Flexibility Driven Scheduling and Mapping for Distributed Real-Time Systems

Paul Pop, Petru Eles, Zebo Peng

Department of Computer and Information Science
Linköpings universitet, Sweden

Outline

- Introduction
- Incremental design process
  - Mapping and scheduling
- Problem formulation
- Mapping strategy
- Experimental results
- Conclusions

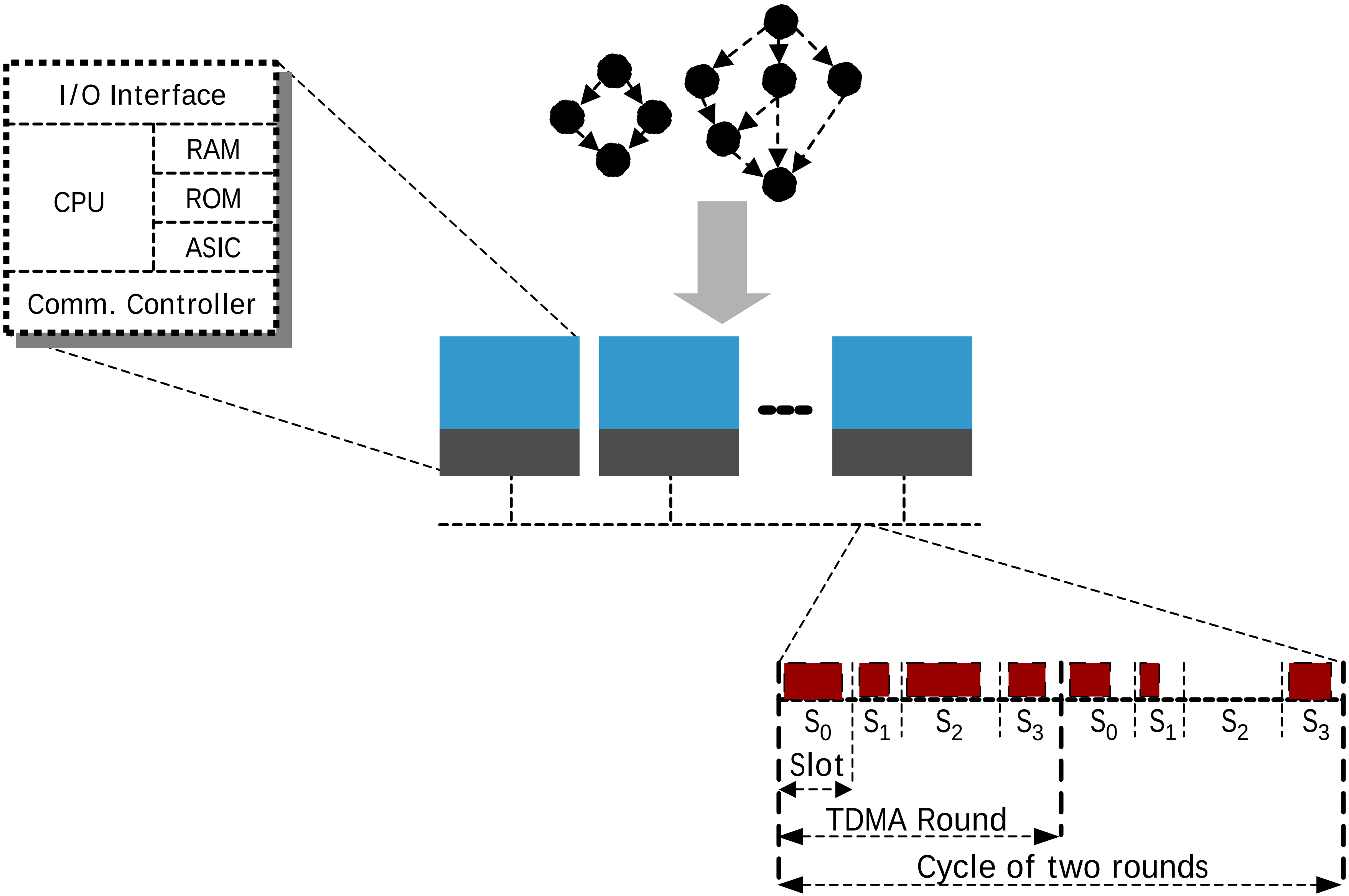
Introduction

- **Characteristics:**
  - Incremental design process, engineering change;
  - Distributed real-time embedded systems; Heterogeneous architectures;
  - Fixed priority pre-emptive scheduling for processes;

    static cyclic scheduling for messages;
  - Communications using a time-division multiple-access (TDMA) scheme:

    H. Kopetz, G. Grünsteidl. TTP-A Protocol for Fault-Tolerant Real-Time Systems. IEEE Computer '94.
- **Contributions:**
  - Mapping and scheduling considered inside an incremental design process;
  - Two design criteria (and their metrics) that drive our mapping strategies to solutions supporting an incremental design process;
  - Two mapping algorithms.
- **Message:**
  - Engineering change can be successfully addressed at system level.

