Multi-Criteria Evaluation of Partitioned EDF-VD for Mixed-Criticality Systems Upon Identical Processors

Paul Rodriguez[1,3] Laurent George[2,3] Yasmina Abdeddaïm[2,4]
Joël Goossens [1]

[1]Université Libre de Bruxelles

[2]University of Paris-Est, LIGM

[3]ECE Paris

[4]ESIEE

Outline

Introduction

Do we need to consider specific task-to-processor heuristics for MC ?

- Classical task allocation heuristics : First Fit (FF), Best Fit (BF), Next Fit (NF), Worst Fit (WF)
- Classical Sorting criteria: Decreasing Utilization (DU), Decreasing Density (DD)
- First Fit with Decreasing Density outperforms all task to processor allocation schemes (Lupu& al. 2010)
- Is it also the case for tasks having different Criticalities ?
- We consider Dual criticalies in this paper (LO and HI)

EDF with DBF tuning

- The schedulability test has to be computationally efficient : partitioning involves testing it very frequently
- EDF VD (Ref Sanjoy) that introduce the concept of intermediate deadline for HI tasks
  - A HI task has two deadlines: $D_i(LO)$ and $D_i(HI)$ with $D_i(LO) = \lambda . D_i(HI)$
  - Tasks with implicit deadlines
- The algorithm by Ekberg and Yi 2013 is EDF with modified deadlines
  - The offline phase (which effectively is a schedulability test) of this algorithm runs in pseudo-polynomial time [EY13]

EDF-VD

- The original EDF-VD algorithm (Baruah &. al 2011) provides an efficient MC schedulability test for implicit deadline systems
- The value of a λ factor is equal to $U_{LO}^{HI}/(1 - U_{HI}^{HI})$ where U_{LO}^{HI} is the utilization of all HI tasks in LO mode
- For all HI tasks, virtual deadlines in LO mode are created equal to λ multiplied by their real deadlines
- The system is scheduled online using EDF with the deadlines corresponding to the current mode
- Schedulability is guaranteed if $U_{LO}^{LO} + min(U_{HI}^{HI}, \lambda) \leqslant 1$

Ekberg-Yi

- The algorithm by Ekberg and Yi [EY13] is EDF with modified deadlines and schedules constrained deadline systems
- The schedulability test of this algorithm runs in pseudo-polynomial time [EY13]
- Ekberg and Yi define two demand bound functions for MC systems, $dbf_{LO}(t)$ and $dbf_{LO}(t)$. Two schedulability conditions must be satisfied for a constrained deadline MC system to be schedulable by EDF :
  - $\forall t \geqslant 0 : dbf_{LO}(t) \leqslant t$
  - $\forall t \geqslant 0 : dbf_{HI}(t) \leqslant t$
- Deadlines are adjusted following a heuristic procedure that tries to satisfy the conditions on both the LO and HI dbf, incrementing and decrementing deadlines one at a time

Partitioning heuristics

- In a single-criticality system, Lupu & al. [LCGG10] identify 32 typical heuristics for partitioning, each being a combination of a task order and a a processor assignment rule.
- Tasks in the system are put in ascending or descending order among deadline, period, density and utilization.
- Tasks are then assigned to processors in that order, using either first fit, best fit, worst fit or next fit as a rule to decide which possible processor will obtain each task.

Mixed-criticality partitioning heuristics

- Simple single-criticality (criticality agnostic) heuristics can be used for mixed-criticality systems as long as one specifies which utilization (HI or LO) is to be considered
- Additional heuristics exist if we consider the option of assigning HI tasks alone first then adding LO tasks with (possibly) another method. The total number of possible heuristics of this kind is 1024.
  - We adopt a Racing algorithm to remove non efficient partitioned heuristics in our experiments
- Our experiments leads to two partitioned heuristics: First Fit with decreasing Density (F_{DD}) and Worst Fit with Decreasing Density (W_{DD})

