Evaluation of Mixed-Criticality Scheduling Algorithms using a Fair Taskset Generator

Saravanan Ramanathan and Arvind Easwaran

Nanyang Technological University, Singapore

July 5, 2016

Outline

1. Mixed-Criticality (MC) System
2. Motivation
3. Taskset Generators
   - Existing Taskset Parameters
   - New Taskset Parameter
   - Existing Generators
4. MC-FairGen
   - MC-FairGen Framework
   - MC-FairGen Generator
5. Evaluation
   - Single-core
   - Multi-core
6. Summary

Outline

1. Mixed-Criticality (MC) System
2. Motivation
3. Taskset Generators
   - Existing Taskset Parameters
   - New Taskset Parameter
   - Existing Generators
4. MC-FairGen
   - MC-FairGen Framework
   - MC-FairGen Generator
5. Evaluation
   - Single-core
   - Multi-core
6. Summary

Mixed-Criticality (MC) System

Task Model: A MC system with two criticality levels (namely LO and HI) is being considered.

τ_i = (T_i,χ_i,C_i^L,C_i^H,D_i)

- $T_i \in \mathbb{R}^+$ is the minimum release separation time,
- $\chi_i \in \{LO, HI\}$ is the criticality level,
- C_i^L and C_i^H are the LO- and HI-criticality Worst-Case Execution Time (WCET) values respectively; we assume $C_i^L \leq C_i^H$ and,
- $D_i \in \mathbb{R}^+$ is the relative deadline.

$u_i^L = C_i^L / T_i$ and $u_i^H = C_i^H / T_i$ denotes the LO-mode and HI-mode utilization.

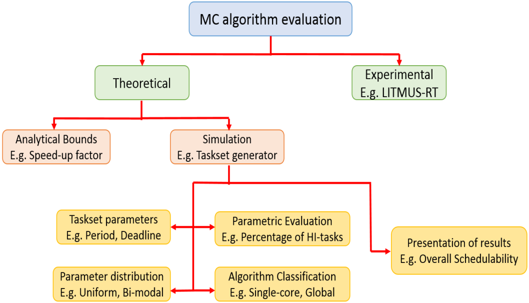
Outline

1. Mixed-Criticality (MC) System
2. Motivation
3. Taskset Generators
   - Existing Taskset Parameters
   - New Taskset Parameter
   - Existing Generators
4. MC-FairGen
   - MC-FairGen Framework
   - MC-FairGen Generator
5. Evaluation
   - Single-core
   - Multi-core
6. Summary

Motivation

Evaluation of MC scheduling algorithm:

- Lack of benchmark applications
- No standardized evaluation platform
- Need for unbiased taskset generator



Outline

1. Mixed-Criticality (MC) System
2. Motivation
3. Taskset Generators
    - Existing Taskset Parameters
    - New Taskset Parameter
    - Existing Generators
4. MC-FairGen
    - MC-FairGen Framework
    - MC-FairGen Generator
5. Evaluation
    - Single-core
    - Multi-core
6. Summary

Existing Taskset Parameters

Factors affecting the performance of a scheduling algorithm

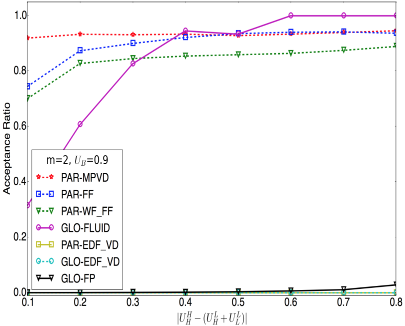
- For non-MC Systems
  - Period
  - Deadline
  - Maximum Individual task utilization
  - Total utilization bound
  - Number of tasks
- For MC Systems
  - Percentage of HI-tasks
  - Criticality Factor ($\frac{C_i^H}{C_i^L}$)
  - Normalized system utilization difference
- Distribution and the range of values

New Taskset Parameter

Normalized system utilization difference

- Total utilization difference $(|U_H^H - (U_H^L + U_L^L)|)$
- HI-criticality utilization difference $(|U_H^H - U_H^L|)$
- LO-mode utilization difference $(|U_H^L - U_L^L|)$

Where $U_L^L \stackrel{\text{def}}{=} \sum_{\tau_i \in \tau_L} u_i^L$,
$U_H^L \stackrel{\text{def}}{=} \sum_{\tau_i \in \tau_H} u_i^L$ and $U_H^H \stackrel{\text{def}}{=} \sum_{\tau_i \in \tau_H} u_i^H$