"Classic" Mapping and Scheduling

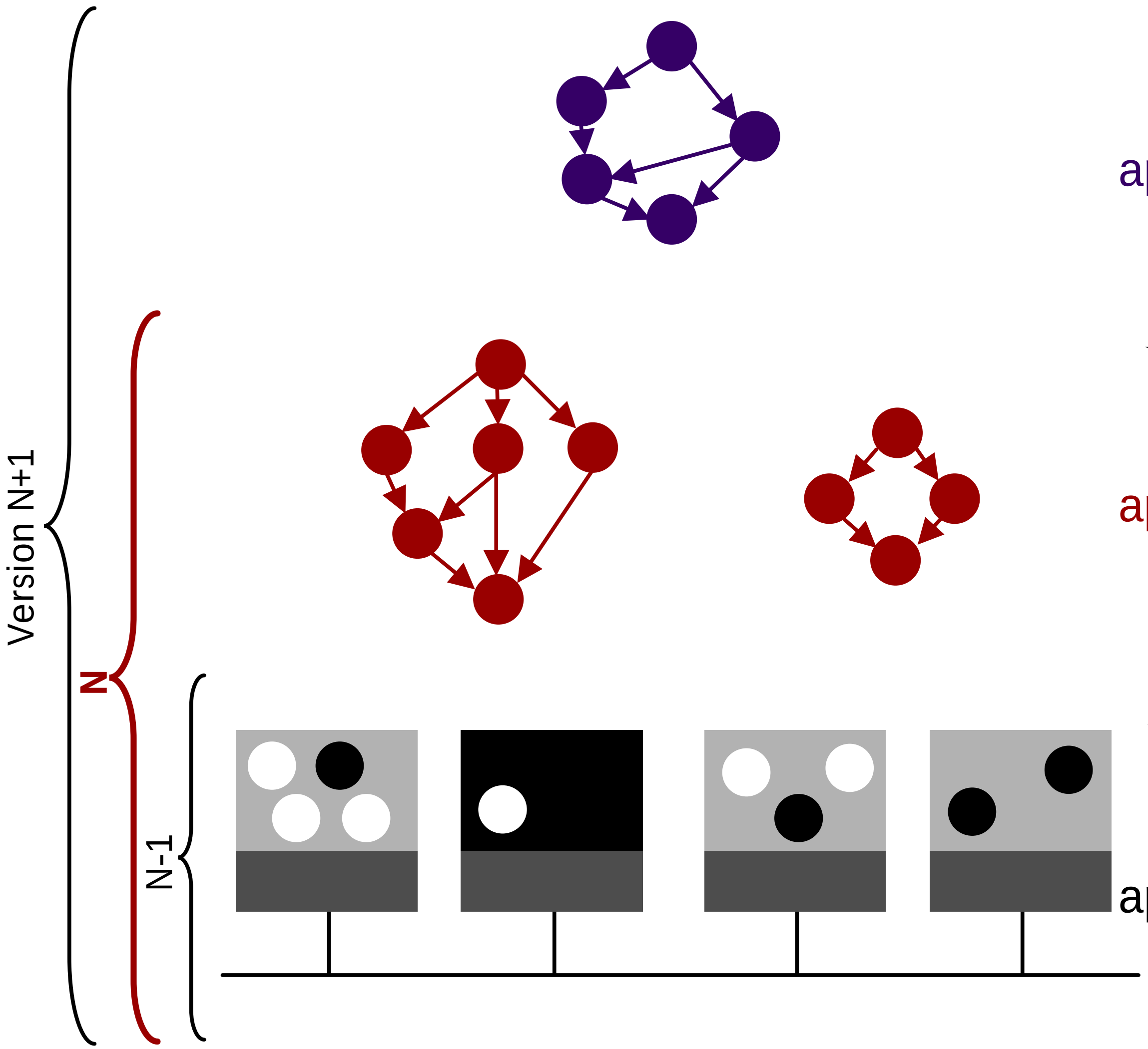


Incremental Design Process

- Start from an already existing system with applications:
  - In practice, very uncommon to start from scratch.

- Implement new functionality on this system (increment):
  - As few as possible modifications of the existing applications, to reduce design and testing time;
  - Plan for the next **increment**:
    It should be easy to add functionality in the future.

Mapping and Scheduling



Future applications

Do not exist yet at Version N!

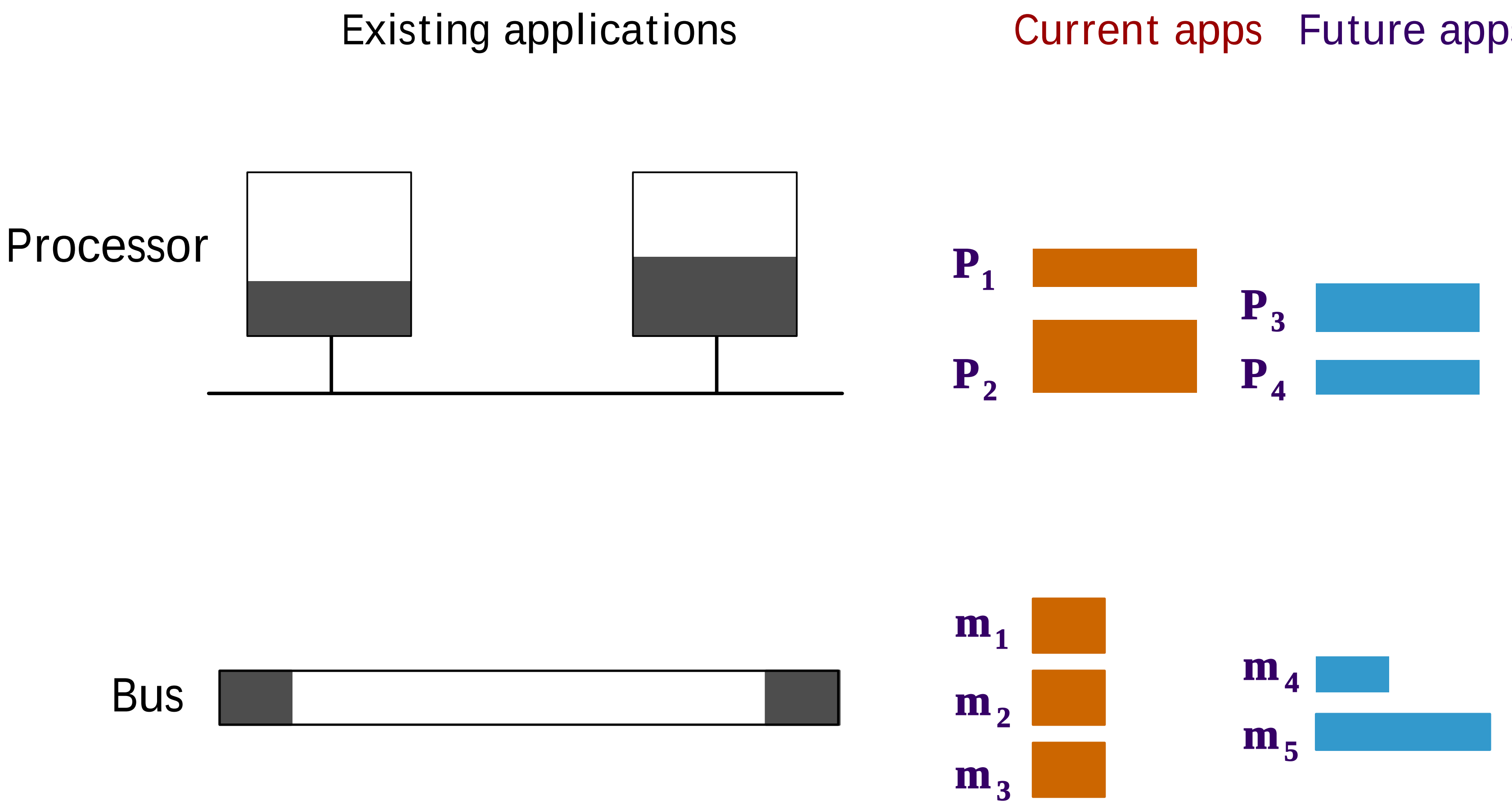
Current applications

Map and schedule so that the future applications will have a chance to fit.

