Detection of scheduling anomalies in real-time system

Blandine Djika†, Alain Plantec*, Frank Singhoff*, Georges Kouamou†

*Lab-STICC, UMR CNRS 6285, Univ. of Brest
†Univ of Yaounde 1, Cameroon

Journées nationales du GDR GPL 2023
June 5 – 9, 2023,
Rennes, France
Outline

1. Context
2. Problem statement and contributions
3. Scheduling anomalies models
4. Monitoring of scheduling anomalies
5. Conclusion
Real-time system?

System made of a set of task that have timing constraints to meet.

Aircraft, Satellites

Electronic device

Industrials
Problem statement (1/2)

A scheduling anomaly refers to a counter-intuitive phenomenon in which increasing the system resources or relaxing the application constraints can lead to missed deadline. (Almeida06)

- When do scheduling anomalies occur?
- How to detect these anomalies in real-time systems?
Problem Statement (2/2)

3 periodics tasks scheduled with a non-preemptive fixed-priority scheduler in an uniprocessor system.

Each task $\tau_i = (R_i(\tau_i), C_i(\tau_i), D_i(\tau_i), T_i(\tau_i), \mu_i(\tau_i))$

$\tau_1 = (0, 1, 3, 3, 99); \tau_2 = (0, 2, 6, 6, 98); \tau_3 = (0, 4, 12, 12, 97);
Contributions

1. Model analysis to specify scheduling anomalies
   - Static constraints
   - Dynamic constraints

2. MONANO: Monitoring library on POSIX/RTEMS
Outline

1. Context
2. Problem statement and contributions
3. Scheduling anomalies models
4. Monitoring of scheduling anomalies
5. Conclusion
Scheduling anomalies model (1/2)

Static constraints
- Design time Verification
- YES
  - Potential Anomalies
- NO
  - No scheduling anomalies

Dynamic constraints
- Runtime Verification
- YES
  - Detection Anomalies
- Detected Scheduling Anomalies
Scheduling anomalies model (2/2)

- 7 types of scheduling anomalies identified from the literature

- 17 Static constraints
  - 8 related to tasks
  - 9 related to the execution platform
  - 19 scenarios combining static constraints

- 7 Dynamic constraints

- Ex.: Reducing task execution time, increasing processor speed, increasing task period, ....
  - Reducing task execution time: 7 scenarios combining static constraints
Outline

1. Context
2. Problem statement and contributions
3. Scheduling anomalies models
4. Monitoring of scheduling anomalies
5. Conclusion
MONANO, a user-level library to validate our approach

Services:
- Store architecture model of monitored applications
- Creation of monitored threads
- Measure and compute metrics to verify dynamic and static constraints
- Anomaly detection and application callback
Monitoring of scheduling anomalies (2/3)

Function pthread_monano_verify_reduce_execution_time
begin
    ....
    if (execution_time(τ_i) < C_i(τ_i)) then
        Callback()
    end
end

Function pthread_monano_signal_departure_time
begin
    start_execution(τ_i) = clock_gettime()
    ...
end

Function pthread_monano_signal_end_time
begin
    end_execution(τ_i) = clock_gettime()
    execution_time(τ_i) = end_execution(τ_i) - start_execution(τ_i) - blocking_time(τ_i) - preemption_time(τ_i)
    ....
    if (blocking_time(τ_j) == 0) then
        preemption_time(τ_j) = (end_execution(τ_i) - start_execution(τ_i)) + preemption_time(τ_j)
    else
        preemption_time(τ_j) = (end_execution(τ_i) - end_blocking(τ_i)) + preemption_time(τ_j)
    end if
    ....
end

- 6 Algorithms to check for dynamic constraints
- 5 Algorithms to compute task execution time during its runtime.
- 1 Algorithm to verify the schedulability of the system
- Etc....
Monitoring of scheduling anomalies (3/3)

- **MONANO Validation**
  - Static and dynamic constraints leading to scheduling anomalies
  - ROSACE: Aircraft flight controller application of 15 tasks developed on RTEMS (Pagetti14)
  - Anomaly detection and application callback
Conclusion

- **Problem statement**
  - Detection of scheduling anomalies in real-time systems

- **Contributions**
  - Model made of constraints to specify scheduling anomalies. Checked both on design and runtime
  - A POSIX/RTEMS Monitoring library called MONANO to detect scheduling anomalies in uniprocessor system

- **Results**
  - Verify that MONANO can be implemented on POSIX/RTEMS
  - Detection of 5 scheduling anomalies for uniprocessor system from the 7 identified from the literature

- **Future work**
  - Complete MONANO evaluation (overhead, scalability)
  - Extend MONANO to multiprocessor system