COMFAST: A Comparative Framework for Analysis of Scheduling Techniques in Multi cores

S. Shah, A. Qahir, M. Safeer, S. Mazahir and O. Hasan

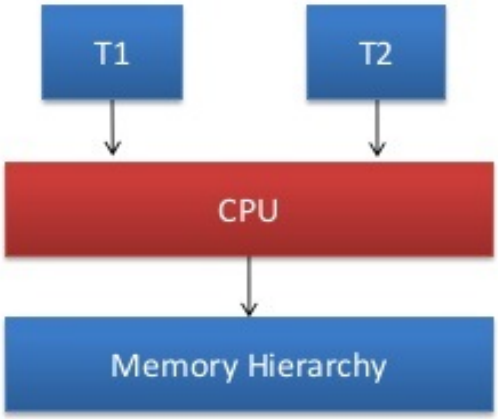
National University of Sciences and Technology (NUST), Pakistan

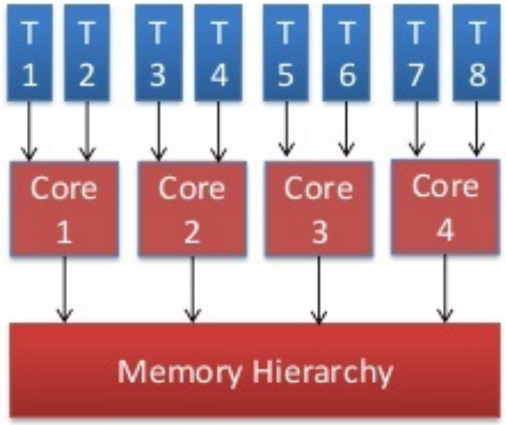
SysCon 2018
Vancouver, Canada

Outline

- Introduction and Motivation
- Proposed Methodology: *COMFAST*
- Results
- Conclusions