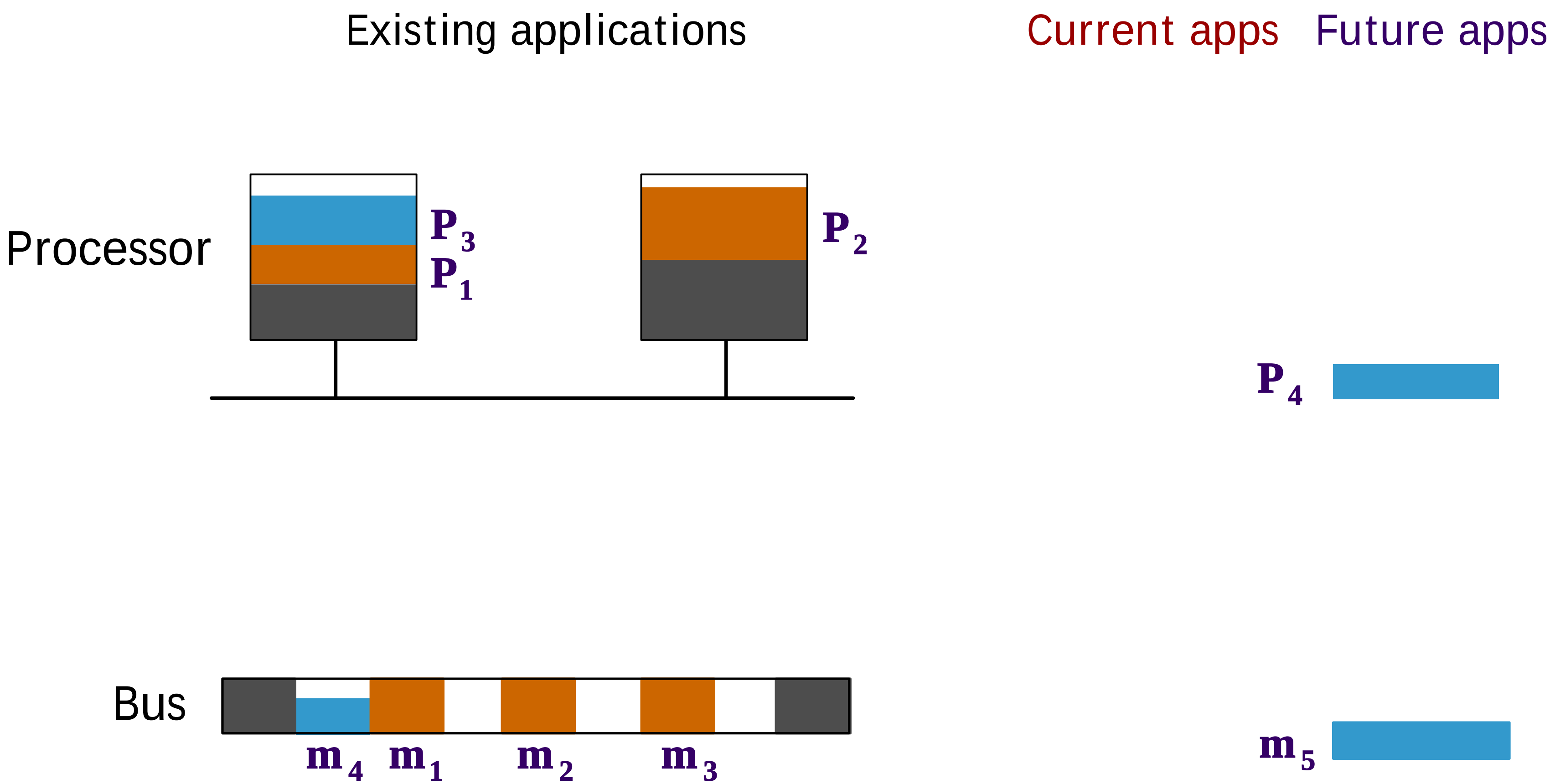
Existing applications

No modifications are performed to the existing applications.

Mapping and Scheduling Example



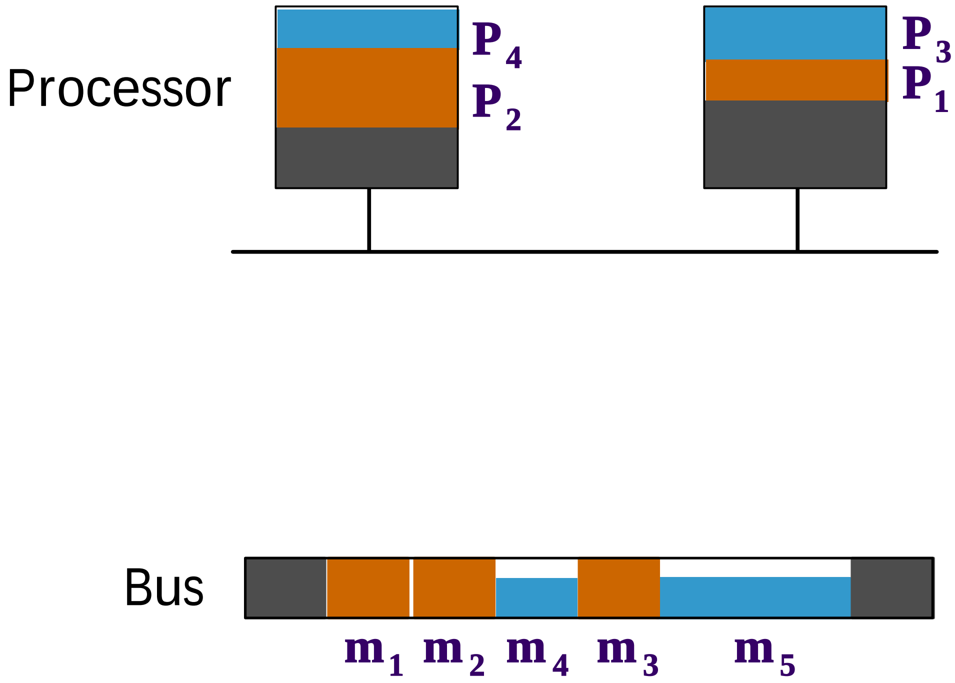
Mapping and Scheduling Example, Cont.



Mapping and Scheduling Example, Cont.

