_c = [6^5, 7^{10}]$		
16	1.59510e-15 $\tilde{n}_c = [6^{12}, 7^4]$	**1.59510e-15** $\tilde{n}_c = [6^{12}, 7^4]$	**1.59510e-15** $\tilde{n}_c = [6^{12}, 7^4]$
		1.57863e-15 $\tilde{n}_c = [5^2, 6^{15}]$	**1.57863e-15** $\tilde{n}_c = [5^2, 6^{15}]$
18	1.75199e-10 $\tilde{n}_c = [5^8, 6^{10}]$	**7.60007e-11** $\tilde{n}_c = [5^{16}, 10^2]$	**7.60007e-11** $\tilde{n}_c = [5^{16}, 10^2]$
19	2.10599e-10 $\tilde{n}_c = [5^{14}, 6^5]$	**1.36000e-10** $\tilde{n}_c = [4^{15}, 10^4]$	**1.36000e-10** $\tilde{n}_c = [4^{15}, 10^4]$

Higher LoC than the maximal LoC achieved for EQC

Maximal LoC

n_c	EOC	Non-Equidistant Checkpointing	
5	2.39998e-10 $\tilde{n}_c = [20^5]$	1.12000e-10 $\tilde{n}_c = [18^4, 28^1]$	1.12000e-10 $\tilde{n}_c = [18^4, 28^1]$
[illegible]	2.07477e-15	2.07477e-15	2.07477e-15
10	1.75999e-15 $\tilde{n}_c = [10^{10}]$	1.55958e-15 $\tilde{n}_c = [8^6, 13^4]$	1.55958e-15 $\tilde{n}_c = [8^6, 13^4]$

n_c	EOC	Non-Equidistant Checkpointing	
14	1.63343e-15 $\tilde{n}_c = [7^{12}, 8^2]$	1.57695e-15 $\tilde{n}_c = [4^3, 8^{11}]$	1.57695e-15 $\tilde{n}_c = [4^3, 8^{11}]$
	1.61335e-15	1.57695e-15	1.57695e-15
19	2.10599e-10 $\tilde{n}_c = [5^{14}, 6^5]$	1.36000e-10 $\tilde{n}_c = [4^{15}, 10^4]$	1.36000e-10 $\tilde{n}_c = [4^{15}, 10^4]$

An LoC requirement can be satisfied with lower number of checkpoints when Non-Equidistant Checkpointing is used

The best case completion time, i.e. when no errors occur, can be reduced when Non-Equidistant Checkpointing is used

The maximal LoC can be achieved for n_c=13

Overview

- Exhaustive Search
- Clustered Checkpointing
- Results
- Conclusion

Conclusion

- By using non-equidistant checkpointing, the LoC can be improved in comparison to the LoC obtained when equidistant checkpointing is used
- An LoC requirement can be satisfied for a lower number of checkpoints if non-equidistant checkpointing is used
- The best case completion time, i.e. the time required for a job to complete when no errors occur, can be reduced while at the same time a given LoC requirement is satisfied

LUND
UNIVERSITY