Multi-core Architectures

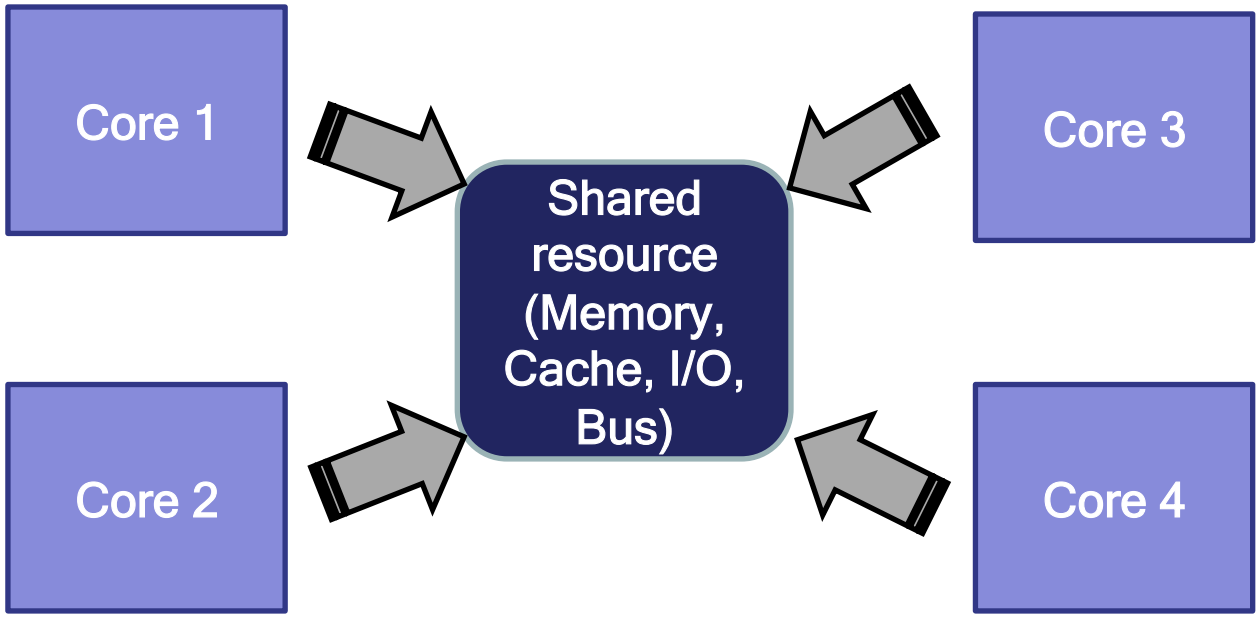




Unicore

Multicore

Resource Contention Problem



- Resource contention can lead to the accumulation of delays, which may lead to
  - Degraded performance
  - Tasks missing their deadlines

Scheduling Algorithms

- Used to allocate tasks to cores, ensuring that the time constraints of all tasks are met
  - Load Balancing [Merkel, 2010]
    - Improve system's performance by migrating tasks from busy cores to idle cores, at the cost of additional migration time
  - Reasonable Allocation Techniques [Huang, 2016]
    - First Fit (first core with available capacity),
    - Best Fit (core with maximum available capacity)
    - Worst Fit (core that has the minimum capacity available)
  - Task Aware Scheduling [Chiang, 2016]
    - Allocate tasks based on their resource requirements to ensure optimal utilization for resources

Evaluation of Scheduling Algorithms

- Worst Case Response Time (WCRT)
  - Sum of its execution time and the delays that are accumulated due to preemptions from higher priority tasks

- Most of the existing analysis approaches make their preferable assumptions for the tasks and system resources
  - Fail to compare different scheduling techniques on a common and fair ground

Proposed Solution: COMFAST

- COMFAST: A Comparative Framework for Analysis of Scheduling Techniques in Multi cores
  - System Model
  - A generic Analysis Approach

COMFAST: System Model

- *m* cores
- Task graph: A set of n tasks
  - $G_i = \{T_1, T_2, \ldots T_n\}$
  - *T*: task

D_i	Task deadline
τ^{BCET}	Best case execution time
τ^{WCET}	Worst case execution time
R_i	Response time
τ^{minS}	Minimum start time
τ^{maxS}	Maximum start time
τ^{minF}	Minimum finish time
τ^{maxF}	Maximum finish time
τ^{N}	Maximum number of resource accesses
τ^{M}	Maximum access duration
τ^{d}	Total resource demand
τ^{pri}	Task priority
τ^{p}	Processor

R is calculated using τ^{WCET} and the surplus time added due to resource contention

$$\tau^d = \tau^N * \tau^M$$

Define dependencies between tasks

Deadline monotonic priority

COMFAST: System Model

- A task graph is represented by its schedulability (G_S)
  - Schedulable (G_S = 1): If all its constituent tasks meet their deadlines
  - Non-schedulable ($G_S = 0$): If any one of its tasks is unable to meet its deadline

- Data set is characterized by acceptance ratio (σ)
  - Ratio of the number of schedulable task graphs to the total number of task graphs in that data set