Existing applications

Current apps Future apps



Problem Formulation

Input

- A set of *existing* applications modelled as process sets.
- A *current* application to be mapped.
- Each process in the application has its own *period, priority* and *deadline*.
- Each process has a *potential set of nodes* to be mapped to and a *WCET*.
- The system architecture is given.

Output

- A **mapping and scheduling of the *current* application**, so that:
  - Requirement a: constraints of the *current* application are satisfied and minimal modifications are performed to the *existing* applications.
  - Requirement b: new *future* applications can be mapped on the resulted system.

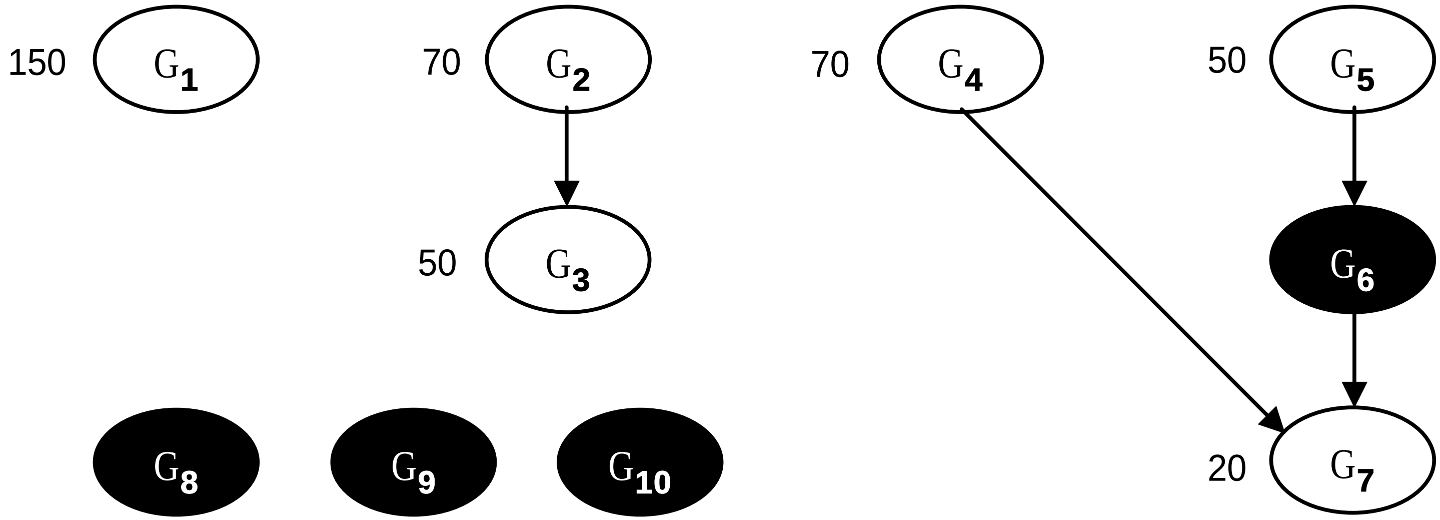
Mapping and Scheduling, Requirement a)

- **Mapping and scheduling of the *current* application**, so that:
  Constraints of the *current* application are satisfied and
  minimal modifications are performed to the *existing* applications.

- **Subset selection problem**
  Select that subset Ω of existing applications which guarantees that the current application fits and the modification cost $R(\Omega)$ is minimized:

$$R(\Omega) = \sum_{\Gamma_i \in \Omega} R_i$$

Characterizing Existing Applications

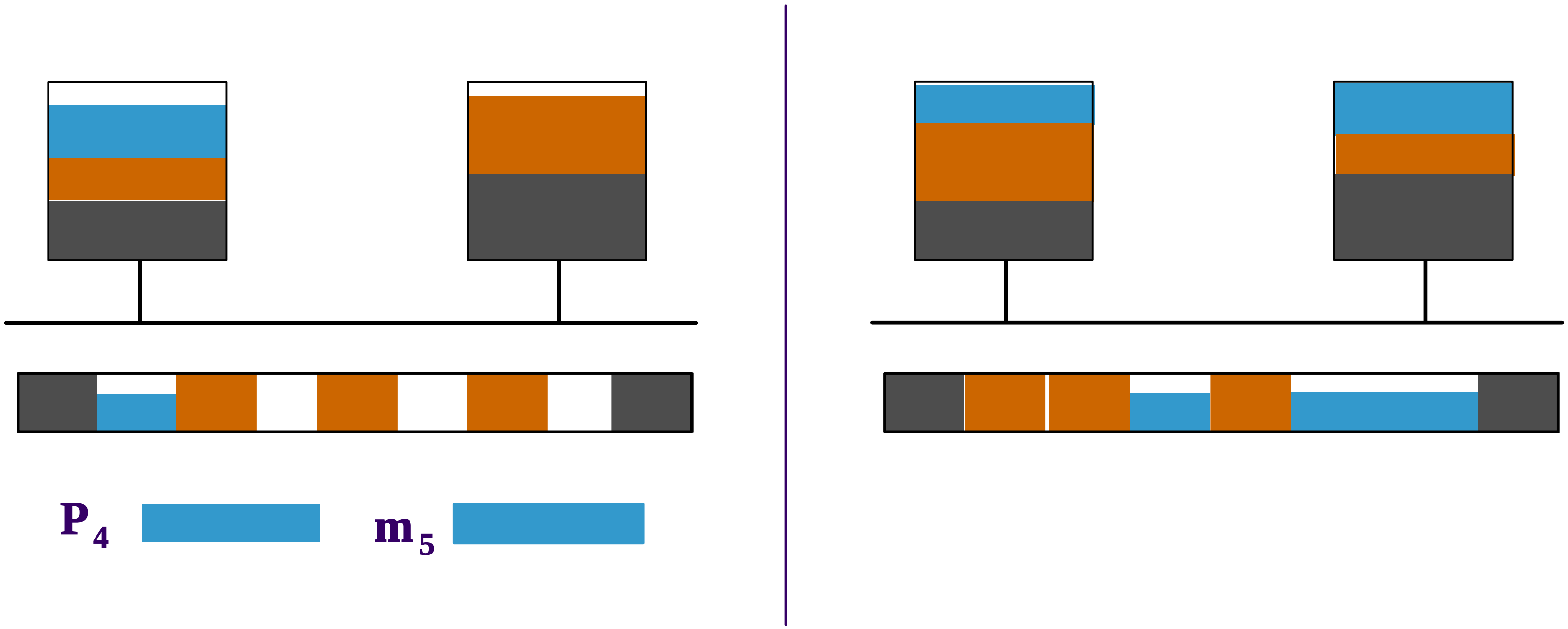


R({Γ_7})=20, R({Γ_3})=50, R({Γ_3 , Γ_7})=70,
R({Γ_4, Γ_7})=90 (the inclusion of Γ_4 triggers the inclusion of Γ_7),
R({Γ_2 , Γ_3})=120, R({Γ_3 , Γ_4 , Γ_7})=140, R({Γ_1})=150, ….