Task set generation

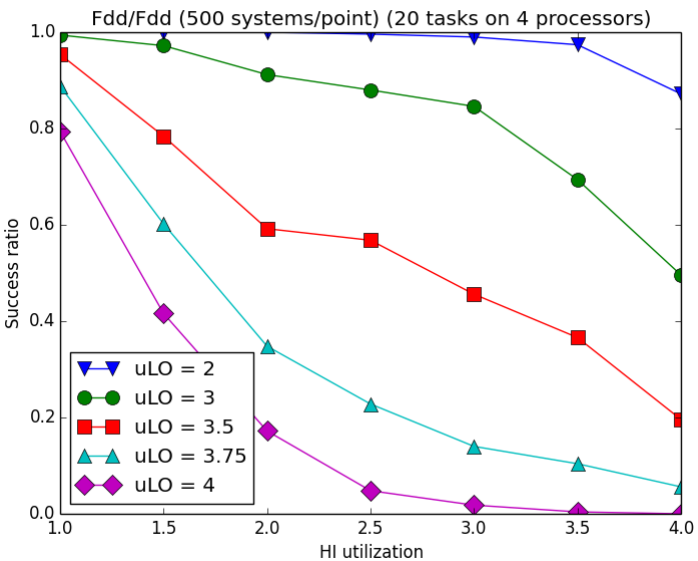
Parameters and task utilization

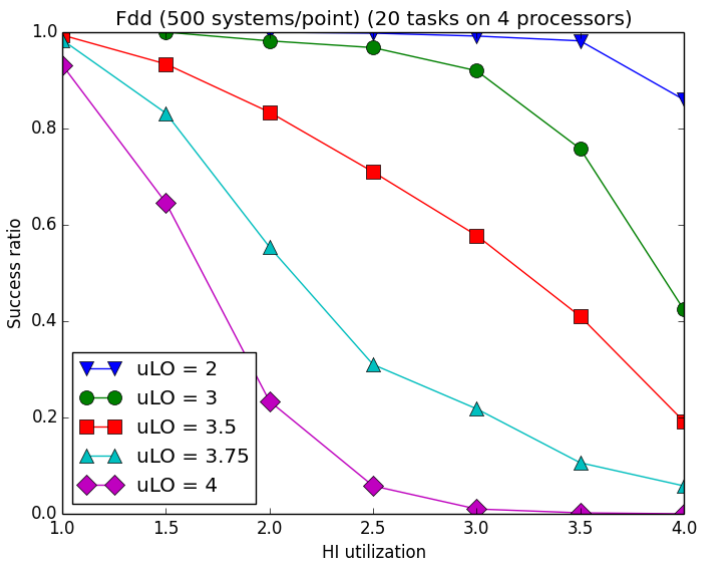
- We generate sporadic tasks according to UUniFast algorithm (extended to multiprocessor systems in Emberson & al. 2010).
- Input parameters for our generator are U(LO), U(HI), the ratio n_{LO}/n_{HI}, T_{min} and T_{max}
- We generate n_{LO} and n_{HI} utilizations and try to make compatible couples (LO/HI) tasks.
- Periods are randomly chosen in $[T_{min}, T_{max}]$ with a log uniform random variable
- Deadlines are randomly chosen in $[WCET, Period]$ for WCET in LO and HI mode.
- We consider 4 processors and periods in $[5, 50]$

Simulation Results

Performance of F_{DD} for HI tasks

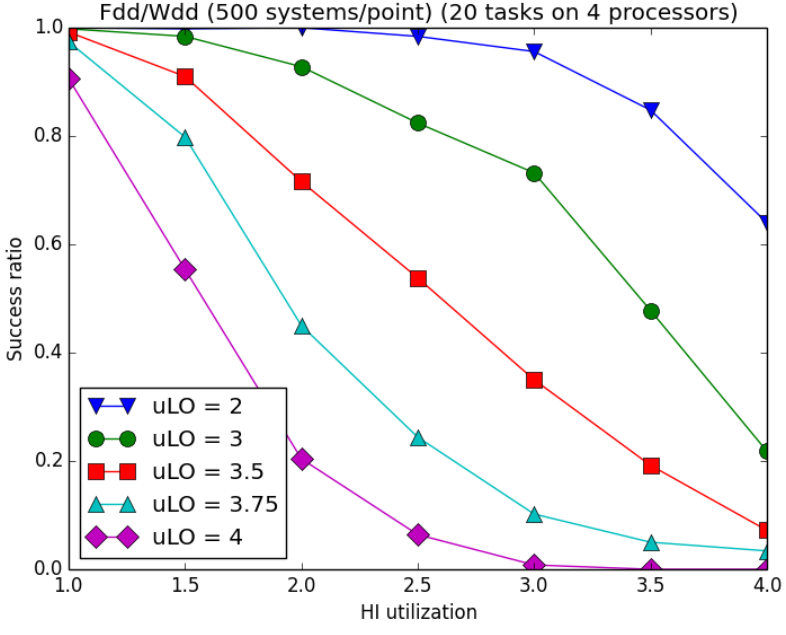
- Decreasing Density is still the best sorting criteria for MC scheduling
- F_{DD} outperforms F_{DD}/F_{DD}
- The price to pay for assigning HI tasks first can reach 20% (obtained when U(HI) is low and U(LO) is high)
- When U(HI) is high, F_{DD}/F_{DD} tends towards FDD