COMFAST

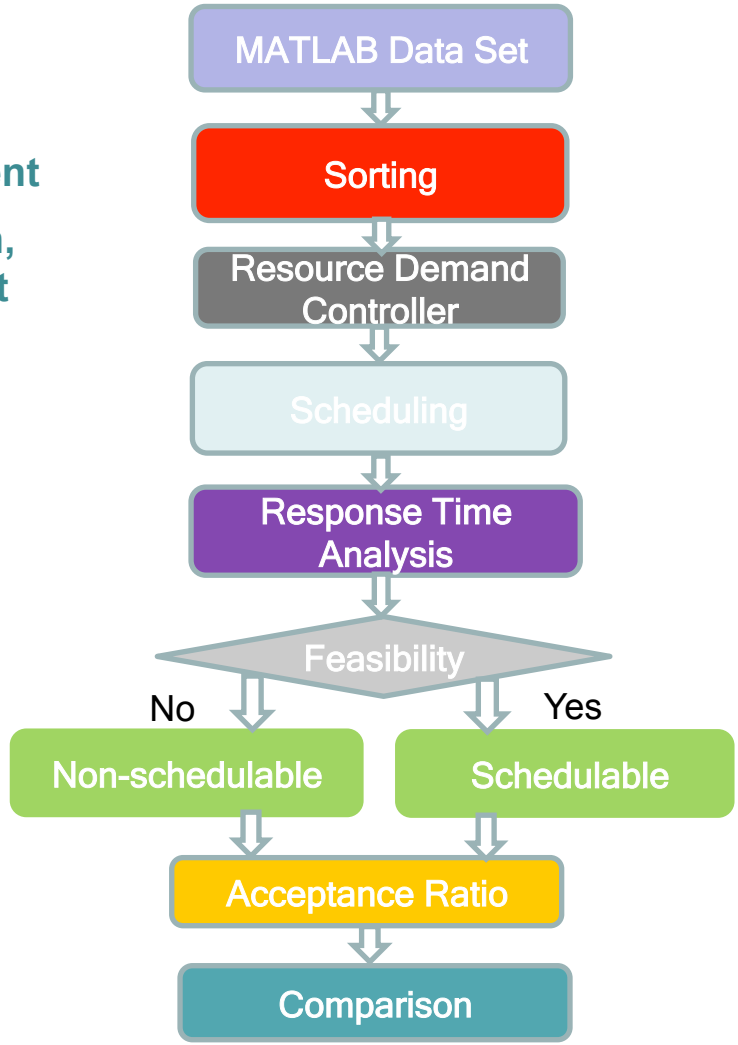
A generic data set that can be mapped to a wide range of real-time applications

