ECRTS 2010 WATERS

Fault Resilience Analysis for Real-Time Systems

George Lima[1], Flávia Maristela[2], Verônica Lima[3]
gmlima@ufba.br, flaviamsn@ifba.edu.br, cadena@ufba.br

[1]Department of Computer Science – Distributed Systems Lab (UFBA)
[2]Department Technology in Electro-electronics (IFBA)
[3]Department of Statistics (UFBA)

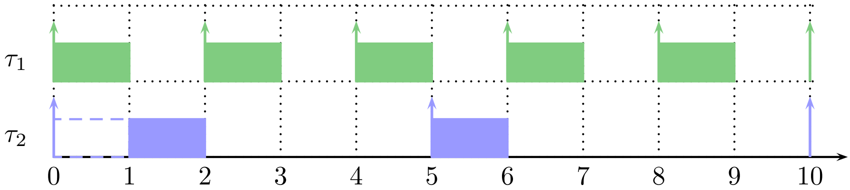
1^{st} International Workshop on Analysis, Tools and Methodologies for Embedded and Real-time Systems

Presentation Overview

- Real-time systems are organized a set of n tasks $\Gamma = \{\tau_1, \ldots, \tau_n\}$
- Each task τ_i has attributes such as:
  - Period
  - Deadline
  - Worst-Case Execution Cost
  - Recovery Execution Cost
- Real-time system tasks in have to meet both their logical and timing requirements.
- In order to guarantee that deadlines are met, all tasks have their execution ordered according to a scheduling policy.
  - Rate Monotonic (RM)
  - Earliest Deadline First (EDF)

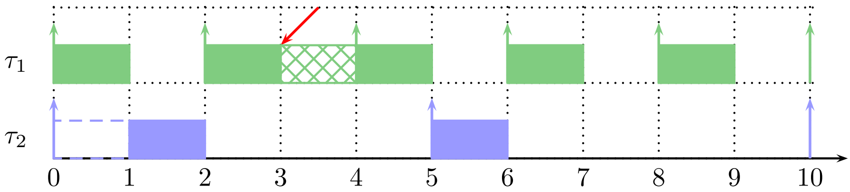
Example

$\Gamma = \{\tau_1, \tau_2\}$, $\mathbf{T} = (2, 5)$, $\mathbf{C} = (1, 1)$, $\mathbf{D} = \mathbf{T}$. *Tasks are scheduled according to Rate Monotonic.*

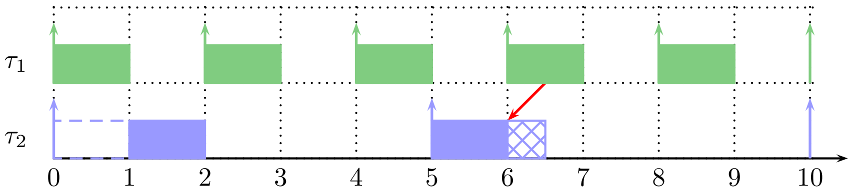


- All application potentially fails $\Rightarrow$ Real-time systems requirements must be met, even in the presence of faults.
- Fault tolerance is provided by executing recovery actions upon error detection.
- Recovery scheme is usually based on time redundancy.

- Recovery based on re-execution of the faulty task.



- Recovery based on alternative tasks.

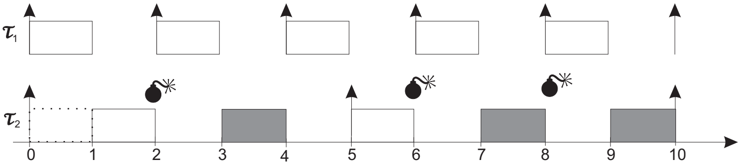


- Usually, fault tolerance for real-time systems is:
    - strictly linked to system and fault models;
    - based on worst case conditions

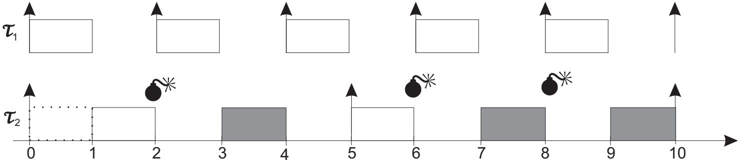
- Usually, fault tolerance for real-time systems is:
  - strictly linked to system and fault models;
  - based on worst case conditions
- HOW TO COMPARE DIFFERENT REAL TIME SYSTEMS FROM RESILIENCE VIEWPOINT?

- Usually, fault tolerance for real-time systems is:
  - strictly linked to system and fault models;
  - based on worst case conditions
- HOW TO COMPARE DIFFERENT REAL TIME SYSTEMS FROM RESILIENCE VIEWPOINT?
- HOW TO MEASURE THE SYSTEM FAULT RESILIENCE?