Existing Generators

Algorithms: EY, ECDF, TG, AMC

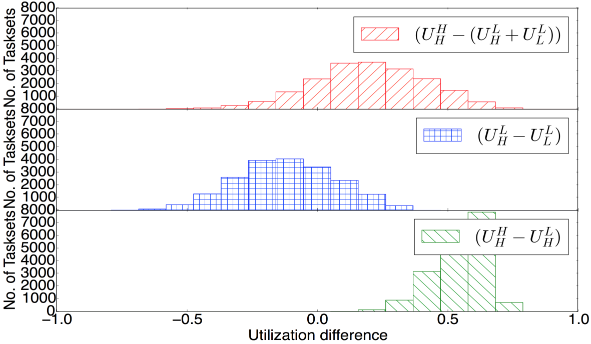


Figure: Normalized system utilization distributions

Existing Generators

Algorithms: MPVD

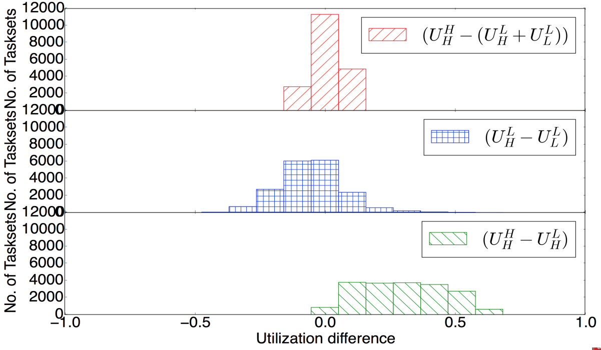


Figure: Normalized system utilization distributions

Existing Generators

Algorithms: EDF-VD,MC-FLUID

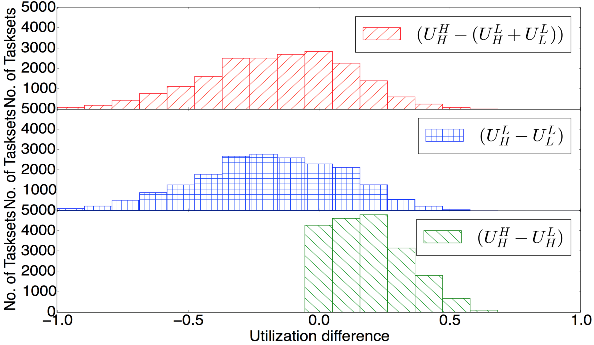
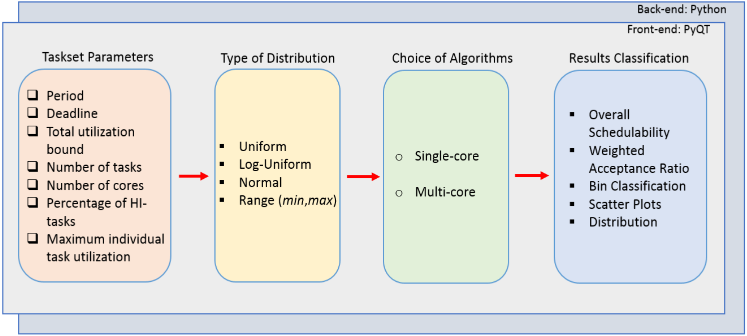


Figure: Normalized system utilization distributions

Outline

1. Mixed-Criticality (MC) System
2. Motivation
3. Taskset Generators
   - Existing Taskset Parameters
   - New Taskset Parameter
   - Existing Generators
4. MC-FairGen
   - MC-FairGen Framework
   - MC-FairGen Generator
5. Evaluation
   - Single-core
   - Multi-core
6. Summary

MC-FairGen Framework



Back-end: Python

Front-end: PyQT

Taskset Parameters

- ❑ Period
- ❑ Deadline
- ❑ Total utilization bound
- ❑ Number of tasks
- ❑ Number of cores
- ❑ Percentage of HI-tasks
- ❑ Maximum individual task utilization

Type of Distribution

- ▪ Uniform
- ▪ Log-Uniform
- ▪ Normal
- ▪ Range (*min*,*max*)

Choice of Algorithms

- ○ Single-core
- ○ Multi-core

Results Classification

- ▪ Overall Schedulability
- ▪ Weighted Acceptance Ratio
- ▪ Bin Classification
- ▪ Scatter Plots
- ▪ Distribution

- Multiple utilization bounds (U_H^H, U_H^L and U_L^L)
- Uses RandFixedSum to choose utilizations

