Real-Time Scheduling on Two-type Heterogeneous Multiprocessors

Gurulingesh Raravi, Björn Andersson, Konstantinos Bletsas, Vincent Nelis











INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO POLITÉCNICO DO PORTO



Researce Real-Tir FCT Researce

Research Centre in Real-Time Computing Systems FCT Research Unit 608

Multiprocessor Real-Time Scheduling

- 1. Assign the tasks to processors
- 2. Schedule the tasks



Multiprocessor Real-Time Scheduling

- 1. Assign the tasks to groups of processors
- 2. Schedule the tasks



System Model

- Implicit-deadline sporadic tasks
- Two-type heterogeneous multiprocessors
- Task Assignment Problem: Main Challenge



Three Migration Models



Scope of This Work



Problem Definition

- Problem P1: Intra-migrative assignment
 - Assign tasks to <u>processor types</u> so that each processor type is utilized to at most 100%.



- Step1: Partition the tasks as *heavy* and *light*
 - Heavy
 - H1 = {cannot be assigned to type-2 processors}
 - H2 = {cannot be assigned to type-1 processors}
 - Light
 - L = {can be assigned to both the processor types}



- Step1: Partition the tasks as *heavy* and *light*
 - H1 = {cannot be assigned to type-2 processors}
 - H2 = {cannot be assigned to type-1 processors}
 - L = {can be assigned to both the processor types}
- Step2: Assign the *heavy* tasks
 - Assign H1 to type-1 and H2 to type-2



- Step3: Assign the *light* tasks
 - Sort the tasks in L in non-increasing order of

utilization of the task on type-2 utilization of the task on type-1



- Step3: Assign the *light* tasks
 - Sort the tasks in L in non-increasing order of

utilization of the task on type-2 utilization of the task on type-1

 Intuition: Left-hand side tasks are more preferable to be assigned to type-1



Sort and Assign (SA)

- Problem P1: Intra-migrative assignment
- Property of SA:
 - At most one task split between type-1 and type-2



First Result – SA

- Problem P1: Intra-migrative assignment
- Theorem 1:
 - The approximation ratio of SA is: $1+\alpha/2 \le 1.5$



First Result – SA



Problem Definition

- Problem P2: Non-migrative assignment
 - Assign tasks to <u>processors</u> so that each processor is utilized to at most 100%.



Our Approach – SA-P

• Take the solution of SA



Do wrap-around assignment



Our Approach – SA-P

• Wrap-around assignment



- Properties of wrap-around assignment
 - At most |type-1| 1 tasks split between type-1
 - At most |type-2| 1 tasks split between type-2
 - At most 1 task split between type-1 and type-2

Second Result – SA-P

- Problem P2: Non-migrative task assignment
- Theorem 2:
 - The "approximation ratio" of SA-P is $1+\alpha \leq 2$



Second Result – SA-P



Simulation Results – SA

- Random task sets
 - 100000 critically feasible task sets
- Results



Histogram for SA

Simulation Results – SA-P

- Random task sets
 - 100000 critically feasible task sets
- Results



Histogram for SA-P

Summary

• Contributions



• Significance

- Intra-migrative assignment

- SA: First solution
- Non-migrative assignment
 - SA-P: better performance compared to SOTA