1. Worst-case based models may not reflect the real capacity of a system to tolerate faults
2. Models do not consider the system overall behavior



1. Worst-case based models may not reflect the real capacity of a system to tolerate faults
2. Models do not consider the system overall behavior



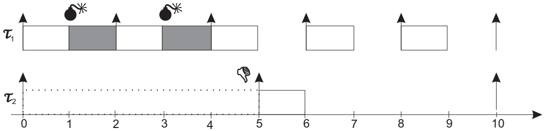
3. Intuitively, the expected number of errors increases with time if they are not co-related.

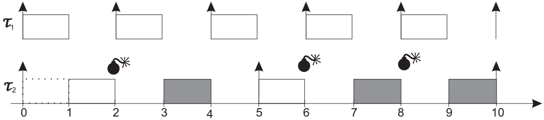
Assumption

The fault resilience of a system is proportional to the number of errors it can deal with.

Assumption

The expected number of error occurrences increases with time.





- The resilience of a given task $\tau_i \in \Gamma$ depends on how its jobs behave when errors take place

Requirement

Fault resilience must be given for individual tasks of the analyzed system.

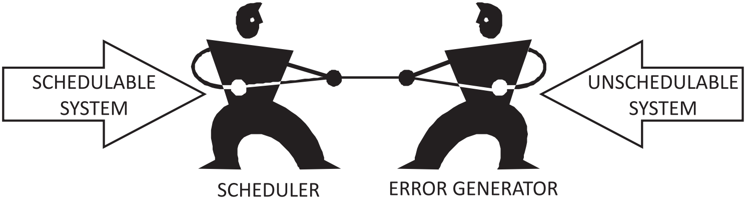
Requirement

The fault resilience of a task must account for the overall behavior of its jobs.

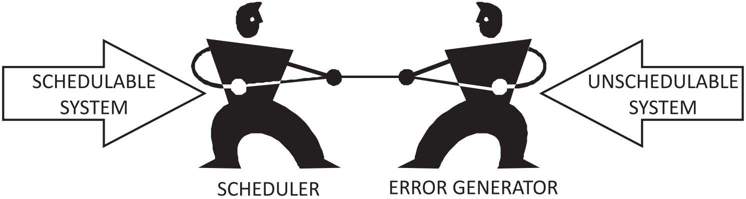
Definition

The fault resilience of a job is measured as the minimum number of errors that make it miss its deadline. The fault resilience distribution of a task is given by the fault resilience of its jobs.

- System is modeled based on two main components

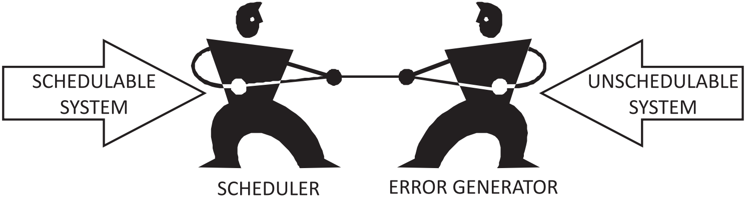


- System is modeled based on two main components



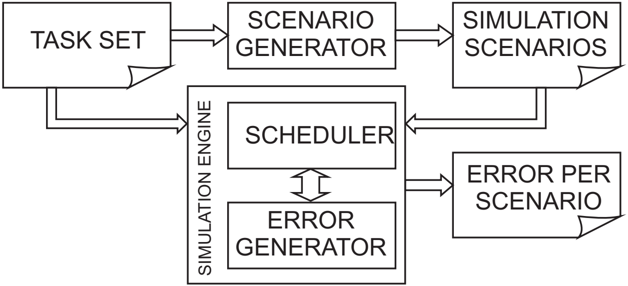
- Main Goal ⇒ Derive a metric that can be able to:
  - Be independent of system model
  - Express system resilience;
  - Compare different fault-tolerant scheduling approach;

- System is modeled based on two main components



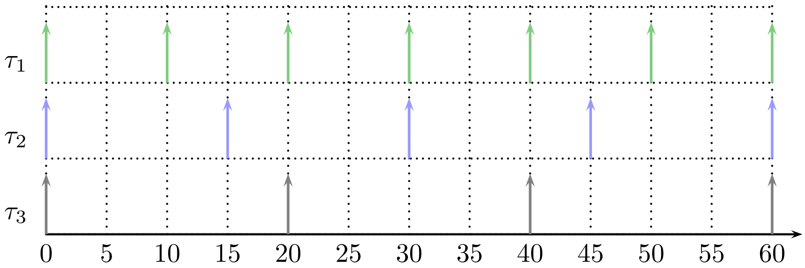
- Main Goal ⇒ Derive a metric that can be able to:
  - Be independent of system model
  - Express system resilience;
  - Compare different fault-tolerant scheduling approach;
- Metric intuition: measure the *effort* that the error generator does to make a given task miss its deadline
- $\mathbf{E}_i = f_i$