The total number of possible subsets is 16.

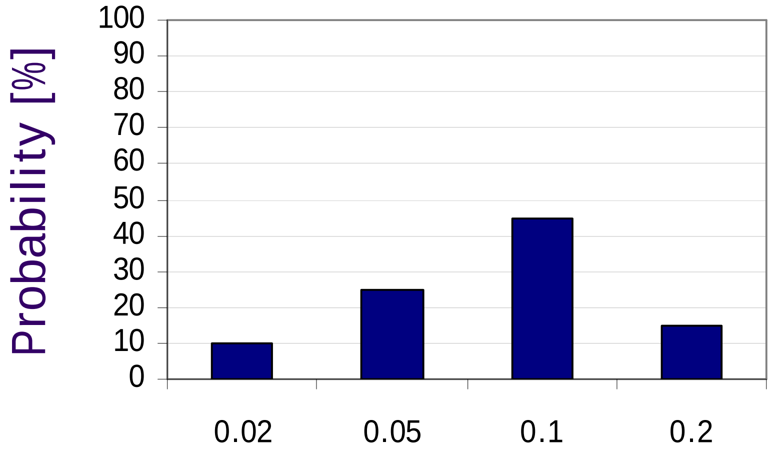
Mapping and Scheduling, Requirement b)

- **Mapping and scheduling of the *current* application**, so that:
  New *future* applications can be mapped on the resulted system.

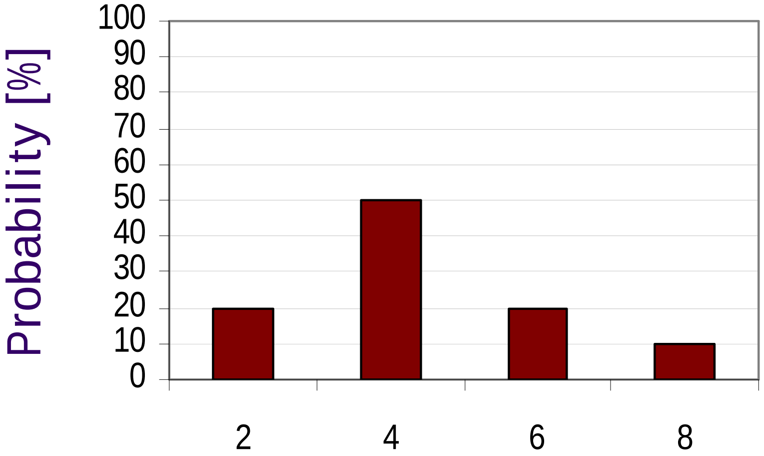


- Design criteria reflect the degree to which a design meets the requirement b);
- Design metrics quantify the degree to which the criteria are met;
- Heuristics to improve the design metrics.

Characterizing Future Applications



Typical *utilization factors*
$U_f = C/T$



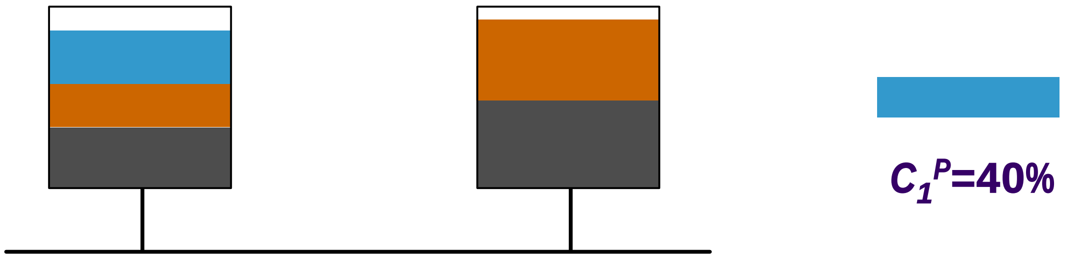
Typical message sizes [bytes]

- Smallest expected period T_{min}
- Expected necessary bandwidth b_{need}

Mapping and Scheduling: Processes

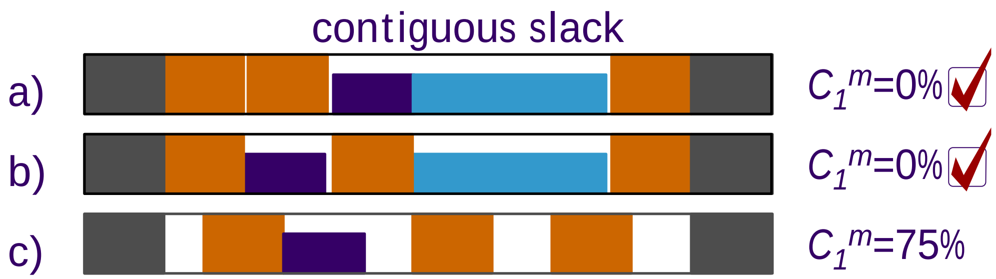
- Design criterion for processes: *available utilization*
  - How well the available utilization of the *current* design alternative accommodate a family of *future* applications that are characterized as outlined before;

- Design metrics for the first design criterion
  - C_1^P for processes
  - How much of the largest *future* application (total available utilization), *cannot* be mapped on the *current* design alternative;
  - *Bin-packing algorithm* using the *best-fit policy*: utilization factors of processes as objects to be packed, and the slack as containers.