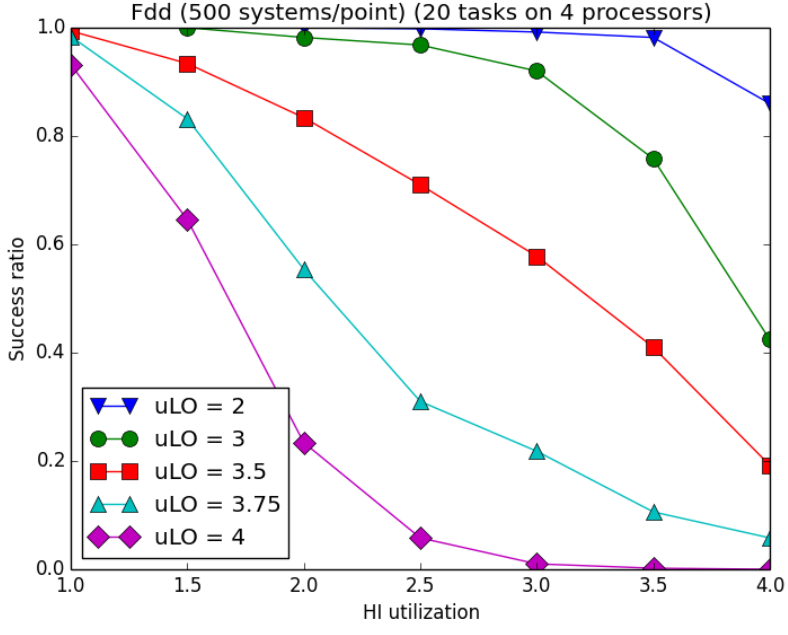




Performance of W_{DD} for HI tasks

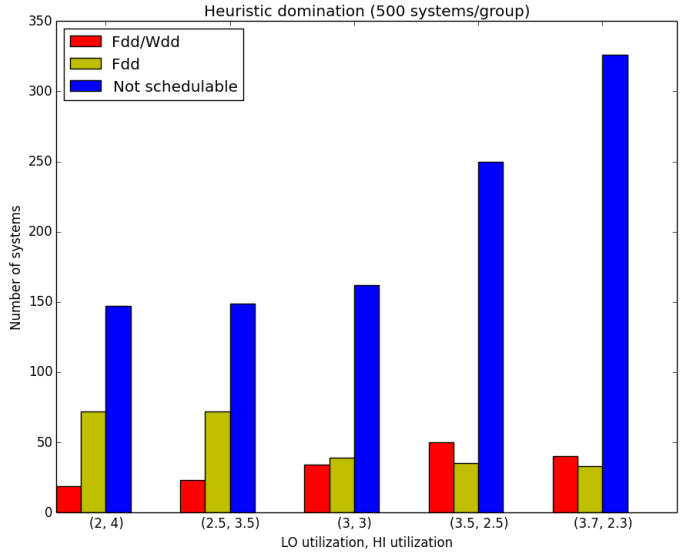
- F_{DD}/W_{DD} is an interesting alternative when U(HI) is less than 2.5 (on 4 processors)
- The price to pay for assigning HI tasks is reduced to 5% (when U(HI) is low and U(LO) is high)





Number of HI tasks schedulable systems (exclusive)

- F_{DD}/W_{DD} outperforms F_{DD} (20%) for scheduling HI tasks when utilization of HI tasks is smaller than 2.5 (on 4 processors)
- F_{DD} should be considered when the utilization of HI tasks is high
- The % of non schedulable tasks ranges from 30% to 66% in our experiments



Conclusion

- In MC Partitioned scheduling (dual criticalities), DD is still an performant sorting criteri a
- F_{DD}/W_{DD} should be considered when U(HI) is not high (less than 40% in our experiments) to maximize the chance of assigning HI tasks to a platform
- F_{DD} performs better when U(HI) is high (tends towards to an MC agnostic behavior)

Further works:

- More simulations needed
- Extend the results to more than two criticalities
- Explore semi-partitioned (restricted migration) partitioned heuristics


Pontus Ekberg and Wang Yi, *Bounding and shaping the demand of generalized mixed-criticality sporadic task systems*, Real-Time Systems (2013), 1–39.


Irina Lupu, Pierre Courbin, Laurent George, and Joël Goossens, *Multi-criteria evaluation of partitioning schemes for real-time systems*, 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2010, pp. 1–8.