- Simulation Environment Framework



Example

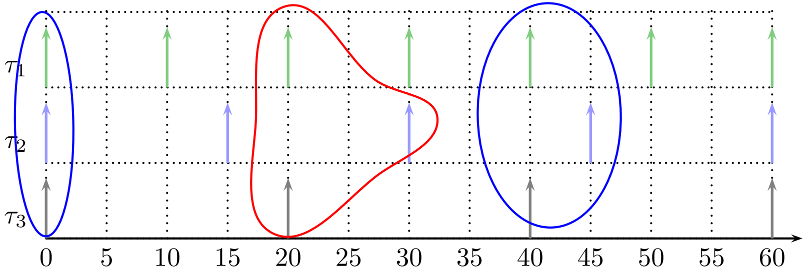
Consider $\Gamma = \{\tau_1, \tau_2, \tau_3\}$ *a set of periodic tasks so that* $\mathbf{T} = (10, 15, 20)$.



- $\mathbf{S} = (0, 0, 0)$ and $\mathbf{S} = (40, 45, 40)$ are simulation scenarios.

Example

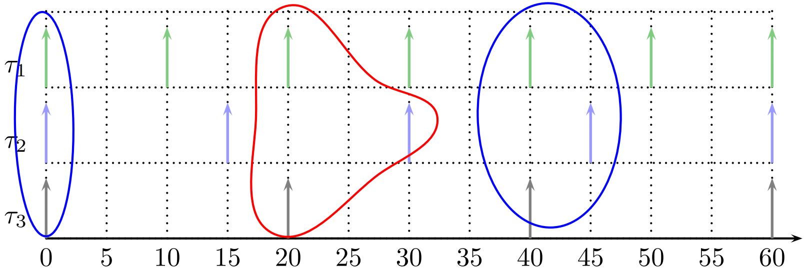
Consider $\Gamma = \{\tau_1, \tau_2, \tau_3\}$ *a set of periodic tasks so that* $\mathbf{T} = (10, 15, 20)$.



- $\mathbf{S} = (0, 0, 0)$ and $\mathbf{S} = (40, 45, 40)$ are simulation scenarios.

Example

Consider $\Gamma = \{\tau_1, \tau_2, \tau_3\}$ *a set of periodic tasks so that* $\mathbf{T} = (10, 15, 20)$.



- $\mathbf{S} = (0, 0, 0)$ and $\mathbf{S} = (40, 45, 40)$ are simulation scenarios.
- $\mathbf{S} = (20, 30, 20)$ is not.

- Goal: Determine the simulation time window

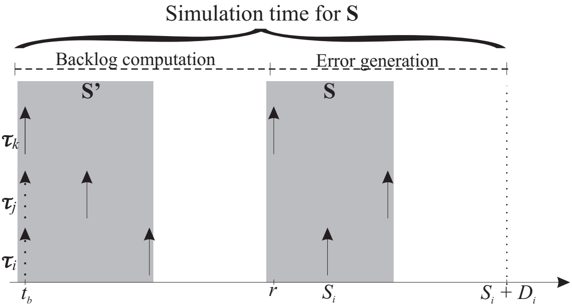
Definition

Tuple $\mathbf{S} = (S_1, \ldots, S_n)$ *is a simulation scenario of a periodic task set* $\Gamma = \{\tau_1, \ldots, \tau_n\}$ *if the following predicate holds:*

$$\text{scenario}(\Gamma, \mathbf{S}) \overset{def}{=} \exists w \in \quad , \forall S_i : (S_i + w) \mod T_i = 0 \wedge \max(\mathbf{S}) - S_i < T_i$$

- Scenario generator implements two kinds of generation procedures
    - Sequential procedures: generate the whole set of simulation scenarios
    - Random generation procedures: generate a subset of simulation scenarios

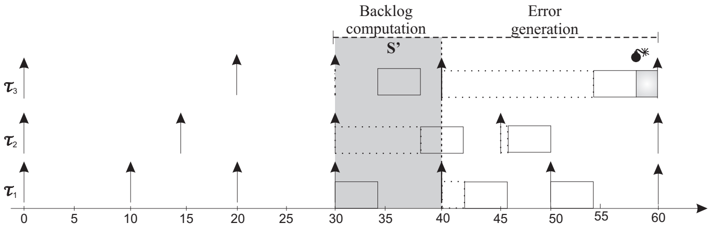
- Goal: Determine the fault resilience for a given task τ_i regarding a specific scenario **S**



Example

Consider $\Gamma = \{\tau_1, \tau_2, \tau_3\}$ *a set of periodic tasks so that* $\mathbf{T} = (10, 15, 20)$. *The simulation scenario* $\mathbf{S} = (50, 45, 40)$ *and we wish to analyze the job of* τ_3 *released at* $t = 40$.