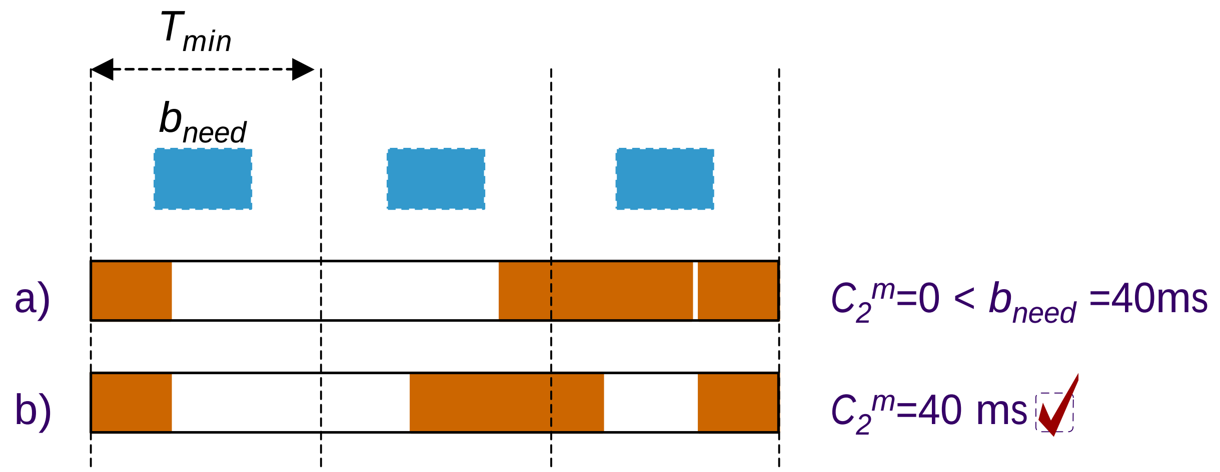
Mapping and Scheduling: Messages

- First design criterion for messages: slack sizes
  - How well the slack sizes of the *current* design alternative accommodate a family of *future* applications that are characterized as outlined before;
  - Tries to cluster the available slack: the best slack would be a contiguous slack.
- Design metrics for the first design criterion
  - C_1^m for messages;
  - How much of the largest *future* application (contiguous slack), *cannot* be mapped on the *current* design alternative;
  - *Bin-packing algorithm* using the *best-fit policy*: processes as objects to be packed, and the slack as containers.



Mapping and Scheduling: Messages, Cont.

- Second design criterion: slack distribution for messages
  - Used for the reduction of design space exploration
  - How well the slack of the *current* design alternative is distributed in time to accommodate the messages of a family of *future* applications;
  - Tries to distribute the slack so that we periodically (T_{min}) have enough necessary bandwith b_{need} for the most demanding future application.
- Design metrics for the second design criterion
  - C_2^m is the sum of minimum *periodic* slack inside a T_{min} period on each processor.



Mapping and Scheduling Strategy

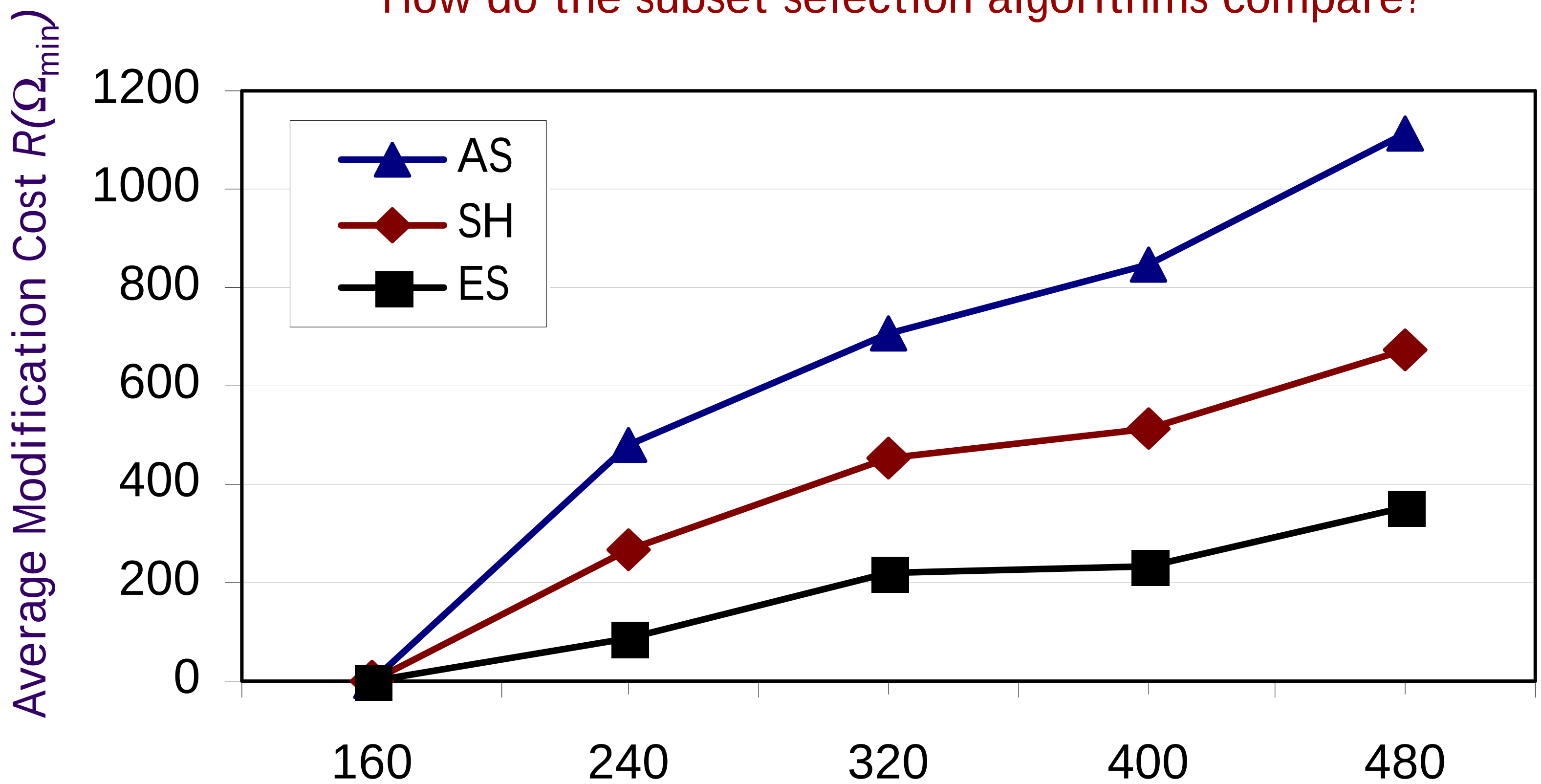
- Initial mapping and scheduling
- Requirement a)
  Minimizing the modification cost $R(\Omega)$, **subset selection**:
  - Exhaustive Search (ES)
  - Ad-Hoc Solution (AH)
  - Subset Selection Heuristic (SH)
- Requirement b)
  Starting from a valid solution, heuristics to minimize the objective function:

$$C = w_1^P (C_1^P) + w_1^m (C_1^m) + w_2^m \max(0, b_{need} - C_2^m)$$

  - Ad-Hoc approach (AH), little support for incremental design.
  - Simulated Annealing (SA), near optimal value for *C*.
  - Mapping Heuristic (MH):
    - Iteratively performs ***design transformations*** that improve the design;
    - Examines only transformations with the *highest potential* to improve the design;
    - Design transformations:
      moving a process to a different processor,
      moving a message to a different slack on the bus.