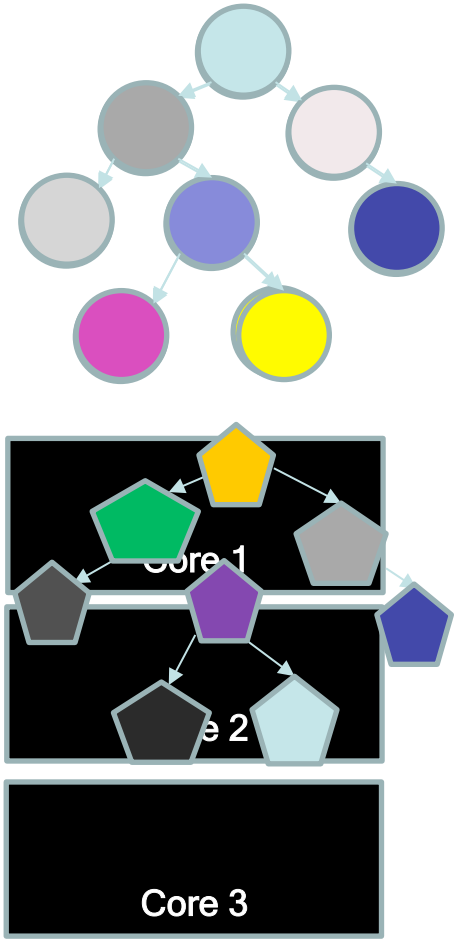
To facilitate priority assignment

Set a resource demand (high, medium, low) for the data set

Check if the response times exceed the deadlines

The number of task graphs that were deemed schedulable



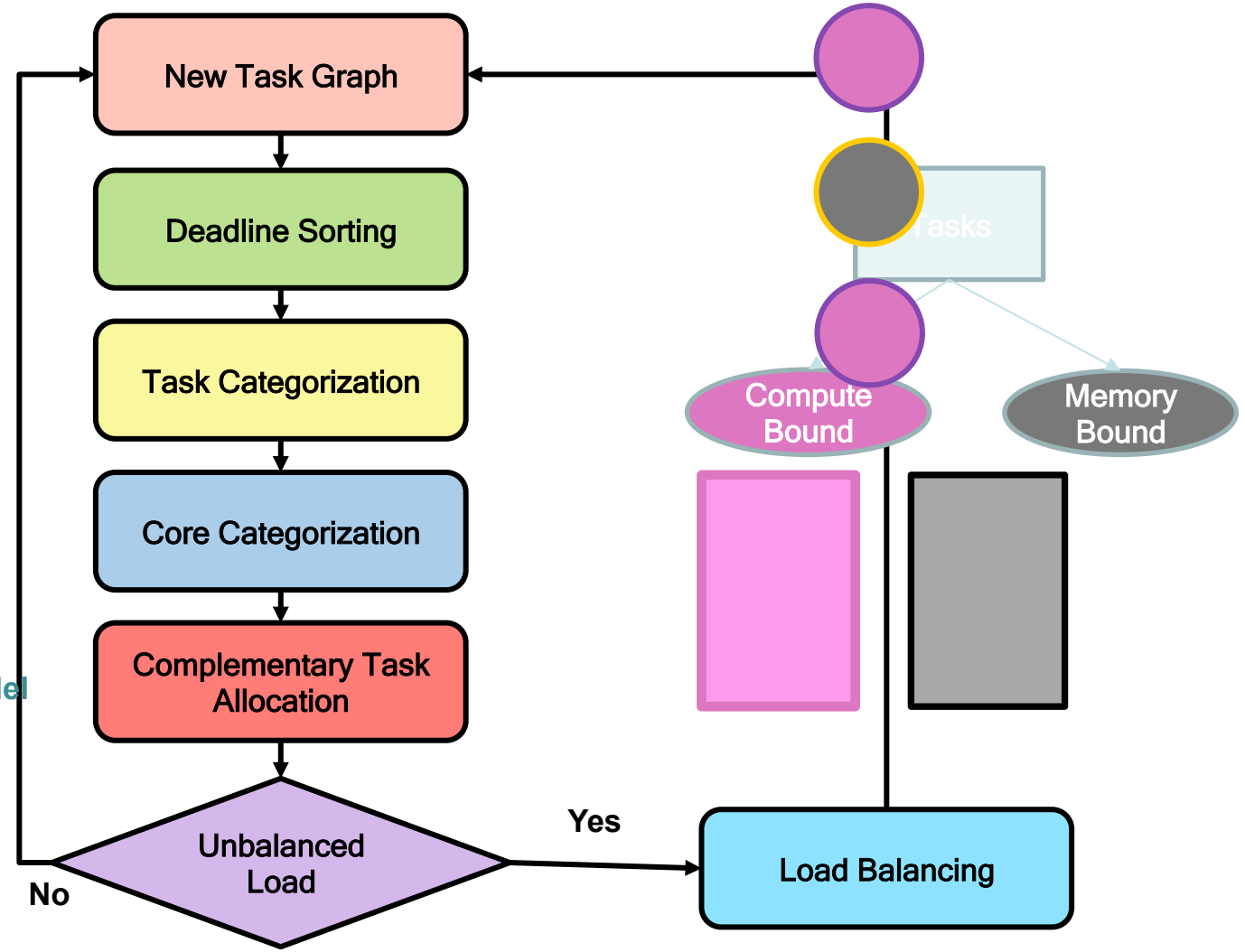


Case Studies

- Dynamic Task Aware Scheduling (DTAS) [Chiang, Journal of Systems and Software (JSS), 2016]

- First Fit (FF) Allocation Using Symmetric Analysis [Huang, Design Automation Conference (DAC), 2016]

Dynamic Task Aware Scheduling (DTAS)