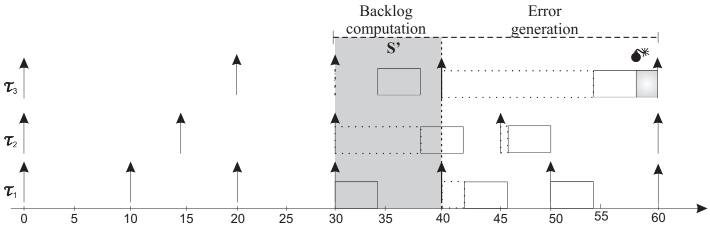
- $\mathbf{S}' = (40, 30, 40)$ and simulation time interval is $[30; 60)$



Example

Consider $\Gamma = \{\tau_1, \tau_2, \tau_3\}$ *a set of periodic tasks so that* $\mathbf{T} = (10, 15, 20)$. *The simulation scenario* $\mathbf{S} = (50, 45, 40)$ *and we wish to analyze the job of* τ_3 *released at* $t = 40$.

- $\mathbf{S}' = (40, 30, 40)$ and simulation time interval is $[30; 60)$



- In this case, $\mathbf{E}_3 = f_3^{\mathbf{S}} = 1$

Example

Let $\Gamma = \{\tau_1, \ldots, \tau_4\}$ *be a task set independent periodic tasks scheduled by Rate Monotonic. Task attributes are* $\mathbf{C} = (30, 35, 25, 30)$, $\mathbf{T} = (100, 175, 200, 300)$ *and* $\mathbf{D} = \mathbf{T}$.

- We focus on task τ_4.

f_4	2	3	4

- Distribution of f_4 gives more information about the system fault resilience.
- We can compute the mean effort $\bar{\mathbf{E}}_4 = 3$.

- Is it possible to compare two different scheduling approaches from fault resilience viewpoint?

- Is it possible to compare two different scheduling approaches from fault resilience viewpoint?
- 10 task sets, with 10 tasks each, were generated
- $C_i = \bar{C}_i = 3$ and $\mathbf{T}$ was randomly generated
- Both RM and EDF scheduling were considered.
- A sample of simulation scenarios was randomly generated.
- We computed the mean effort $\bar{\mathbf{E}}_i$ for each task τ_i, $i = 1, 2, \ldots 10$.
- In order to determine sample size, we assumed a sample error $|\bar{\mathbf{E}}_i^* - \bar{\mathbf{E}}_i| = 5\text{x}10^{-3}$

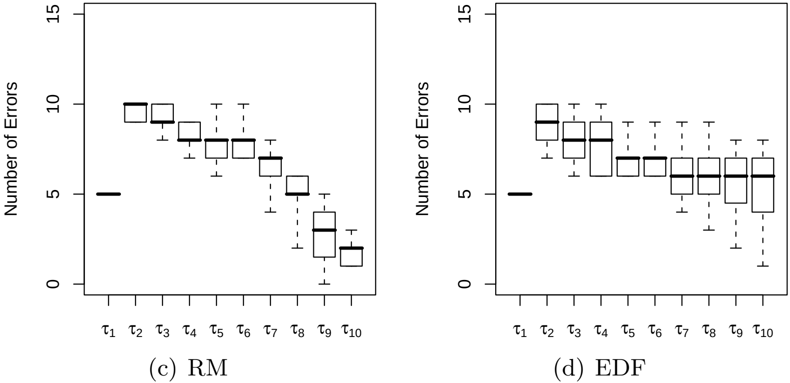


Figure: Fault resilience distribution

- 40 task sets composed of 30 tasks each were randomly generated.
- Periods and execution times were randomly select in the intervals $[10; 800]$ and $[3; 30]$, respectively.
- The calculated hyperperiod was of the order 10^{15}.
- Sample error equal to 5×10^{-3}
- $\alpha = 5\%$

GOAL: estimate the fault resilience for each system task.

Table: Fault resilience estimation

% CPU	RM $\bar{f}_i$	RM CI	EDF $\bar{f}_i$	EDF CI
55-65	6.35	[6.237,6.552]	6.05	[6.010,6.136]
65-75	3.01	[2.996,3.050]	3.25	[3.192,3.294]
75-85	2.33	[2.300,2.358]	2.87	[2.801,2.907]
85-95	1.95	[1.890,1.960]	2.13	[2.023,2.223]

- Some aspects must be further investigated:
  - Considering task sets where not all tasks are periodic;
  - Taking spacial redundancy;
  - Other strategies to estimate the backlog.
- Extend the model to derive probabilistic schedulability bounds for real-time systems.