Experimental Results

How do the subset selection algorithms compare?

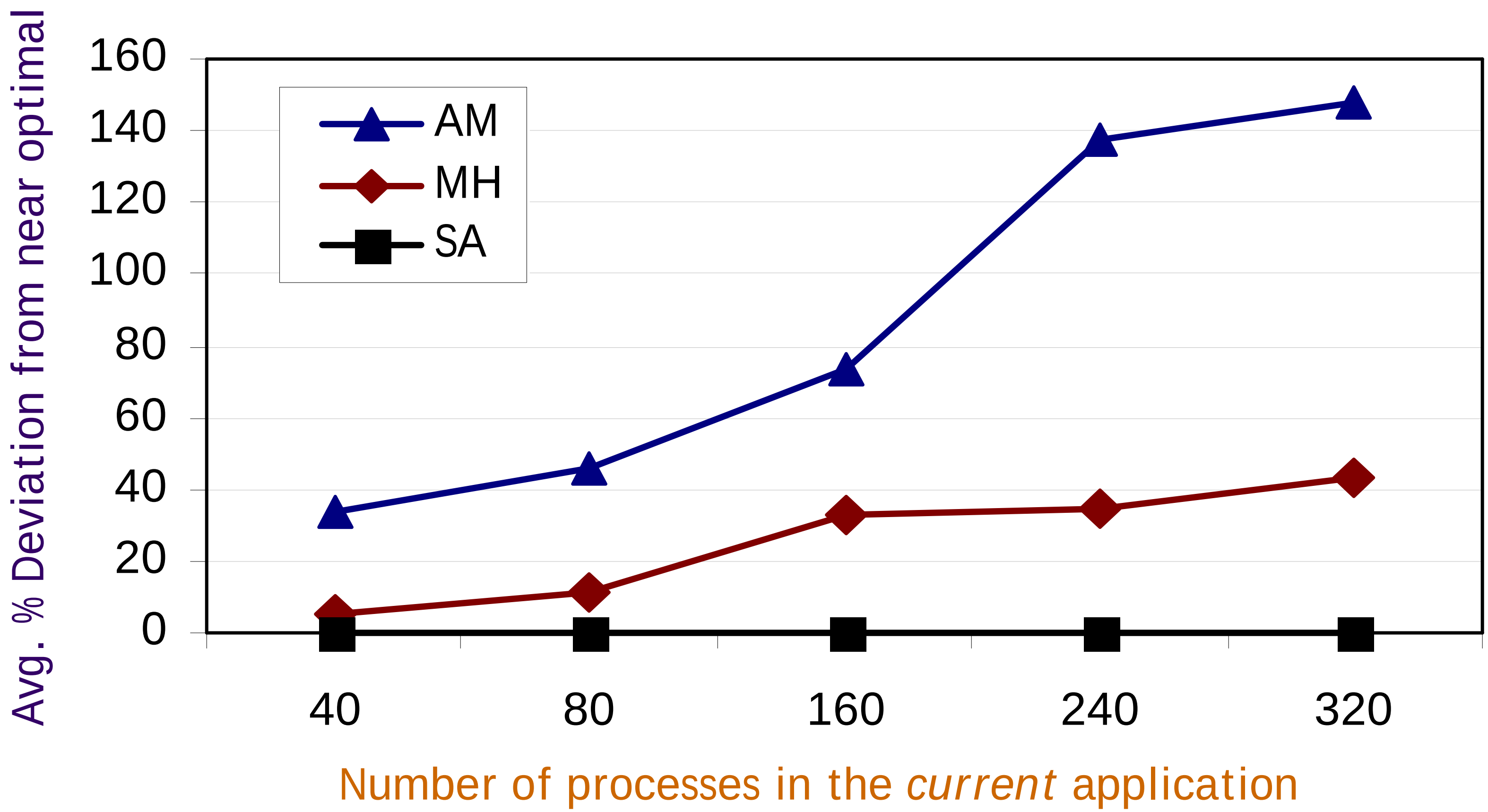


Number of processes in the *current* application

existing applications: 400

Experimental Results

How does the **quality** (cost function) of the mapping heuristic (MH) compare to the ad-hoc approach (AM) and the simulated annealing (SA)?