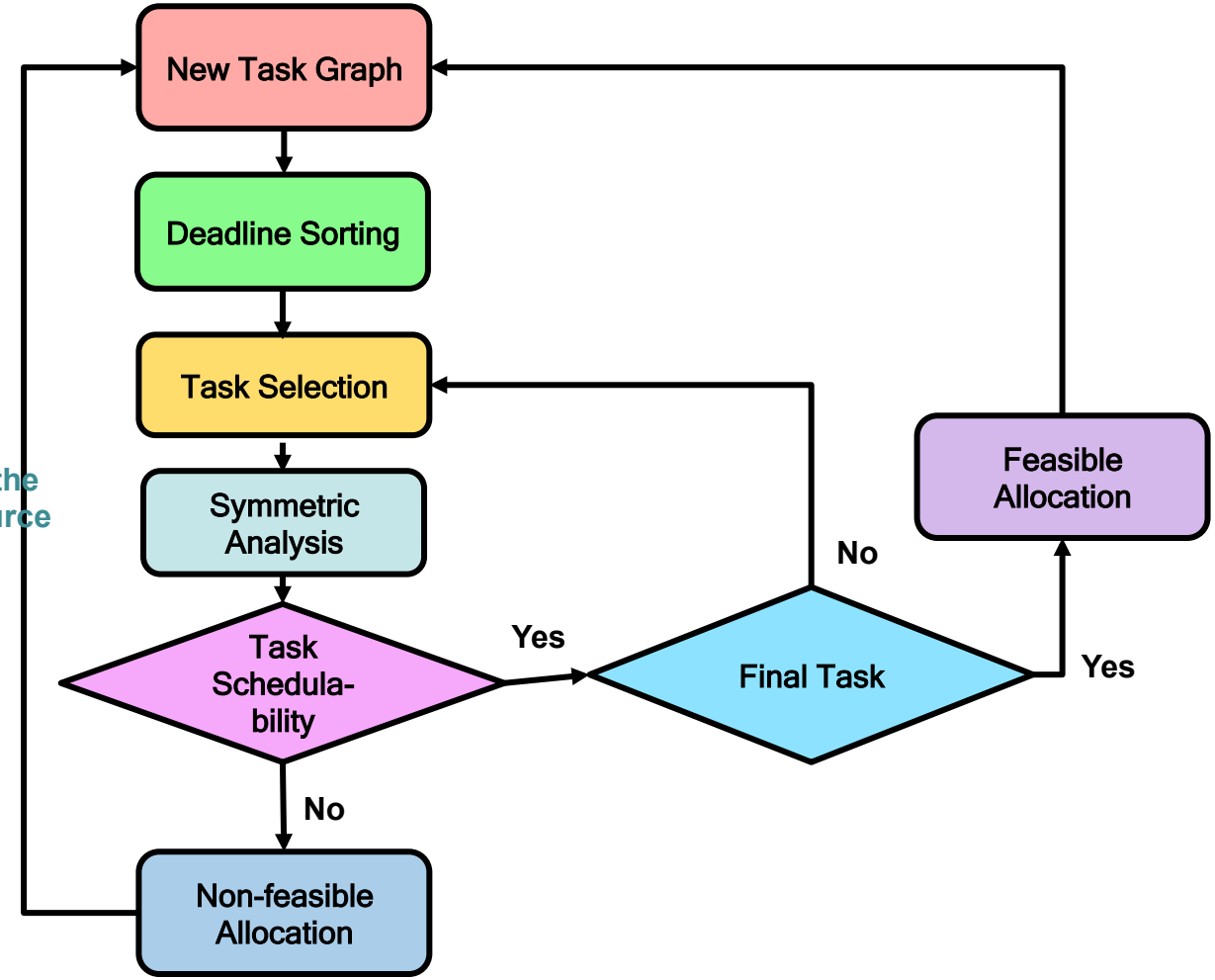


Run tasks with different resource demands in parallel

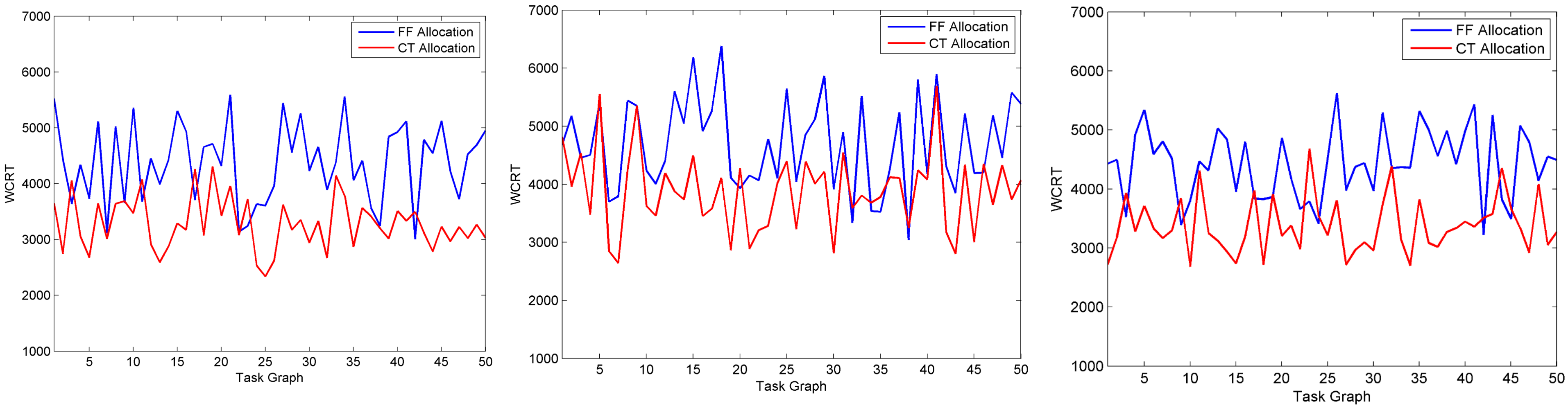
First Fit (FF) Allocation

Calculate times spent on the core and the shared resource

Check if tasks meet their deadlines based on symmetric analysis



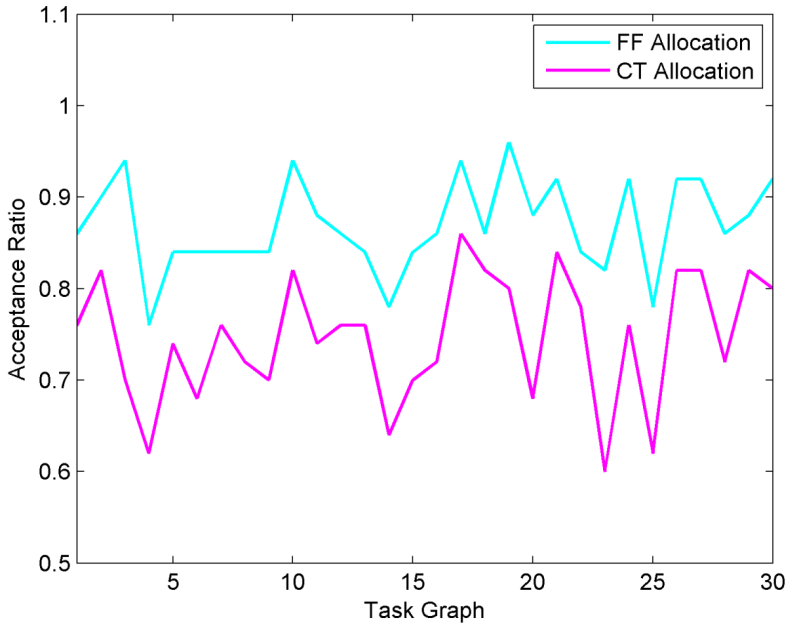
Experimental Results: WCRT



- **DTAS** gives **better** results than FF Allocation
  - **35.38% better** for rare resource accesses
  - **26.56% better** for medium resource accesses
  - **14.68% better** for frequent resource accesses

Experimental Results: Acceptance Ratios

- First Fit Allocation yields better results than complementary tasks by 15.3% at higher frequencies



Conclusions

- COMFAST: A unified platform for comparison of scheduling techniques that focus primarily on minimizing contention
  - A generic data set
    - Task sets with varying resource requirements
  - Comparison is done based on
    - Worst case response times
    - Schedulability
- Ongoing and Future work
  - Optimization of scheduling algorithms based on our findings
  - More case studies

Thanks!