Taskset Generator

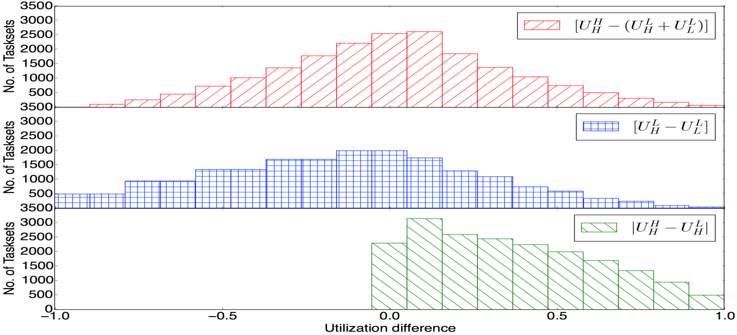


Figure: Normalized system utilization distributions of MC-FairGen

Outline

1. Mixed-Criticality (MC) System
2. Motivation
3. Taskset Generators
   - Existing Taskset Parameters
   - New Taskset Parameter
   - Existing Generators
4. MC-FairGen
   - MC-FairGen Framework
   - MC-FairGen Generator
5. Evaluation
   - Single-core
   - Multi-core
6. Summary

Experimental Results

Single-core, constrained deadline, uniform distribution

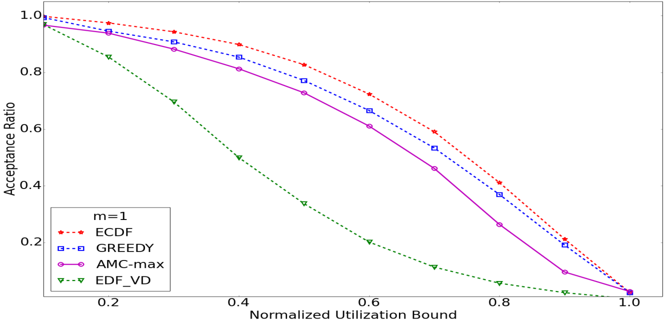


Figure: Overall Schedulability

Experimental Results

Single-core, constrained deadline, log-uniform distribution

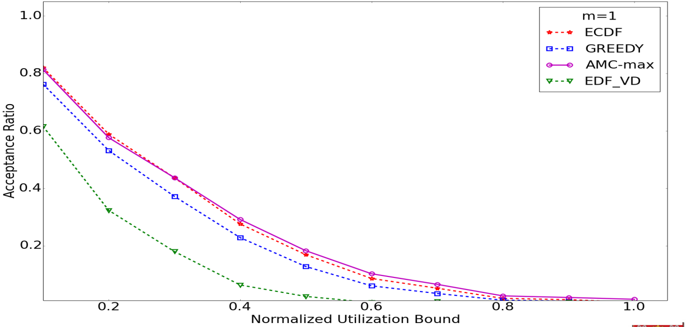


Figure: Overall Schedulability

Experimental Results

Multi-core, implicit deadline, uniform distribution

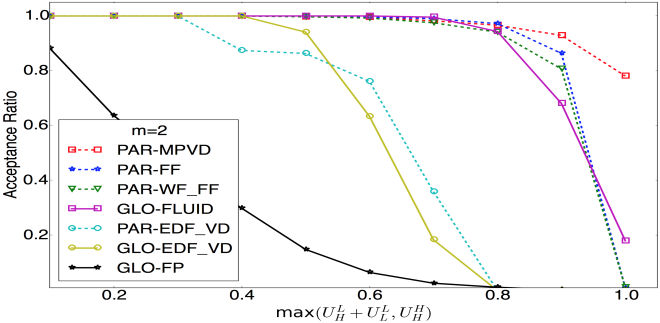


Figure: Overall Schedulability

Outline

1. Mixed-Criticality (MC) System
2. Motivation
3. Taskset Generators
   - Existing Taskset Parameters
   - New Taskset Parameter
   - Existing Generators
4. MC-FairGen
   - MC-FairGen Framework
   - MC-FairGen Generator
5. Evaluation
   - Single-core
   - Multi-core
6. Summary

Summary

- Bias in existing taskset generators
- MC-FairGen Framework
    - Comprehensive list of factors that affect MC algorithms
    - New taskset generator with multiple utilization bounds
    - Parameter specific evaluation of existing MC algorithms
- Future Work
    - Common evaluation platform for MC systems
    - Dynamic parameter control and support parallelization

NANYANG TECHNOLOGICAL UNIVERSITY

Thank you..!

Questions..?

NANYANG TECHNOLOGICAL UNIVERSITY