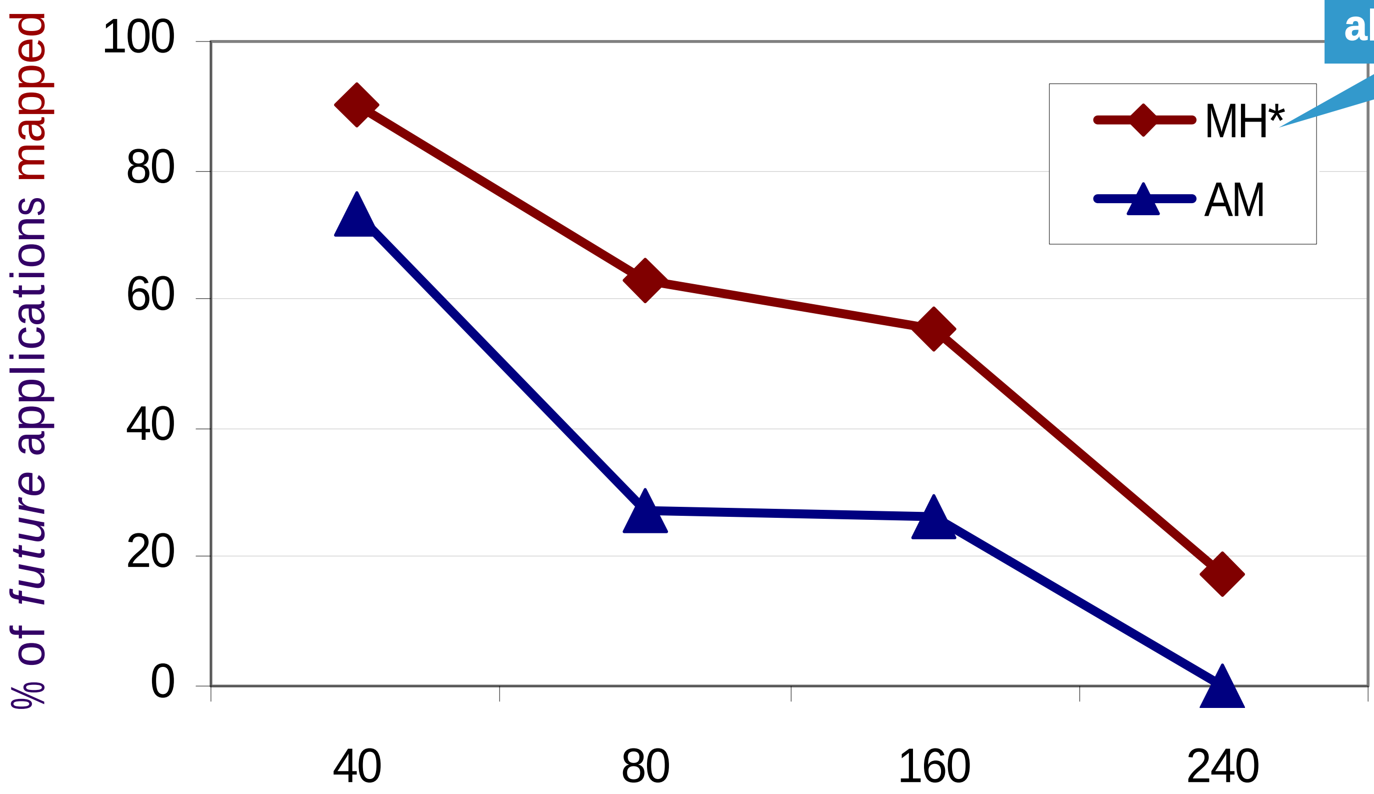


existing applications: 400

Experimental Results, Cont.

Are the mapping strategies proposed facilitating the implementation of future applications?

No modifications allowed



Number of processes in the *current* application

existing applications: 400, future application: 80

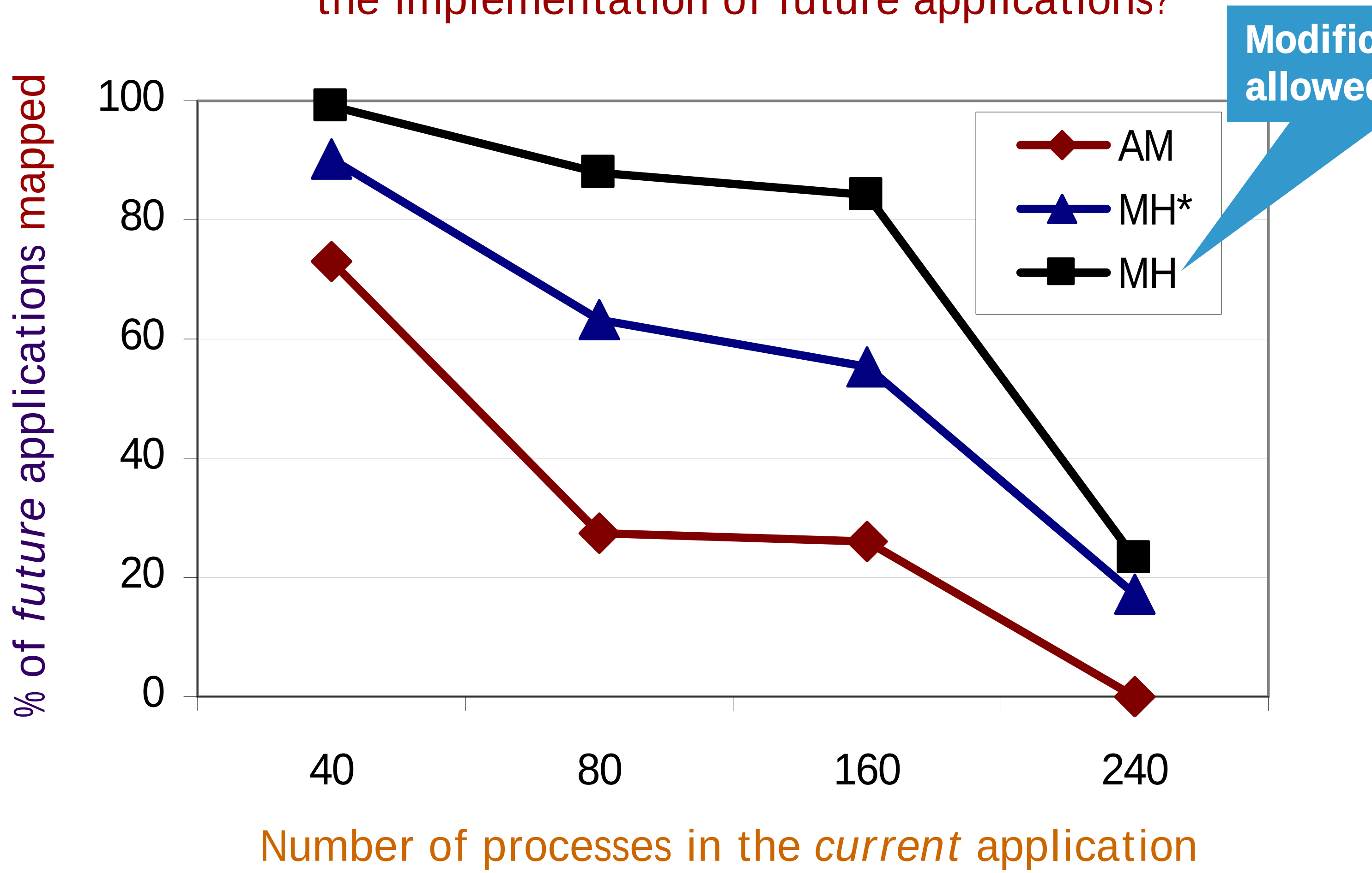
Future application

Current application

Existing application

Experimental Results, Cont.

Are the mapping strategies proposed facilitating the implementation of future applications?



existing applications: 400, future application: 80

Conclusions

- Distributed real-time embedded systems
  - Fixed priority pre-emptive scheduling for processes
  - Static cyclic scheduling for messages (TDMA)

- Mapping and scheduling considered inside an incremental design process
  - Constraints of the *current* application are satisfied and minimal modifications are performed to the *existing* applications
  - New *future* applications can be mapped on the resulted system

- Mapping strategy
  - Design criteria+metrics which drive mapping strategies to solutions supporting an incremental design process
  - Iterative improvement mapping algorithm; subset selection algorithm